

# Ejercicio 4 - Docker Compose

Proyecto Docker: Sara García Barbas y Gerald Alexis Rueda Tejedo

[Introducción](#)

[¿Qué es Docker Compose?](#)

[¿Qué es `htop`?](#)

[Paso 1](#)

[Paso 2](#)

[Paso 3](#)

## Introducción

En este tutorial desplegaremos `htop` en un contenedor Docker utilizando **Docker Compose**. `htop` es una herramienta interactiva que permite monitorear procesos en sistemas basados en Unix de manera visual e intuitiva.

Docker Compose facilita la gestión de contenedores, permitiendo definir y ejecutar aplicaciones multicontenedor con un solo archivo de configuración. Configurar `htop` en un contenedor es útil para monitorear el rendimiento de un sistema de forma aislada o dentro de un entorno de desarrollo.

Al finalizar este tutorial, tendremos un contenedor con `htop` en ejecución y podremos usarlo para visualizar el consumo de CPU, memoria y otros recursos en tiempo real.

## ¿Qué es Docker Compose?

**Docker Compose** es una herramienta que permite definir y gestionar aplicaciones multi-contenedor en **Docker** utilizando un solo archivo de configuración, generalmente llamado `docker-compose.yml`.

Con **Docker Compose**, en lugar de ejecutar y configurar cada contenedor manualmente con comandos de `docker run`, podemos definir todos los servicios, volúmenes, redes y variables de entorno en un archivo YAML. Luego, con un solo comando (`docker-compose up`), Docker se encarga de construir y ejecutar toda la aplicación.

## ¿Qué es `htop` ?

**htop** es una herramienta de monitoreo de procesos interactiva para sistemas **Unix** (Linux, macOS, BSD) que permite visualizar en tiempo real el uso de CPU, memoria, procesos y otros recursos del sistema. Es una alternativa más avanzada y amigable a `top`, con una interfaz más intuitiva y colorida.

## Paso 1

En nuestra máquina virtual abrimos un terminal y comprobamos si tenemos `docker-compose` instalado

```
docker-compose --version
```

```
cliente@clienteLinux ~$ docker-compose --version
No se ha encontrado la orden «docker-compose», pero se puede instalar con:
sudo snap install docker          # version 27.2.0, or
sudo apt install docker-compose  # version 1.29.2-1
Consulte «snap info docker» para ver más versiones.
cliente@clienteLinux ~$
```

Si no lo tenemos instalado, ejecutaremos:

```
sudo apt install docker-compose -y
```

```
cliente@clienteLinux ~$ sudo apt install docker-compose -y
[sudo] contraseña para cliente:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
docker-ce docker-ce-cli python3-attr python3-distutils python3-docker
python3-dockerpty python3-docopt python3-dotenv python3-jjsonschema
python3-lib2to3 python3-pyrsistent python3-setuptools python3-texttable
python3-websocket
```

Si comprobamos la versión, vemos que ahora ya si lo tenemos instalado:

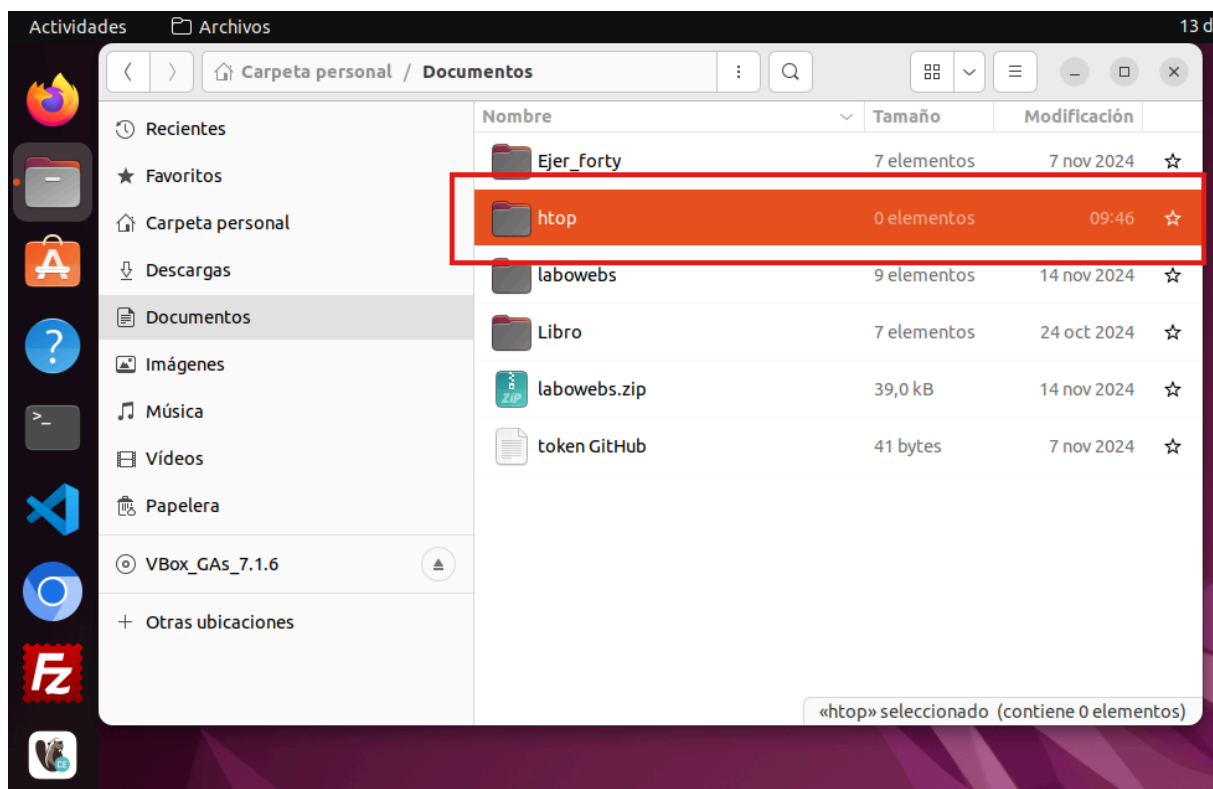
```
docker-compose --version
```

```
cliente@clienteLinux ~$ docker-compose --version
docker-compose version 1.29.2, build unknown
cliente@clienteLinux ~$
```

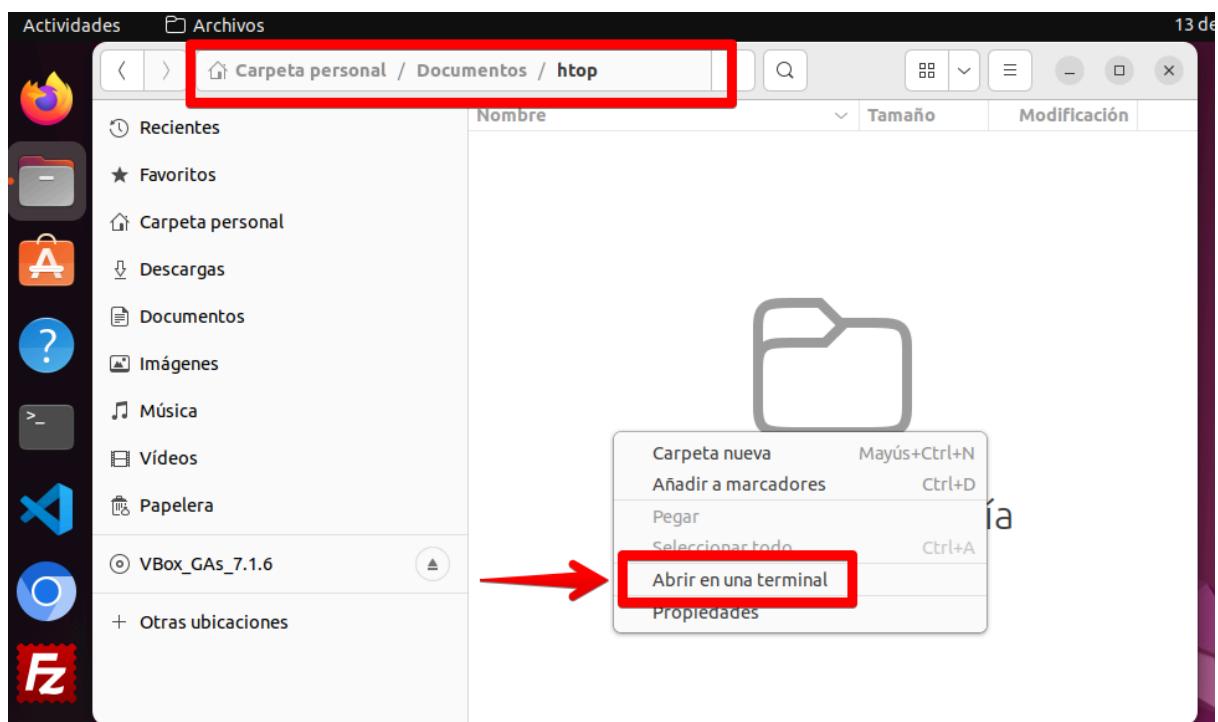
## Paso 2

En nuestra máquina virtual creamos un nuevo directorio al que llamaremos

`http`, y este será sobre el que trabajaremos. Yo he creado el directorio en `/home/cliente/Documentos/http`

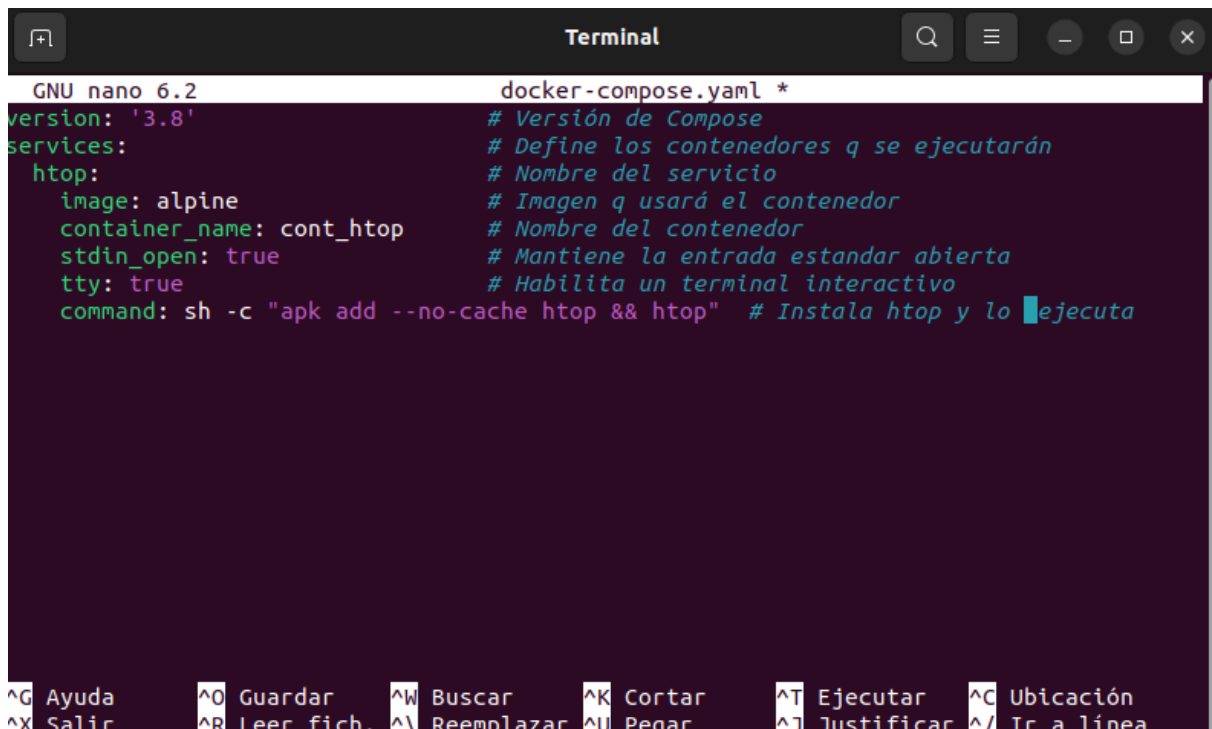


Desde el directorio que acabamos de crear, abrimos un terminal



Vamos a crear nuestro archivo `compose.yaml` y añadirle el siguiente contenido:

nano docker-compose.yaml



```
GNU nano 6.2 docker-compose.yaml *
version: '3.8'
services:
  htop:
    image: alpine
    container_name: cont_htop
    stdin_open: true
    tty: true
    command: sh -c "apk add --no-cache htop && htop" # Instala htop y lo ejecuta
```

## ¿Por qué la imagen Alpine?

Alpine Linux es una distribución de Linux minimalista y liviana, ideal para contenedores Docker por lo siguiente:

- **Es pequeña** (~5 MB), reduciendo así el tiempo de descarga y arranque.
- **Es segura.**
- **Es eficiente**, usa `musl` y `busybox` en lugar de `glibc` y herramientas más pesadas.

## ¿Qué es Alpine?

Es un sistema operativo basado en Linux, diseñado para ser **ligero y seguro**. Se usa mucho en Docker porque consume menos recursos que distribuciones como Ubuntu o Debian.

La combinación de **Alpine Linux** con **htop** en Docker es útil por:

1. **Ligereza:** Alpine es una imagen mínima (~5 MB), lo que hace que el contenedor sea rápido y eficiente.

2. **Monitorización sin sobrecarga:** `htop` permite visualizar el uso de CPU, memoria y procesos sin consumir muchos recursos.
3. **Simplicidad:** Solo necesitamos instalar `htop` con `apk add --no-cache htop` y ejecutarlo.

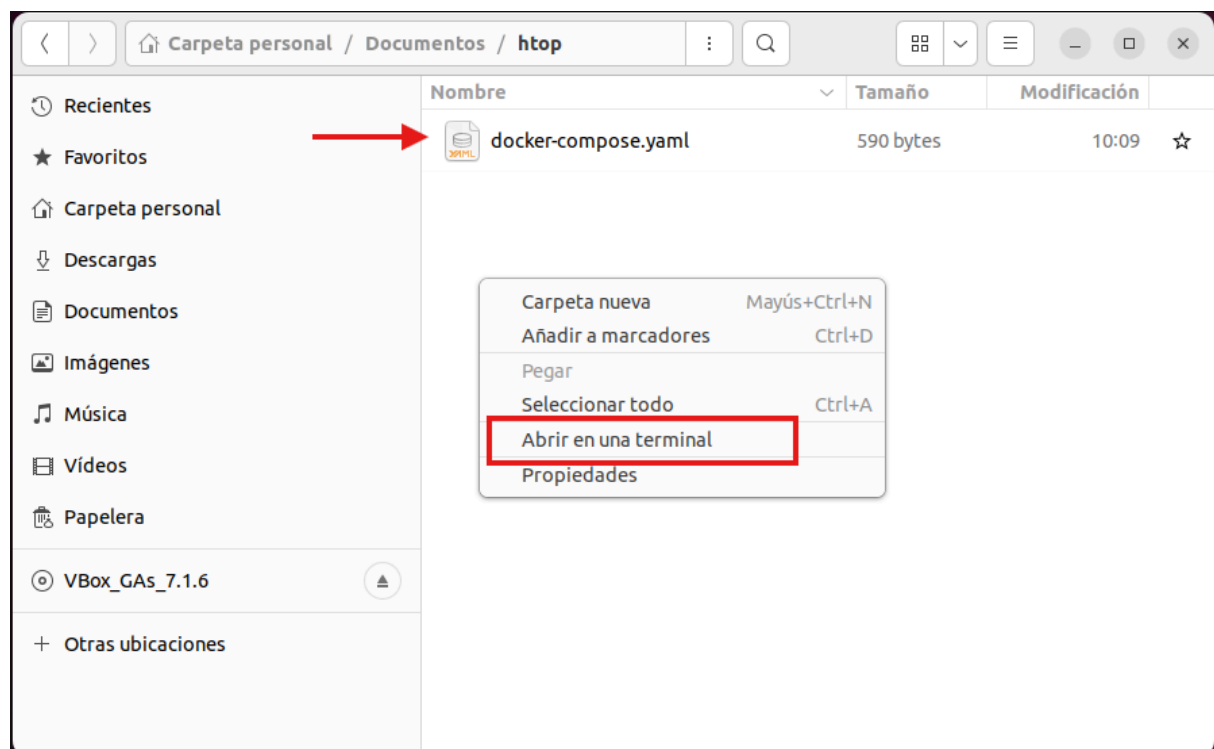
Esta combinación es ideal para depurar o monitorear procesos dentro de un contenedor sin inflar su tamaño.

### ¿No podemos usar solo `htop` ?

No se puede usar `htop` solo en Docker porque **necesita un sistema operativo base**. Docker no ejecuta aplicaciones de forma independiente, sino dentro de contenedores basados en imágenes de sistemas como **Alpine, Ubuntu o Debian**.

## Paso 3

Abrimos un terminal en la carpeta donde guardamos el archivo `docker-compose.yml` y ejecutamos:



```
docker-compose up
```

```
cliente@clienteLinux ~/Documentos/htop$ docker-compose up
Pulling htop (alpine:)...
latest: Pulling from library/alpine
1f3e46996e29: Downloading [>] 36.88k
1f3e46996e29: Downloading [=====] 2.568M
1f3e46996e29: Extracting [>] 65.54kB
1f3e46996e29: Extracting [=====] 1.376MB
1f3e46996e29: Extracting [=====] 3.642MB
1f3e46996e29: Extracting [=====] 3.642MB
1f3e46996e29: Pull complete
Digest: sha256:56fa17d2a7e7f168a043a2712e63aed1f8543aeafdcee47c58dcffe38ed51099
Status: Downloaded newer image for alpine:latest
Creating cont_htop ... done
Attaching to cont_htop
cont_htop | fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
cont_htop | fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
```

Ahora ejecutamos:

```
docker ps
docker exec -it 6b149a8f1cee htop
```

```
cliente@clienteLinux ~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
6b149a8f1cee	alpine	"sh -c 'apk add --no..."	31 minutes ago	Up About a minute

```
cliente@clienteLinux ~$ docker exec -it 6b149a8f1cee htop
```

Se nos iniciará la interfaz de `htop` y ya podremos usarlo:

```
0[| 0.7%] Tasks: 2, 0 thr, 0 kthr; 0 running
1[| 0.7%] Load average: 0.00 0.03 0.05
2[|| 4.3%] Uptime: 01:11:57
Mem[||||||| 1.17G/15.6G]
Swp[ 0K/2.00G]

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1 root 20 0 1772 1408 1024 S 0.0 0.0 0:00.42 htop
14 root 20 0 1776 1408 1024 R 0.0 0.0 0:00.03 htop

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

Si presionamos la tecla F1 o sobre el texto Help veremos las distintas funcionalidades de htop y las teclas asociadas a dichas funcionalidades:



```
Terminal
htop 3.3.0 - (C) 2004-2019 Hisham Muhammad. (C) 2020-2024 htop dev team.
Released under the GNU GPLv2+. See 'man' page for more info.

CPU usage bar: [low/normal/kernel/guest          used%]
Memory bar:    [used/shared/compressed/buffers/cache used/total]
Swap bar:      [used/cache/frontswap             used/total]

Type and layout of header meters are configurable in the setup screen.

Process state: R: running; S: sleeping; t: traced/stopped; Z: zombie; D: disk sleep

# : hide/show header meters          S-Tab: switch to previous screen tab
Tab: switch to next screen tab        Space: tag process
Arrows: scroll process list           c: tag process and its children
Digits: incremental PID search        U: untag all processes
F3 /: incremental name search          F9 k: kill process/tagged processes
F4 \: incremental name filtering        F7 ]: higher priority (root only)
F5 t: tree view                       F8 [: lower priority (+ nice)
p: toggle program path                a: set CPU affinity
m: toggle merged command              e: show process environment
Z: pause/resume process updates       i: set IO priority
u: show processes of a single user    l: list open files with lsof
H: hide/show user process threads    x: list file locks of process
K: hide/show kernel threads          s: trace syscalls with strace
O: hide/show processes in containers w: wrap process command in multiple lines
F: cursor follows process            Y: set scheduling policy
+ - *: expand/collapse tree/toggle all F2 C S: setup
N P M T: sort by PID, CPU%, MEM% or TIME F1 h ?: show this help screen
I: invert sort order                 F10 q: quit
F6 > .: select sort column

Press any key to return.
```

Las principales acciones que podremos realizar son:

### 1. Ver el uso del sistema

- **CPU:** Muestra el uso de cada núcleo de la CPU en tiempo real con gráficos de barras.
- **Memoria:** Visualizar el uso de la RAM y swap.
- **Carga del sistema:** Muestra la carga promedio del sistema (load average).

### 2. Gestionar procesos

- **Ver procesos activos:** Muestra una lista completa de los procesos en ejecución, con detalles como PID, usuario, uso de CPU y memoria.
- **Ordenar procesos:** Podremos ordenar los procesos por uso de CPU, memoria, nombre, etc. (Tecla F6).

### 3. Buscar y filtrar procesos

- **Buscar un proceso:** Presionando F3 y escribiendo el nombre del proceso para buscarlo.

- **Filtrar procesos:** Usar F4 para aplicar filtros a los procesos que se están mostrando, por ejemplo, por nombre.

#### 4. Matar procesos

- **Matar un proceso:** Seleccionando un proceso y presionando F9 para terminarlo (se pueden elegir diferentes señales para terminar el proceso, como SIGTERM o SIGKILL).

#### 5. Ver la jerarquía de procesos

- **Modo árbol:** Cambia la vista para mostrar los procesos en forma de árbol jerárquico con F5, lo que ayuda a ver qué procesos son padres e hijos.

#### 6. Cambiar la prioridad de los procesos

- **Cambiar la prioridad:** Usando F7 y F8 para ajustar la prioridad de un proceso (cambiar su "nice value").

#### 7. Personalización

- **Configurar la visualización:** Presionando F2 abriremos el menú de configuración, donde podremos cambiar cómo se muestran los procesos y ajustar las opciones de visualización.

#### 8. Ver estadísticas detalladas

- También podremos ver detalles sobre el uso del sistema, como las interrupciones, los contextos de los procesos, el uso de discos, etc.