



 **https://**



MEMORIA DE PRÁCTICAS SERVIDOR WEB (HTTPS)

DESPLIEGUE DE APLICACIONES WEB

Sandra Ruiz Jiménez
2º DAW

Tabla de contenido

1.INTRODUCCIÓN	1
2.CREACIÓN DE BIND MOUNT	1
3.BÚSQUEDA EN DOCKER HUB	2
4.INSTALANDO EL CONTENEDOR APACHE.....	2
5.CONFIGURACIÓN DEL CONTENEDOR.....	3
5.1. ACTUALIZACIÓN DEL SERVIDOR	3
5.2. INSTALACIÓN SSL	4
6.COMPROBACIÓN EN EL NAVEGADOR.....	6

1.INTRODUCCIÓN

En esta práctica vamos a configurar un contenedor Docker para que se pueda acceder a una página web con el protocolo https.

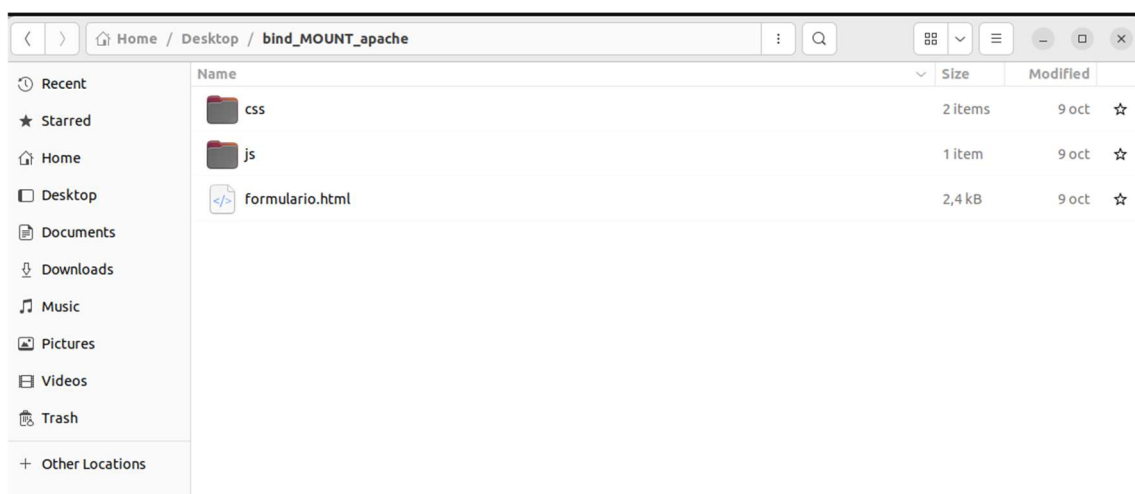
Además, el contenedor docker tendrá que estar basado en la imagen httpd de Docker Hub y usar un bind mount para publicar la página web.

2.CREACIÓN DE BIND MOUNT

Para esta práctica, elegimos conseguir la persistencia de los datos de los contenedores usando bind mount. De este modo, lo que estamos haciendo es "mapear" una parte de nuestro sistema de ficheros, de la que normalmente tenemos el control, con una parte del sistema de ficheros del contenedor.

Con el uso de bind mount en lugar de volúmenes, otras aplicaciones que no sean docker podrán acceder a los ficheros. Esto podría considerarse algo negativo, pero precisamente los bind mounts son el mecanismo que vamos a preferir para la fase de desarrollo, ya que las aplicaciones que accederán a esos ficheros serán los IDEs que usemos para modificar el código y, así, desde nuestro propio equipo podremos probar el código sin instalar nada en nuestro sistema y modificaremos los archivos que tenemos en local, que a la misma vez están en el contenedor.

Para ello, vamos a crear una carpeta en la que alojaremos la página web que queremos montar. Esta carpeta será la que usaremos como ruta de origen para el bind mount de nuestro contenedor Apache, cuando llegue el momento de lanzarlo.



3.BÚSQUEDA EN DOCKER HUB

Entramos en Docker Hub y buscamos el servidor Apache.

https://hub.docker.com/_/httpd

Tenemos la opción de descargarnos la imagen previamente con el comando **docker pull httpd** o, por el contrario, que será lo que haremos en esta práctica, usar el comando **docker run**, por lo que no será necesario descargar la imagen previamente, ya que si esta no se encuentra en el sistema, la busca y descarga la última versión disponible.

4.INSTALANDO EL CONTENEDOR APACHE

Para instalar el contenedor debemos tener en cuenta algunos parámetros que deberemos usar en su instalación, por ejemplo:

- El mapeado de los puertos
Queremos mapear el puerto 80, que es por el que escucha Apache. Como el objetivo de esta práctica es mostrarla por https, tendremos que mapear el puerto 443.
- El uso de bind mount
Como ya se ha dicho anteriormente, usaremos un bind mount en el contenedor, por lo que debemos usar el parámetro:

--mount type=bind,src=ruta_origen_local,dst=ruta_contenedor_apache

La ruta de destino será la carpeta htdocs del contenedor Apache.

Por lo tanto, el comando que usaremos quedaría de la siguiente forma:

```
docker run --name apacheHTTPS -p 80:80 -p 443:443 --mount  
type=bind,src=/home/sandra/Desktop/bind_MOUNT_apache,dst=/usr/local/apache  
2/htdocs httpd
```

```
sandra@sandra-UX410UAK:~$ docker run --name apacheHTTPS -p 80:80 -p 443:443 --mount  
type=bind,src=/home/sandra/Desktop/bind_MOUNT_apache,dst=/usr/local/apache2/  
htdocs httpd  
AH00558: httpd: Could not reliably determine the server's fully qualified domain  
name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress thi  
s message  
AH00558: httpd: Could not reliably determine the server's fully qualified domain  
name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress thi  
s message  
[Tue Oct 25 14:37:53.630861 2022] [mpm_event:notice] [pid 1:tid 140211898568000]  
AH00489: Apache/2.4.54 (Unix) configured -- resuming normal operations  
[Tue Oct 25 14:37:53.635560 2022] [core:notice] [pid 1:tid 140211898568000] AH00  
094: Command line: 'httpd -D FOREGROUND'
```

Usando el comando anterior, el contenedor se debe haber creado y levantado. Para comprobar que es así, usamos el comando **docker ps**, que muestra los contenedores activos en este momento.

```
sandra@sandra-UX410UAK:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
4cf6a35e1972   httpd     "httpd-foreground"      43 seconds ago Up 41 seconds 0
.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp
HTTPS
sandra@sandra-UX410UAK:~$
```

5.CONFIGURACIÓN DEL CONTENEDOR

Accedemos al contenedor porque vamos a trabajar en la consola del servidor. Para ello necesitamos que el contenedor esté levantado, por lo que ejecutamos el comando **docker exec -it apacheHTTPS /bin/bash**

```
sandra@sandra-UX410UAK:~$ docker exec -it apacheHTTPS /bin/bash
root@4cf6a35e1972:/usr/local/apache2#
```

Una vez que tenemos el terminal podemos trabajar desde el propio sistema.

5.1. ACTUALIZACIÓN DEL SERVIDOR

El siguiente paso es actualizarlo, para lo que usamos **apt update**

```
root@4cf6a35e1972:/usr/local/apache2# apt update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8184 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [193 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.6 kB]
Fetched 8599 kB in 2s (3593 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@4cf6a35e1972:/usr/local/apache2#
```

Como más tarde tendremos que modificar algún archivo, instalamos el comando nano, que no se encuentra en el servidor, con el comando **apt install nano**


```

root@4cf6a35e1972:/usr/local/apache2# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libncursesw6
Suggested packages:
  gpm hunspell
The following NEW packages will be installed:
  libgpm2 libncursesw6 nano
0 upgraded, 3 newly installed, 0 to remove and 4 not upgraded.
Need to get 824 kB of archives.
After this operation, 3087 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main amd64 libncursesw6 amd64 6.2+20201114-2 [132 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 nano amd64 5.4-2+deb11u1 [656 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 libgpm2 amd64 1.20.7-8 [35.6 kB]
Fetched 824 kB in 0s (3757 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libncursesw6:amd64.
(Reading database ... 7140 files and directories currently installed.)
Preparing to unpack .../libncursesw6_6.2+20201114-2_amd64.deb ...
Unpacking libncursesw6:amd64 (6.2+20201114-2) ...
Selecting previously unselected package nano.
Preparing to unpack .../nano_5.4-2+deb11u1_amd64.deb ...
Unpacking nano (5.4-2+deb11u1) ...
Selecting previously unselected package libgpm2:amd64.
Preparing to unpack .../libgpm2_1.20.7-8_amd64.deb ...
Unpacking libgpm2:amd64 (1.20.7-8) ...
Setting up libgpm2:amd64 (1.20.7-8) ...
Setting up libncursesw6:amd64 (6.2+20201114-2) ...
Setting up nano (5.4-2+deb11u1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group editor) does n't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group pico) doesn't exist
Processing triggers for libc-bin (2.31-13+deb11u4) ...
root@4cf6a35e1972:/usr/local/apache2#

```

5.2. INSTALACIÓN SSL

Tenemos que entrar en el fichero de configuración **/usr/local/apache2/conf/httpd.conf** y descomentar las siguientes líneas de código:

```

...
#LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
...
#LoadModule ssl_module modules/mod_ssl.so
...
#Include conf/extra/httpd-ssl.conf
...

```

Para ello escribimos `nano conf/httpd.conf` y descomentamos las líneas de arriba.

```
root@4cf6a35e1972:/usr/local/apache2# nano conf/httpd.conf
root@4cf6a35e1972:/usr/local/apache2#
```

A continuación, reiniciamos el contenedor para que los cambios realizados se actualicen, con el comando **`apachectl restart`**.

```
root@4cf6a35e1972:/usr/local/apache2# apachectl restart
AH00526: Syntax error on line 144 of /usr/local/apache2/conf/extra/httpd-ssl.conf:
SSLCertificateFile: file '/usr/local/apache2/conf/server.crt' does not exist or is
empty
root@4cf6a35e1972:/usr/local/apache2#
```

Vemos que nos aparece un error sobre el certificado SSL, que no existe o que está vacío. Si comprobamos el contenido de `/conf` con **`conf ls`**, vemos que no aparece ningún certificado porque aún no lo hemos instalado.

```
root@4cf6a35e1972:/usr/local/apache2# cd conf
root@4cf6a35e1972:/usr/local/apache2/conf# ls
extra httpd.conf magic mime.types original
root@4cf6a35e1972:/usr/local/apache2/conf#
```

Ahora instalaremos el certificado:

- **`apt install openssl`**

```
root@4cf6a35e1972:/usr/local/apache2/conf# apt install openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (1.1.1n-0+deb11u3).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
root@4cf6a35e1972:/usr/local/apache2/conf#
```

- **`openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout server.key -out server.crt`**

Nos pedirán algunos datos que debemos introducir.


```

root@4cf6a35e1972:/usr/local/apache2/conf# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout server.key -out server.crt
Generating a RSA private key
....+++++
.....+++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:SEVILLA
Locality Name (eg, city) []:CARMONA
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DESPLIEGUE
Organizational Unit Name (eg, section) []:DESPLIEGUE_WEB
Common Name (e.g. server FQDN or YOUR name) []:DESPLIEGUE
Email Address []:sandra.ruizjimenez86@gmail.com
root@4cf6a35e1972:/usr/local/apache2/conf#

```

- Si ahora comprobamos el contenido de /conf con **ls**, vemos que ya existen los ficheros **server.crt** y **server.key**.

```

root@4cf6a35e1972:/usr/local/apache2/conf# ls
extra httpd.conf magic mime.types original server.crt server.key
root@4cf6a35e1972:/usr/local/apache2/conf#

```

- Volvemos a reiniciar el contenedor para que se apliquen los cambios: **apachectl restart**

```

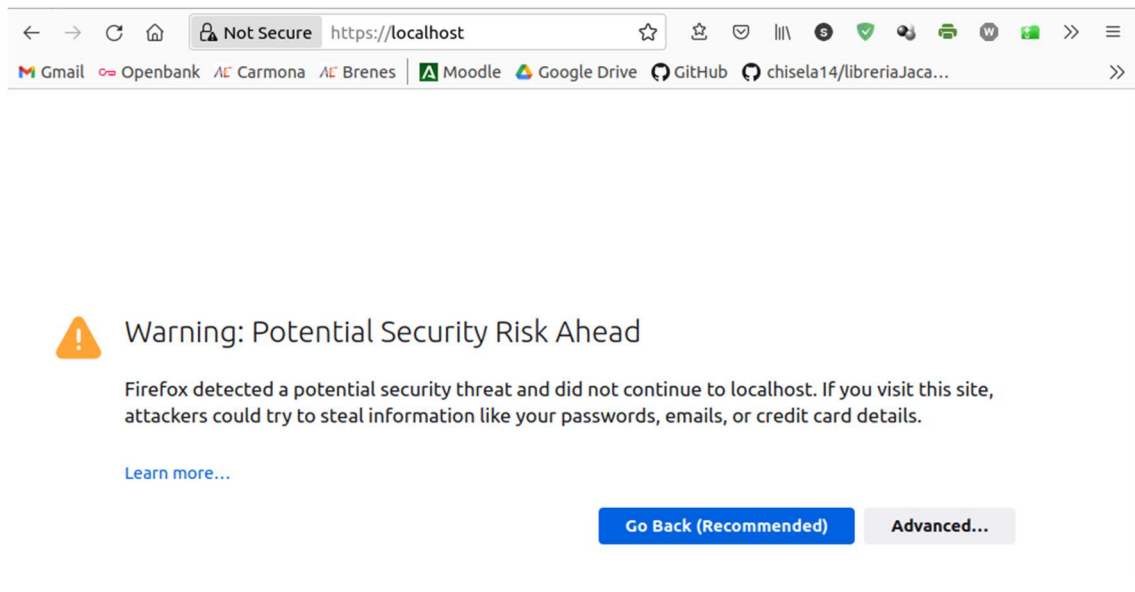
root@4cf6a35e1972:/usr/local/apache2/conf# apachectl restart
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
root@4cf6a35e1972:/usr/local/apache2/conf#

```

6.COMPROBACIÓN EN EL NAVEGADOR

Si escribimos en la barra del navegador **localhost/nombre_fichero**, tiene que aparecernos la web que se escucha a través del puerto 80 que es el del servidor Apache.

Ahora vamos a comprobar que también podemos visualizarla a través de https. Para ello escribiremos en la barra del navegador **https://localhost/nombre_fichero**
 Nos deben saltar unos avisos del navegador para informarnos de que el sitio al que queremos acceder no es seguro. Este aviso sale porque el certificado que tenemos es autogenerado.



Damos permiso para que el navegador acceda y debemos ver la página cargada por https.

A screenshot of a web browser window showing a 'Sign up' form. The address bar displays 'https://localhost/formulario.html'. The form is titled 'Sign up' and contains the following fields and elements: 'Nombre de usuario:' with a text input field; 'Email:' with a text input field; 'Sexo:' with two radio buttons labeled 'Hombre' and 'Mujer'; 'Edad:' with a dropdown menu; a checkbox labeled 'Acepto los términos y condiciones'; 'Contraseña:' with a text input field; 'Confirmar contraseña:' with a text input field; and a blue button labeled 'Enviar' at the bottom.