

Prácticas Docker

1. Docker logs

- Arrancamos un contenedor basado en la imagen NGINX en modo background

```
docker run -d --name nginx1 nginx  
a0796441c01d02aabb7cf9fff556129c1391589d7f961ff13979d830f7c885  
27
```

- Comprobamos que está funcionando

```
docker ps  
  
CONTAINER ID          IMAGE          COMMAND  
CREATED              STATUS        PORTS  
NAMES  
7ab2e60b7093         nginx         "nginx -g 'daemon  
of...' 22 seconds ago Up 21 seconds 80/tcp  
nginx1
```

- Comprobamos si hay algún log. Debe salir vacío

```
docker logs nginx1
```

- Nos conectamos al contenedor desde otro terminal

```
docker exec -it nginx1 bash  
root@7ab2e60b7093:/#
```

- Instalamos el comando wget en el contenedor

```
apt-get update  
apt-get install wget
```

- Por supuesto, podemos entrar en una Shell si es necesario, indicando que es interactivo

```
docker exec -it nginx1 bash  
root@a0796441c01d:/# ls  
bin    dev    home  lib64  mnt    proc   run    srv    tmp    var  
boot  etc    lib   media  opt    root   sbin   sys    usr
```

- Lanzamos un "wget" contra el servidor nginx del contenedor

```
wget localhost
```

```
--2018-03-21 13:57:47-- http://localhost/
Resolving localhost (localhost)... 127.0.0.1, ::1
Connecting to localhost (localhost)|127.0.0.1|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 612 [text/html]
Saving to: 'index.html'

index.html
100%[=====
=====>]      612  --.-KB/s    in 0s

2018-03-21 13:57:47 (1.85 MB/s) - 'index.html' saved
[612/612]
```

- Comprobamos que tenemos ya salida en el log

```
docker logs nginx1

127.0.0.1 - - [21/Mar/2018:13:57:47 +0000] "GET /
HTTP/1.1" 200 612 "-" "Wget/1.18 (linux-gnu)" "-"
```

- Comprobamos con el comando Docker top los procesos que se están ejecutando en el contenedor
- Podemos ver que está el nginx y además la bash con la que hemos entrado desde Docker exec

```
docker top nginx1
```

UID	PID	PPID	C	STIME	TTY	TIME
CMD						
root	7892	7877	0	17:29	?	00:00:00
nginx: master process nginx -g daemon off;						
101	7941	7892	0	17:29	?	00:00:00
nginx: worker process						
root	7962	7877	0	17:29	pts/0	00:00:00
bash						

- Vamos ahora a lanzar un sleep desde la bash.

```
sleep 500
```

- Comprobamos el top de nuevo

```
docker top nginx1
```

UID	PID	PPID	C	STIME	TTY	TIME
CMD						

root	7892	7877	0	mar21	?	00:00:00
nginx: master process nginx -g daemon off;						
101	7941	7892	0	mar21	?	00:00:00
nginx: worker process						
root	7962	7877	0	mar21	pts/0	00:00:00
bash						
root	10686	7962	0	00:04	pts/0	00:00:00
sleep 500						

- Vemos ahora las estadísticas de uso del contenedor

nocker stats nginx1						
CONTAINER ID		NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET
I/O	BLOCK I/O	PIDS				
7ab2e60b7093	nginx1	0.00%	1.918MiB / 3.859GiB	0.05%	23.4kB /	
0B	25.9MB / 0B	4				

- Cortamos el sleep
- Generamos un fichero grande dentro del contenedor

```
dd if=/dev/zero of=f1.dat bs=1024 count=100000000
```

- Accedemos de nuevo a las estadísticas y vemos el incremento en las mismas

CONTAINER ID	NAME	CPU %	MEM USAGE
/ LIMIT	MEM %	NET I/O	BLOCK I/O
PIDS			
7ab2e60b7093	nginx1	99.62%	2.035MiB /
3.859GiB	0.05%	26.8kB / 0B	26MB / 0B
4			

- Por último, matamos el contenedor en vez de pararlo

```
docker kill nginx1

nginx1

docker ps

CONTAINER ID          IMAGE                  COMMAND                  CREATED
STATUS                PORTS                 NAMES

docker ps -a

CONTAINER ID          IMAGE                  COMMAND                  PORTS
CREATED              STATUS                NAMES
7ab2e60b7093         nginx                 "nginx -g 'daemon of..."  9
hours ago            Exited (137) 8 seconds ago
nginx1
```

-