



PUESTA EN PRODUCCIÓN SEGURA

Unidad 3. Actividad 4



14 DE DICIEMBRE DE 2023
CARLOS DÍAZ MONTES
ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

Enunciado.....	2
Ejercicio.....	2

Enunciado

Actividad. Siguiendo los pasos anteriores instale un servidor Apache en un Docker y redireccionar las comunicaciones de la maquina origen del puerto 8080 al puerto 80 del docker. Capture todos los pasos.

Ejercicio

Me he creado un contenedor llamado web:

```
pps@localhost:~/dockerweb> sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
web                  latest             3b911fef0092        9 seconds ago      233MB
alpine               latest             f8c20f8bbcb6        6 days ago         7.38MB
hello-world          latest             9c7a54a9a43c        7 months ago       13.3kB
pps@localhost:~/dockerweb>
```

Nos creamos un fichero Dockerfile donde guardaremos lo siguiente:

```
GNU nano 7.2 Dockerfile
FROM ubuntu:latest
MAINTAINER CARLOS cdiazmaguadulce@gmail.com
RUN apt-get update
RUN apt-get -y install apache2
EXPOSE 80
CMD /usr/sbin/apache2ctl -D FOREGROUND
```

Ahora usamos build -t y el nombre que queremos del contenedor:

```
pps@localhost:~/dockerweb> nano Dockerfile
pps@localhost:~/dockerweb> sudo nano Dockerfile
pps@localhost:~/dockerweb> sudo docker build -t web ~/dockerweb/
[+] Building 70.8s (7/7) FINISHED
=> [internal] load build definition from Dockerfile                                0.2s
=> => transferring dockerfile: 201B                                              0.1s
=> [internal] load .dockerignore                                                  0.2s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest                 3.4s
=> [1/3] FROM docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af9362  7.3s
=> => resolve docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af9362  0.1s
=> => sha256:5e8117c0bd28aecad06f7e76d4d3b64734d59c1a0a44541d18060cd8fba30c50 29.55MB / 29.55MB  3.2s
=> => sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936225685bb 1.13kB / 1.13kB  0.0s
```

Seleccionamos los puertos:

```
pps@localhost:~/dockerweb> sudo docker run -d -p 8085:80 web
ef6372425573ec334ebd3a6bc0a61a3a31994b38580f2b3317901dd0741adf42
```

Y comprobamos que funciona:

