

P7: Docker

Exercise1– El objetivo de este ejercicio es que se familiarice con la operación y los conceptos básicos de los contenedores virtuales.

1. Ejecute un comando en su host para crear y arrancar un contenedor usando la imagen hello-world. Anote el comando que ha utilizado. A continuación, ejecute un comando en su host para comprobar todos los contenedores que se han generado en su equipo. Escriba el comando utilizado, el resultado del mismo y describa el significado de cada columna.

//from terminal

docker run hello-world #también se puede hacer con create y start

docker ps -a

```
dumre ➤ dumre ➤ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

dumre ➤ dumre ➤ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
32d7ed255be3   hello-world    "/hello"                 55 seconds ago    Exited (0) 54 seconds ago          flamboyant_lumiere
```

La primera columna es el identificador de cada contenedor.

La segunda columna son los nombres de la imagen usada, va seguido de un nombre generado de una base de datos(por defecto) o el que se le aporta.

La tercera columna indica hace cuanto tiempo se ha creado.

Y la última columna muestra en qué estado está y si el proceso ha muerto, hace cuanto tiempo.

2. Ejecute de nuevo un comando en su host para crear y arrancar un nuevo contenedor usando la imagen hello-world. ¿Hay alguna diferencia respecto a la ejecución anterior? Nuevamente liste todos los contenedores que se han generado en su equipo, liste también las imágenes en su repositorio local.

//from terminal

docker run hello-world

docker ps -a

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
43aa1a0a54a2	hello-world	"/hello"	4 seconds ago	Exited (0) 4 seconds ago		happy_edison
32d7ed255be3	hello-world	"/hello"	3 minutes ago	Exited (0) 3 minutes ago		flamboyant_lumiere

Ahora la imagen como ya está instalada, se carga más rápido.

//from terminal

docker images #listar images del repositorio local

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
simtools	latest	5925781a7810	10 days ago	7.51GB
nginx	latest	76c69feac34e	3 weeks ago	142MB
debian	latest	d8cacd17cfdc	3 weeks ago	124MB
debian	9	662c05203bab	5 months ago	101MB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB

3. Averigüe el tamaño y el número de capas de la imagen hello-world.

//from terminal

docker image #para saber el tamaño

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
simtools	latest	5925781a7810	10 days ago	7.51GB
nginx	latest	76c69feac34e	3 weeks ago	142MB
debian	latest	d8cacd17cfdc	3 weeks ago	124MB
debian	9	662c05203bab	5 months ago	101MB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB

//from terminal

docker history hello-world #para ver las capas

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
feb5d9fea6a5	14 months ago	/bin/sh -c #(nop) CMD ["/hello"]	0B	
<missing>	14 months ago	/bin/sh -c #(nop) COPY file:50563a97010fd7ce...	13.3kB	

4. Borre todos los contenedores creados en su host.

//from terminal

docker container prune #borra todos los contenedores parados

```
dumre > dumre > ~ > docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
43aa1a0a54a2daf018e30599f4c0c88a518ca1fb861ce9274cb32cbea9503d58
32d7ed255be3ac586d9d24abfa4a4412a91e858a2b55d0f587e562a1a757702f

Total reclaimed space: 0B
```

#alternativa

docker rm \$(docker ps -aq) #borra todos los contenedores parados.

docker kill \$(docker ps -q) #para todos los procesos

docker rm -f \$(docker ps -a -q) # el combo

5. Cree un nuevo contenedor a partir de la imagen de hello-world, en este caso con nombre mycontainer y hostname mycont. Intente arrancar un nuevo contenedor con el mismo comando, ¿qué sucede?

//from terminal

docker run hello-world

docker create --name mycontainer --hostname mycont hello-world

```
dumre > dumre > ~ > docker create --name micontenedor --hostname mycont hello-world
75a16b87203fca02c73e293be4f5ffbc50b2e92184775545715ebbea7ab3b20

dumre > dumre > ~ > docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
75a16b87203f   hello-world "/hello"               4 seconds ago   Created                               micontenedor
54cf820b8d6a   hello-world "/hello"               2 minutes ago   Exited (0) 2 minutes ago           focused_bose
```

se crean 2 contenedores con la imagen de hello-world.

6. Cree dos nuevos contenedores con el mismo hostname, ¿hay algún problema?

//from terminal

docker create --name mycontainer1 --hostname mycont hello-world

docker create --name mycontainer2 --hostname mycont hello-world

```
dumre > dumre > ~ > docker create --name mycontainer1 --hostname mycont hello-world
docker create --name mycontainer2 --hostname mycont hello-world
b16cc9014d774e3286e2d78faa04a2386cd70dc41a6a3dfbbf8bfa89742f0766
e5e9e450deab09791a8ecdfe14198ea7c170b8430b78ff3dffa299236ee2cbf2

dumre > dumre > ~ > docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
e5e9e450deab   hello-world "/hello"               10 seconds ago   Created                               mycontainer2
b16cc9014d77   hello-world "/hello"               10 seconds ago   Created                               mycontainer1
75a16b87203f   hello-world "/hello"               3 minutes ago    Created                               micontenedor
54cf820b8d6a   hello-world "/hello"               6 minutes ago    Exited (0) 6 minutes ago           focused_bose
```

No da ningún problema con el hostname para 2 contenedores.

//from terminal

docker create --name mycontainer1 --hostname suma hello-world

```
dumre > dumre > ~ > docker create --name mycontainer1 --hostname suma hello-world
Error response from daemon: Conflict. The container name "/mycontainer1" is already in use by container
"b16cc9014d774e3286e2d78faa04a2386cd70dc41a6a3dfbbf8bfa89742f0766". You have to remove (or rename) that
container to be able to reuse that name.
```

Da error si ponemos el mismo contenedor (name).

7. Ejecute un contenedor a partir de una imagen de debian de forma interactiva con una bash como proceso principal.

//from terminal

docker run -it debian /bin/bash

```
dumre dumre ~ docker run -it debian /bin/bash
root@80241258944e:/#
```

d=daemon (para ejecutarse en background)

it=interactive

#para salir de modo interactivo: **ctrl+p+q**

8. Conéctese al contenedor que acaba de crear ejecutando otro proceso bash y cree en ese contenedor un fichero llamado test.log.

//from terminal

docker attach 80241258944e

touch test.log

ls -l

```
dumre dumre ~ docker attach 80241258944e
root@80241258944e:/# touch test.log
root@80241258944e:/# touch test.log
root@80241258944e:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys test.log tmp usr var
root@80241258944e:/# ls -l
total 64
drwxr-xr-x  2 root root 4096 Oct 24 00:00 bin
drwxr-xr-x  2 root root 4096 Sep  3 12:10 boot
drwxr-xr-x  5 root root 360 Nov 20 10:11 dev
drwxr-xr-x  1 root root 4096 Nov 20 10:11 etc
drwxr-xr-x  2 root root 4096 Sep  3 12:10 home
drwxr-xr-x  8 root root 4096 Oct 24 00:00 lib
drwxr-xr-x  2 root root 4096 Oct 24 00:00 lib64
drwxr-xr-x  2 root root 4096 Oct 24 00:00 media
drwxr-xr-x  2 root root 4096 Oct 24 00:00 mnt
drwxr-xr-x  2 root root 4096 Oct 24 00:00 opt
dr-xr-xr-x 350 root root  0 Nov 20 10:11 proc
drwx----- 2 root root 4096 Oct 24 00:00 root
drwxr-xr-x  3 root root 4096 Oct 24 00:00 run
drwxr-xr-x  2 root root 4096 Oct 24 00:00 sbin
drwxr-xr-x  2 root root 4096 Oct 24 00:00 srv
dr-xr-xr-x 13 root root  0 Nov 20 10:11 sys
-rw-r--r--  1 root root  0 Nov 20 10:14 test.log
drwxrwxrwt  2 root root 4096 Oct 24 00:00 tmp
drwxr-xr-x 11 root root 4096 Oct 24 00:00 usr
drwxr-xr-x 11 root root 4096 Oct 24 00:00 var
```

#comentaris

docker exec -it CONTAINER ID /bin/bash

te unes como otro proceso, si matas este, se mata todo.

docker attach CONTAINER ID

te unes al proceso principal, si matas este, encara hi ha el principal en ejecución.

9. Ejecute un comando para visualizar los cambios del sistema de ficheros del contenedor respecto a la imagen y observe como aparece el fichero anteriormente creado

//from terminal

docker diff 80241258944e #diff permite ver todos los cambios del sistema de ficheros

```
dumre dumre ~ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
80241258944e   debian        "/bin/bash"             23 minutes ago Up 23 minutes                cranky_haslett
e5e9e450deab   hello-world   "/hello"                32 minutes ago Created                mycontainer2
b16cc9014d77   hello-world   "/hello"                32 minutes ago Created                mycontainer1
75a16b87203f   hello-world   "/hello"                36 minutes ago Created                micontenedor
54cf820b8d6a   hello-world   "/hello"                38 minutes ago Exited (0) 38 minutes ago focused_bose

dumre dumre ~ docker diff 80241258944e
A /test.log
```

Exercise 2– El objetivo de este ejercicio es que se familiarice con la ejecución de servicios de red en contenedores virtuales.

1. Cree un contenedor nginx en modo "dettached" (o "daemon"). Navegue a la página principal servida por el contenedor y a una inexistente. Obtenga y comente los logs del servidor nginx.

//from terminal

docker run -d nginx

docker inspect 82b97681cecc | grep IPAddress

```
dumre dumre ~ docker run -d nginx
82b97681ceccc5417cf14494bfb2eeb72fb519146a865c54f172f7b546dfac08

dumre dumre ~ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
82b97681cecc   nginx    "/docker-entrypoint...." 6 seconds ago  Up 5 seconds  80/tcp       eloquent_mestorf

dumre dumre ~ docker inspect 82b97681cecc | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "172.17.0.2",
"IPAddress": "172.17.0.2",
```

//en el browser (chrome/ mozilla)

<http://172.17.0.2>



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

—alternativa—

#lo de arriba es para usuarios linux. Para windows o mac, hay que hacer port forwarding.

//terminal

docker run -d -p 8080:80 nginx

//web

localhost:8080

//from terminal

docker logs 82b97681cecc

```
dumre dumre ~ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
82b97681cecc   nginx    "/docker-entrypoint...." 16 minutes ago  Up 16 minutes  80/tcp       eloquent_mestorf

dumre dumre ~ docker logs 82b97681cecc
/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: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: 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: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/11/20 10:54:56 [notice] 1#1: using the "epoll" event method
2022/11/20 10:54:56 [notice] 1#1: nginx/1.23.2
2022/11/20 10:54:56 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/11/20 10:54:56 [notice] 1#1: OS: Linux 5.15.0-53-generic
2022/11/20 10:54:56 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/11/20 10:54:56 [notice] 1#1: start worker processes
2022/11/20 10:54:56 [notice] 1#1: start worker process 29
2022/11/20 10:54:56 [notice] 1#1: start worker process 30
2022/11/20 10:54:56 [notice] 1#1: start worker process 31
2022/11/20 10:54:56 [notice] 1#1: start worker process 32
2022/11/20 10:54:56 [notice] 1#1: start worker process 33
2022/11/20 10:54:56 [notice] 1#1: start worker process 34
2022/11/20 10:54:56 [notice] 1#1: start worker process 35
2022/11/20 10:54:56 [notice] 1#1: start worker process 36
172.17.0.1 - - [20/Nov/2022:10:59:20 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36" "-"
2022/11/20 10:59:20 [error] 29#29: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "172.17.0.2", referer: "http://172.17.0.2/"
172.17.0.1 - - [20/Nov/2022:10:59:20 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://172.17.0.2/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36" "-"
```

da error 404,555, errores de peticiones http, no such file or directory.

2. Ejecute una bash en el contenedor y modifique el fichero index.html.

//from terminal

docker exec -it b618b089ce0f /bin/bash

#se ha de actualizar, y instalar nano para poder modificar fichero index.html. (sudo apt update/install nano)

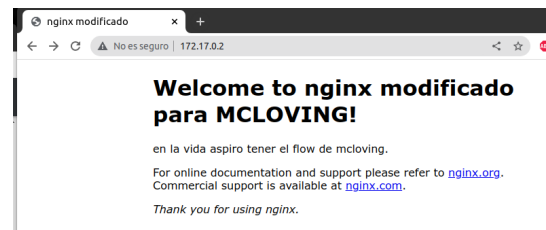
el fichero index.html se encuentra en usr/share/nginx/html/

```
GNU nano 5.4                                usr/share/nginx/html/index.html
<!DOCTYPE html>
<html>
<head>
<title>nginx modificado</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx modificado para MCLOVING!</h1>
<p>en la vida aspiro tener el flow de mcloving.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

#con nano modificamos index.html



3. Cree en el host un fichero index.html a su gusto y cópielo al contenedor utilizando el comando docker cp. Describa los pasos realizados.

//from terminal local del ordenador

nano index2.html

#despues de editar

docker cp ./index2.html b618b089ce0f:/usr/share/nginx/html

```
GNU nano 5.4                                index2.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
<h1 class="">Gokarna <br> Dumre
</h1>
<h2>
TELECOMMUNICATION <br>ENGINEER
</h2>
<h3>
Profile Info
</h3>

<p>
Hello my name is Gokarna Dumre. Im 22 years old and im Nepali. Currently im studying Telecoms in UPC, im in the 1st year of my bachelors degree.
<br>
<br>
<strong>soft skills</strong>
<ul> <!-- ul: unordenate list -->
<li>Team Worker</li> <!-- li: list -->
<li>Proactive</li>
<li>optimist</li>
</ul>
</p>
<a href="https://www.instagram.com/itsgorkiwiski/" target="_blank">@itsgorkiwiski
</a>
</table>
```

```
dumre ~ nano index2.html

dumre ~ ls
Descargas Escritorio index2.html Plantillas Simtools-Docker synth-shell
Documentos Imágenes Música Público snap Videos

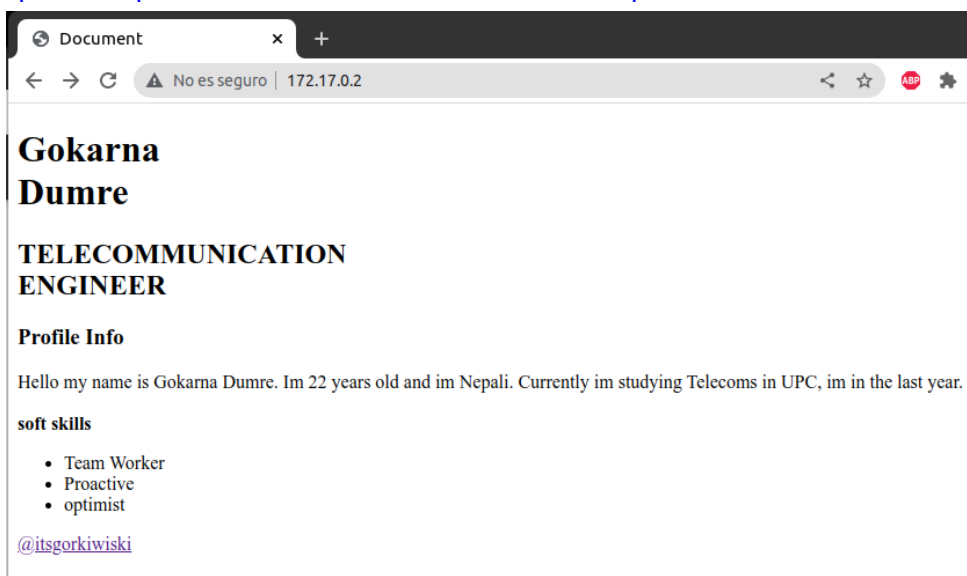
dumre ~ docker cp ./index2.html b618b089ce0f:/usr/share/nginx/html
```

```
root@b618b089ce0f:/usr/share/nginx/html# ls
50x.html index.html
root@b618b089ce0f:/usr/share/nginx/html# ls
50x.html index.html index2.html
root@b618b089ce0f:/usr/share/nginx/html#
```

Primero se ha creado el fichero index2.html en el ordenador local, despues se ha copiado dicho fichero al contenedor.

```
root@b618b089ce0f:/usr/share/nginx/html# ls
50x.html index.html index2.html
root@b618b089ce0f:/usr/share/nginx/html# cp index2.html index.html
root@b618b089ce0f:/usr/share/nginx/html# rm index2.html
root@b618b089ce0f:/usr/share/nginx/html# ls
50x.html index.html
root@b618b089ce0f:/usr/share/nginx/html#
```

Copio el index2 a index y eliminó index2. (index.html es el que se ve en el browser).
#para ir rápido, en vez de nombrar index2, si se pone index no hará falta copiar 2 veces.



4. Ahora arranque un contenedor nginx con un volumen que conecte un directorio de su host con el "document root" del contenedor nginx.

//from terminal local del ordenador

mkdir html

cp ./index2.html html

ls html

```
dumre ➤ dumre ➤ ~ ➤ ls
Descargas  Escritorio  index2.html  Plantillas  Simtools-Docker  synth-shell
Documentos  Imágenes    Música       Público     snap             Videos

dumre ➤ dumre ➤ ~ ➤ mkdir html

dumre ➤ dumre ➤ ~ ➤ cp ./index2.html html

dumre ➤ dumre ➤ ~ ➤ cat html/
cat: html/: Es un directorio

dumre ➤ dumre ➤ ~ ➤ ls html/
index2.html
```

docker run -dp 3000:3000 -v /home/dumre/html:/usr/share/nginx/html nginx

#en linux no cal poner -p, solo windows/mac

```
dumre ➤ dumre ➤ ~ ➤ docker run -dp 3000:3000 -v /home/dumre/html:/usr/share/nginx/html nginx
15ae0947d3f668cd38896eb60629a1be21fe523265ef322b4e0c5c50ad3fec52

dumre ➤ dumre ➤ ~ ➤ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
15ae0947d3f6   nginx    "/docker-entrypoint...." 58 seconds ago Up 57 seconds 80/tcp, 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp  dazzling_brattain
b618b089ce0f   nginx    "/docker-entrypoint...."  About an hour ago Up About an hour 80/tcp  pensive_raman
```

docker exec -it 15ae0947d3f6 /bin/bash

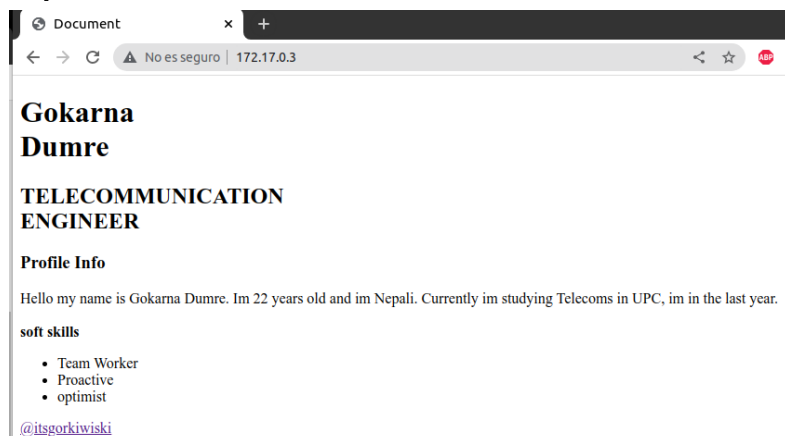
```
dumre ➤ dumre ➤ ~ ➤ docker exec -it 15ae0947d3f6 /bin/bash
root@15ae0947d3f6:/# cd /usr/share/nginx/html
root@15ae0947d3f6:/usr/share/nginx/html# ls
index2.html
root@15ae0947d3f6:/usr/share/nginx/html# mv index2.html index.html
root@15ae0947d3f6:/usr/share/nginx/html#
```

docker inspect 15ae0947d3f6 | grep IPAddress

```
dumre ➤ dumre ➤ ~ ➤ docker inspect 15ae0947d3f6 | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "172.17.0.3",
"IPAddress": "172.17.0.3",
```

//chrome/firefox

http://172.17.0.3/



vemos que hemos copiado un fichero local de nuestro ordenador al contenedor nginx.

5. Utilizando un volumen, cambie la configuración de un contenedor nginx para que el proceso principal de dicho contenedor sea ejecutado en el puerto 8080.

//from terminal #cal poner -p en windows/mac

docker run -d -v /home/dumre/html:/usr/share/nginx/html nginx

```
dumre> dumre> ~ docker run -d -v /home/dumre/html:/usr/share/nginx/html nginx
cbbd44c0e42e639e24a728e1fc997f96b4ff40cf61c41a72638fba05e13dbc4c

dumre> dumre> ~ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
cbbd44c0e42e   nginx    "/docker-entrypoint...."  8 seconds ago  Up 7 seconds  80/tcp      agitated_swartz

dumre> dumre> ~ docker inspect cbbd44c0e42e | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "172.17.0.2",
"IPAddress": "172.17.0.2",
```

docker exec -it be2add7c3f2d /bin/bash

#el nano hay que descargarlo

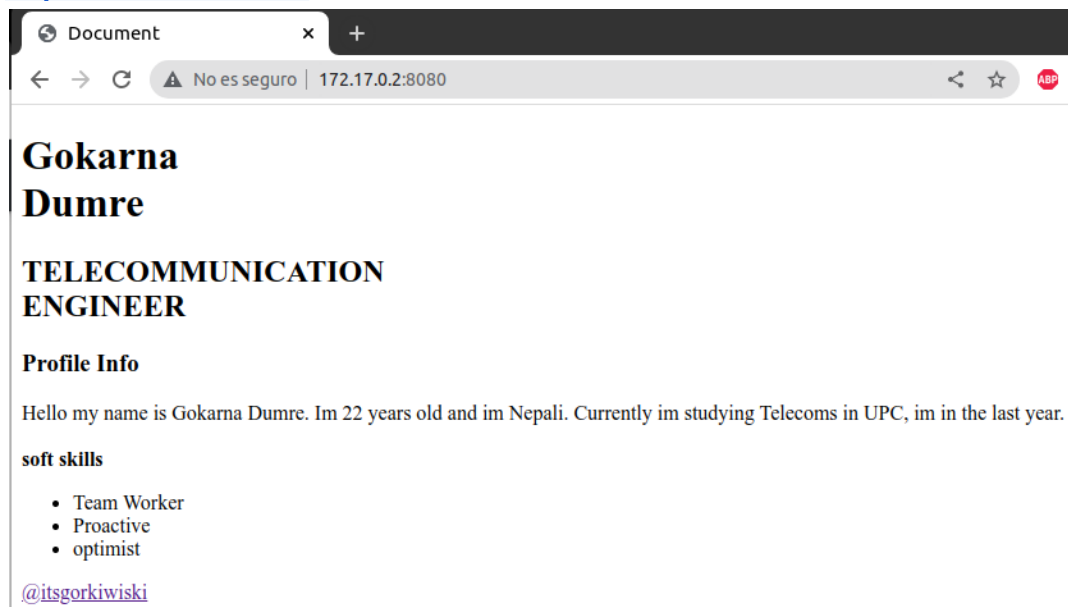
```
root@cbbd44c0e42e:/usr/share/nginx/html# cd /etc/nginx/conf.d/
root@cbbd44c0e42e:/etc/nginx/conf.d# ls
default.conf
root@cbbd44c0e42e:/etc/nginx/conf.d# nano default.conf
root@cbbd44c0e42e:/etc/nginx/conf.d# nano default.conf
root@cbbd44c0e42e:/etc/nginx/conf.d# cat default.conf
server {
    listen      8080;
    listen     [::]:8080;
    server_name localhost;

    #access_log /var/log/nginx/host.access.log  main;
```

docker stop cbbd44c0e42e

docker start cbbd44c0e42e

<http://172.17.0.2:8080/>



-v sirve para guardar datos, y que se pueda recuperarlos al reiniciar el contenedor. si no especifico el puerto 8080 dará error.

6. ¿Con qué comando se puede redireccionar el puerto 8081 del host al puerto 8080 del contenedor? realice dicha redirección y comente de que formas podría ahora acceder desde el host al contenido servidor por el contenedor nginx.

Seguimos el mismo procedimiento que el ejercicio anterior.

```
root@cbbd44c0e42e:/etc/nginx/conf.d# cat default.conf
server {
    listen      8080;
    listen  [::]:8080;
    server_name localhost;

    #access_log  /var/log/nginx/host.access.log  main;
```

pero ahora cambiando el 1er 8080 por 8081. El resultado sería el mismo.

<http://172.17.0.2:8081>

Exercise3– En este ejercicio se familiarizará con la creación de imágenes para contenedores.

1. Partiendo de una imagen Ubuntu, debe crear una imagen propia descrita en un Dockerfile. En su imagen debe instalar el paquete netcat-traditional. Además debe realizar una configuración en la que netcat arranque en el puerto 12345 y funcione como un servidor de echo.

En su Dockerfile:

- Deberá usar ENTRYPOINT con netcat como servidor de echo:
/bin/netcat -l -p 12345 -e /bin/cat
Donde: -l es para escuchar
-k es para seguir ejecutándose con múltiples llamadas
-p para especificar el puerto
-e para que se ejecute el programa indicado tras la conexión
- También deberá utilizar EXPOSE para documentar el puerto que ha utilizado.

```
//from terminal
mkdir Dokimages
cd Dokimages
nano Dockerfile
```

FROM ubuntu:20.04

RUN apt-get update && apt-get install -y netcat-traditional

ENTRYPOINT ["/bin/netcat","-l","-p","12345","-e","/bin/cat"]

EXPOSE 12345

#FROM:from el sistema operativo sobre el q correrá la imagen

#RUN:lo que quiero que se ejecute

#ENTRYPOINT especifica el ejecutable que usará el contenedor, y CMD se corresponde con los parámetros a usar con dicho ejecutable.

#EXPOSE: el puerto usado

//from terminal

docker build -t netcat . #el punto es para decir que la imagen esta en el directorio actual

```
dumre > dumre ~ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
f3a129f2c48c   netcat    "/bin/netcat -l -p 12345" 2 minutes ago  Up 2 minutes  12345/tcp

dumre > dumre ~ docker inspect f3a129f2c48c | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "172.17.0.2",
"IPAddress": "172.17.0.2",
```

docker run images

```
dumre > dumre ~/Dokimages docker run netcat
hoe
```

//another terminal

nc 172.17.0.2 12345

```
dumre > dumre ~ nc 172.17.0.2 12345
he
he
suuuuuu
suuuuuu
```

2. Comente las capas y el tamaño de la imagen que ha creado.

//from terminal

docker history netcat

```
dumre dumre ~ docker history netcat
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
a8a862bf31c1	34 minutes ago	/bin/sh -c #(nop) EXPOSE 12345	0B	
c26b97409294	34 minutes ago	/bin/sh -c #(nop) ENTRYPOINT ["/bin/netcat"...	0B	
fc3a5d83fffc	34 minutes ago	/bin/sh -c apt-get update && apt-get install...	40.6MB	
a8780b506fa4	2 weeks ago	/bin/sh -c #(nop) CMD ["bash"]	0B	
<missing>	2 weeks ago	/bin/sh -c #(nop) ADD file:29c72d5be8c977aca...	77.8MB	

5 capas, 1 para cada entryptpoint.

//from terminal

docker images

```
dumre dumre ~ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
netcat	latest	a8a862bf31c1	36 minutes ago	118MB
simtools	latest	5925781a7810	10 days ago	7.51GB
ubuntu	latest	a8780b506fa4	2 weeks ago	77.8MB
nginx	latest	76c69feac34e	3 weeks ago	142MB
debian	latest	d8cadc17cfdc	3 weeks ago	124MB
debian	9	662c05203bab	5 months ago	101MB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB

La imagen de netcat ocupa la suma de lo que ocupan sus capas.

size_image=Sum(size_layers)

3. Arranque un contendedor a partir de su imagen en modo "dettached".

//from terminal

docker run -d netcat

```
dumre dumre ~ docker run -d netcat
77bf452be6bafee4e4862cfc18a52a39135822261f0ebf8167f46a309b701b52
```

4. Ejecute un netcat en modo cliente en su host para conectarse al contenedor donde se ejecuta el netcat como servidor de echo y compruebe el correcto funcionamiento. Describa los pasos realizados.

```
dumre dumre ~ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
77bf452be6ba	netcat	"/bin/netcat -lkp 12..."	5 minutes ago	Up 5 minutes	12345/tcp	agitated_meninsky
f3a129f2c48c	netcat	"/bin/netcat -lkp 12..."	27 minutes ago	Up 27 minutes	12345/tcp	inspiring_meninsky

```
dumre dumre ~ docker inspect 77bf452be6ba | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "172.17.0.3",
"IPAddress": "172.17.0.3",
```

//another terminal

nc 127.17.0.3 12345

literamente es como el apartado 2. y sale lo mismo

5. Finalmente, a partir de una imagen de Ubuntu, cree una imagen que contenga el navegador de consola lynx. Describa los pasos realizados.

//from terminal

mkdir lynx

cd lynx

nano Dockerfile

FROM ubuntu

RUN apt-get update && apt-get install -y lynx

ENTRYPOINT ["/bin/lynx"]

EXPOSE 1234

```
dumre ➤ dumre ➤ ~/docker_images ➤ mkdir lynx
dumre ➤ dumre ➤ ~/docker_images ➤ cd lynx/
dumre ➤ dumre ➤ ~/docker_images/lynx ➤ ls
dumre ➤ dumre ➤ ~/docker_images/lynx ➤ nano Dockerfile
dumre ➤ dumre ➤ ~/docker_images/lynx ➤ docker build -t lynx .x
unable to prepare context: path ".x" not found
dumre ➤ dumre ➤ ~/docker_images/lynx ➤ docker build -t lynx .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM ubuntu:20.04
20.04: Pulling from library/ubuntu
eaead16dc43b: Pull complete
Digest: sha256:450e066588f42ebe1551f3b1a535034b6aa46cd936fe7f2c6b0d72997ec61dbd
Status: Downloaded newer image for ubuntu:20.04
--> 680e5dfb52c7
```

//from terminal

docker run -it lynx sport.es

```
dumre@dumre: ~/docker_images/lynx
SPORT | Noticias del Barça, La Liga, fútbol y otros deportes (pl of
#RSS de la portada de SPORT alternate alternate alternate alternate alternate alternate alternate
IFRAME: https://www.googletagmanager.com/ns.html?id=GTM-TTM3HX
Temas
* Mundial 2022 Directo
* Dnde ver el Mundial
* De Jong
* Lesión Benzema Mundial
* Stream Luis Enrique
* Futuro Mbappé
* Test Eric García
* Mercado fichajes
* Messi Cristiano
* Partidos Copa del Rey
sport
Edición
* ESEspaña
* USEstados Unidos
* VEVenezuela
* COColombia
* CLChile
```

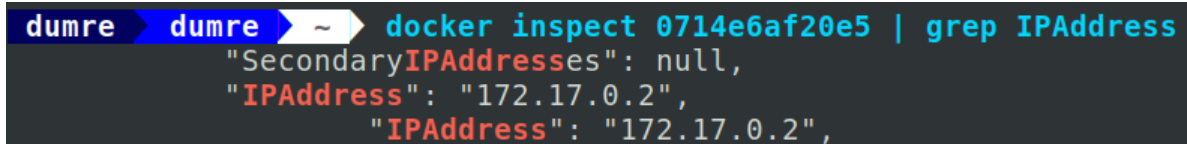
Exercise4– El objetivo de este ejercicio es que se familiarice con la creación de infraestructura de red virtual para entornos de contenedores.

1. Cree y arranque un contenedor a partir de la imagen que contiene lynx que ha creado. También cree y arranque un contenedor a partir de la imagen de nginx, ambos contenedores en el bridge por defecto. Navegue desde el contenedor que contiene lynx al contenedor nginx. Describa los pasos realizados.

//from terminal

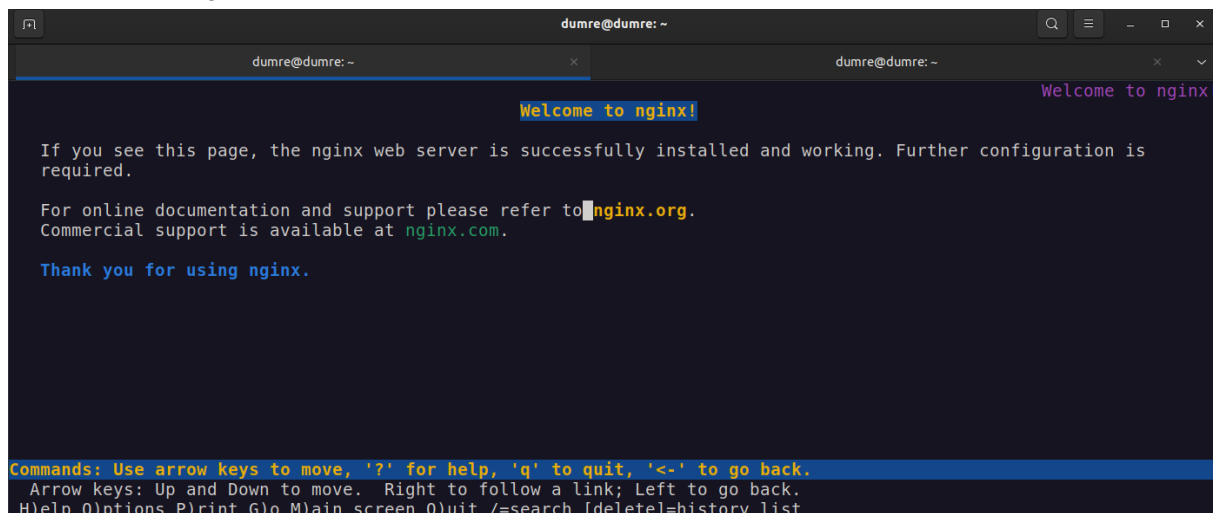
docker run -d nginx

docker inspect 0714e6af20e5 | grep IPAddress



```
dumre > dumre > ~ > docker inspect 0714e6af20e5 | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "172.17.0.2",
"IPAddress": "172.17.0.2",
```

docker run -it lynx 172.17.0.2



```
dumre@dumre: ~
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is
required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

2. Defina una red personalizada y arranque de nuevo dos contenedores como los anteriores en esa red. Navegue desde el contenedor que contiene lynx al contenedor nginx. Describa las diferentes formas en que puede realizar la conexión para esta navegación y describa los pasos realizados. Comente las direcciones IP obtenidas y sus diferencias con las del bridge por defecto. Obtenga la dirección IP del servidor y del cliente y comente la máscara obtenida y porqué es diferente a los contenedores inicializados en el bridge por defecto

//fromterminal

docker network create --driver bridge my-net

docker network ls

docker inspect

```
dumre > dumre ~ docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
c56ccaf930a8    bridge   bridge  local
f13b2da400d5    host     host    local
112a8daf77be    my-net   bridge  local
ae38bb3b6426    none     null    local

dumre > dumre ~ docker network inspect my-net | grep Subnet
"Subnet": "172.18.0.0/16",

dumre > dumre ~ docker network inspect bridge | grep Subnet
"Subnet": "172.17.0.0/16",

dumre > dumre ~ docker network inspect bridge | grep Gateway
"Gateway": "172.17.0.1"

dumre > dumre ~ docker network inspect my-net | grep Gateway
"Gateway": "172.18.0.1"
```

```
dumre > dumre ~ docker create --name nginx --network my-net nginx
c6e25e9aa8caa4854e9921b7181cf23191306ac9be7d0b462a505a3491e3553e

dumre > dumre ~ docker container start nginx
nginx

dumre > dumre ~ docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
c56ccaf930a8    bridge   bridge  local
f13b2da400d5    host     host    local
112a8daf77be    my-net   bridge  local
ae38bb3b6426    none     null    local

dumre > dumre ~ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
c6e25e9aa8ca   nginx    "/docker-entrypoint..." 34 seconds ago Up 14 seconds 80/tcp         nginx
ed2ac296748d   lynx     "/bin/lynx 172.17.0.2"    About an hour ago Exited (0) About an hour ago      compassionate_kepler
402f50a783b5   lynx     "/bin/lynx 172.17.0.2"    About an hour ago Exited (0) About an hour ago      sleepy_napier
0714e6af20e5   nginx    "/docker-entrypoint..." About an hour ago Up About an hour 80/tcp         loving_torvalds

dumre > dumre ~ docker inspect c6e25e9aa8ca | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "",
"IPAddress": "172.18.0.2",
```

docker run -it --name lynx --network my-net lynx 172.18.0.2

```
dumre@dumre: ~
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.
```

La las mascaras tanto del bridge por defecto, como el de my-net son 16, y vemos que tienen el netID diferente, 17 y 18. Como hemos arrancado nginx con el bridge my-net, la ip de nginx es una subred del bridge my-net,(tienen el mismo netID).

Y las ip son diferentes a los contenedores de bridge por defecto, porque como los contenedores son de 2 drivers (bridges) distintos.

Exercise5– El objetivo de este ejercicio es que utilice el orquestador de contenedores docker-compose.

1. Replique la estructura virtual anterior para contenedores lynx y nginx mediante un fichero docker-compose.

```
version: '3'
services:
  server:
    image: nginx
    networks:
      - mynet
  client:
    image: lynx
    stdin_open: true
    tty: true
    networks:
      - mynet
networks:
  mynet: # If not specified a docker-compose creates a user-network with a default name. >
    driver: bridge
```

```
#docker run -it --name lynx --network my-net lynx 172.18.0.2
```

//from terminal

```
GNU nano 6.2                                     docker-compose.yaml
version: '3'
services:
  server:
    image: nginx
    networks:
      - mynet
  client:
    image: lynx
    stdin_open: true
    tty: true
    networks:
      - mynet
networks:
  mynet: # If not specified a docker-compose creates a user-network with a default name.
    driver: bridge

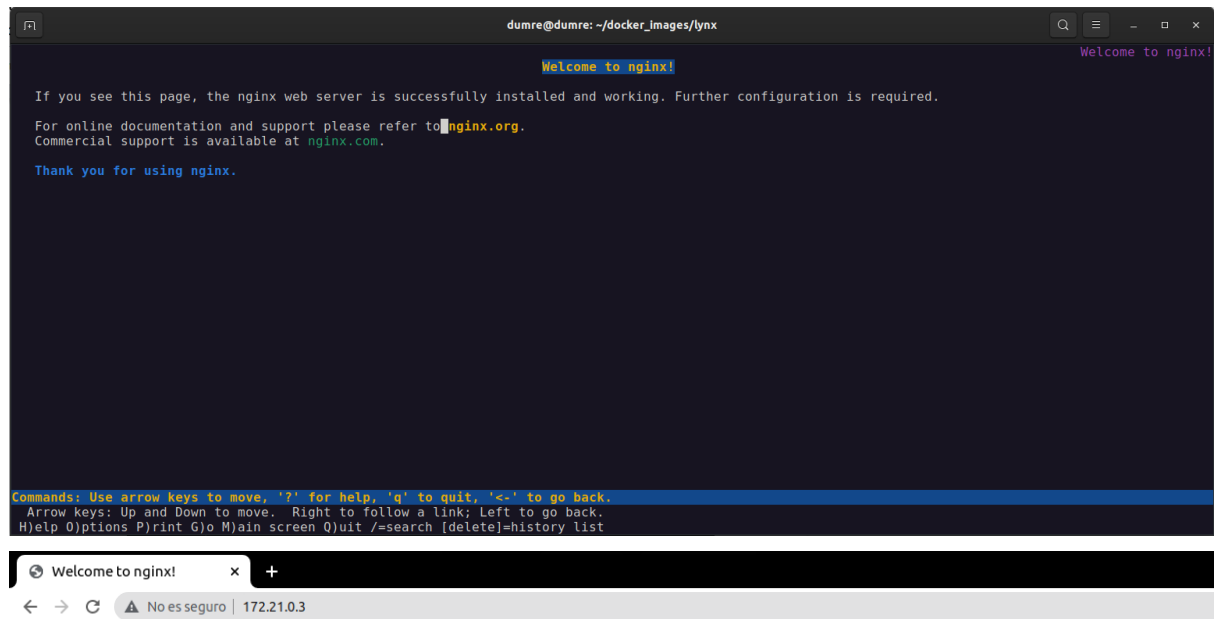
#docker run -it --name lynx --network my-net lynx 172.18.0.2
```

```
dumre dumre ~/docker_images/lynx docker-compose up -d
Creating lynx_server_1 ... done
Creating lynx_client_1 ... done

dumre dumre ~/docker_images/lynx docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
73438014473b   lynx     "/bin/lynx 172.21.0.3"   3 minutes ago Up 3 minutes
a553f0bf7fd9   nginx    "/docker-entrypoint..." 4 minutes ago Up 4 minutes
1c15c009863a   lynx     "/bin/lynx"              4 minutes ago Exited (1) 4 minutes ago

dumre dumre ~/docker_images/lynx docker inspect a553f0bf7fd9 | grep IPAddress
"SecondaryIPAddresses": null,
"IPAddress": "",
"IPAddress": "172.21.0.3",
```

docker-compose run client 172.21.0.3



Welcome to nginx!

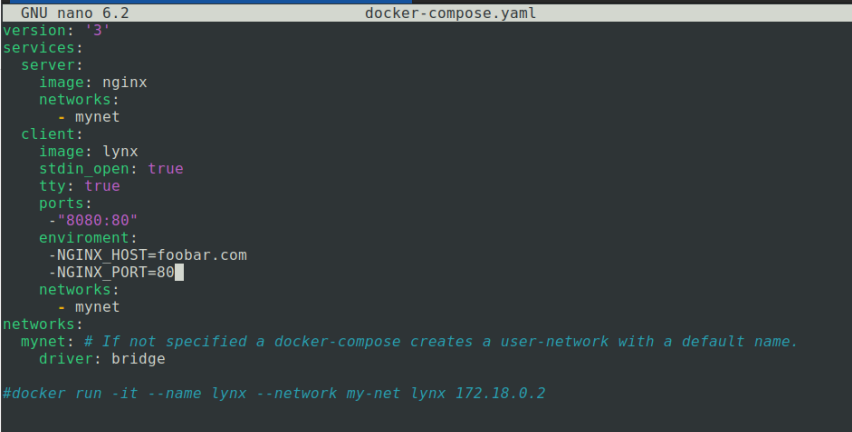
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

2. En el fichero docker-compose, configure el contenedor nginx para que escuche en el puerto 8080 pero que dicha configuración se realice mediante una variable de entorno. Para ello, puede realizar la configuración que figura en la documentación en https://hub.docker.com/_/nginx.

```
version: '3'
services:
  serv:
    image: "nginx"
    volumes:
      - ./templates:/etc/nginx/templates
    ports:
      - "8080:80"
    environment:
      - NGINX_HOST= foobar.com
      - NGINX_PORT=80
    networks:
      - mynet
  cli:
    image: "lab35"
    stdin_open: true
    tty: true
    networks:
      - mynet
networks:
  mynet:
    driver: bridge
```



```
GNU nano 6.2 docker-compose.yaml
version: '3'
services:
  server:
    image: nginx
    networks:
      - mynet
  client:
    image: lynx
    stdin_open: true
    tty: true
    ports:
      - "8080:80"
    environment:
      - NGINX_HOST=foobar.com
      - NGINX_PORT=80
    networks:
      - mynet
networks:
  mynet: # If not specified a docker-compose creates a user-network with a default name.
    driver: bridge

#docker run -it --name lynx --network my-net lynx 172.18.0.2
```

docker-compose build

docker-compose up -d

docker ps

docker inspect (id) //para saber la ip

docker-compose run server 8080

docker-compose run client 172.20.0.2