

Rubén López Vega

Módulo: Despliegue de aplicaciones web

Práctica nº 6: Docker. Introducción y almacenamiento.

2º Curso DA2D1E

Instalamos la versión de docker desktop para windows y comprobamos que se ha realizado correctamente.

```
Windows PowerShell
PS C:\Users\rivega\Desktop\docker Clase> docker -v
Docker version 19.03.5, build 633a0ea
PS C:\Users\rivega\Desktop\docker Clase>
```

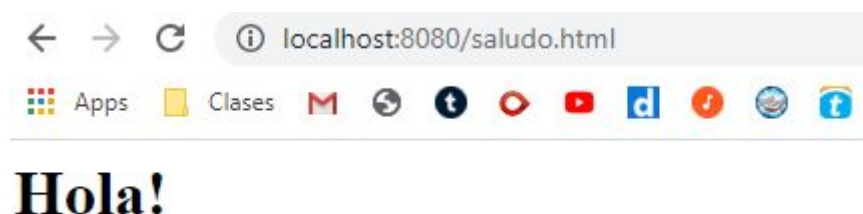
Descargamos la imagen de Apache con el comando **docker pull httpd**.
Generamos un fichero **saludo.html** y el **Dockerfile** y construimos la imagen con **docker build -t [nombre] [ruta Dockerfile]**

```
PS C:\Users\rivega\Desktop\docker Clase> docker build -t apache-server ./
Sending build context to Docker daemon 3.584kB
Step 1/2 : FROM httpd:2.4
2.4: Pulling from library/httpd
Digest: sha256:b783a610e75380aa152dd855a18368ea2f3becb5129d0541e2ec8b662cbd8afb
Status: Downloaded newer image for httpd:2.4
--> c562eeace183
Step 2/2 : COPY ./htdocs/ /usr/local/apache2/htdocs/
--> 0558e63cda0c
Successfully built 0558e63cda0c
Successfully tagged apache-server:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host.
All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recom
mended to double check and reset permissions for sensitive files and directories.
PS C:\Users\rivega\Desktop\docker Clase>
```

Iniciamos nuestro contenedor para que escuche al puerto 8080

```
PS C:\Users\rivega\Desktop\docker Clase> docker run -dit --name my-running-app -p 8080:80 apac
he-server
b7e9080a7e4577403de77d72381f8c417f0081dc26650e5f31c45764bb49ac14
PS C:\Users\rivega\Desktop\docker Clase>
```

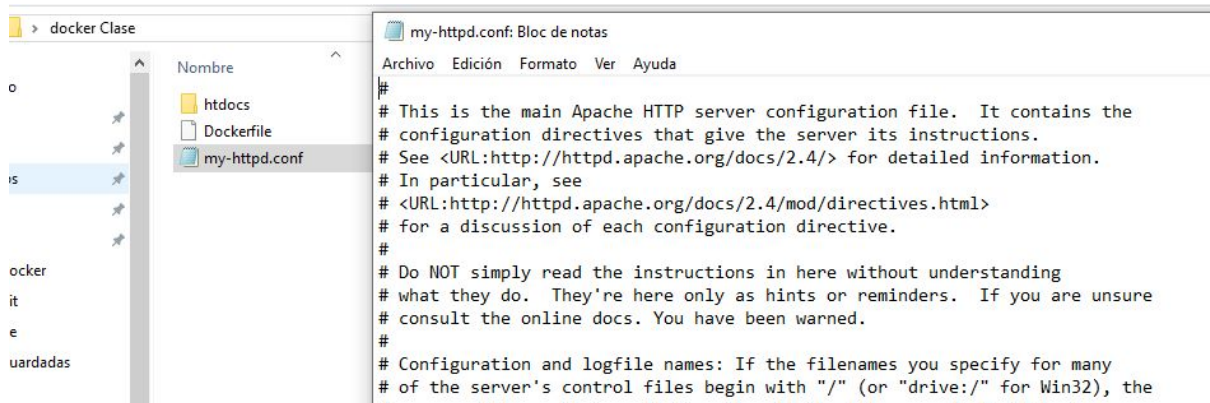
Y comprobamos que responde con nuestro fichero **saludo.html**



También podemos montar la imagen sin Dockerfile con **docker run -dit --name my-apache-app2 -p 8080:80 -v /usr/local/apache2/htdocs/ httpd:2.4**

```
PS C:\Users\rivega\Desktop\docker Clase> docker run -dit --name my-apache-app2 -p 8080:80 -v /usr/local/apache2/htdocs/ httpd:2.4
b66fedbdadde07766bd9398c7ac79a7eeef6f34ccc27accece91c10aed0bc50
PS C:\Users\rivega\Desktop\docker Clase>
```

Para modificar el fichero de configuración **httpd.conf**, generamos primero una copia con **docker run --rm httpd:2.4 cat /usr/local/apache2/conf/httpd.conf > my-httpd.conf**



Modificamos el fichero y para volver a incluirlo, deberemos sustituirlo en el fichero Dockerfile añadiendo la línea **COPY ./my-httpd.conf /usr/local/apache2/conf/httpd.conf** o con el comando **cp ./my-httpd.conf my-apache-app2:/usr/local/apache2/conf/httpd.conf**

Ahora vamos a generar un volumen para poner una página web en él y comprobamos donde se almacenan los datos de dicho volumen.

```
PS C:\Users\r\vega\Desktop\docker Clase> docker volume create --name volumen_test
volumen_test
PS C:\Users\r\vega\Desktop\docker Clase> docker volume inspect volumen_test
[
  {
    "CreatedAt": "2020-02-25T15:34:43Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/volumen_test/_data",
    "Name": "volumen_test",
    "Options": {},
    "Scope": "local"
  }
]
PS C:\Users\r\vega\Desktop\docker Clase>
```

Como desde windows no tenemos un acceso directo al volumen, asique lo vinculamos a un contenedor para poder introducir en él los datos que queramos. Copiaremos el contenido de nuestra carpeta htdocs a la carpeta donde se ha montado el volumen.

```
PS C:\Users\r\vega\Desktop\docker Clase> docker run -dit --name my-apache-app2 -p 8080:80 -v volumen_test:/data httpd:2.4
98371854c92f9e6dfa211f230c257d6d7295579b4e8c577d7192665f2d0d2408
PS C:\Users\r\vega\Desktop\docker Clase> docker cp ./htdocs/. my-apache-app2:/data
```

Comprobamos que en dicha carpeta se ha subido el contenido de nuestro htdocs, el fichero **saludo.html**.

Para ello entramos en la consola del contenedor y navegamos hasta **/data**.

```
PS C:\Users\r1vega\Desktop\docker Clase> docker exec -it my-apache-app2 /bin/bash
root@98371854c92f:/usr/local/apache2# cd ..
root@98371854c92f:/usr/local# cd ..
root@98371854c92f:/usr# cd ..
root@98371854c92f:/# ls
bin boot data dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@98371854c92f:/# cd data
root@98371854c92f:/data# ls
bash chown dd dnsdomainname false gunzip ln mkdir mount pidof rm sed su touch
cat cp df domainname fgrep gzexe login mknod mountpoint pwd rmdir sh sync true
chgrp dash dir echo findmnt gzip ls mktemp mv rbash run-parts sleep tar umount
chmod date dmesg egrep grep hostname lsblk more nisdomainname readlink saludo.html stty tempfile
root@98371854c92f:/data# exit
```

Ahora, esta información está en el volumen, que es independiente del contenedor. Para comprobar esto, vamos a montar otro contenedor de la misma manera que el anterior y con el mismo volumen.

```
PS C:\Users\r1vega\Desktop\docker Clase> docker run -dit --name my-apache-app4 -p 8085:80 -v volume_test:/data httpd:2.4
cc50312c517cf4dc384ad88bff3c0481a67831c591a049132114b3834ac2f06b
```

Ahora al ir a la ruta donde hemos montado el contenedor, este deberá contener la misma información que el anterior.

```
PS C:\Users\r1vega\Desktop\docker Clase> docker exec -it my-apache-app4 /bin/bash
root@cc50312c517c:/usr/local/apache2# cd ..
root@cc50312c517c:/usr/local# cd ..
root@cc50312c517c:/usr# cd ..
root@cc50312c517c:/# ls
bin boot data dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@cc50312c517c:/# cd data
root@cc50312c517c:/data# ls
bash chown dd dnsdomainname false gunzip ln mkdir mount pidof rm sed su
cat cp df domainname fgrep gzexe login mknod mountpoint pwd rmdir sh sync
chgrp dash dir echo findmnt gzip ls mktemp mv rbash run-parts sleep tar umount
chmod date dmesg egrep grep hostname lsblk more nisdomainname readlink saludo.html stty tempfile
root@cc50312c517c:/data#
```