**Unidaddidáctica nº4-ActividaddeDesarrollo-**

**"Servicio FTP: Ubuntu Server "**

**Servicios en Red e Internet –2º ASIR**

**I.E.S. Miguel Herrero**

**Curso2021-22**

**Objetivos:**En esta práctica vamos a instalar y configurar un servidor FTP (ssh/sftp y ftp/vsftpd) en un SO libre, Linux Ubuntu Serverasí mismo vamos a configurar los servicios FTP para subir archivos a los servidores webs. También vamos a implementar un servidor FTP/VSFTPD en dockers.

**Duración:**4 h

**Recursos:**

**Vsftpd, instalación y configuración básica de un servidor FTP->**https://ubunlog.com/vsftpd-instalar-un-servidor-ftp-ubuntpu/?utm\_source=dlvr.it&utm\_medium=facebook

**How To Install an FTP server (vsftpd) on Ubuntu 20.04/20.10->**

https://devanswers.co/install-ftp-server-vsftpd-ubuntu-20-04/

**How To Configure SFTP for a Web Server Document** Root->

https://devanswers.co/configure-sftp-web-server-document-root/

**Modo y formato de entrega:**Entregar a través de la plataforma online ***google classroom*** en pdf llamado A2 UD4 Tu nombre y apellidos.pdf generado a partir de este y en el cual deberás añadir las respuestas a las cuestiones planteadas así como las capturas de pantalla, logs y demás evidencias necesarias para demostrar que la práctica se ha realizado cumpliendo los requisitos solicitados en el enunciado.

**Escenario:**

En esta práctica deberás instalar y configurar un servidor ftp/sftp en Ubuntu Server.Para ello vamos a utilizar las mismas máquinas y mismo escenario que en lasActividades 2 y 3 de la unidad 2. **Todas las máquinasestarán en la misma subred privada interna ASIRXXX**, con dirección de red **192.168.XXX.0/24 y 2001:DB8:1:XXX::/64**, donde XXX son los dígitos de tu nombre de usuario del dominio.



**AMPLIACIÓN 3**

**Ejercicio 8: Configura un servidor VSFTPD con dockers y acceso al “*documentroot*” del servidor web. Usa la imagen panubo/vsftpd:**

**Recursos:**

**panubo/vsftpd->**

https://hub.docker.com/r/panubo/vsftpd/

Lee la documentación de la imagen y configura de manera adecuada un contendor con el servicio FTP/VSFTPD de forma que un usuario llamado **webdev\_asirXXX.es**pueda llevar a cabo el mantenimiento todos los sitios 4,5,6 y 7 del servidor web (servidor 3).

**Nota**: No olvides parar el servicio VSFTPD en la VPS (máquina virtual servidor 3).

-Conéctate al servidor **VSFTPD**del docker (¿Cuál es su FQDN?) mediante ***FTP*** desde uno de los clientes 1 o 2 como usuario ***webdev\_asirXXX.es***utilizando un **cliente FTP**. Modifica en local la web index.html, súbela mediante ftp al servidor web y accede al sitio web desde un navegador.

**Vamos a llevar a cabo la ampliación 3 paso a paso**

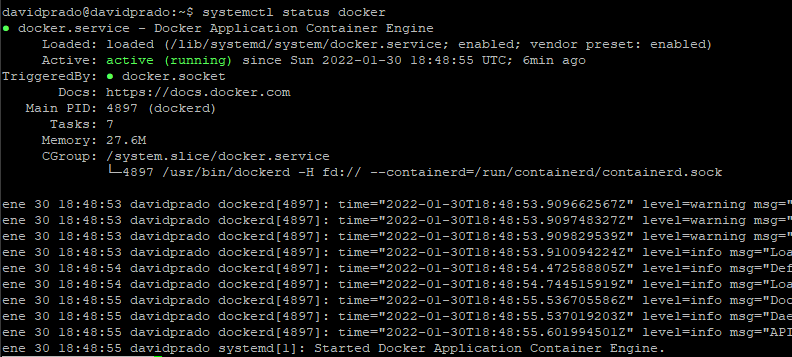
**(en rojo lo modificado):**

**1ª PARTE**

**Objetivo**: Mantenimiento del sitio/servidor web de tu máquina virtual (host) con un servidor FTP (vsftpd) en contenedor “*dockerizado*”..

- Instala docker en una máquina virtual Linux.

- Verifica que Docker está instalado



**Pista**: docker --v…..

-Crea en $HOME (/home/tu\_usuario) un directorio llamado **servidorFtp** y en su interior los siguientes archivos:

**servidorFtp**

**├── vsftpd.conf**

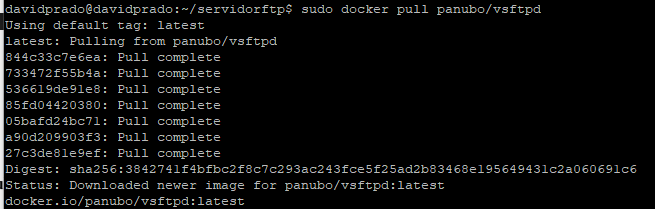
**├── docker-compose.yml**

**├── Dockerfile**

****

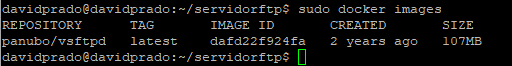
- Descarga la imagen **panubo/vsftpd**

**Pista**: docker p…..

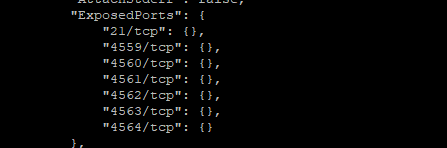


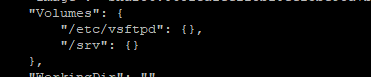
- Muestra las imágenes que tienes descargadas

**Pista**: docker im….



-Haz un inspect de la imagen de panubo/vsftpd e indica qué puertos (ExposedPorts) y volúmenes (Volumes) expone a título informativo





- Acude a la documentación de la imagen panubo/vsftpd en docker-hub y responde a las siguientes cuestiones:

- ¿Qué aplicaciones tiene en general el servidor vsftpd?

Un servicio FTP

- Explica para que sirven las siguientes variables de entorno:

FTP\_USER: define un usuario FTP por defecto

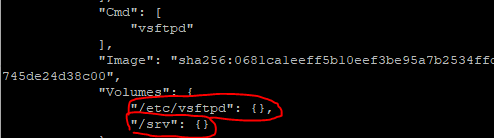
FTP\_PASSWORD: define una contraseña

FTP\_CHOWN\_ROOT:

- Indica cuál es el directorio en la imagen donde está ubicadala configuración y raíz del servicio ftp:

-Configuración del servidor FTP (vsftpd.conf):

-Raiz(root) servidor FTP:



**Pista:**Volumes del inspect

- Ejecuta un contenedor demonio, de nombre ***servidorFtp1***, a partir de la imagen ***panubo/vsftpd*** de forma que lleves el mantenimiento de algún sitio web ubicado en tu máquina virtual o host (por ejemplo recuerda que en *nginx* el directorio del sitio web por defecto es: /var/www/html). Utiliza el **usuario**: asirXXXy **contraseña**:1234

**Pista (**sustituye los interrogantes por el correspondiente valor**)**:***docker run -d - -name ¿? -p 20:20 -p 21:21 -p 4559-4564:4559-4564 -v ¿?:/srv -e FTP\_USER=¿? -e FTP\_PASSWORD=¿? -e FTP\_CHOWN\_ROOT=true panubo/vsftpd***

******

******

**Nota**: La variable de entorno FTP\_CHOWN\_ROOT=true asigna como propietario y grupo al directorio del contenedor /srv (De esa manera se puede escribir/subir archivos al servidor FTP). PERO NO LO HACE DE MANERA RECURSIVA. Ello puede generar problemas a la hora de subir archivos al servidor FTP. Lo ideal sería que el directorio a mapear estuviera, **vacío**; sin embargo eso no es habitual.

Por ello, tienes que acceder al contenedor, ***servidorFtp1***, y establecer como usuario y grupo propietario del directorio raíz ,/srv, a ftp:ftp. Para ello sigue los siguientes pasos:

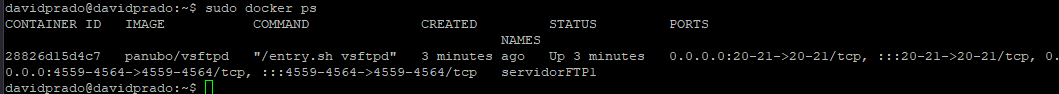
1º.- Inicia en el contenedor ***servidorFtp1*** una terminal interactiva (bash)

**Pista**:docker exec -it …..…….

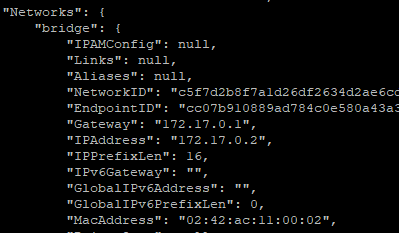
-Ejecuta en el contenedor**servidorFTP1** el siguiente comando: ***chown -R ftp:ftp /srv***

.- Lista los contenedores que se están ejecutando.

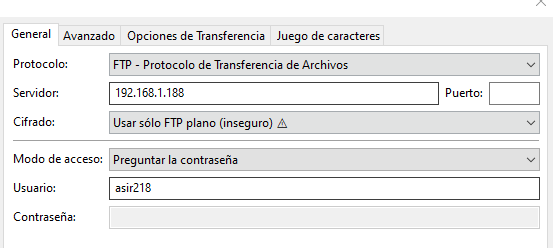
**Pista**: docker p….

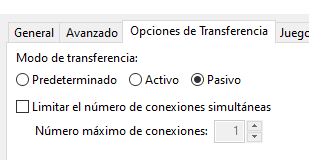


- Haz un inspect del contenedor ***servidorFtp1***e indica la red a la que está conectado, Ip, máscara y puerta de enlace



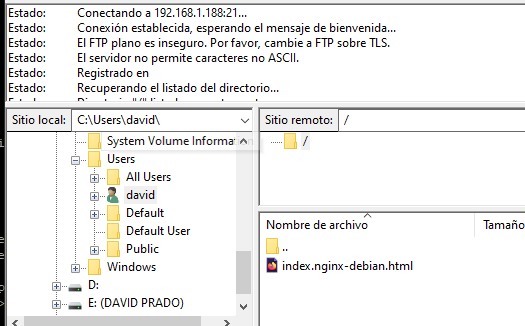
- Conectate al servidor ftp





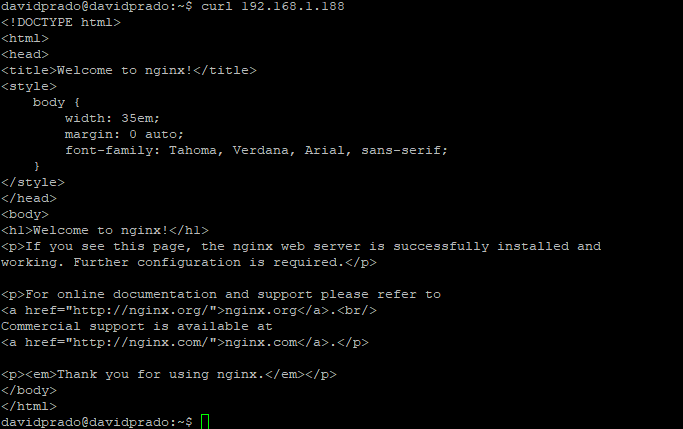
**Pista**: ftp ip\_servidor\_ftp (No utilices localhost)

- Muestra el contenido de directorio raíz

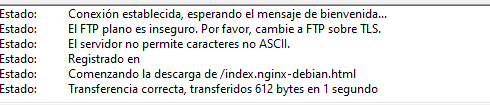


**Nota**: Si no puedes mostrarlo, activa el modo pasivo en el cliente

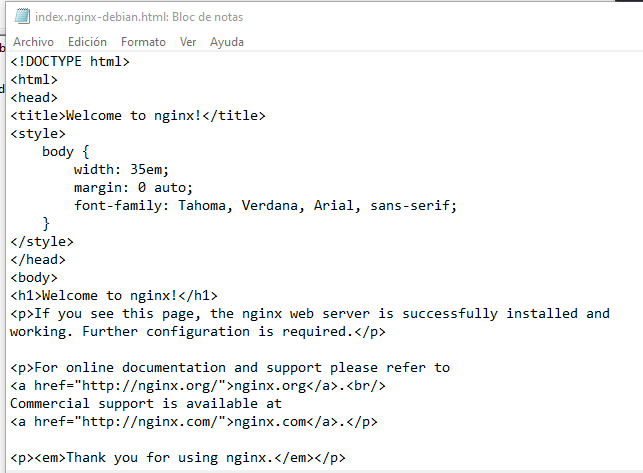
El anterior index



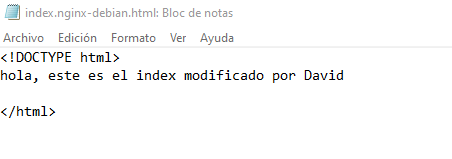
- Descarga el index.html



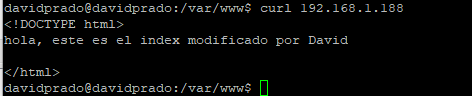
- Modifica el index.html



Lo modifico



- Sube el index.html



-Borra el contenedor ***servidorFtp1***

**Pista**: docker r…….

~~-Crea un nuevo contenedor~~ ***~~servidorFtp1~~***~~con la misma configuración que antes pero añadiendo la variable de entorno: FTP\_CHOWN\_ROOT=true~~

~~-Prueba ahora a descargar el index.html, modificarlo y subirlo.~~

~~- Conéctate con el navegador al sitio web y comprueba que se ha actualizado la página web.~~

**2ª PARTE**

**Objetivo**: Instalación de un servidor/sitio web *nginx* en contenedor “dockerizado” y llevar a cabo su mantenimiento con un servidor FTP vsftpd también en contenedor.

**SERVIDOR WEB EN CONTENEDOR**

-Crea en $HOME (/home/tu\_usuario) un directorio llamado ***servidorWeb1*** y en su interior los siguientes archivos y directorios:

**servidorWeb1**

**├── conf.d**

**│ └── default.conf**

**├── docker-compose.yml**

**├── Dockerfile**

**└── html**

**└── index.html**

-El contenido del index.html es el siguiente:

index.html:

<!DOCTYPE html>

<html>

<head>

<title>Servidor web 1</title>

</head>

<body>

<h1>Bienvenido al servidor web 1</h1>

</body>

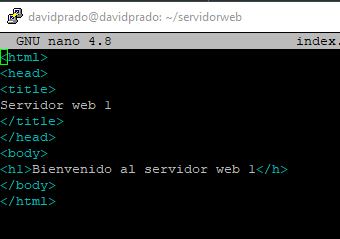
</html>

Creo una carpeta que contendrá el Index del html



Creo el index





- Acude a la documentación de la imagen *nginx* en docker-hub y responde a las siguientes cuestiones:

- ¿Qué aplicaciones/usos tiene en general el servidor nginx y en particular la imagen nginx que acabas de descargar?

- Indica la ruta (ubicación) de los siguientes archivos en el contenedor:

-Directorio ubicación sitio web por defecto (index.html):

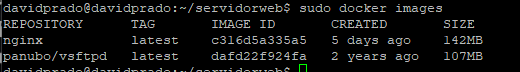
-Fichero de ubicación de nginx (nginx.conf):

-Fichero de configuración del sitio web por defecto (default.conf):

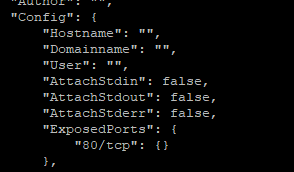
- Descarga la imagen nginx



-Muestra las imágenes que tienes tu registro local

****

-Haz un inspect de la imagen de nginx para ver qué puertos (ExposedPorts) expone.

****

- Crea un contenedor demonio, llamado ***servidorWeb1***, accesible con el puerto 8081, usando la imagen oficial de **nginx**. (Expón el puerto 8081 de tu host al puerto 80 del contenedor) y monta el directorio del sitio web por defecto (/usr/share/…) en el directorio html creado anteriormente

Arranco el contenedor con la carpeta montada del index



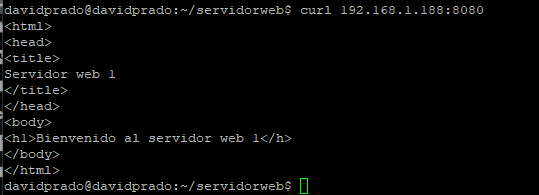
**Pista**: docker run -d -p…..- -name…….

-Muestra la lista de contenedores y comprueba que ***servidorWeb1*** se está ejecutando

****-Inicia en el contenedor ***servidorWeb1*** una terminal interactiva (bash) y muestra en el mismo su dirección IPv4, puerta de enlace y muestra el sitio web (curl ….)

**Pista**:docker exec…….

-Accede al navegador web desde el host (máquina virtual) y comprueba que el servidor web 1 está funcionando.

****

**SERVIDOR FTP EN CONTENEDOR**

-Borra el contenedor ***servidorFtp1***

**Pista**:docker r……

- Ejecuta un nuevo contenedor demonio, de nombre ***servidorFtp2***, a partir de la imagen *panubo/vsftpd* de forma que lleves el mantenimiento del ***servidorWeb1***. Utiliza el **usuario**: asirXXXy **contraseña**:1234

**Pista**: ***docker run -d - -name ¿? -p 20:20 -p 21:21 -p 4559-4564:4559-4564 -v ¿?:/srv -e FTP\_USER=¿? -e FTP\_PASSWORD=¿? -e FTP\_CHOWN\_ROOT=true panubo/vsftpd***

Nota: No te olvides acceder al contenedor, ***servidorFtp2***, y establecer como usuario y grupo propietario del directorio raíz ,/srv, a ftp:ftp al igual que hiciste en servidorFtp1

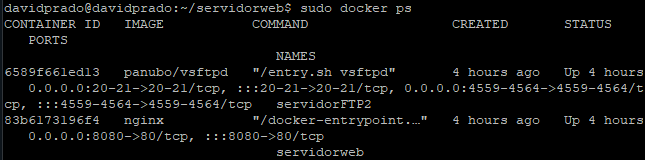
Inicio el FTP con la carpeta montada del servidor web



Cambio el sueño de /srv a ftp



-Muestra la lista de contenedores y comprueba que ***servidorFtp2*** se está ejecutando

****

-Inicia en el contenedor ***servidorFTP2*** una terminal interactiva (bash) y haz un ping al contenedor ***servidorWeb1***. (ping….).

**Pista**:docker exec…….

-Utiliza el contenedor***servidorFtp2*** para descargar el index.html del ***servidorweb1***, modificarlo y subirlo al ***servidorWeb1***.

index.html (modificado):

<!DOCTYPE html>

<html>

<head>

<title>Servidor web 1</title>

</head>

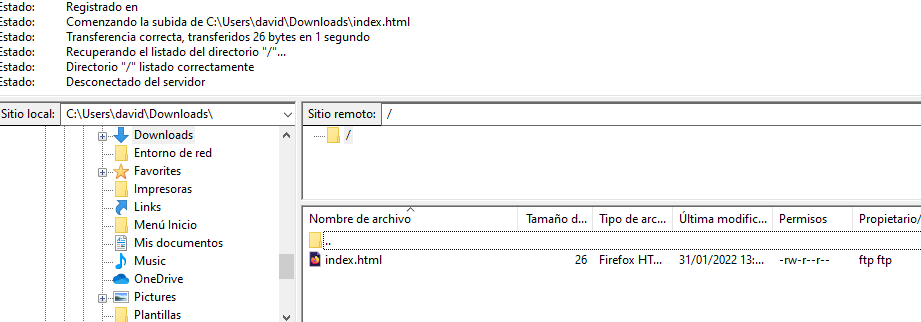
<body>

<h1>Bienvenido al servidor web 1 DEFINITIVO</h1>

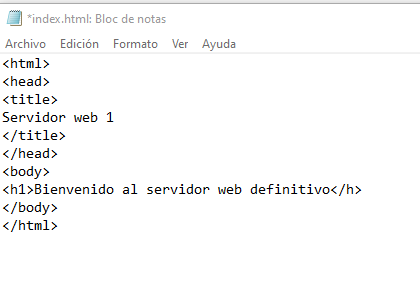
</body>

</html>

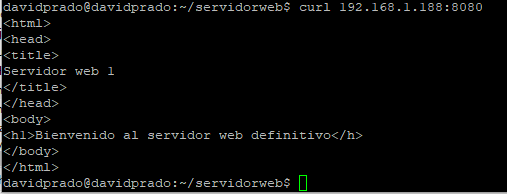
Inicio sesión en el servidor desde filezilla



Y modifico el index



-Accede alnavegador web del host (máquina virtual) y comprueba que el servidor web 1 esta funcionando y modificado.

****

**3ª PARTE**

**Objetivo**. Vamos a generar los contenedores ***servidorWeb1*** y ***servidorFtp2*** con **docker-compose** y repetir los pasos de la 2ª parte, es decir, EN VEZ DE GENERAR LOS CONTENEDORES CON **DOCKER RUN** como en la 1ª y 2ª parte LOS VAMOS A CONSTRUIR CON LA HERRAMIENTA **DOCKER-COMPOSE** configurando los escenarios en **docker-compose.yml.** Esta segunda manera es la forma habitual de arrancar los contenedores.

-Borra los contenedores ***servidorWeb1*** y ***servidorFtp2***

**Pista**: docker r.....

-Instala docker-compose.

- Verifica que docker-compose está instalado

**Pista**: docker-compose --….

**SERVIDOR WEB EN CONTENEDOR**

**ESCENARIO CON DOCKER-COMPOSE**

-Vamos a crea el escenario del servidor web 1. Ubícate en el directorio **$HOME/*servidorWeb1***

-Edita el archivo ***docker-compose.yml***

-Copia las siguientes líneas en docker-compose.yml:

version: "3.9"

services:

web:

container\_name: ***servidorWeb1***

image: nginx:latest

volumes:

- ./html:/usr/share/nginx/html

ports:

- 8081:80

networks:

contenedores:

ipv4\_address: 192.168.20.101

networks:

contenedores:

name: red\_contenedores

ipam:

config:

- subnet: 192.168.10.0/24

-Explica el significado de cada una de las líneas anteriores del archivo docker-compose.yml

-Levanta el escenario con docker-compose en modo demonio (detach)

**Pista**: docker-compose u………

-Muestra los contenedores en ejecución con docker-compose

**Pista**: docker-compose p…..

-Muestra un listado de los contenedores con docker

**Pista**: docker p……..

Muestra un listado de las redes

**Pista**: docker ne……..

-Accede alnavegador web del host y comprueba que el servidor web 1 esta funcionando.

**Pista**:http://localhost.....

**SERVIDOR FTP EN CONTENEDOR**

**ESCENARIO CON DOCKER-COMPOSE**

-Vamos a crear el ESCENARIO del servidor ftp 2. Ubícate en el directorio  **$HOME/servidorFtp**

-Edita el archivo Dockerfile

-Copia las siguientes líneas

FROM panubo/vsftpd

RUN apt update && apt install -y nano

RUN chown -R ftp:ftp /srv

EXPOSE 21

-Explica el significado de cada una de las líneas anteriores. ¿Qué estamos haciendo en el Dockerfile? ¿Te suena de algo?

- Genera una imagen de nombre *tu\_usuario\_docker\_hub/vsftpd*

**Pista**: docker build -t…….

- Haz un listado de las imágenes y comprueba que se ha creado la imagen: tu\_usuario\_docker\_hub/vsftpd

Pista: docker i……..

-Edita el archivo docker-compose.yml

-Copia las siguientes líneas en docker-compose.yml

version: "3.9"

services:

vsftpd:

image: tu\_usuario\_docker\_hub/vsftpd

container\_name: ***servidorFtp2***

environment:

- FTP\_USER=asirXXX

- FTP\_PASSWORD=1234

- FTP\_CHOWN\_ROOT=true

volumes:

- ../servidorWeb1/html:/srv

ports:

- 0.0.0.0:21:21

- 0.0.0.0:20:20

- 4559-4564:4559-4564

networks:

contenedores:

ipv4\_address: 192.168.20.100

networks:

contenedores:

external: true

name: red\_contenedores

-Explica el significado de cada una de las líneas anteriores

-Levanta el escenario con docker-compose

**Pista:**docker-compose u……

-Muestra los contenedores en ejecución con docker-compose

**Pista:**docker-compose p…….

-Muestra los contenedores con docker

**Pista:**docker p…..

-Muestra un listado de las redes

**Pista:**docker n……..

- Conéctate al servidor ftp

**Pista**: ftp ip\_servidor\_ftp (No utilices localhost)

- Muestra el contenido de directorio raíz

- Descarga el index.html

- Modifica el index.html

index.html (modificado de nuevo):

<!DOCTYPE html>

<html>

<head>

<title>Servidor web 1</title>

</head>

<body>

<h1>Bienvenido al servidor web 1 REQUETE DEFINITIVO</h1>

</body>

</html>

- Sube el index.html

**Nota**: En el caso de que no te deje subir el archivo index.html, accede al contenedor, ***servidorFtp2***, y establecer como usuario y grupo propietario del directorio raíz ,/srv, a ftp:ftp al igual que hiciste en la 1ª y 2ª parte

- Establece las reglas DNAT adecuada en el router mikrotik para que el profesor pueda acceder a tus sitios: FTP (***servidorFTP2***) y WEB (***servidorWeb1***). Cuando hayas configurado de manera adecuada la reglas DNAT avisa al profesor para que verifique su funcionamiento

- Una vez haya realizado las comprobaciones adecuadas el profesor, ubícate en los directorios respectivos ·$HOME/servidorWeb1 y $HOME/servidorFtp y realiza las siguientes acciones:

Detén ambos escenarios.

**Pista**: docker-compose s….

- Borra ambos escenarios

**Pista**: docker-compose r….

**PRÁCTICA FINALIZADA**