

Ejercicio - inicial

Ejercicio - inicial

Tarea 1:

Tarea 2:

Tarea 3:

Tarea 4:

Crear un contenedor demonio a partir de la imagen `nginx`, el contenedor se debe llamar **`servidor_web`** y se debe acceder a él utilizando el puerto 8181 del ordenador donde tengas instalado `docker` y podamos comprobar que el contenedor está funcionando.

Tarea 1:

Pantallazo donde se vea la creación del contenedor y podamos comprobar que el contenedor está funcionando.

```
docker run -d --name servidor_web -p 8181:80 nginx
```

```
manuelmc09@cliente:~$ docker run -d --name servidor_web -p 8181:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a2abf6c4d29d: Pull complete
a9edb18cadd1: Pull complete
589b7251471a: Pull complete
186b1aaa4aa6: Pull complete
b4df32aa5a72: Pull complete
a0bcbecc962e: Pull complete
Digest: sha256:0d17b565c37bcdb895e9d92315a05c1c3c9a29f762b011a10c54a66cd53c9b31
Status: Downloaded newer image for nginx:latest
0350a6e5b799df189f7503a23b7e25c622986e7d0097adc79f706c0857c571a2
manuelmc09@cliente:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
0350a6e5b799	nginx	"/docker-entrypoint..."	14 seconds ago	Up 11 seconds	0.0.0.0:8181->80/tcp, :::8181->80/tcp
servidor_web					

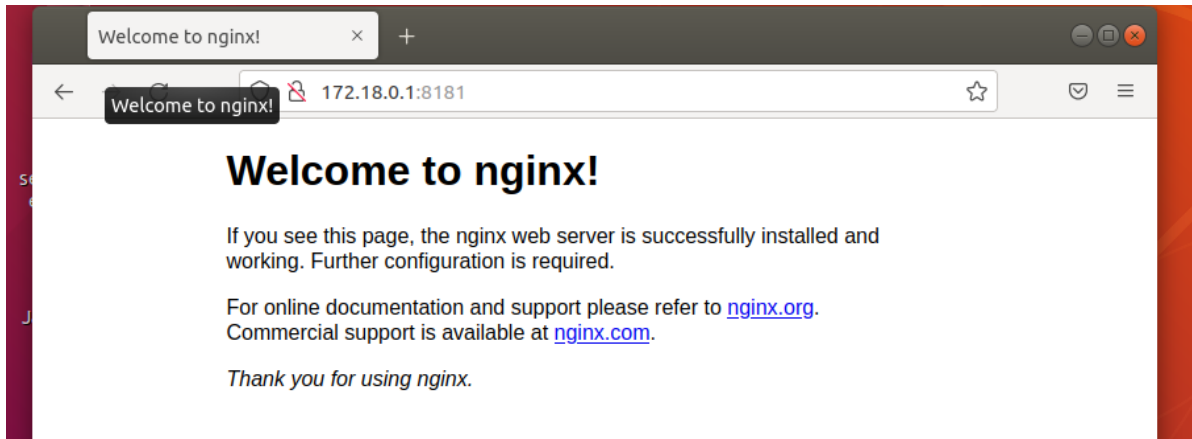
Tarea 2:

Pantallazo donde se vea el acceso al servidor web utilizando un navegador web (recuerda que tienes que acceder a la `ip` del ordenador donde tengas instalado `docker`)

En nuestro caso la dirección `ip` es: 172.18.0.1:8181

```
ip address
```

```
manuelmc09@cliente:~$ ip address
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 08:00:27:ef:6a:f7 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.10/24 brd 172.17.0.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::1465:e2a2:256:3ad/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:63:3b:13:cc brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:63ff:fe3b:13cc/64 scope link
        valid_lft forever preferred_lft forever
```



Tarea 3:

Pantallazo donde se vean las imágenes que tienes en tu registro local

```
docker images
```

```
manuelmc09@cliente:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    605c77e624dd   2 weeks ago   141MB
```

Tarea 4:

Pantallazo donde se vea cómo se elimina el contenedor (recuerda que antes debe estar parado el contenedor).

Paramos el contenedor primero:

```
docker stop servidor_web
```

```
manuelmc09@cliente:~$ docker stop servidor_web
servidor_web
```

Comprobamos a continuación que existe el contenedor servidor_web:

```
docker ps -a
```

```
manuelmc09@cliente:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0350a6e5b799	nginx	"/docker-entrypoint..."	23 minutes ago	Exited (0) 29 seconds ago		servidor_web

Eliminamos el contenedor y comprobamos que ya no está:

```
docker rm servidor_web
```

```
manuelmc09@cliente:~$ docker rm servidor_web
servidor_web
```

```
docker ps -a
```

```
manuelmc09@cliente:~$ docker ps -a
```

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

```
manuelmc09@cliente:~$
```