



Ejercicio 4

CIFP La Laboral - Módulo Despliegue de Aplicaciones Web

 **Autores:** Pelayo Rodríguez e Iker Pérez

 **Fecha de entrega y exposición:** Viernes, 21 de febrero de 2025.

 **Repositorio GitHub:**



ÍNDICE

1. Crear el archivo docker-compose.yaml
 - 1.1. Con comandos
 - 1.2. Sin comandos
2. Iniciar el contenedor con Docker Compose
 - 2.1. Verificación del contenedor en ejecución
 - 2.2. Acceso al contenedor
3. Detener y eliminar el contenedor
 - 3.1. Parar la imagen
 - 3.2. Eliminar contenedores

Ejercicio 4: Docker Compose

Crear el archivo `docker-compose.yaml` con comandos

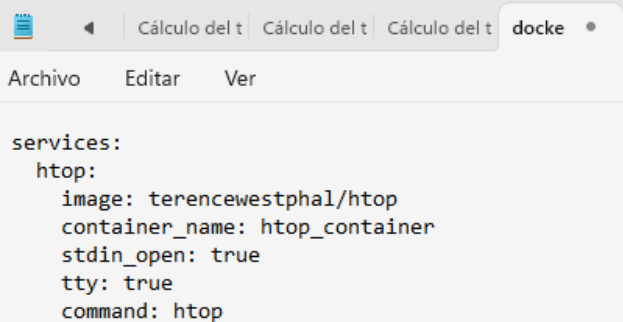
Abrimos una terminal y creamos una nueva carpeta para el proyecto, dentro de esta carpeta, creamos el archivo `docker-compose.yaml`. Para esto, usamos bloc de notas(o cualquier otro editor de texto):

```
$mkdir htop-docker && cd htop-docker  
$notepad docker-compose.yaml
```

```
C:\Windows\System32\htop-docker>docker version  
Client:  
Version:      27.4.0  
API version:  1.47  
Go version:   go1.22.10  
Git commit:   bde2b89  
Built:        Sat Dec  7 10:40:21 2024  
OS/Arch:      windows/amd64  
Context:      desktop-linux  
  
Server: Docker Desktop 4.37.1 (178610)  
Engine:  
Version:      27.4.0  
API version:  1.47 (minimum version 1.24)  
Go version:   go1.22.10  
Git commit:   92a8393  
Built:        Sat Dec  7 10:38:57 2024  
OS/Arch:      linux/amd64  
Experimental: false  
containerd:  
Version:      1.7.21  
GitCommit:    472731909fa34bd7bc9c087e4c27943f9835f111  
runc:  
Version:      1.1.13  
GitCommit:    v1.1.13-0-g58aa920  
docker-init:  
Version:      0.19.0  
GitCommit:    de40ad0
```

```
C:\Windows\System32\htop-docker>notepad docker-compose.yaml
```

```
C:\Windows\System32\htop-docker>
```



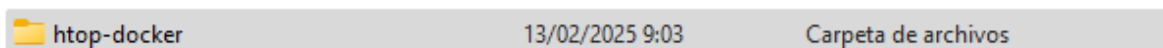
```
services:  
  htop:  
    image: terencewestphal/htop  
    container_name: htop_container  
    stdin_open: true  
    tty: true  
    command: htop
```

Pegamos el siguiente contenido en el archivo y luego lo guardamos y cerramos el bloc de notas:

```
services:
  htop:
    image: terencewestphal/htop
    container_name: htop_container
    stdin_open: true
    tty: true
    command: htop
```

Crear el archivo `docker-compose.yaml` sin comandos

- Abrimos el **Explorador de archivos**.
- Creamos una carpeta llamada **htop-docker** en la misma ubicacion que lo estabamos haciendo con los comandos.



Abrimos

Visual Studio Code, **Bloc de notas**, o cualquier editor de texto y escribimos:

A screenshot of the Visual Studio Code editor. The title bar shows 'docker-compose.yaml'. The editor window displays the following YAML content:

```
1 services:
2   htop:
3     image: terencewestphal/htop
4     container_name: htop_container
5     stdin_open: true
6     tty: true
7     command: htop
```

Iniciar el contenedor con Docker Compose

Ejecutamos el siguiente comando para iniciar

htop en un contenedor y verificamos que el contenedor está corriendo:

```
$docker-compose up -d
$docker ps
```

```
C:\Windows\System32\http-docker\http-docker>docker-compose up -d
[+] Running 1/1
✔ Container http_container Started 0.2s

C:\Windows\System32\http-docker\http-docker>docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED        STATUS        PORTS        NAMES
ecb37130dce7   terencewestphal/http     "http http"            14 seconds ago Up 14 seconds        http_container
```

```
$docker exec -it http_container htop
```

```
1 [ | 1.3%] 5 [ 0.0%] 9 [ 0.0%] 13 [ 0.0%]
2 [ | 1.3%] 6 [ 0.0%] 10 [ 0.0%] 14 [ 0.0%]
3 [ 0.0%] 7 [ 0.0%] 11 [ 0.0%] 15 [ 0.0%]
4 [ 0.0%] 8 [ 0.0%] 12 [ 0.0%] 16 [ 0.0%]
Mem[|||||] 660M/15.6G Tasks: 2, 0 thr; 1 running
Swp[ ] 0K/4.00G Load average: 0.02 0.03 0.00
Uptime: 00:30:01

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
7 root 20 0 3668 1192 932 R 0.0 0.0 0:00.01 http
1 root 20 0 3528 1228 976 S 0.0 0.0 0:00.06 htop htop

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

Detener y eliminar el contenedor

Primero, paramos la imagen y la borramos:

Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

Local Hub repositories

6.88 MB / 465.32 MB in use 3 Images Last refresh: 43 minutes ago


Q Search


<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	○ mariadb	lts-ubi9	90e4bcac427b	8 days ago	465.31 MB	▶ ⋮ 🗑
<input type="checkbox"/>	● hello-world	latest	74cc54e27dc4	22 days ago	10.07 KB	▶ ⋮ 🗑
<input type="checkbox"/>	● terencewestphal/http	latest	d317eb56ee39	7 years ago	6.87 MB	▶ ⋮ 🗑

Y luego una vez paradas/borradas podemos borrar el conte



Containers [Give feedback](#)














View all your running containers and applications. [Learn more](#)

Container CPU usage 
0.05% / 1600% (16 CPUs available)

Container memory usage 
816KB / 15.19GB

[Show charts](#)

  Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	 sad_galileo	1db91595bbe9	hello-world		0%	7 days ago	  
<input type="checkbox"/>	  htop-docker	-	-	-	0.05%	14 minutes ago	  
<input type="checkbox"/>	 htop_container	ecb37130dce7	terencewestphal/htop		0.05%	14 minutes ago	  

1 2