

Introducción a Docker

# UD 07. Caso práctico 02 - (Extra) Certificado Autofirmado para Nginx desde Portainer CE

---



Fons Social Europeu

L'FSE inverteix en el teu futur

Autor: Jose Antonio Villegas Ariza

Actualizado Abril 2022


## Licencia





**Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

<b>Introducción</b>	<b>3</b>
<b>Bibliografía</b>	<b>8</b>

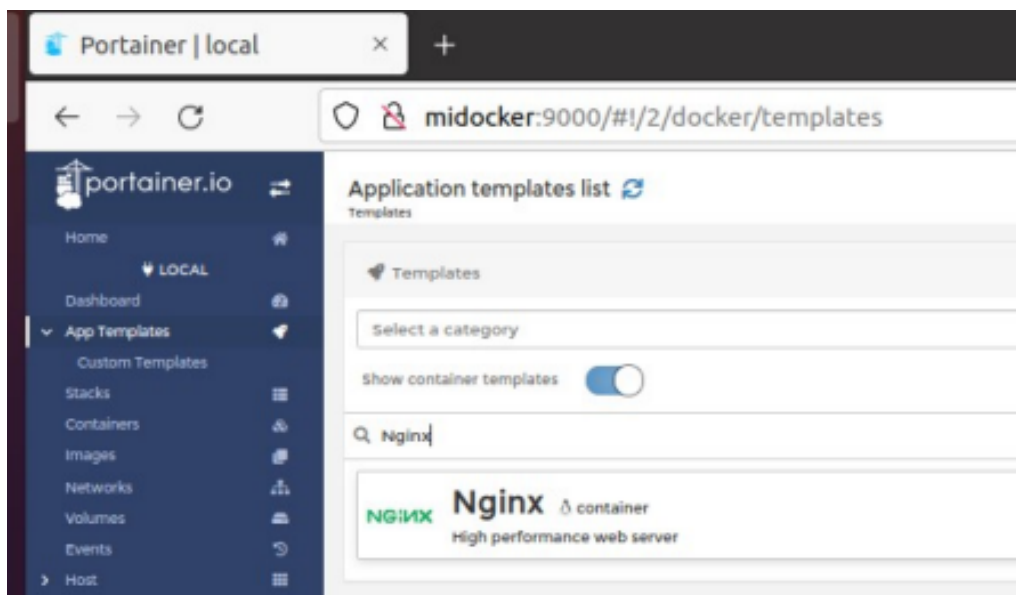
## UD07. CASO PRÁCTICO 02

## 1. INTRODUCCIÓN

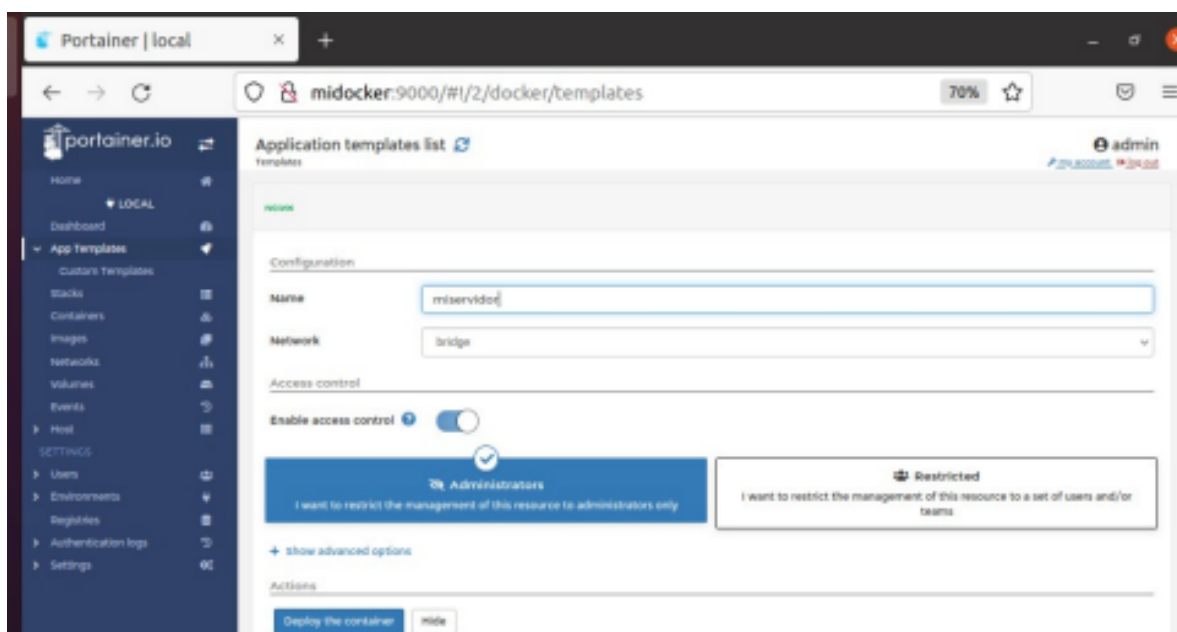
Siguiendo las instrucciones del caso práctico asociado a este documento extra, hemos puesto en marcha el contenedor con Portainer CE y vamos a poner en marcha mediante una plantilla un servidor Nginx e instalar un certificado autofirmado.

En las capturas de pantalla incluidas en este ejemplo, se ha utilizado el host “**midocker**” al ser el host asociado a la máquina, pero podéis utilizar para la realización del caso el host “**localhost**”.

Crearemos un servidor web con la plantilla Nginx seleccionando la opción “**App Templates**” en el menú lateral y utilizando la caja de búsqueda obtenemos la plantilla para crear un contenedor con el servidor web Nginx.



Seleccionamos “**Nginx**” y rellenamos e introducimos el nombre del servidor “**Nginx**”. A continuación hacemos clic en el botón “**Deploy the container**” tal como se aprecia en la siguiente captura de pantalla.



Tras ello, podemos abrir el desplegable **“Show advanced options”**. Esta pantalla nos indica que Portainer CE crea por defecto un par de volúmenes de datos locales con las carpetas **“/etc/nginx”** y **“/usr/share/nginx/html”** mapeadas en ellos respectivamente.

hide advanced options

**Port mapping** [map additional port](#)

Portainer will automatically assign a port if you leave the host port empty.

host	e.g. 80 or 1.2.3.4:80 (optional)	→	container	80	TCP	UDP	
host	e.g. 80 or 1.2.3.4:80 (optional)	→	container	443	TCP	UDP	

**Volume mapping** [map additional volume](#)

Portainer will automatically create and map a local volume when using the auto option.

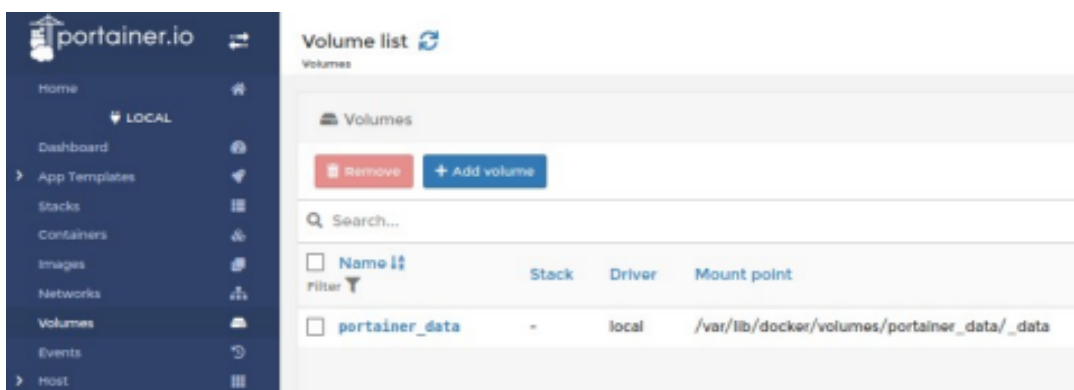
container	/etc/nginx	Auto	Volume	Bind	
container	/usr/share/nginx/html	Auto	Volume	Bind	

**Hosts file entries** [add additional entry](#)

**Labels** [add label](#)

**Hostname**

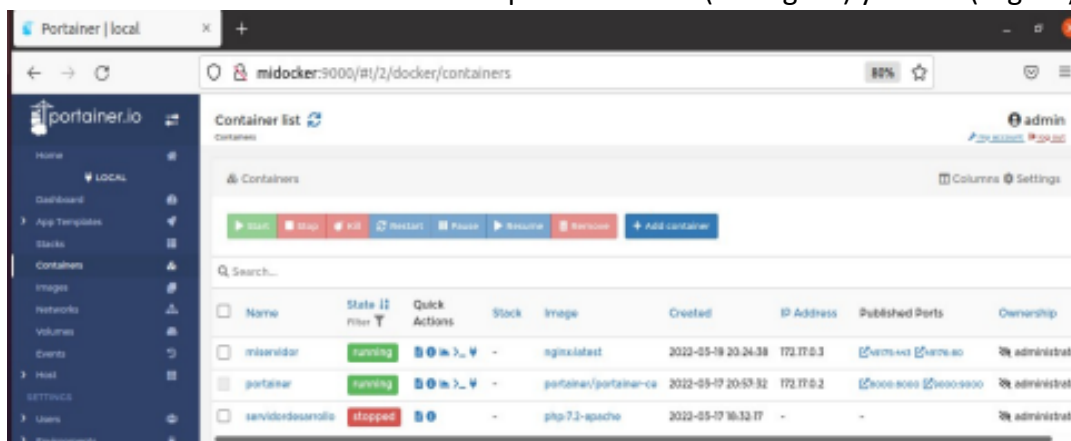
Antes de crear el contenedor voy a mostrar los volúmenes de datos existentes.



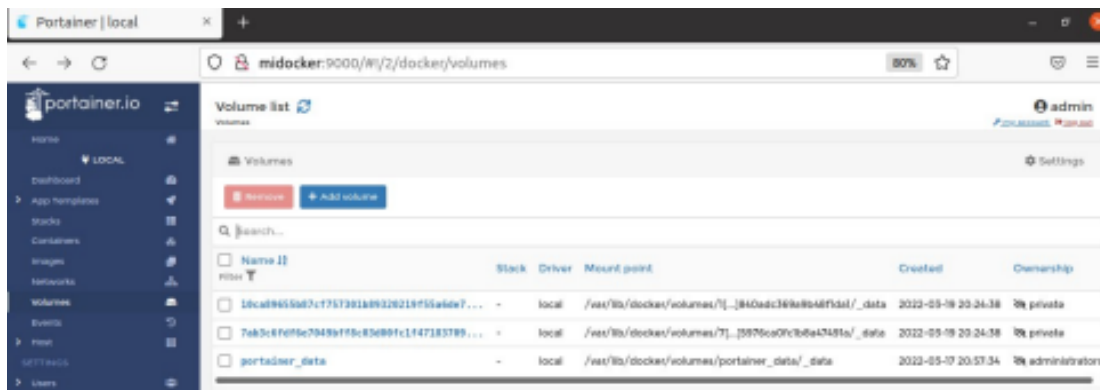
Como se puede apreciar, solo hay un único volumen de datos que se creó cuando se lanzó por primera vez el contenedor con Portainer CE.

Tras este pequeño inciso podemos realizar el despliegue haciendo clic en el botón etiquetado como **“Deploy the container”** que aparece en una de las imágenes anteriores.

La siguiente imagen nos indica que el contenedor **“miservidor”** está en ejecución y que es accesible externamente a través de los puertos 49176 (no seguro) y 49175 (seguro).

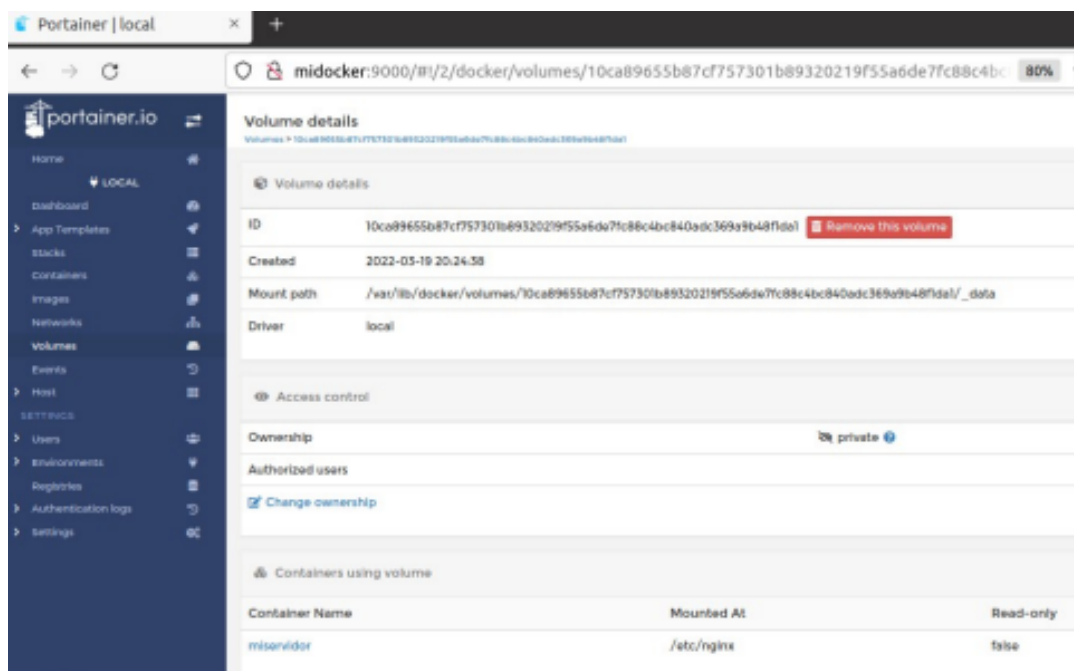


Antes de comprobar el acceso al servidor desde el navegador vamos a ver que efectivamente se han creado los volúmenes de datos locales y los mapeos correspondientes.



Se puede apreciar que aparecen 2 nuevos volúmenes de datos que no teníamos. Si seguimos el enlace del primer volumen, por ejemplo, podemos ver las características de dicho volumen.

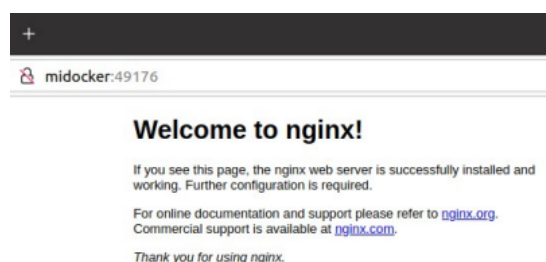
Vemos que el volumen de datos está montado en la carpeta **“/etc/nginx”** del contenedor, por lo que si eliminamos el contenedor esta información no desaparecerá, a no ser que borremos el volumen de datos expresamente.



Desde el navegador de mi máquina cliente comprobamos el acceso al servidor web.

El acceso “no seguro” funciona perfectamente como se puede apreciar.

Acceso al servidor NGINX : <http://localhost:49176> (en la imagen, <http://midocker:49176> debido a que es el nombre de host de la máquina).

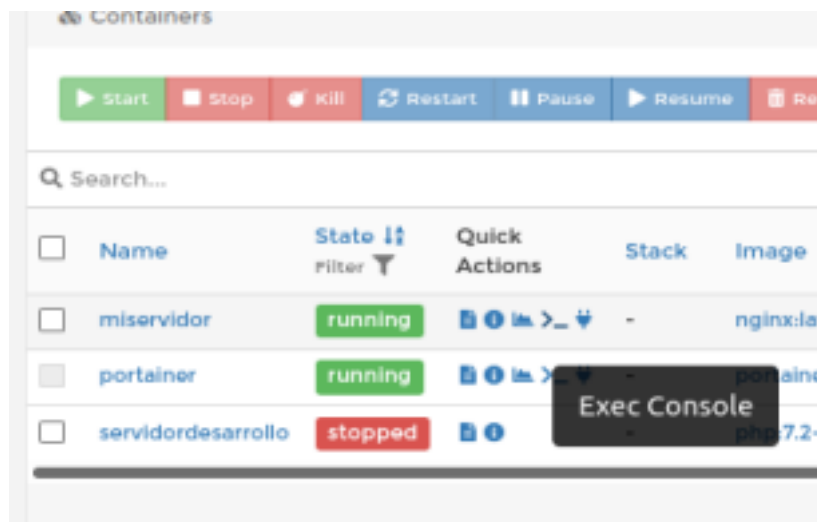


Sin embargo, cuando probamos con el acceso seguro resulta que no funciona.



Tras entrar dentro del contenedor e investigar un poco me he dado cuenta de que “Nginx” no está configurado para trabajar en modo seguro a pesar de que se ha expuesto el puerto 443.

Desde el mismo Portainer CE he entrado al contenedor con un terminal y he cambiado el fichero de configuración “/etc/nginx/conf.d/default.conf”.



La siguiente captura de pantalla nos aparece mostrando la shell del contenedor.



Una vez dentro del contenedor he llevado a cabo las siguientes operaciones:

### 1) Actualizar la lista de paquetes de los repositorios

```
apt-get update
```

## 2) Instalar el editor “Nano”:

```
apt-get install nano
```

## 3) Crear el un certificado y la clave privada para la conexión segura con openssl.

Previamente, creamos la carpeta “/etc/nginx/certificados” y nos situamos en ella.

Tras ello, generamos los certificados con:

```
openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out nginx-certificate.crt -keyout nginx.key
```

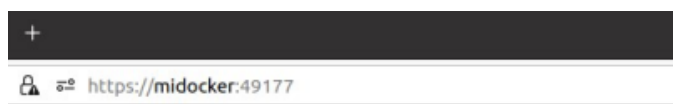
Tras ello, sustituimos el contenido del fichero “/etc/nginx/conf.d/default.conf” por el siguiente:

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /usr/share/nginx/html;  
    index index.html;  
    server_name _;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}  
  
server {  
    listen 443 ssl default_server;  
    listen [::]:443 ssl default_server;  
    ssl_certificate /etc/nginx/certificados/nginx-certificate.crt;  
    ssl_certificate_key /etc/nginx/certificados/nginx.key;  
    root /usr/share/nginx/html;  
    index index.html;  
    server_name _;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Tras realizar estos cambios, paramos el contenedor y lo iniciamos de nuevo.

Tras esto, vemos que ya podemos acceder desde el navegador tanto en modo “no seguro” como en modo “seguro”.

A continuación mostramos los accesos (con esta configuración, cada vez que se inicia el servidor asigna un puerto externo aleatorio):



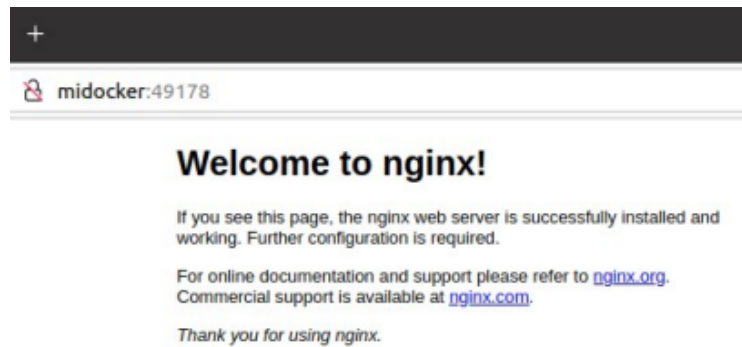
### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

Thank you for using nginx.

El acceso “no seguro” también funciona.



Recordad que, aunque se elimine el contenedor, no se eliminan los volúmenes de datos a no ser que lo hagamos expresamente. Cuando vuelvo a crear el contenedor he de ir a opciones y hay que seleccionar los dos volúmenes creados, ya que si no lo hacemos creará dos nuevos volúmenes y por lo tanto, volveremos a tener la configuración incorrecta.

La siguiente imagen muestra lo que hay que hacer.

— Hide advanced options

**Port mapping** [+ map additional port](#)

Portainer will automatically assign a port if you leave the host port empty.

host	container	TCP	UDP
e.g. 80 or 1.2.3.4:80 (optional)	80	<input type="checkbox"/>	<input type="checkbox"/>
e.g. 80 or 1.2.3.4:80 (optional)	443	<input type="checkbox"/>	<input type="checkbox"/>

**Volume mapping** [+ map additional volume](#)

Portainer will automatically create and map a local volume when using the auto option.

container	volume	Writable	Read-only
/etc/nginx	10ca89655b87cf757301b89320219f55a6de7fc88c4bc540adc369a5	<input type="checkbox"/>	<input type="checkbox"/>
/usr/share/nginx/html	7ab3c6fd6e7049bff8c03d00fc147183709218348fe5976ca0fc1b8a	<input type="checkbox"/>	<input type="checkbox"/>

**Hosts file entries** [+ add additional entry](#)

**Labels** [+ add label](#)

Comprobamos que podemos acceder con los dos modos

En cambio, si elimino el contenedor y no selecciono los volúmenes, cuando lo vuelvo a crear al no tener la configuración actualizada, solo puedo acceder en modo “no seguro”.

## 2. BIBLIOGRAFÍA

- [1] Docker Docs <https://docs.docker.com/>
- [2] Portainer CE Documentation <https://documentation.portainer.io/>