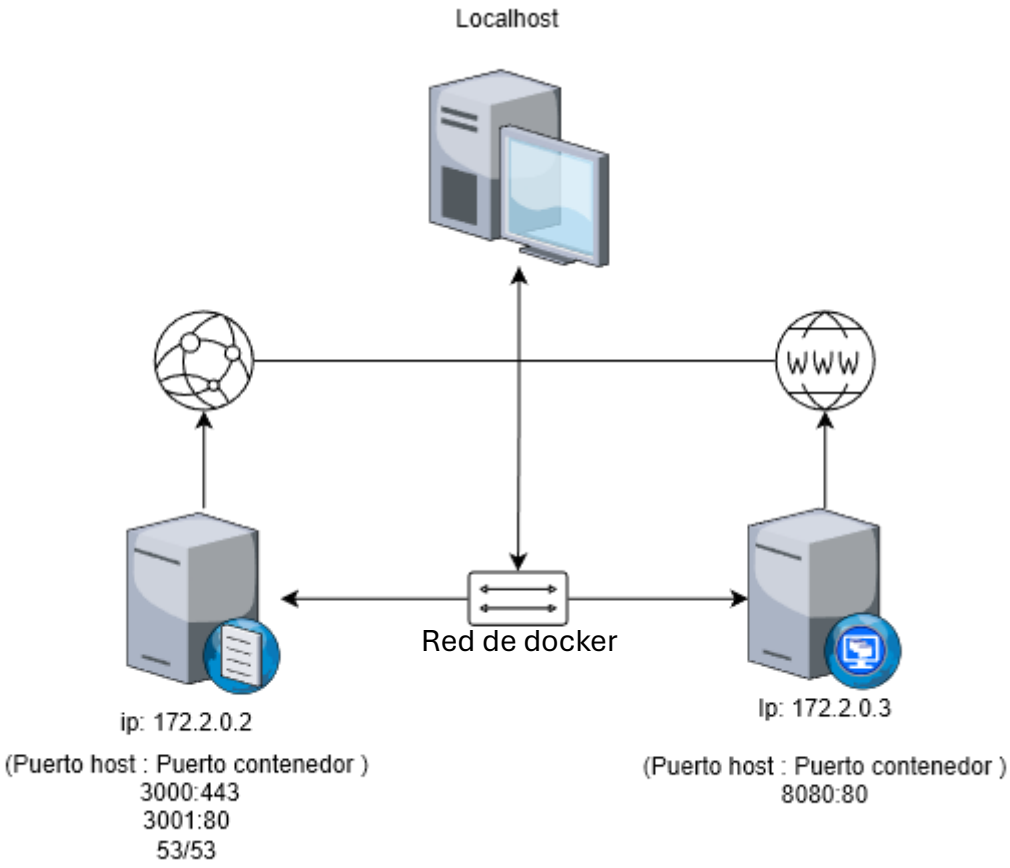
PROYECTO DOCKER

Por: Carlos Pilar

índice

1. Arquitectura de red



Esta es la arquitectura de red que usan los contenedores, primero el host que hace de cliente y es el que aloja los contenedores, después el dns que tiene la ip fija dentro del bridge de Docker **172.2.0.2** y usa los puertos **3000, 3001** para el **80** y **443** dentro del contenedor donde se puede ver el panel de administración web,

Luego el servidor web que usa el puerto **8080** del host que redirecciona al **80** y obtiene su ip por el dhcp de Docker también este servidor se le da el nombre de dominio **mywebserver.local** que es resuelto por el DNS.

2. Estructura del docker-compose.

```yaml
#docker compose de los contenedores.
services:

  #contenedor de dns pihole
  pihole:
    container_name: pihole-dns
    image: pihole/pihole:latest
    ports:
      # DNS Ports
      - "53:53/tcp"
      - "53:53/udp"
      # Puerto http que usara el contenedor usara la 30001
      - "30001:80/tcp"
      # Puertos por defecto para la interfaz de administracionweb usara la 30000
      - "30000:443/tcp"
    environment:
      # Esta es la Time Zone que usa el dns
      TZ: 'Europe/Madrid'
      # Pone una contraseña a la interfaz web
      FTLCONF_webserver_api_password: 'admin123'
    # Volumenes, se usaran para guardar los datos para que se mantengas entre las diferentes actualizaciones
    volumes:
      # Para que se mantengas los archivos de la base de datos de pihole
      - './etc-pihole:/etc/pihole'
    networks:
       mi_red:
        ipv4_address: 172.2.0.2 #esta es la ip fija que va a usar el dns para que el ubuntu pueda preguntarle por nombre de dominio dentro de la red
    restart: unless-stopped
  # Servicio que ejecuta un servidor web Apache
  apache:
    image: httpd:latest # Imagen oficial de Apache HTTP Server
    container_name: apache # Nombre del contenedor
    ports:
      - 8080:80 # Expone el puerto 80 del contenedor como 8080 en el host
    volumes:
      - ./web:/usr/local/apache2/htdocs # Carpeta local "web" servida como raíz del sitio weba aqui puedes poner tus paginas para que las muestre tu servidor
    cap_add:
      # El valor net admin para que sea adminitrador de la red
      - NET_ADMIN
    networks:
       mi_red: # Conecta este contenedor a la misma red que Windows
        ipv4_address: 172.2.0.3 #esta es la ip fija que va a usar el servidor que tiene el nombre personalizado en el dns
    dns:
      - 172.2.0.2 # aqui usa como dns el contenedor de pihole
    restart: unless-stopped
# Definición de la red personalizada de Docker
networks:
  mi_red:
    driver: bridge
    ipam:
      config:
        - subnet: 172.2.0.0/16 #esta es la direccion que usaran los contenedores dentro de la red mired
```

En el Docker compose primero se crea el contenedor de **pihole** que es el **dns**, se configuran todos los parámetros necesarios para su configuración como ip fija, contraseña de el panel de administración, puertos, ruta de archivos de configuración …

Después de crea el contenedor de apache usando la imagen **httpd** que es un Ubuntu que viene con apache instalado y configurado, a este se le da una ip fija también para luego darle un cname en el dns y por ultimo en networks: se establece la red que creo para que estos contenedores se comuniquen entre ellos.

Ambos contenedores se crean con **restart: unless-stopped** lo que hace esto es que los contenedores se reinicien siempre salvo que se apaguen manualmente

3. Como crear los contenedores

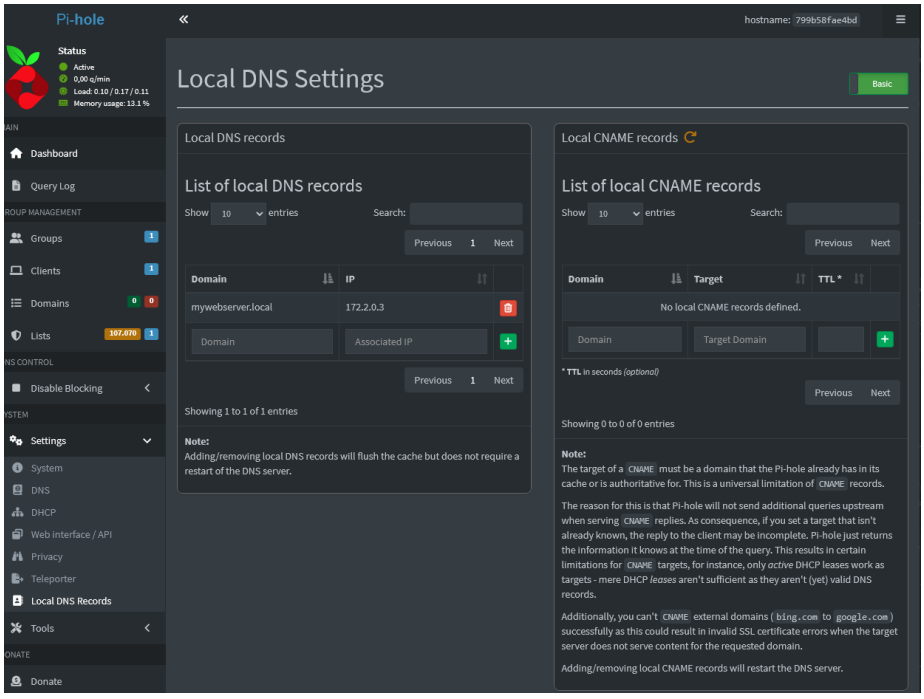
El primer paso para crear los contenedores es abrir una terminal dentro de la carpeta que tenemos el Docker-compose.yml y usamos el comando **Docker-compose up -d .**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\Usuario\OneDrive - Educacyl\2° DAW\despliege de app web\Trabajo-Docker> docker compose up -d
```

Una vez ejecutado el comando podemos ver que descarga las imágenes y las inicia con su configuración correspondiente.

```
PS C:\Users\Usuario\OneDrive - Educacyl\2° DAW\despliege de app web\Trabajo-Docker> docker compose up -d
[+] Running 18/18
 ✔pihole Pulled                                                                  4.3s
   ✔23a93ec442d6 Pull complete                                                   2.8s
   ✔884daf2d2109 Pull complete                                                   0.4s
   ✔88219ec7dad4 Pull complete                                                   0.2s
   ✔2a7ca71b9ea5 Pull complete                                                   0.4s
   ✔c99479331001 Pull complete                                                   0.7s
   ✔f9086e5091f3 Pull complete                                                   2.7s
   ✔4fa2e90b6d8d Pull complete                                                   0.5s
   ✔2d35ebdb57d9 Pull complete                                                   1.1s
   ✔07951c985d56 Pull complete                                                   2.5s
   ✔5a5ecbda1d12 Pull complete                                                   0.5s
 ✔apache Pulled                                                                  4.8s
   ✔fab98c44430d Pull complete                                                   0.5s
   ✔233dec01418c Pull complete                                                   0.2s
   ✔13c22d886563 Pull complete                                                   3.2s
   ✔4870f70a8556 Pull complete                                                   3.4s
   ✔4f4fb700ef54 Pull complete                                                   0.1s
   ✔d7ecded7702a Pull complete                                                   3.1s
[+] Running 3/3
 ✔Network trabajo-docker_mi_red  Created                                         0.0s
 ✔Container pihole-dns           Started                                         1.7s
 ✔Container apache               Started                                         1.5s
PS C:\Users\Usuario\OneDrive - Educacyl\2° DAW\despliege de app web\Trabajo-Docker>
```

Ahora para asignar un nombre de dominio a nuestro servidor web lo que tenemos que hacer es acceder al panel de administración de pihole y configurar allí el nombre asignándole la ip de nuestro contenedor.



4. comprobación de los servicios.

Lo primero vamos a probar a ver si pihole resuelve el nombre correctamente:

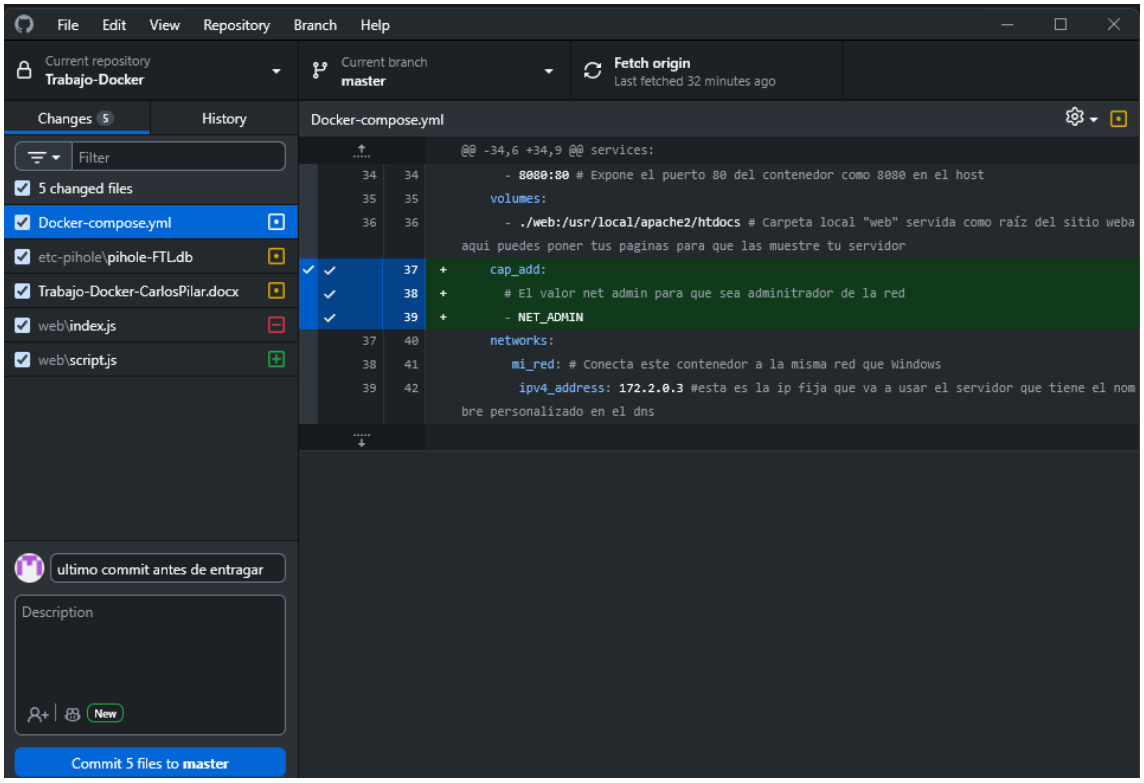
Este es un nslookup dentro del el servidor apache, se puede ver que responde con la ip correcta cuando se pregunta por el nombre de dominio

```
# nslookup mywebserver.local
Server:         127.0.0.11
Address:        127.0.0.11#53

Name:   mywebserver.local
Address: 172.2.0.3
```

Para probar el servidor web he metido una pagina en la carpeta web que se crea al crear el contenedor que es donde se albergan las páginas que va a mostrar el servidor. Y accediendo desde nuestro host podemos ver que se muestra la pagina correctamente

5. documentación con git y github.

El repositorio del trabajo lo e iniciado con la aplicación de github desktop y hay e ido haciendo los commit:



Y para subirlo github simplemente le doy al push y ya lo puedo ver en mi github.

Link: https://github.com/CarlosJP2004/Trabajo_Docker