

Configuración de seguridad en Apache siguiendo el protocolo SSL/TLS - HTTPS

Si se quiere correr el tráfico web sobre SSL, la manera más sencilla es copiar (cp) o montar (mnt) creando un volumen (-v) tu server.crt (certificado) y server.key (llave) en /usr/local/apache2/conf/ y luego personalizar el archivo /usr/local/apache2/conf/httpd.conf descomentando las líneas necesarias.

El archivo conf/extra/httpd-ssl.conf usará los certificados añadidos y configurará el servicio Apache para escuchar en el puerto 443, tanto en el host como en el contenedor. Es importante asegurarse de haber añadido el parámetro -p 443:443 en el contenedor.

1. Crear estructura de carpetas en tu host

Estructura recomendada del proyecto:

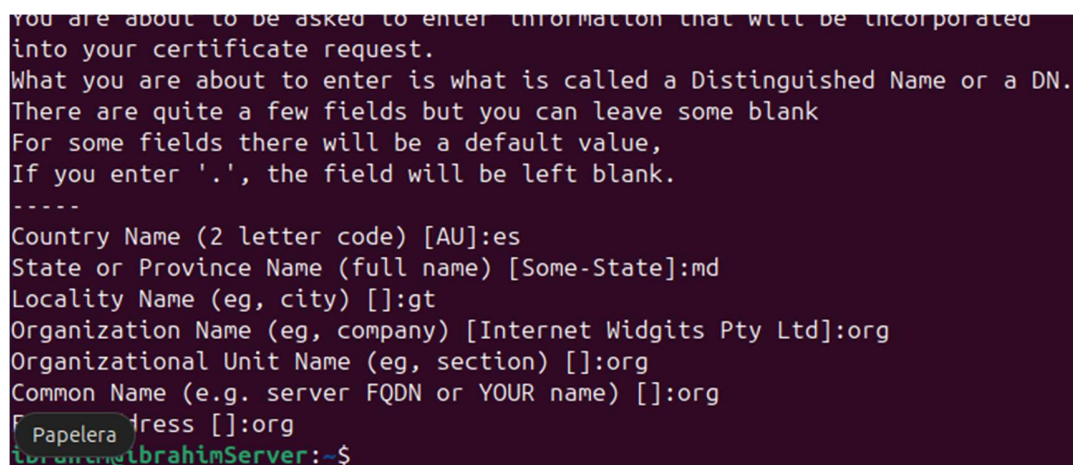
```
apache-ssl/
├── Dockerfile
├── certs/
│   ├── server.crt
│   └── server.key
├── html/
│   └── index.html
```

- certs/: Aquí pondremos el certificado SSL (.crt) y la clave privada (.key).

Para pruebas puedes generar un certificado autofirmado con el siguiente comando:

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

- html/: Contendrá tu sitio web (por ejemplo, un index.html simple).



```
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]:md
Locality Name (eg, city) []:gt
Organization Name (eg, company) [Internet Widgits Pty Ltd]:org
Organizational Unit Name (eg, section) []:org
Common Name (e.g. server FQDN or YOUR name) []:org
Email Address []:org
Papelera
ibrahimServer:~$
```

2. Crear el Dockerfile

El Dockerfile definirá la configuración del contenedor de Apache con soporte HTTPS.

```
FROM httpd:2.4
```

```
# Copiamos nuestro sitio web
```

```
COPY html/ /usr/local/apache2/htdocs/
```

```
# Activamos módulos SSL y rewrite
```

```
RUN apt-get update && apt-get install -y openssl \
```

```
&& sed -i '/LoadModule ssl_module/s/^#//g' /usr/local/apache2/conf/httpd.conf \
```

```
&& sed -i '/Include conf\extra\httpd-ssl.conf/s/^#//g' /usr/local/apache2/conf/httpd.conf
```

```
# Copiamos certificados al contenedor
```

```
COPY certs/server.crt /usr/local/apache2/conf/server.crt
```

```
COPY certs/server.key /usr/local/apache2/conf/server.key
```

```
# Configuración SSL
```

```
RUN sed -i 's|^SSLCertificateFile.*|SSLCertificateFile
```

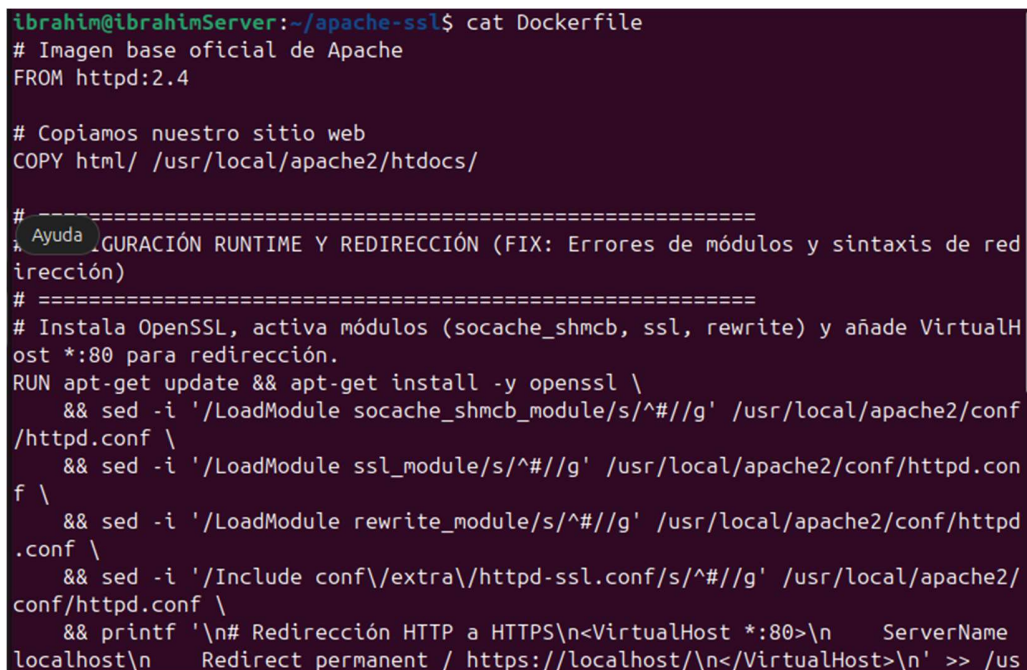
```
"/usr/local/apache2/conf/server.crt"|' /usr/local/apache2/conf/extra/httpd-ssl.conf \
```

```
&& sed -i 's|^SSLCertificateKeyFile.*|SSLCertificateKeyFile
```

```
"/usr/local/apache2/conf/server.key"|' /usr/local/apache2/conf/extra/httpd-ssl.conf
```

EXPOSE 443

```
CMD ["httpd-foreground"]
```



```
ibrahim@ibrahimServer:~/apache-ssl$ cat Dockerfile
# Imagen base oficial de Apache
FROM httpd:2.4

# Copiamos nuestro sitio web
COPY html/ /usr/local/apache2/htdocs/

# =====
# Configuración RUNTIME Y REDIRECCIÓN (FIX: Errores de módulos y sintaxis de red
# irección)
# =====
# Instala OpenSSL, activa módulos (socache_shmcb, ssl, rewrite) y añade VirtualH
ost *:80 para redirección.
RUN apt-get update && apt-get install -y openssl \
    && sed -i '/LoadModule socache_shmcb_module/s/^#//g' /usr/local/apache2/conf
/httpd.conf \
    && sed -i '/LoadModule ssl_module/s/^#//g' /usr/local/apache2/conf/httpd.con
f \
    && sed -i '/LoadModule rewrite_module/s/^#//g' /usr/local/apache2/conf/httpd
.conf \
    && sed -i '/Include conf\extra\httpd-ssl.conf/s/^#//g' /usr/local/apache2/
conf/httpd.conf \
    && printf '\n# Redirección HTTP a HTTPS\n<VirtualHost *:80>\n    ServerName
localhost\n    Redirect permanent / https://localhost/\n</VirtualHost>\n' >> /us
```

3. Construir y ejecutar el contenedor

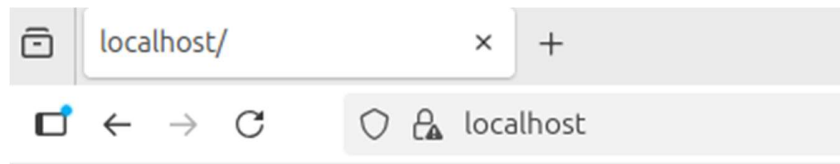
Ejecuta los siguientes comandos en tu terminal:

```
docker build -t apache-ssl .  
docker run -d -p 443:443 apache-ssl
```

De esta manera tendrás un contenedor de Apache operando con SSL. Despliega la página de localhost y verifica el comportamiento en HTTPS.

```
ibrahim@ibrahimServer:~/apache-ssl$ sudo docker build -t apache-ssl-final .  
[sudo] contraseña para ibrahim:  
Lo siento, pruebe otra vez.  
[sudo] contraseña para ibrahim:  
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.  
  
Install the buildx component to build images with BuildKit:  
https://docs.docker.com/go/buildx/  
  
Sending build context to Docker daemon 10.24kB  
Step 1/6 : FROM httpd:2.4  
--> 4613a77dcb46  
Step 2/6 : COPY html/ /usr/local/apache2/htdocs/  
--> 75dcaf47a8fd  
Step 3/6 : RUN apt-get update && apt-get install -y openssl && sed -i '/LoadModule socache_shmcb_module/s/^#//g' /usr/local/apache2/conf/httpd.conf && sed -i '/LoadModule ssl_module/s/^#//g' /usr/local/apache2/conf/httpd.conf && sed -i '/LoadModule rewrite_module/s/^#//g' /usr/local/apache2/conf/httpd.conf && sed -i '/Include conf/extra/httpd-ssl.conf/s/^#//g' /usr/local/apache2/  
ibrahim@ibrahimServer:~/apache-ssl$ sudo docker run -d -p 80:80 -p 443:443 -v "$PWD/certs":/usr/local/apache2/certs/ apache-ssl-final  
6b5fcce57fdce046e52690d73869eae773b31042416ad28752c70b3bb7c38ba7  
ibrahim@ibrahimServer:~/apache-ssl$
```





Sitio Seguro con HTTPS

Segunda parte del ejercicio

Crea un manual explicando los pasos realizados y añade las capturas de imagen correspondientes.

Además, investiga y documenta los siguientes puntos:

1. Cómo redirigir automáticamente de HTTP a HTTPS.
2. Uso de volúmenes para certificados, para poder usar certificados reales de Let's Encrypt o regenerarlos sin reconstruir la imagen.
3. Estructura limpia para entornos de producción o desarrollo.

LA SOLUCION PROPORCIONADA INCLUYE LA PRIMERA Y SEGUNDA PARTE