

title:Crear una imagen con dockerfile
authors: Jairo Zenteno and Daniel Vazquez

Ejercicio- Crear una imagen con DockerFile

Ejercicio- Crear una imagen con DockerFile

Bloque de código DockerFile.
Creamos la imagen
Creamos un repositorio en docker-hub
Subimos nuestra imagen a docker hub

Bloque de código DockerFile.

```
FROM ubuntu:[version]
MAINTAINER jairo jairo@correo.com #Creador de la imagen
ENV DEBIAN_FRONTEND noninteractive
RUN apt-get update
RUN apt-get -y install apache2 #Establecemos que no sea interactiva e instalamos el servidor web apache2
COPY ./public_html /var/www/html
expose 81 #expuesto en el puerto 81
CMD /usr/sbin/apache2ctl -D FOREGROUND #ejecuta el servicio
```

```
jairo@jairo-linux:~/docker-apache$ cat Dockerfile
FROM ubuntu:20.04
MAINTAINER jairo jairo@jairo.com
ENV DEBIAN_FRONTEND noninteractive
RUN apt-get update
RUN apt-get -y install apache2
COPY ./public_html /var/www/html
expose 82
CMD /usr/sbin/apache2ctl -D FOREGROUND
```

La plantilla que descargamos esta en el directorio en el que se encuentra el archivo Dockerfile

Con la variable de entorno `COPY ./public_html /var/www/html` exportamos nuestra plantilla al document_root de apache2.

Creamos la imagen

Una vez configurado el dockerfile, creamos nuestra imagen con apache2, usando los comandos:

```
docker build -t [nombre_de_la_imagen]web .
#IMPORTANTE NOS TENEMOS QUE SITUAR EN EL DIRECTORIO DONDE TENEMOS EL DOCKERFILE O EN CASO DE QUERER EJECUTARLO DESDE OTRO DIRECTORIO TENEMOS QUE PONER LA RUTA COMPLETA DE DONDE SE ENCUENTRA EL DOCKER FILE, ejemplo
#docker build -t web /home/jairo/docker-apache
```

```
root@jairo-linux:/home/jairo/docker-apache# docker build -t miweb .
Sending build context to Docker daemon 3.072kB
Step 1/7 : FROM ubuntu:20.04
--> d13c942271d6
Step 2/7 : MAINTAINER jairo jairo@correo.com
--> Using cache
--> b3e42bc88be2
Step 3/7 : ENV DEBIAN_FRONTEND noninteractive
--> Using cache
--> 52d83ec5f762
Step 4/7 : RUN apt-get update
--> Using cache
--> 9b8837f703cb
Step 5/7 : RUN apt-get -y install apache2
--> Using cache
--> bdf332867163
```

Comprobamos que se ha creado la imagen

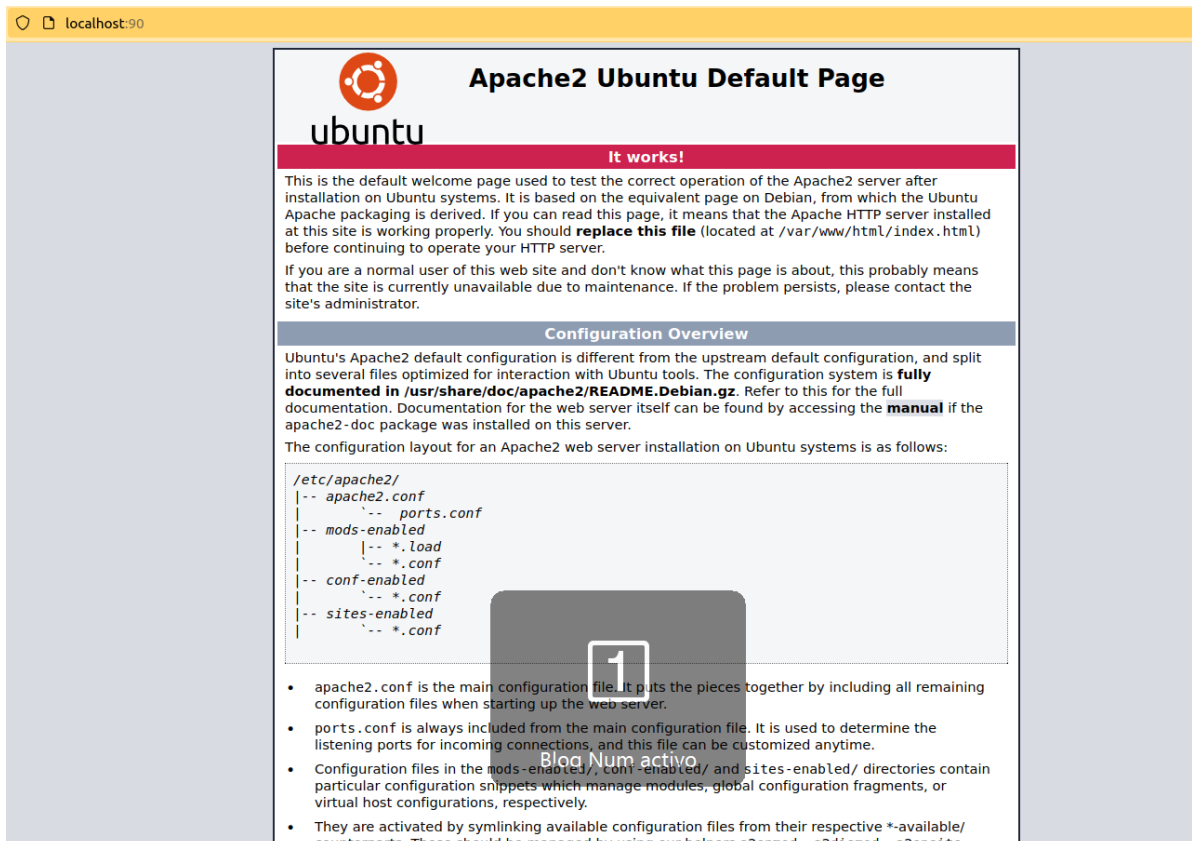
```
docker images
```

```
root@jairo-linux:/home/jairo/docker-apache# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
bl4ckr0se/apache2   v2                 df8591f532db       3 hours ago       219MB
bl4ckr0se/web        latest             df8591f532db       3 hours ago       219MB
miweb                latest             df8591f532db       3 hours ago       219MB
web                  latest             df8591f532db       3 hours ago       219MB
mariaadb             latest             d462573d8688       3 weeks ago       410MB
```

Creamos el contenedor a partir de la imagen creada.

```
docker run -d -p 90:80 web #web nombre de la imagen creada
```

Comprobamos que esta funcionando en el navegador accediendo al <http://localhost:90>



Para poder importar nuestra plantilla html al servidor hice uso de los comandos:

```
docker run -d -p 90:80 -v  
/home/jairo/Documentos/DAW2/Despliegue/univers:/var/www/html --name pagweb web
```

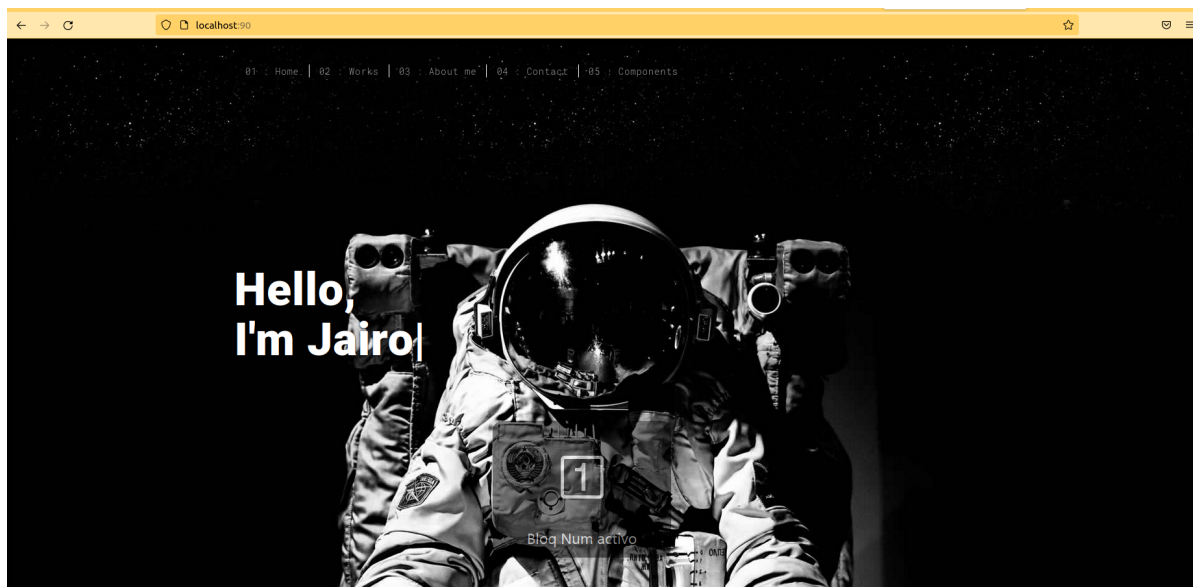
Tuve que borrar el contenedor anterior.

Con el parámetro `-v`, estamos compartiendo los ficheros que tenemos en nuestro localhost en el directorio `/home/jairo/Documentos/DAW2/Despliegue/univers` a nuestro servidor web en el directorio `:/var/www/html`

Comprobamos que ha funcionado.

```
root@jairo-linux:/home/jairo/docker-apache# docker exec -it pagweb bash  
root@06d840138d27:/# cd /var/www/html/  
root@06d840138d27:/var/www/html# ls  
about.html  components.html  index.html  main.3f6952e4.css.gz  main.70a66962.js.gz  sample  works.html  
assets      contact.html      main.3f6952e4.css  main.70a66962.js     main.70a66962.map  work.html  
root@06d840138d27:/var/www/html#
```

Accedemos a localhost:90 y vemos que tenemos nuestra plantilla.



Creamos un repositorio en docker-hub




bl4ckr0se/web

Imagen del servidor apache

Last pushed: never

Tags and Scans
VULNERABILITY SCANNING - **DISABLED** [Enable](#)

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest		2 minutes ago	2 minutes ago

[See all](#)

Subimos nuestra imagen a docker hub

1. Para poder subir nuestra imagen a docker-hub tenemos que iniciar sesion desde el cli con los comandos:

```
docker login
#o también lo podemos hacer con
docker login -u [user_de_docker-hub]
```

```
root@jairo-linux:/home/jairo# docker login
Username: bl4ckr0se
Password:
```

2. Ahora tenemos que hacer un tag de nuestra imagen

```
#docker tag imageId [usuario_docker-hub]/[repositorio de docker images]:
[tag] es alternativo pero aquí ira el tag de la imagen en nuestro caso seria
"latest"
docker tag df8591f532db bl4ckr0se/web
```

```
root@jairo-linux:/home/jairo# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
web            latest    df8591f532db   3 hours ago    219MB
mariadb        latest    d462573d8688   3 weeks ago    410MB
ubuntu        20.04     d13c942271d6   3 weeks ago    72.8MB
ubuntu        latest    d13c942271d6   3 weeks ago    72.8MB
phpmyadmin     latest    e0c502901f60   5 weeks ago    529MB
php            7.4-apache 0a4f19d60710   5 weeks ago    469MB
adminer        latest    e7ff5f92e509   6 weeks ago    88.9MB
fedora         latest    b78af7a83692   2 months ago   153MB
root@jairo-linux:/home/jairo# docker tag df8591f532db bl4ckr0se/web
```

3. Subimos la imagen a docker-hub con los comandos:

```
docker push bl4ckr0se/web
#donde bl4ckr0se/web es el repositorio previamente creado desde la GUI
```

```
root@jairo-linux:/home/jairo# docker push bl4ckr0se/web
Using default tag: latest
The push refers to repository [docker.io/bl4ckr0se/web]
4dc3844dc99e: Pushed
6016509d17ed: Pushed
0eba131dffd0: Pushed
latest: digest: sha256:c2b51fddf184365103e438de4795e940f2a3c720f6480903cb551f86adfce9ee size: 953
```

4. Vemos en la GUI que nos subió correctamente la imagen a docker

The screenshot shows the Docker Hub interface for the repository `bl4ckr0se/web`. The 'Tags' tab is selected, displaying a list of tags. The 'latest' tag is the only one shown, pushed 'an hour ago' by `bl4ckr0se`. The tag details include a digest of `c2b51fddf184`, OS/ARCH of `linux/amd64`, and a compressed size of `84.66 MB`. A 'docker pull bl4ckr0se/web:latest' command is shown with a copy icon. The interface also includes a search bar, navigation links, and a 'Delete' button for the tag list.

TAG	DIGEST	OS/ARCH	LAST PULL	COMPRESSED SIZE
latest	c2b51fddf184	linux/amd64	an hour ago	84.66 MB