

---

## Sistemas Informáticos

### Tarea Docker 1: Creación y muestra de contenedor daemon en comodísimos pasos

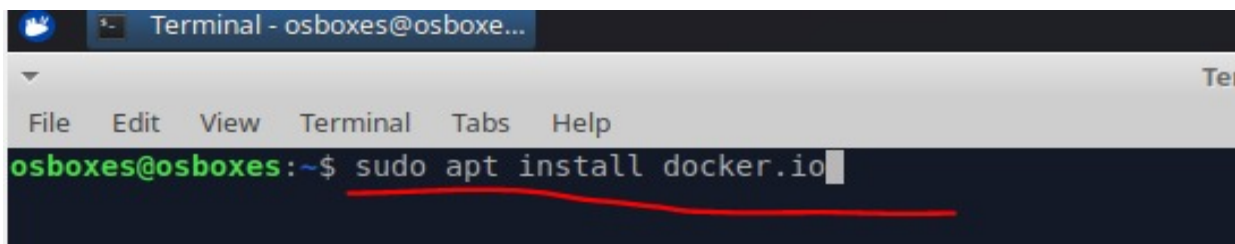
En este ejercicio crearemos un contenedor “daemon” en el sistema operativo Docker para Linux. En este y los siguientes ejercicios utilizaremos Xubuntu.

- 1) Utilizaremos la imagen nginx
- 2) Llamaremos al contenedor “servidor\_web”
- 3) Le asignaremos el puerto 8181
- 4) Mostraremos mediante pantallazos los pasos a seguir desde la misma instalación de Docker hasta, finalmente, comprobar que podemos acceder al contenedor creado

---

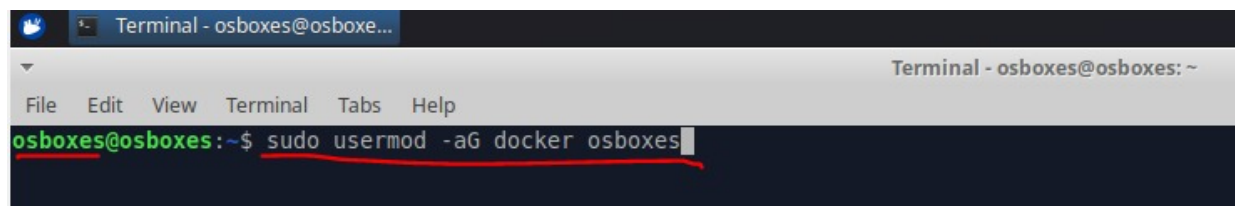
#### Preámbulo: Instalación de Docker y concesión de permiso de Administrador

Desde la consola de Ubuntu, escribimos el comando “sudo apt install docker.io” y pulsamos Enter. Esto instalará Docker.

A screenshot of a terminal window titled "Terminal - osboxes@osboxe...". The terminal shows the command `osboxes@osboxes:~$ sudo apt install docker.io` being entered. A red underline is drawn under the command text.

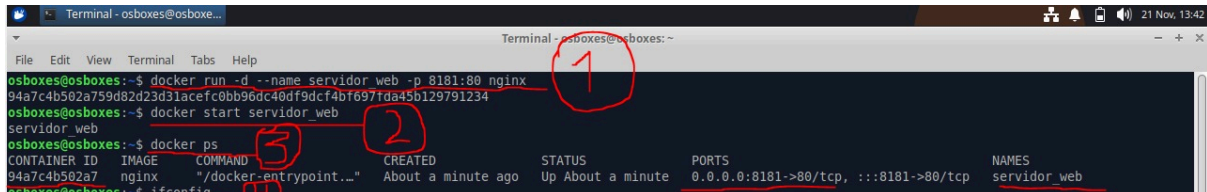
A continuación, escribiremos el siguiente comando: **sudo usermod -aG docker** seguido de nuestro nombre de usuario. En este caso, es “osboxes”.

Hecho esto, tenemos permiso de administrador. De lo contrario, tendríamos que escribir “sudo” delante de cada comando lo que resultaría demasiado molesto.

A screenshot of a terminal window titled "Terminal - osboxes@osboxe...". The terminal shows the command `osboxes@osboxes:~$ sudo usermod -aG docker osboxes` being entered. A red underline is drawn under the command text.

## Creación del contenedor

He aquí un pantallazo con una serie de números en rojo. Iremos explicando cada uno a continuación:



```
osboxes@osboxes:~$ docker run -d --name servidor_web -p 8181:80 nginx
94a7c4b502a759d82d23d31acefc0bb96dc40df9dcf4bf697fda45b129791234
osboxes@osboxes:~$ docker start servidor_web
servidor_web
osboxes@osboxes:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
94a7c4b502a7	nginx	"/docker-entrypoint..."	About a minute ago	Up About a minute	0.0.0.0:8181->80/tcp, :::8181->80/tcp	servidor_web

- 1)** Escribimos el comando **`docker run -d --name servidor_web -p 8181:80 nginx`**. Este comando crea el contenedor, le asigna de nombre “servidor\_web” (que es el que hemos elegido para este ejercicio) y le asigna el puerto 8181 del ordenador donde lo hayamos hecho en imagen nginx.
- 2)** A continuación introducimos el comando **`docker start servidor_web`**. Esto arrancará el contenedor.
- 3)** Introducimos el comando **`docker ps`**. Esto mostrará la lista de contenedores creados.
  - Debajo de CONTAINER ID veremos el número de identificación del contenedor.
  - Debajo de PORTS veremos los puertos asignados a la hora de escribir el comando.
  - Debajo de NAMES veremos el nombre asignado al contenedor.

## Comprobación de IP y conectividad

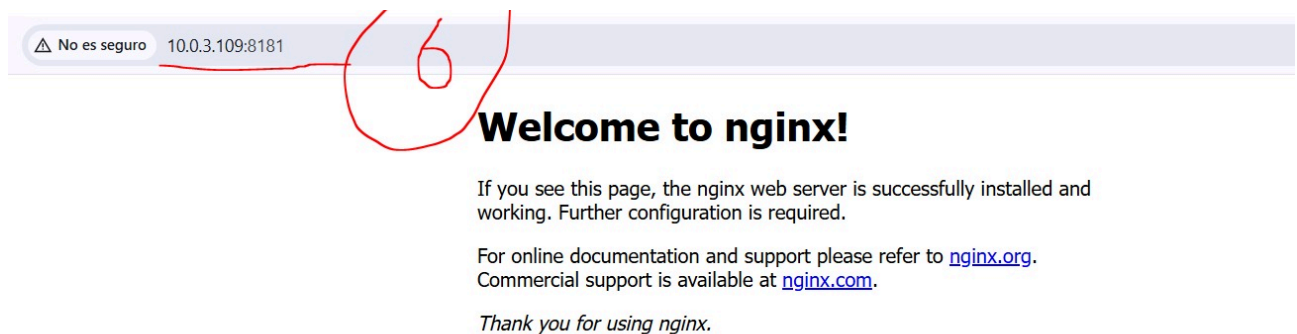
Una vez creado el contenedor y comprobado sus atributos, toca comprobar que tenemos acceso al servidor web.

```
osboxes@osboxes:~$ ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:b7ff:feea:b476 prefixlen 64 scopeid 0x20<link>
    ether 02:42:b7:ea:b4:76 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 4005 (4.0 KB)
    TX errors 0 dropped 23 overruns 0 carrier 0 collisions 0

veth-pair: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.109 netmask 255.255.0.0 broadcast 10.0.255.255
    inet6 fe80::60dc:a57e:7f12:6b59 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:89:18:65 txqueuelen 1000 (Ethernet)
    RX packets 12579 bytes 2181987 (2.1 MB)
    RX errors 0 dropped 83 overruns 0 frame 0
    TX packets 596 bytes 219319 (219.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**4)** Introducimos el comando “ifconfig” para mostrar el listado de direcciones IP. Nos interesa la que empieza por el número “10.0” (esto pertenece al tema de redes, así que por ahora no toca explicarlo). En la imagen está subrayada en rojo. (Lo consideramos como paso **5**)

**6)** Escribimos la dirección IP en el buscador web seguido de dos puntos y el puerto asignado a nuestro contenedor. En nuestro caso, es 10.0.3.109:8181. Podemos ver en pantalla que, en efecto, hemos podido conectarnos a la red.



## Eliminar el contenedor

Una vez sabemos que la creación ha sido exitosa y tenemos conexión, procederemos a eliminar el contenedor.

```
osboxes@osboxes:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    60c8a892f36f   7 weeks ago    192MB
osboxes@osboxes:~$ docker stop 94a7
94a7
osboxes@osboxes:~$ docker rm servidor_web
servidor_web
osboxes@osboxes:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
osboxes@osboxes:~$
```

- 7) Antes, escribimos el comando **docker images**. Este muestra las imágenes que tenemos. Ahora mismo solo tenemos nginx, en la que hemos creado el contenedor servidor\_web.
- 8) No podemos eliminar un contenedor arrancado sin detenerlo primero. Escribimos el comando **docker stop** seguido del nombre del contenedor o de los cuatro primeros dígitos de su ID. En este paso hemos utilizado el ID.
- 9) Ahora borramos el contenedor con **docker rm** seguido del nombre del contenedor o del ID. Para mostrar que ambas opciones son igualmente válidas, aquí he utilizado el nombre.
- 10) Para comprobar que ha sido eliminado con éxito, introducimos el comando **docker ps**. La lista que aparecerá deberá estar vacía, como en la imagen.