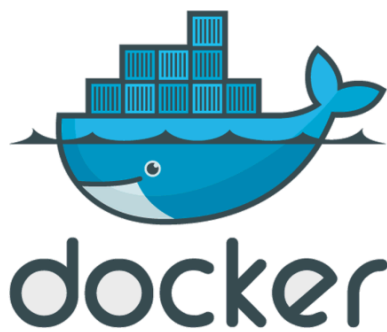


INSTALACIÓN ALPINE LINUX

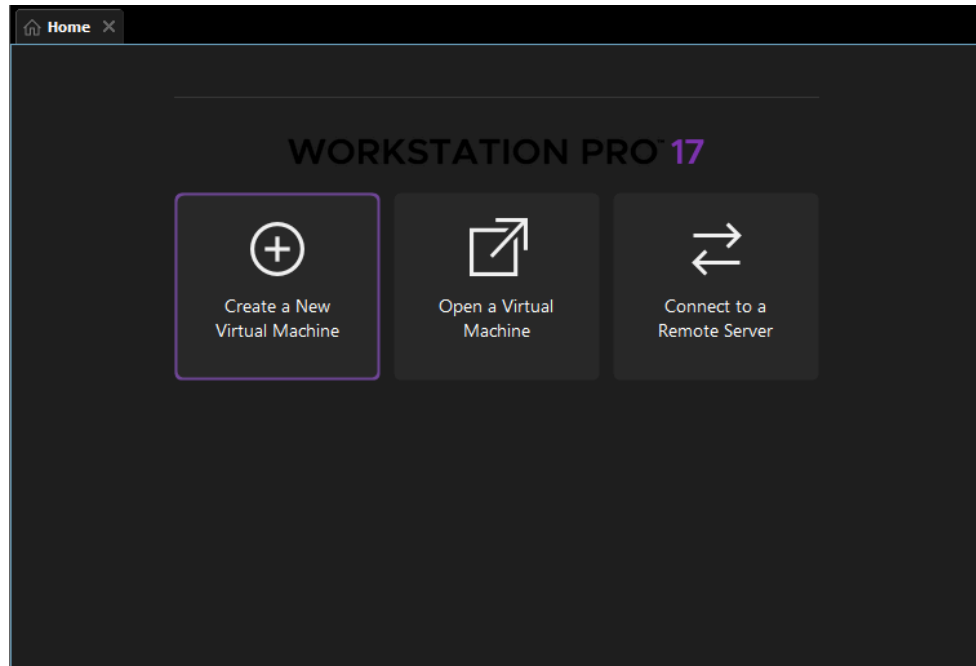
1.Creación de la máquina virtual.	2
2.Configuración de la máquina virtual.	6
3.Instalación y creación de docker y el contenedor con apache	7
4.Servidor configurado y funcionando	8



1.Creación de la máquina virtual.

Para crear la máquina virtual lo primero es utilizar una aplicación de virtualización, en mi caso he utilizado VMWare.

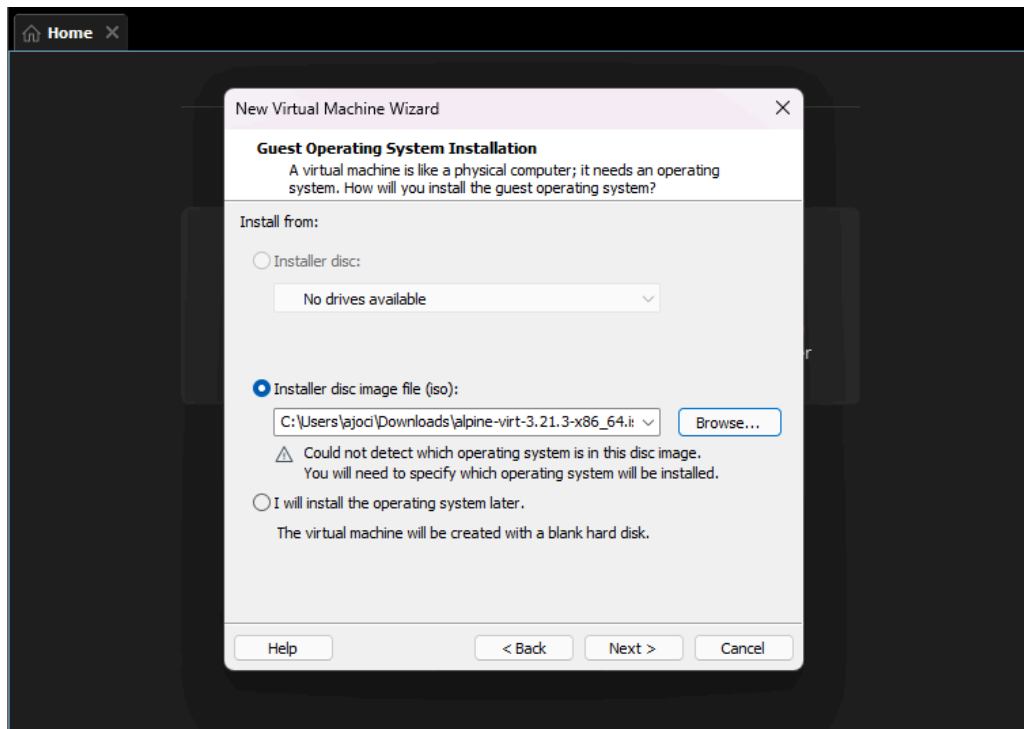
El primer paso es pulsar en crear una nueva máquina virtual.



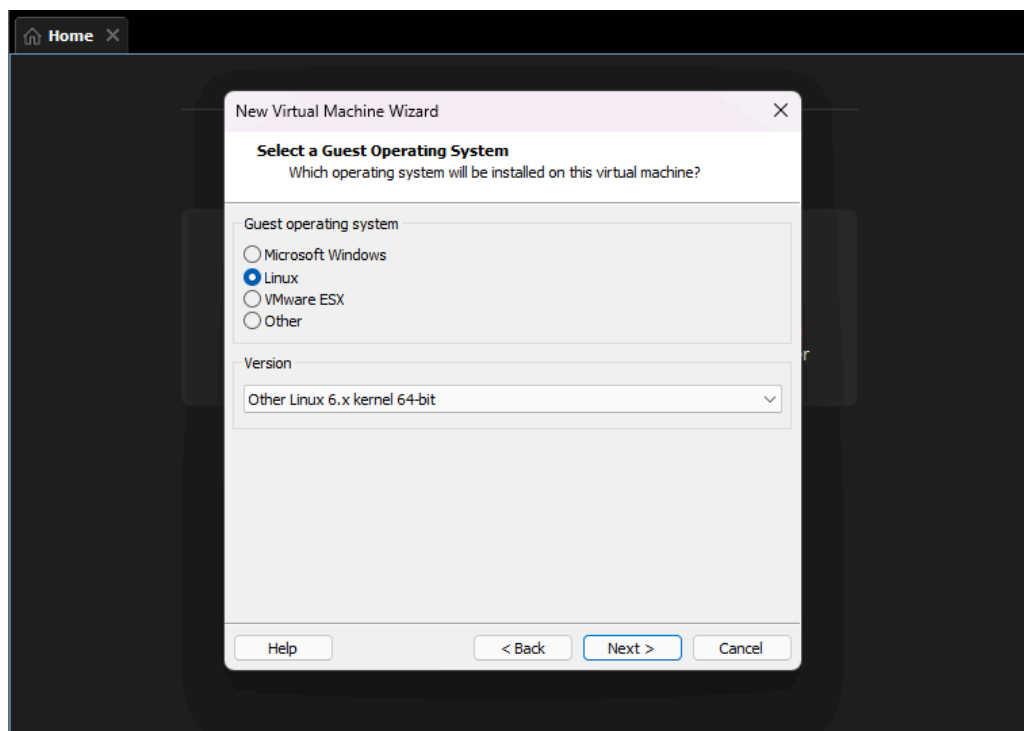
Para empezar la instalación hay que darle al botón de abajo, Custom, para poder configurar la máquina virtual como más me interese en función de lo que requiera el sistema operativo que voy a instalar.



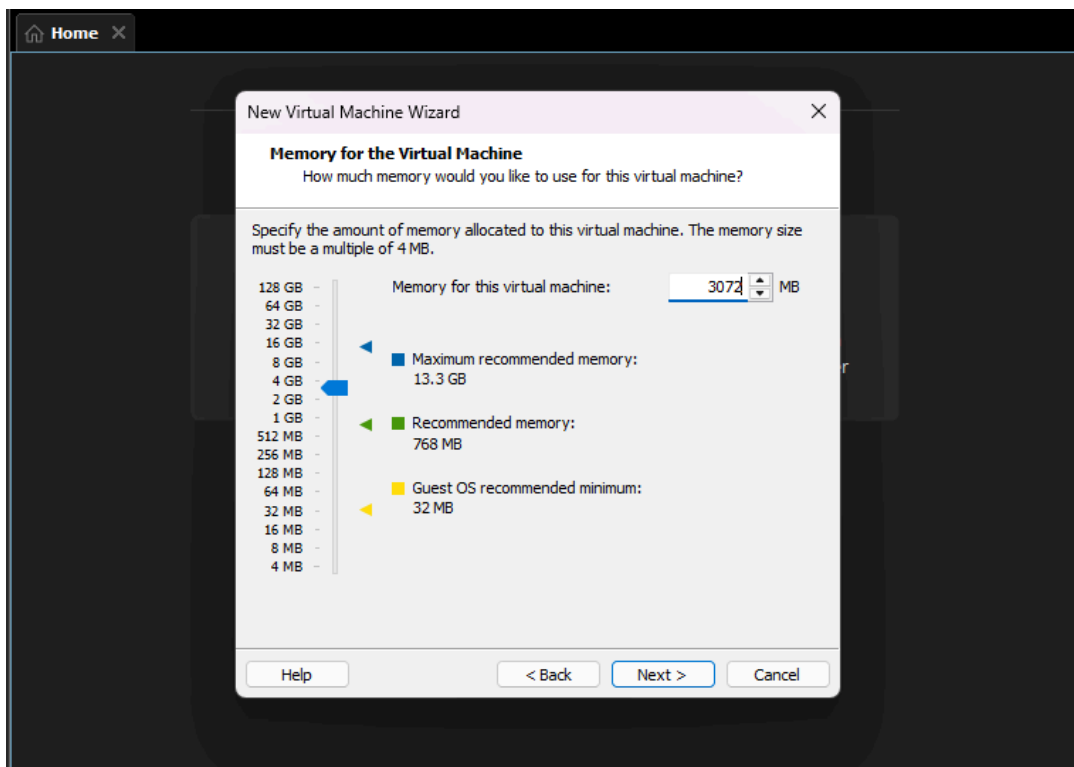
El siguiente paso importante es añadir la ruta donde está la ISO que voy a utilizar, en este caso he utilizado una máquina de alpine, ya que consume pocos recursos y en ella puedo instalar un contenedor de docker con apache para montar el servidor web.



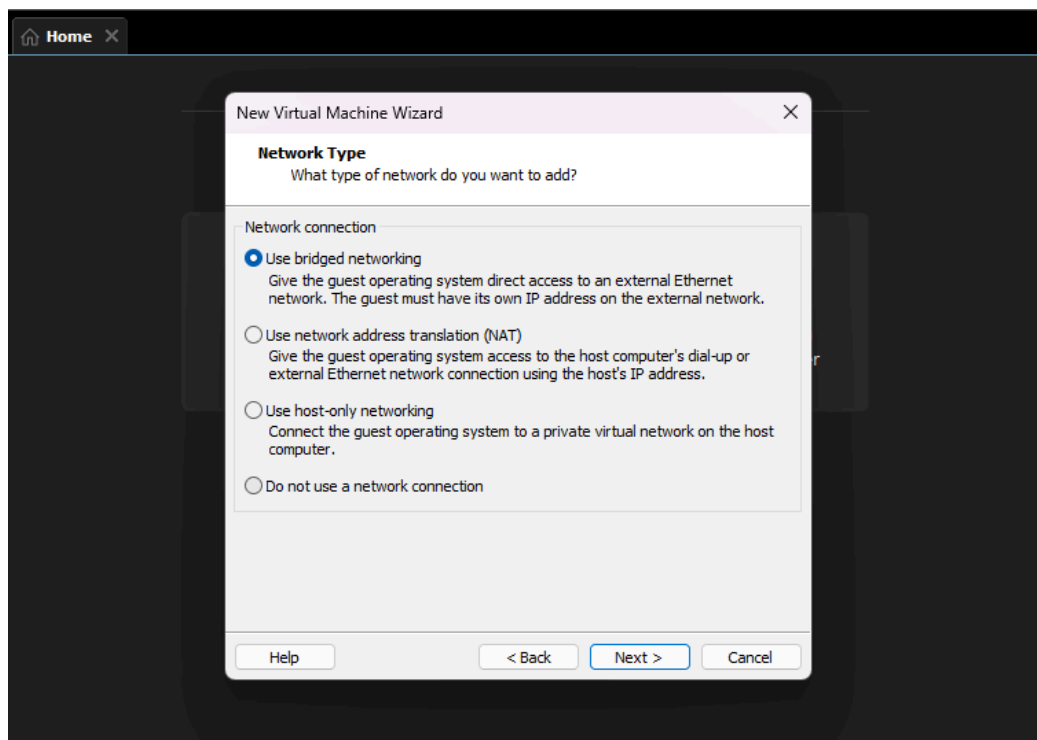
A continuación especifico cual es el sistema operativo que voy a instalar, como he dicho antes es un alpine por tanto hay que seleccionar un Linux de 64-bit.



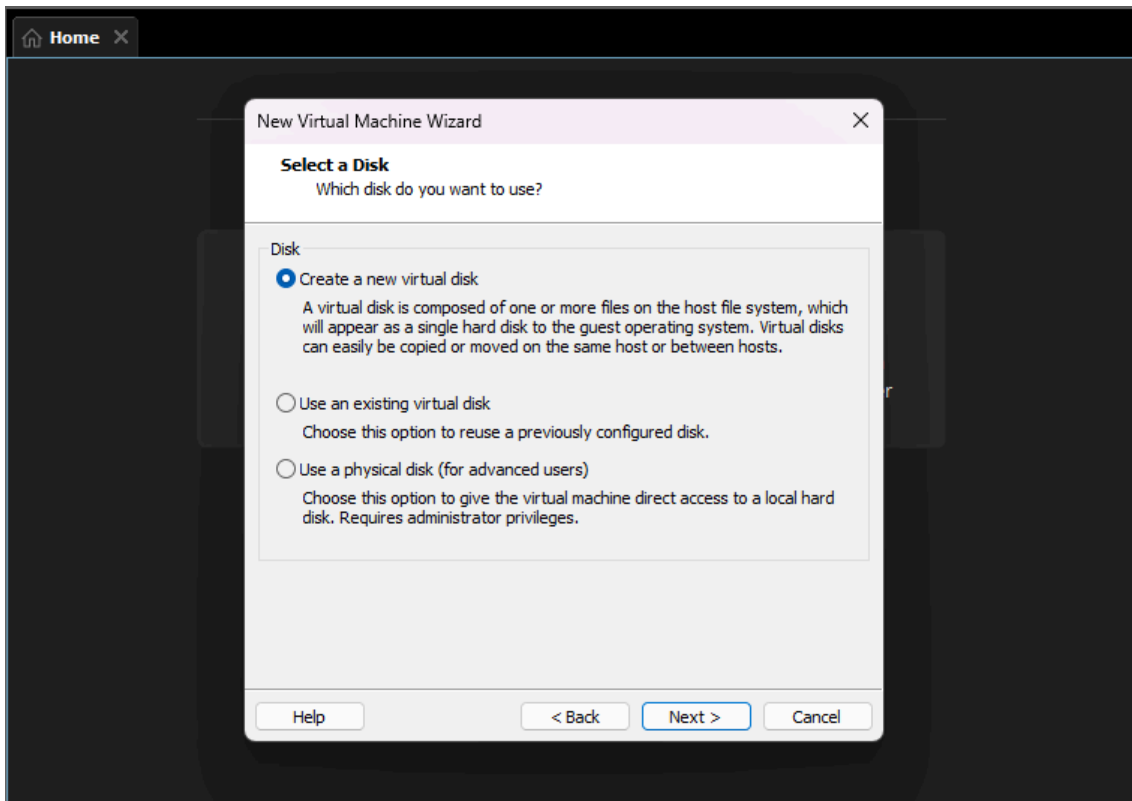
A la hora de configurar la RAM hay que comprobar cuales son los requisitos mínimos del sistema operativo, en este caso he puesto 3GB para no tener ningún problema, pero se podría montar hasta con 2GB



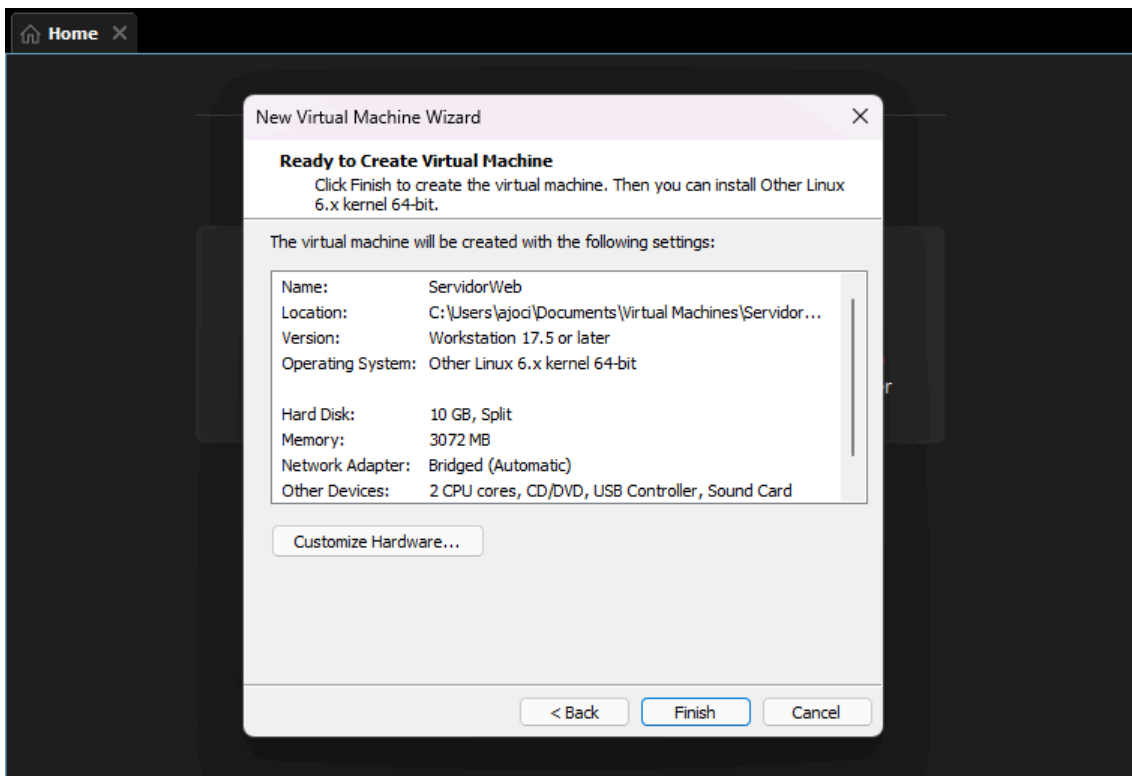
El tipo de red he seleccionado el Bridget, esto permite que la máquina sea accesible desde otros dispositivos en la misma red, así podré acceder más tarde a la página web como se ve en las capturas finales.



Todas las máquinas necesitan un disco virtual propio, para esto selecciono la primera opción y le añado 10 GB de almacenamiento.



En esta última captura se puede ver un resumen de toda la configuración que he hecho para la máquina.



2. Configuración de la máquina virtual.

Al iniciar la máquina el primer comando que hay que poner es *setup-alpine* para poder configurar en la máquina cosas como el Hostname, el idioma del teclado, etc...

```
Welcome to Alpine Linux 3.21
Kernel 6.12.13-0-virt on an x86_64 (/dev/tty1)

localhost login: root
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <https://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

localhost:~# setup-alpine

ALPINE LINUX INSTALL
-----

Keymap
-----
af  al   am  ara  at   az   ba   bd   be   bg   br   brai  by   ca   ch   cm
fr  gb   ge  gh   gr  hr   hu   id   ie   il   in   iq   ir   is   it   jp
nd  me  nk  ml   nm  mt   my   ng   nl   no   nz   ph   pk   pl   pt   ro
tw  ua   us  uz   vn

Select keyboard layout: [none] es

es-ast      es-cat      es-deadtilde  es-dvorak      es-nodeadkeys es-winkeys      es

Select variant (or 'abort'): es-winkeys

* Caching service dependencies ...
* Setting keymap ...

Hostname
-----
Enter system hostname (fully qualified form, e.g. 'foo.example.org') [localhost] ServidorWEB
```

Una vez terminada la configuración, hay que ver que se ha configurado todo bien y comprobar que ip tiene la máquina como se ve en la imagen es la 192.168.1.148, esta será la IP que tendré que poner en el navegador para acceder a mi página más tarde.

```
servidorweb:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:00:f7:1d brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.148/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fdd4:3f69:b500:e4ff:20c:29ff:fe00:f71d/64 scope global dynamic flags 100
        valid_lft 1758sec preferred_lft 1758sec
    inet6 fe80::20c:29ff:fe00:f71d/64 scope link
        valid_lft forever preferred_lft forever
servidorweb:~# _
```

3.Instalación y creación de docker y el contenedor con apache

Usando el comando **apk add docker-cli-compose** se instala docker en la máquina para poder montar el contenedor apache, en la siguiente imagen se ve como se instalan 23 paquetes.

```

servidorweb:~# apk add docker docker-cli-compose
(1/23) Installing ca-certificates (20241121-r1)
(2/23) Installing libseccomp (2.5.5-r0)
(3/23) Installing runc (1.3.0-r1)
(4/23) Installing containerd (2.1.1-r0)
(5/23) Installing libffi (3.4.7-r0)
(6/23) Installing libintl (0.22.5-r0)
(7/23) Installing libmount (2.40.4-r1)
(8/23) Installing pcre2 (10.43-r0)
(9/23) Installing glib (2.82.5-r0)
(10/23) Installing log_proxy (0.7.4-r0)
(11/23) Installing containerd-openrc (2.1.1-r0)
(12/23) Installing libnsl (1.0.5-r2)
(13/23) Installing libnftnl (1.2.8-r0)
(14/23) Installing libxtables (1.8.11-r1)
(15/23) Installing iptables (1.8.11-r1)
(16/23) Installing iptables-openrc (1.8.11-r1)
(17/23) Installing tini-static (0.19.0-r3)
(18/23) Installing docker-engine (28.2.1-r0)
Executing docker-engine-28.2.1-r0.pre-install
(19/23) Installing docker-openrc (28.2.1-r0)
(20/23) Installing docker-cli (28.2.1-r0)
52%

```

Ahora hay que crear un archivo .sh que sea ejecutable en esta captura se puede ver el contenido del archivo, a continuación explico la función de cada línea.

- **docker container run -dit** : Crea y ejecuta un contenedor en segundo plano (-d), con terminal interactiva (-it).
- **-p B080:B0** : Mapea un puerto del host al puerto del contenedor.
- **--hostname apache** : Asigna el nombre de host apache al contenedor.
- **--name apache** : Nombra el contenedor como apache para gestionarlo fácilmente.
- **-u /root/apache:/usr/local/apache2/htdocs/** : Monta la carpeta /root/apache del host en /usr/local/apache2/htdocs/ del contenedor.
- **--restart unless-stopped** : Reinicia automáticamente el contenedor si se cierra.
- **httpd:2.4** : Usa la imagen oficial de Apache versión 2.4.

```
servidorweb:~# cat script.sh
docker container run -dit \
    -p 8080:80 \
    --hostname apache \
    --name apache \
    -v /root/apache:/usr/local/apache2/htdocs/ \
    --restart unless-stopped \
httpd:2.4
```

Para ejecutar el script creado anteriormente hay que utilizar el comando `sh script.sh`.

```
servidorweb:/# sh script.sh
Unable to find image 'httpd:2.4' locally
2.4: Pulling from library/httpd
61320b01ae5e: Pull complete
be60498bea0a: Pull complete
4f4fb700ef54: Pull complete
8f86928406fd: Pull complete
162ef2c73af1: Pull complete
8dbbd44856ed: Pull complete
Digest: sha256:09cb4b94edaaa796522c545328b62e9a0db60315c7be9f2b4e02204919926405
Status: Downloaded newer image for httpd:2.4
817610c239fc7a0713b85d1ef971cc0a955a20d24abf1af72b96990100cd98d8
servidorweb:/#
```

Una vez ejecutado el comando hay que comprobar que el contenedor se ha montado correctamente y si está funcionando esto se puede ver poniendo `docker container ls`

```
servidorweb:/# docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
817610c239fc   httpd:2.4      "httpd-foreground"      45 seconds ago Up 45 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   apache
servidorweb:/#
```

4. Servidor configurado y funcionando

En estas dos últimas capturas se puede ver cómo accedo al servidor desde mi máquina real poniendo la IP de la máquina virtual y el puerto del contenedor, al no estar creado todavía el archivo .html no se muestra nada pero una vez se creen y los coloque en la ruta `/root/apache/` de la máquina virtual se podrá visualizar la página.



En esta última captura se puede ver el resultado desde el navegador, lo único que he cambiado la IP ya que tuve que hacer la máquina virtual de nuevo, el resto de la página web se puede ver en el archivo `VisualizacionPaginaWeb.pdf`.