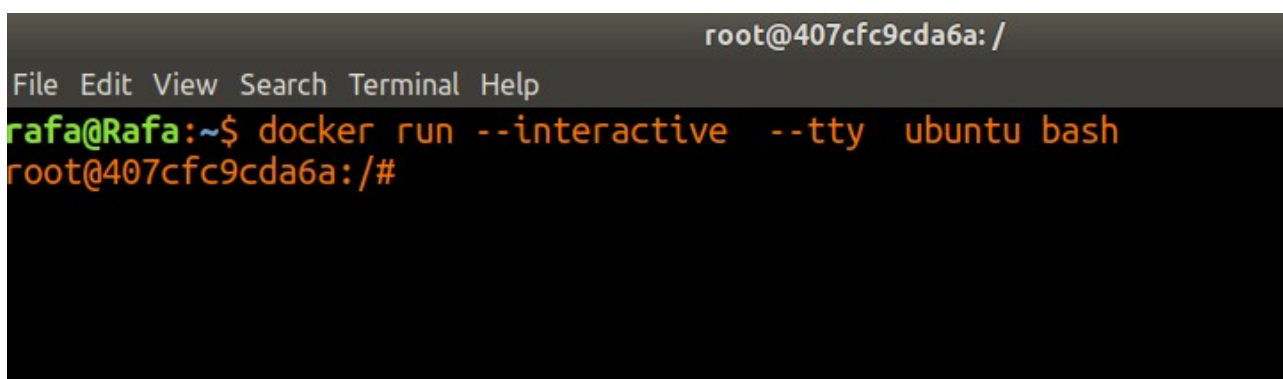


Memoria

Servidor Apache con HTTPS

En esta memoria aprenderemos a descargar, configurar y lanzar un contenedor de docker con ubuntu al que instalaremos el servidor de apache2 y crearemos un certificado autofirmado para poder usar el protocolo seguro https.

El primer paso es conseguir una imagen de docker con el sistema operativo ubuntu. Podemos conseguirlo con el siguiente comando que lo ejecutará de forma interactiva para que podamos acceder a el desde la terminal. He buscado para que sirva el atributo `-tty` y he entendido que es una especie de entrada salida desde la terminal (o shell), además al poner `bash` al final ejecutaremos el contenedor para empezar a trabajar con el directamente. Todos los comandos usados en esta memoria son de dos tutoriales accesibles al final de esta memoria.



```
root@407cfc9cda6a: /  
File Edit View Search Terminal Help  
rafa@Rafa:~$ docker run --interactive --tty ubuntu bash  
root@407cfc9cda6a:/#
```

Importante De ahora en adelante se llamara al contenedor desde el nombre ubuntu, aunque en esta captura se me olvidó poner el parametro `--name ubuntu` para que esto sea posible.

En esta ocasion se me olvidó hacer el bind mount, se haria introduciendo el parametro:

--mount type=bind,src=home/rafa/Desktop/volumenUbuntu/,dst=/var/www/html
Donde src es la carpeta de origen (carpeta local) y dst la carpeta de destino(Docker ubuntu)

Una vez tengamos nuestro contenedor, si se para o lo paramos, podemos arrancarlo con `docker start ubuntu`, y acceder a la terminal con `exec -it ubuntu /bin/bash` (para acceder de forma interactiva)

```
rafa@Rafa:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
rafa@Rafa:~$ docker start ubuntu
ubuntu
rafa@Rafa:~$ docker exec -it ubuntu
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
rafa@Rafa:~$ docker exec -it ubuntu /bin/bash
root@17a6c49943bd:/#
```

Este SO como es evidente viene con lo minimo, asi que lo primero que necesitamos es actualizar llos repositorios de las aplicaciones y servicios de ubuntu con el comando `apt update`.

```
root@17a6c49943bd:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 B]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [114 B]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 B]
```

Podemos instalar nano o otros editores de texto desde terminal. Nos hara falta mas adelante para modificar algunos archivos. Ademas tendremos que instalar tambien el servidor de Apache2.

```
root@17a6c49943bd:/# apt install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not
Need to get 269 kB of archives.
After this operation, 868 kB of additional disk space
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd
B]
Fetched 269 kB in 2s (130 kB/s)
debconf: delaying package configuration, since apt-u
Selecting previously unselected package nano.
(Reading database ... 4127 files and directories cur
Preparing to unpack .../nano_4.8-1ubuntu1_amd64.deb
Unpacking nano (4.8-1ubuntu1) ...
Setting up nano (4.8-1ubuntu1) ...
update-alternatives: using /bin/nano to provide /usr/
update-alternatives: warning: skip creation of /usr/
associated file /usr/share/man/man1/nano.1.gz (of l
update-alternatives: using /bin/nano to provide /usr/
update-alternatives: warning: skip creation of /usr/
associated file /usr/share/man/man1/nano.1.gz (of lin
root@17a6c49943bd:/# apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

Cuando instalemos apache nos hara algunas preguntas sobre nuestra zona horaria.

```
Please select the geographic area in which you live. Subsequent configuration
questions will narrow this down by presenting a list of cities, representing the ti
zones in which they are located.

 1. Africa      4. Australia  7. Atlantic  10. Pacific  13. Etc
 2. America    5. Arctic    8. Europe   11. SystemV
 3. Antarctica 6. Asia      9. Indian   12. US
Geographic area: 8

Please select the city or region corresponding to your time zone.

 1. Amsterdam  14. Copenhagen  27. London      40. Riga        53. Ulyanovsk
 2. Andorra    15. Dublin      28. Luxembourg  41. Rome        54. Uzhgorod
 3. Astrakhan  16. Gibraltar  29. Madrid      42. Samara      55. Vaduz
 4. Athens     17. Guernsey    30. Malta       43. San_Marino  56. Vatican
 5. Belfast    18. Helsinki    31. Mariehamn   44. Sarajevo    57. Vienna
 6. Belgrade   19. Isle_of_Man 32. Minsk       45. Saratov     58. Vilnius
 7. Berlin     20. Istanbul    33. Monaco      46. Simferopol  59. Volgograd
 8. Bratislava 21. Jersey      34. Moscow      47. Skopje      60. Warsaw
 9. Brussels   22. Kaliningrad 35. Nicosia     48. Sofia       61. Zagreb
10. Bucharest  23. Kiev        36. Oslo        49. Stockholm   62. Zaporozhye
11. Budapest   24. Kirov       37. Paris       50. Tallinn     63. Zurich
12. Busingen   25. Lisbon      38. Podgorica   51. Tirane
13. Chisinau   26. Ljubljana   39. Prague      52. Tiraspol
Time zone: 29
```

Una vez tengamos nuestro apache instalado vamos a necesitar tambien una herramienta llamada systemctl que sirve para gestionar servicios.

```
root@17a6c49943bd:/# apt install systemctl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be inst
  libmpdec2 libpython3-stdlib libpython3.8-mir
  python3 python3-minimal python3.8 python3.8-
Suggested packages:
```

Con esta herramienta podremos comprobar el estado, arrancar y parar apache2.

```
root@17a6c49943bd:/# systemctl status apache2
apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2
   Active: inactive (dead)
root@17a6c49943bd:/# systemctl start apache2
root@17a6c49943bd:/# systemctl status apache2
apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2
   Active: active (running)
root@17a6c49943bd:/#
```

A continuacion vamos a activar los modulos de ssl y rewrite, los utilizaremos mas tarde.

```
root@17a6c49943bd:/# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-si
gned certificates.
To activate the new configuration, you need to run:
  service apache2 restart
root@17a6c49943bd:/# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  service apache2 restart
root@17a6c49943bd:/# service apache2 restart
* Restarting Apache httpd web server: apache2
/usr/sbin/apachectl: 99: ulimit: error setting limit (Operation not permitted)
Setting ulimit failed. See README.Debian for more information.
AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[ OK ]
root@17a6c49943bd:/#
```

Con este ultimo comando reiniciaremos apache para recargar la nueva configuracion.

Ahora vamos a entrar a este archivo de configuracion usando nano:

```
root@17a6c49943bd: /# nano /etc/apache2/apache2.conf
```

Al final de este documento tendremos que escribir esta etiqueta, que sirve para direccionar nuestra pagina principal al contenido de esta carpeta.

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
<Directory /var/www/html>  
AllowOverride All  
</Directory>
```

A continuacion vamos a crearnos nuestro certificado autofirmado para poder usar el el protocolo seguro https. Lo primero es saber donde vamos a guardarlo, asi que vamos a crear una carpeta en esa misma ruta que aparece en el ejemplo, y una vez dentro ejecutamos el comando de creacion de la clave.

```

root@17a6c49943bd:/# mkdir /etc/apache2/certificate
root@17a6c49943bd:/# cd /etc/apache2/certificate
root@17a6c49943bd:/etc/apache2/certificate# openssl req -new -newkey rsa:4096 -x
sha256 -days 365 -nodes -out apache-certificate.crt -keyout apache.key
Generating a RSA private key
.....++++
.....++++
writing new private key to 'apache.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]:

```

A continuacion nos pedira unos datos que son totalmente irrelevantes.

```

Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:ES
Locality Name (eg, city) []:ES
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ES
Organizational Unit Name (eg, section) []:ES
Common Name (e.g. server FQDN or YOUR name) []:150.150.150.150
Email Address []:ES
root@17a6c49943bd:/etc/apache2/certificate# ls
apache-certificate.crt  apache.key
root@17a6c49943bd:/etc/apache2/certificate#

```

En el tutorial que leí me parecio entender que en Common name podia poner la ip que quisieras, pero parece que lo entendi mal ya que para acceder finalmente a la pagina desde https o http tuve que usar la ip de localhost, en mi caso 172.17.0.2.

Una vez creado, con ls comprobamos que los archivos estan en la carpeta que queremos.

Para mas informacion sobre los paremetros de creacion de la clave, encontré esta maravillosa [pagina](#).

Por último vamos a editar este fichero de configuración y a reiniciar de nuevo el servicio.

```
root@17a6c49943bd:/etc/apache2/certificate# nano /etc/apache2/sites-enabled/000-default.conf
root@17a6c49943bd:/etc/apache2/certificate# service apache2 restart
* Restarting Apache httpd web server apache2
/usr/sbin/apache2ctl: 99: ulimit: error setting limit (Operation not permitted)
Setting ulimit failed. See README.Debian for more information.
AH00558: apache2: 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
[ OK ]
root@17a6c49943bd:/etc/apache2/certificate#
```

Dentro de este archivo tendremos que escribir lo siguiente:

Aquí está el archivo, antes de nuestra configuración.

```
1 <VirtualHost *:80>
2     ServerAdmin webmaster@localhost
3     DocumentRoot /var/www/html
4     ErrorLog ${APACHE_LOG_DIR}/error.log
5     CustomLog ${APACHE_LOG_DIR}/access.log combined
6 </VirtualHost>
```

Aquí está el archivo, después de nuestra configuración.

```
1 <VirtualHost *:443>
2     ServerAdmin webmaster@localhost
3     DocumentRoot /var/www/html
4     ErrorLog ${APACHE_LOG_DIR}/error.log
5     CustomLog ${APACHE_LOG_DIR}/access.log combined
6     SSLEngine on
7     SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
8     SSLCertificateKeyFile /etc/apache2/certificate/apache.key
9 </VirtualHost>
```

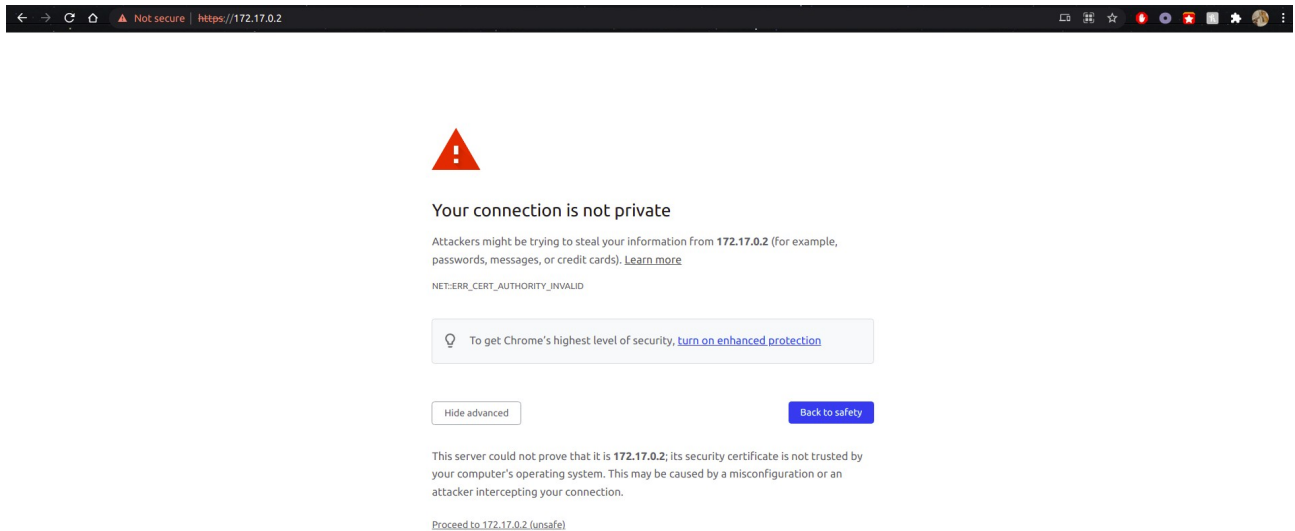
Opcionalmente, es posible que desee redirigir a los usuarios HTTP a la versión HTTPS de su sitio web.

En este caso, utilice la siguiente configuración.

```
1 <VirtualHost *:80>
2     RewriteEngine On
3     RewriteCond %{HTTPS} !=on
4     RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]
5 </VirtualHost>
6 <VirtualHost *:443>
7     ServerAdmin webmaster@localhost
8     DocumentRoot /var/www/html
9     ErrorLog ${APACHE_LOG_DIR}/error.log
10    CustomLog ${APACHE_LOG_DIR}/access.log combined
11    SSLEngine on
12    SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
13    SSLCertificateKeyFile /etc/apache2/certificate/apache.key
14 </VirtualHost>
```


Mi archivo de configuracion en este ejemplo es exactamente igual a la ultima captura, con redireccionamiento de http a https.

Una vez el servicio de apache2 este de nuevo levantado vamos a poner nuestro localhost . El localhost se puede averiguar poniendo Hostname -I en la terminal de ubuntu (**el del docker**).



En esta ultima captura podemos ver que accedemos correctamente desde el protocolo https, evidentemente al ser un certificado autofirmado google chrome nos avisa de que no es fiable pero como es nuestra pagina podemos continuar con normalidad y apareceria la pagina principal de Apache2.

Para guardar una imagen de nuestro contenedor solo tendremos que usar el comando docker commit (nombre/id) (nombre de la imagen). Como podemos ver en el siguiente ejemplo:

```
rafa@rafa:~$ docker commit ubuntu ubuntu-apache2
sha256:4fdbef2fa5230de91e6f7364e8c1e838a4ee5830d20683669758a32866203239
rafa@rafa:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu-apache2	latest	4fdbef2fa523	30 seconds ago	244MB

Bibliografia

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>

*Instalar Apache

<https://techexpert.tips/es/apache-es/habilitar-https-en-apache/>

*Activar https (certificado)

<https://www.openssl.org/docs/man1.0.2/man1/openssl-req.html>

*explicacion comando openssl