



Ejercicio 1

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



ÍNDICE

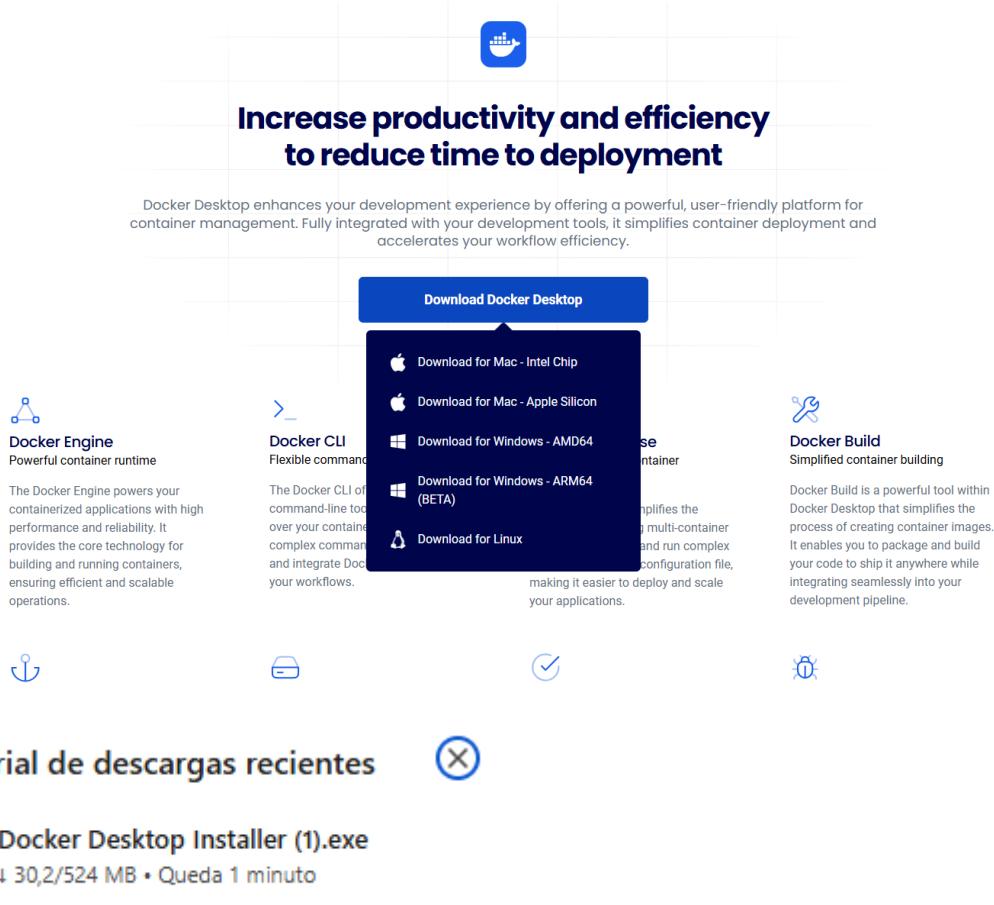
- [Ejercicio 1: Manual de Docker Desktop](#)
- [Proceso paso a paso de instalación](#)
- [Navegación por la interfaz principal](#)
- [Operaciones básicas con contenedores](#)
- [Gestión de imágenes Docker](#)
 - [Configuración de redes y volúmenes](#)
 - [Herramientas de diagnóstico](#)
- [**Disk Usage \(Uso de Disco\)**](#)
- [**Logs Explorer \(Explorador de Logs\)**](#)



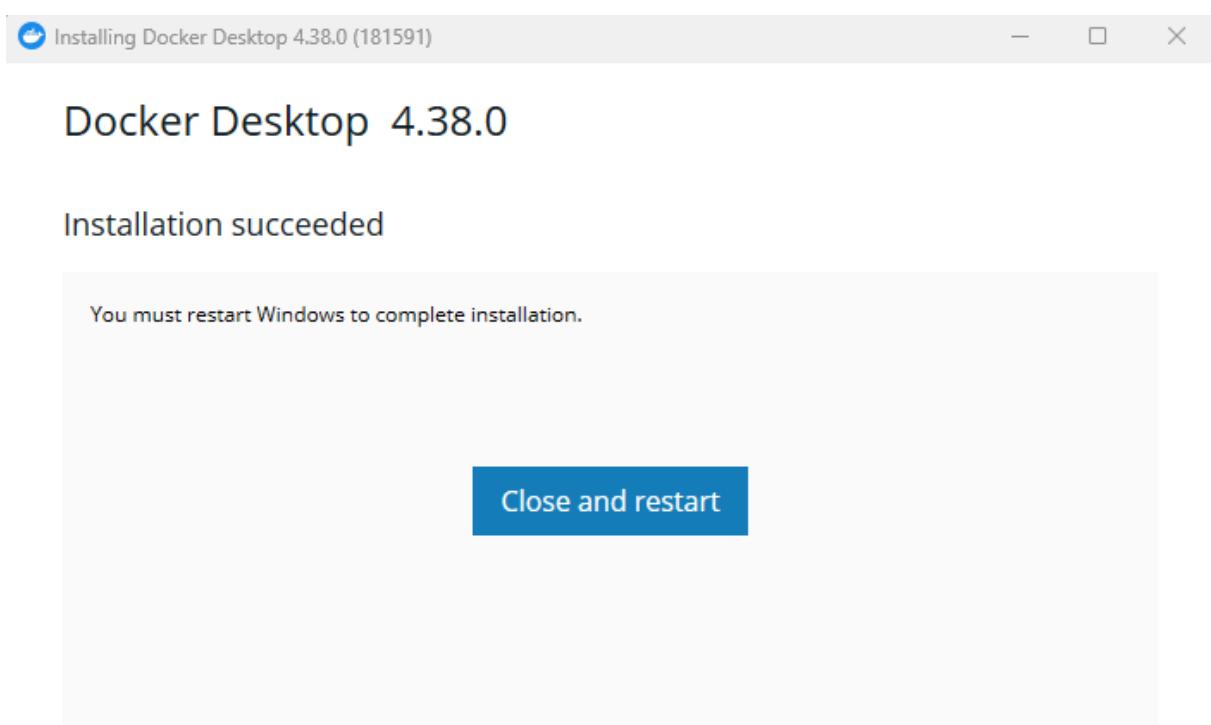
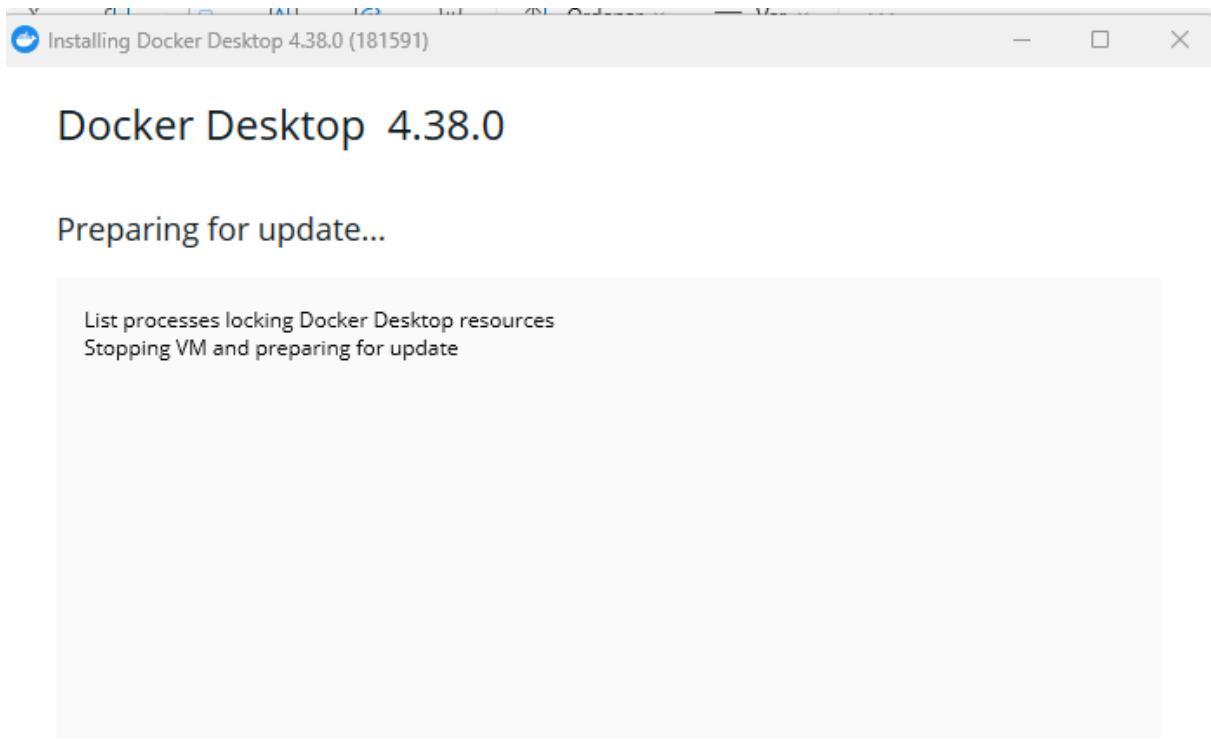
Ejercicio 1: Manual de Docker Desktop

Proceso paso a paso de instalación

- Abrimos el navegador y accedemos a la página oficial de Docker:
 [Página oficial de Docker Desktop](#)
- Bajamos un poco pinchamos en Download Docker Desktop y lo descargamos para Windows.

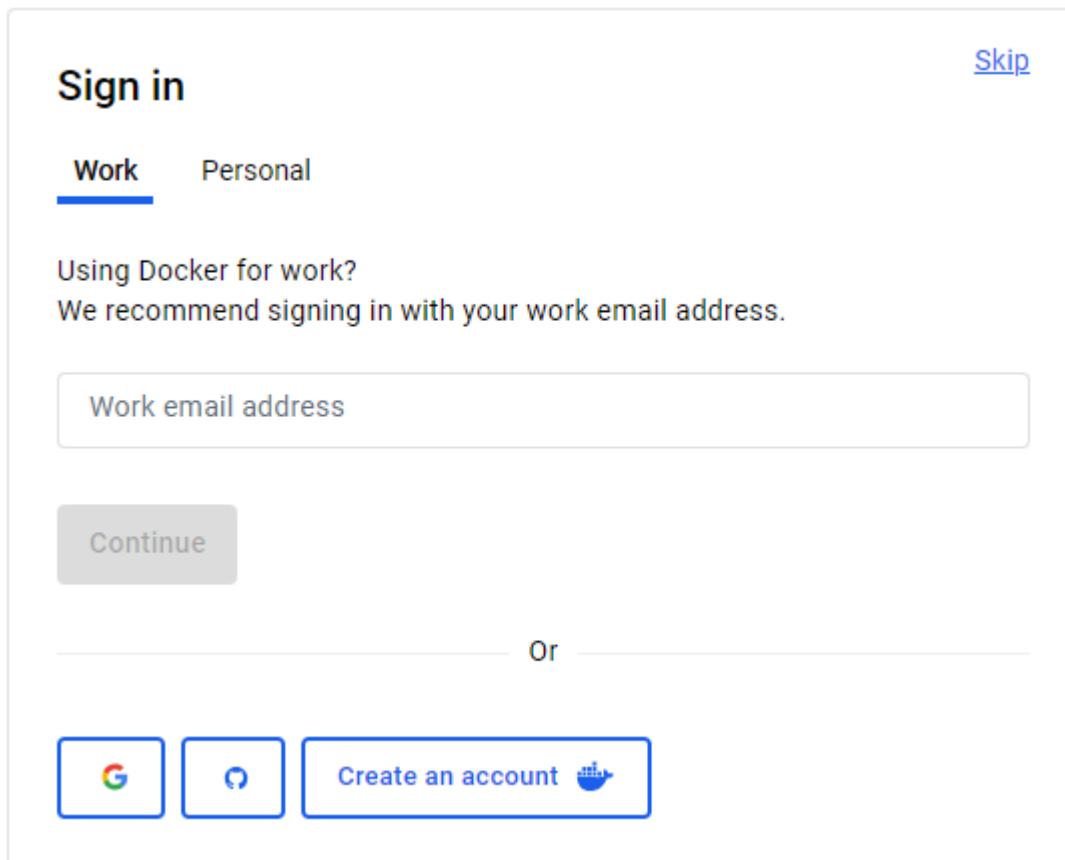


Una vez descargado, pinchamos en el instalador y comenzamos con la descarga:



Una vez realizada la instalación, accedemos a Docker Desktop:

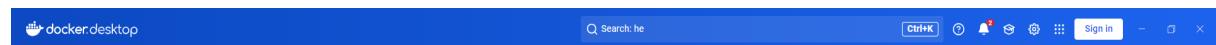
Nada mas entrar nos pedirá iniciar sesión o registrarnos:



Una vez hecho, ya accedemos a Docker Desktop y a todas sus funcionalidades:

The screenshot displays the Docker Desktop application window. On the left is a sidebar with various icons and links: Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, Extensions (Manage), Disk usage, Logs Explorer, and Portainer (DEB). The main area is titled 'Containers' with a 'Give feedback' link. It says 'View all your running containers and applications' and 'Learn more'. Below this, there are sections for 'Container CPU usage' and 'Container memory usage', both stating 'No containers are running.' There is also a 'Show charts' link. A search bar and a filter button ('Only show running containers') are present. A table lists three containers: 'happy_engelbart' (nginx:latest, 2 days ago), 'angry_moser' (hello-world, 2 days ago), and 'http-docker' (status unknown, N/A). Each row has an 'Actions' column with three icons: a triangle, a vertical ellipsis, and a trash can.

💻 Navegación por la interfaz principal



Containers Give feedback

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

Container CPU usage (1) Container memory usage (1)

No containers are running.

Search Only show running containers

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
angry_moser	hello-world	Exited	N/A	1 hour ago		

[Show charts](#)

Images Give feedback

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

Local Hub repositories

10.07 KB / 0 Bytes in use 1 images Last refresh: 1 hour ago

Search

Name	Tag	Status	Created	Size	Actions
hello-world	latest	In use	21 days ago	10.07 KB	

Volumes Give feedback

Manage your volumes, view usage, and inspect their contents. [Learn more](#)

Containers can use volumes to store data

All data in a container is lost once it is removed. Containers use volumes to persist data.

[Create a volume](#)

Builds Give feedback

Build container images and artifacts from source code. [Learn more](#)

Selected builder: desktop-linux [Import builds](#) [Builder settings](#)

Build history Active builds

Show only my builds

ID	Name	Builder	Status	Duration	Created	Author	
	No build record found	If you can't see your builds here make sure your builder instances are running.					

Docker Scout [Give feedback](#)

Advanced image analysis with Docker Scout

Want to use Docker Scout on your remote repositories? [Set up your integrations now](#)

Understand your application's dependencies, analyze the vulnerabilities, and act quickly with suggested remediation options. [Learn more](#) and [upgrade](#).

Sample image Vulnerabilities

hello-world:latest Not analyzed [Analyze image](#)

Advanced image analysis can be accessed by viewing any of the images on the [Images view](#).

How to access Advanced image analysis

[docker desktop](#)

Local Docker Engine

- Containers
- Images
- Volumes
- Builds
- Dev Environments
- Docker Scout
- Docker Run Cloud
- Extensions

Images [Give feedback](#)

Local Hub

Search Content Categories Sort by Recently added

Name	Tag	Created	Last updated	Image ID
docker/scout-demo-service	fix	22 days ago	22 days ago	6657e9d8214f
docker/scout-demo-service	main	22 days ago	47.50 MB	6657e9d8214f

Last refresh: 2 minutes ago

Manage Extensions [Give feedback](#)

Docker Extensions let you use third-party tools within Docker Desktop to extend its functionality. [Learn more](#)

My Extensions [Browse](#) [Create](#) [Request an extension](#)

Search

Extension	Author	Description	Reviews	Action
SQL container manager	DrewSt.Tech • drewst/docker-sql-extension:0.2.0	Create, connect, and manage SQL dev containers	Not reviewed ↓ 7.3K	Install ⋮
VNC Viewer	pgmystery • pgmystery/docker-extension-vnc:1.1.0	Docker Desktop Extension for connecting to a VNC Server Container or Remote Host and control it over an built-in-view.	Not reviewed ↓ 4.0K	Install ⋮
Daytona	Daytona • daytonaio/docker-extension:0.52.1	Docker Extension for using Daytona.	Not reviewed ↓ 1.1K	Install ⋮
KRS	Kubetools • kubetools/krs-docker-extension:0.0.3	A GenAI-powered Kubetools Recommender System for Kubernetes clusters.	Not reviewed ↓ 2.5K	Install ⋮
Plate Recognizer Installer	Plate Recognizer • platerecognizer/installer:0.0.1	Plate Recognizer Installer for Stream and Snapshot SDKs	Not reviewed ↓ 800	Install ⋮

Engine running | RAM: 0.93 GB CPU: 0.00% Disk: 1.08 GB used (limit 1006.85 GB) Terminal [New version available](#)

📦 Operaciones básicas con contenedores

Ver Contenedores Disponibles

- Abrimos **Docker Desktop**.
- En la barra lateral izquierda, seleccionamos **Containers**.
- Se muestra una lista con los contenedores creados, indicando su estado (corriendo, detenido, etc.).

Iniciar un Contenedor Detenido

- Ubicamos el contenedor en la lista.
- Si el estado es "Exited", hacemos clic en los tres puntos (:) a la derecha del contenedor.
- Seleccionamos Restart para reiniciarlo.

```
2025-02-12 10:13:19 Hello from Docker!
2025-02-12 10:13:19 This message shows that your installation appears to be working correctly.
2025-02-12 10:13:19
2025-02-12 10:13:19 To generate this message, Docker took the following steps:
2025-02-12 10:13:19 1. The Docker client contacted the Docker daemon.
2025-02-12 10:13:19 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
2025-02-12 10:13:19   (amd64)
2025-02-12 10:13:19 3. The Docker daemon created a new container from that image which runs the
2025-02-12 10:13:19   executable that produces the output you are currently reading.
2025-02-12 10:13:19 4. The Docker daemon streamed that output to the Docker client, which sent it
2025-02-12 10:13:19   to your terminal.
2025-02-12 10:13:19
2025-02-12 10:13:19 To try something more ambitious, you can run an Ubuntu container with:
2025-02-12 10:13:19 $ docker run -it ubuntu bash
2025-02-12 10:13:19
2025-02-12 10:13:19 Share images, automate workflows, and more with a free Docker ID:
2025-02-12 10:13:19 https://hub.docker.com/
2025-02-12 10:13:19
2025-02-12 10:13:19 For more examples and ideas, visit:
2025-02-12 10:13:19 https://docs.docker.com/get-started/
2025-02-12 10:13:19
```

Detener un Contenedor en Ejecución

- Usamos el contenedor en ejecución.
- Le damos a los tres puntos (:).
- Selecciona **Pause** o **Stop** según sea necesario.

Eliminar un Contenedor

- Nos aseguramos de que el contenedor está detenido.
- Hacemos clic en los tres puntos (:) a la derecha del contenedor.
- Seleccionamos el ícono de **Eliminar** para eliminarlo.

A screenshot of the Docker desktop application. At the top, there's a search bar and a filter button 'Only show running containers'. Below is a table with columns: Name, Image, Status, Port(s), CPU (%), Last started, Actions, and a Delete button. One row is selected, showing 'angry_moser' as the name, 'hello-world' as the image, 'Exited' as the status, and 'N/A' for CPU. The 'Actions' column contains a context menu with options like 'Delete' highlighted.

Ver Detalles de un Contenedor

- Hacemos clic en los tres puntos (:) a la derecha del contenedor.
- Seleccionamos view details.
- Al entrar, podemos ver información como logs, configuración y estadísticas de uso.

A screenshot of the Docker desktop application. It shows the same interface as the previous image, but the context menu is now open over the 'angry_moser' container, with 'View details' highlighted.

A screenshot of the Docker logs for the 'angry_moser' container. The 'Logs' tab is selected. The log output is as follows:

```
2025-02-12 10:13:00 Hello from Docker!
2025-02-12 10:13:00 This message shows that your installation appears to be working correctly.
2025-02-12 10:13:00
2025-02-12 10:13:00 To generate this message, Docker took the following steps:
2025-02-12 10:13:00 1. The Docker client contacted the Docker daemon.
2025-02-12 10:13:00 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
2025-02-12 10:13:00   (amd64)
2025-02-12 10:13:00 3. The Docker daemon created a new container from that image which runs the
2025-02-12 10:13:00   executable that produces the output you are currently reading.
2025-02-12 10:13:00 4. The Docker daemon streamed that output to the Docker client, which sent it
2025-02-12 10:13:00   to your terminal.
2025-02-12 10:13:00
2025-02-12 10:13:00 To try something more ambitious, you can run an Ubuntu container with:
2025-02-12 10:13:00 $ docker run -it ubuntu bash
2025-02-12 10:13:00
2025-02-12 10:13:00 Share images, automate workflows, and more with a free Docker ID:
2025-02-12 10:13:00 https://hub.docker.com/
2025-02-12 10:13:00
2025-02-12 10:13:00 For more examples and ideas, visit:
2025-02-12 10:13:00 https://docs.docker.com/get-started/
2025-02-12 10:13:00
```

```

1 v {
2   "Id": "743da717dfd7c605f01ef26d7ae3263d812eefa31676235b527500f27ba38d32",
3   "Created": "2025-02-12T08:00:04.8352248252",
4   "Path": "/hello",
5   "Args": [],
6   "State": {
7     "Status": "exited",
8     "Running": false,
9     "Paused": false,
10    "Restarting": false,
11    "OOMKilled": false,
12    "Dead": false,
13    "Pid": 0,
14    "ExitCode": 0,
15    "Error": "",
16    "StartedAt": "2025-02-12T09:13:19.031056818Z",
17    "FinishedAt": "2025-02-12T09:13:19.248875412"
18  },
19  "Image": "sha256:74c5c54e27dc1bb10d4b2226072d464509f2f222f1a3ce7454a59661a1d44602",
20  "ResolveConfPath": "/var/lib/docker/containers/743da717dfd7c605f01ef26d7ae3263d812eefa31676235b527500f27ba38d32/resolv.conf",
21  "HostnamePath": "/var/lib/docker/containers/743da717dfd7c605f01ef26d7ae3263d812eefa31676235b527500f27ba38d32/hostname",
22  "HostsPath": "/var/lib/docker/containers/743da717dfd7c605f01ef26d7ae3263d812eefa31676235b527500f27ba38d32/hosts",
23  "LogPath": "/var/lib/docker/containers/743da717dfd7c605f01ef26d7ae3263d812eefa31676235b527500f27ba38d32/logs",
24  "Name": "angry_moser",
25  "RestartCount": 0,
26  "Driver": "overlay2",
27  "Platform": "linux",
28  "MountLabel": "",
29  "ProcessLabel": "",
30  "AppArmorProfile": "",
31  "ExecIDs": null,
32  "HostConfig": {
33    "Binds": null,
34    "ContainerIDFile": "",
35    "LogConfig": {
36      "Type": "json-file",
37      "Config": {}
38    }
39  }
40}

```

Copiar el Comando de Ejecución de un Contenedor

- Hacemos clic en los tres puntos (:) junto al contenedor.
- Seleccionamos **Copy docker run** para copiar el comando con el que fue creado.

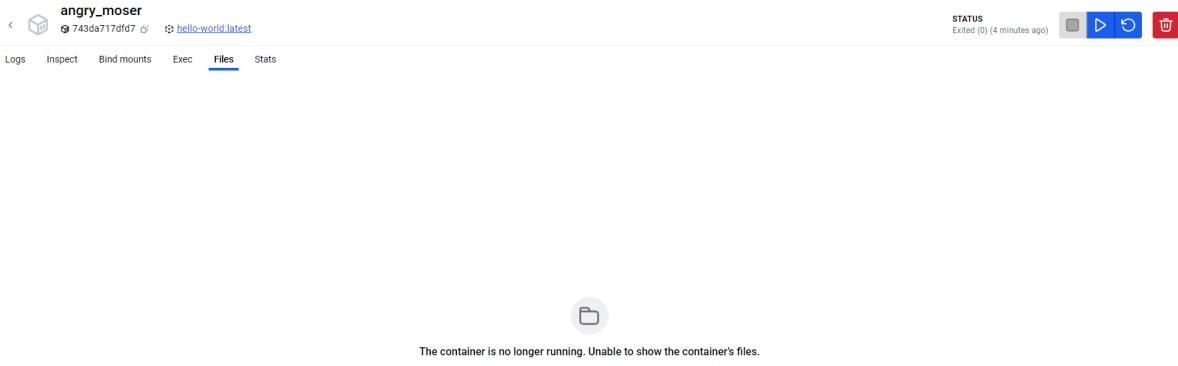
Name	Image	Status	Port(s)
angry_moser 743da717dfd7	hello-world	Exited	

Name	Image	Status	Port(s)
angry_moser 743da717dfd7	hello-world	Exited	

Ver Archivos de un Contenedor

- Hacemos clic en los tres puntos (:).
- Seleccionamos **View files** para explorar los archivos dentro del contenedor.

Name	Image	Status	Port(s)
angry_moser 743da717dfd7	hello-world	Exited	



Gestión de imágenes Docker

Ver las Imágenes Disponibles

- En la barra lateral izquierda, seleccionamos **Images**.
- Se nos muestra la lista de imágenes descargadas en el sistema.

Local Hub repositories						
		Status			Actions	
Name	Tag	Status	Created	Size	More	Delete
hello-world 74cc54e27dc4	latest	In use	21 days ago	10.07 KB		

Descargar una Imagen desde Docker Hub

- Descargamos la imagen manualmente con:

```
$docker pull nginx:latest
```

- Vamos a la pestaña **Images**.
- Hacemos clic en el botón **Pull** (Descargar).
- Escribimos el nombre de la imagen y su versión, por ejemplo:

```
$nginx:latest
```

- Haz clic en **Pull** para descargar la imagen.

The screenshot shows the Docker interface with the 'Images' tab selected. There are two images listed: 'nginx:latest' and '97662d24417b'. A context menu is open over the 'nginx' row, with the 'Delete' option highlighted.

Eliminar una Imagen

1. Encontramos la imagen que queremos borrar en la pestaña **Images**.
2. Hacemos clic en el ícono de los tres puntos (⋮) a la derecha.
3. Seleccionamos **Delete** para eliminar la imagen.

The screenshot shows the Docker interface with the 'Images' tab selected. There are three images listed: 'hello-world', 'nginx', and '97662d24417b'. The 'hello-world' image is selected, and its 'Delete' button is highlighted.

No se pueden eliminar imágenes que están en uso por un contenedor en ejecución:

The screenshot shows the Docker interface with the 'Images' tab selected. There are two images listed: 'hello-world' and '97662d24417b'. The 'hello-world' image is selected, and its 'Delete' button is highlighted, but it is marked as 'In use'.

Crear una Imagen Personalizada

- Escribimos un **Dockerfile** con la configuración de la imagen.
- Usamos el siguiente comando en la terminal:

```
$ docker build -t mi-imagen .
```

- Posteriormente, la imagen nos aparece en la pestaña **Images**.

Ejecutar un Contenedor desde una Imagen

- Encontramos la imagen que queremos ejecutar.
- Hacemos clic en **Run**.
- Configuramos los puertos y volúmenes si es necesario y hacemos clic en **Run**.

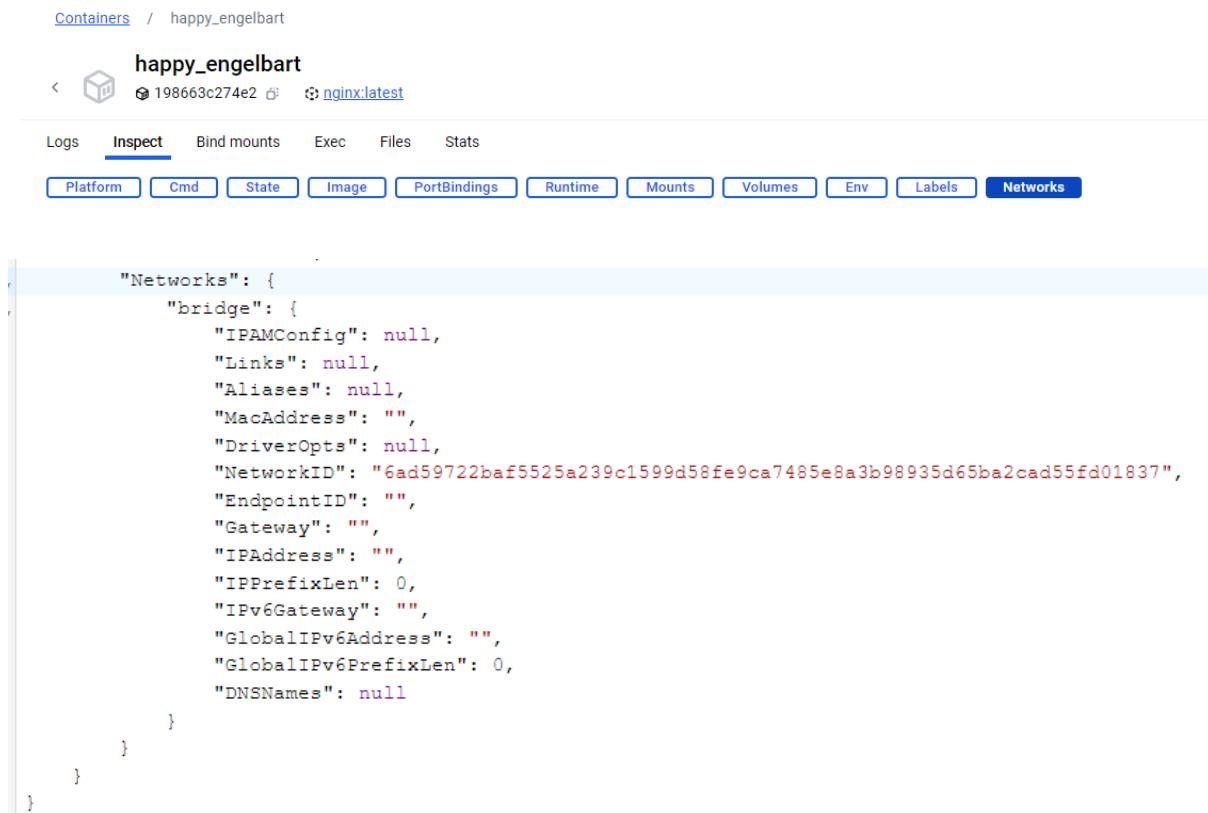
The screenshot shows the Docker interface for creating a new container. At the top, it says "Run a new container" and "nginx:latest". Below that, there's a section titled "Optional settings" with a "Container name" input field. A note below it says "A random name is generated if you do not provide one." Under "Ports", it says "Enter '0' to assign randomly generated host ports." There's a "Host port" input field set to ":80/tcp". In the "Volumes" section, there are two input fields: "Host path" and "Container path", separated by a plus sign. Below that is the "Environment variables" section with "Variable" and "Value" input fields, also separated by a plus sign. At the bottom right are "Cancel" and "Run" buttons.

Containers / happy_engelbart		STATUS
	198669c274e2 ⚡	nginx:latest
Logs	Inspect	Running (0 seconds ago)
Bind mounts	Exec	
Files	Stats	
<pre>2025-02-12 11:05:37 /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration 2025-02-12 11:05:37 /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/ 2025-02-12 11:05:37 /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh 2025-02-12 11:05:37 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf 2025-02-12 11:05:37 /docker-entrypoint.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf 2025-02-12 11:05:37 /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh 2025-02-12 11:05:37 /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh 2025-02-12 11:05:37 /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh 2025-02-12 11:05:37 /docker-entrypoint.sh: Configuration complete; ready for start up 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: starting the "goli" event method 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: nginx/1.27.4 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14) 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: OS: Linux 5.15.46.4-microsoft-standard-WSL2 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576/1048576 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker processes 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 29 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 30 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 31 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 32 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 33 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 34 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 35 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 36 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 37 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 38 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 39 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 40 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 41 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 42 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 43 2025-02-12 11:05:37 005/02/12 10:05:37 [notice] 1#1: start worker process 44</pre>		

💡 Configuración de redes y volúmenes

Ver las redes disponibles

- En la barra lateral, seleccionamos **Containers**.
- Hacemos clic en un contenedor existente y seleccionamos **View details**.
- En la pestaña **Network**, vemos las redes asociadas al contenedor.

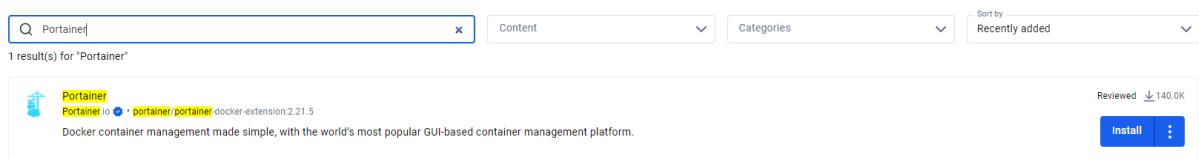


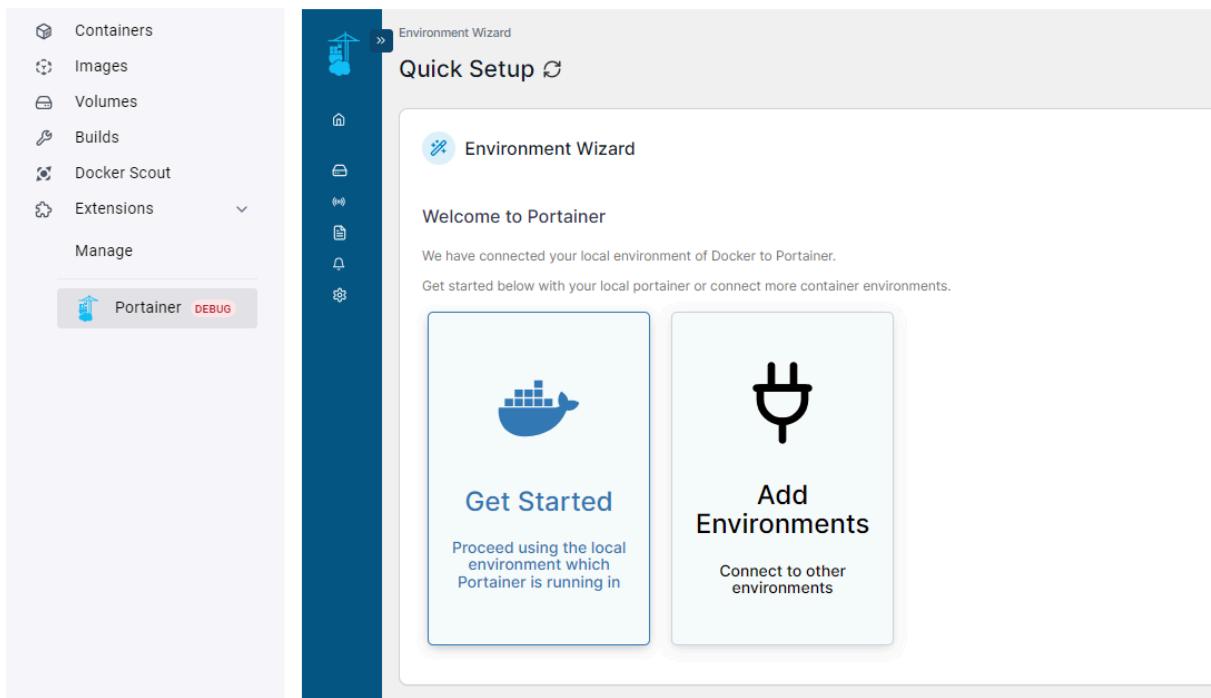
```
"Networks": {  
    "bridge": {  
        "IPAMConfig": null,  
        "Links": null,  
        "Aliases": null,  
        "MacAddress": "",  
        "DriverOpts": null,  
        "NetworkID": "6ad59722baf5525a239c1599d58fe9ca7485e8a3b98935d65ba2cad55fd01837",  
        "EndpointID": "",  
        "Gateway": "",  
        "IPAddress": "",  
        "IPPrefixLen": 0,  
        "IPv6Gateway": "",  
        "GlobalIPv6Address": "",  
        "GlobalIPv6PrefixLen": 0,  
        "DNSNames": null  
    }  
}
```

Crear una Nueva Red

Docker Desktop no ofrece una opción nativa para añadir redes directamente desde su interfaz, pero podemos utilizar extensiones de terceros para ampliar su funcionalidad.

- Hacemos clic en la pestaña **Extensions** en la barra lateral izquierda.
- En el Marketplace de Extensiones, buscamos **Portainer**.
- Hacemos clic en **Install** para añadir la extensión.

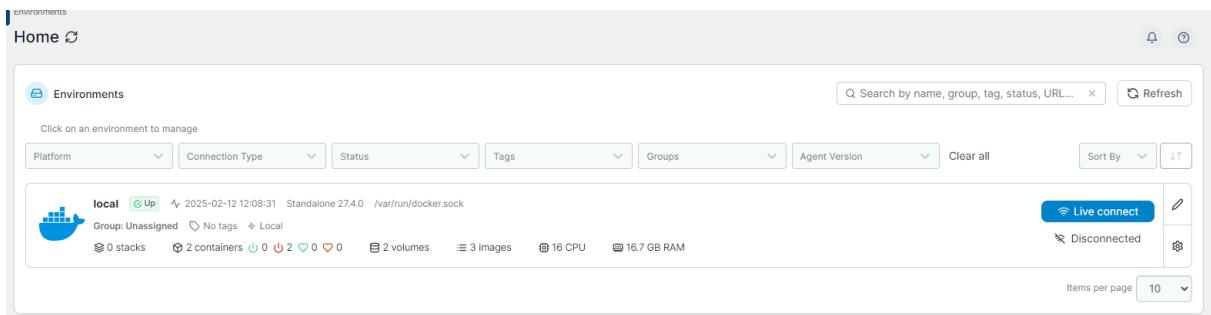




- Configuramos el entorno

Al iniciar Portainer por primera vez, verás la pantalla de "Quick Setup" (como en tu imagen). Aquí puedes:

- **"Get Started"**: Gestionamos el entorno local de Docker donde se ejecuta Portainer.



- **"Add Environments"**: Agregamos otros entornos remotos, como clústeres de Kubernetes o instancias Docker en servidores.



- Administramos contenedores

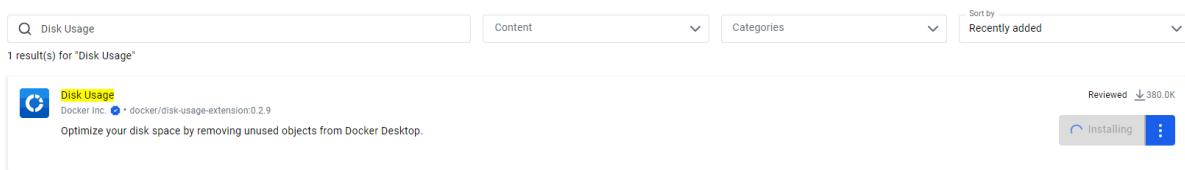
Desde el panel de control de Portainer, puedes:

- Crear, iniciar, detener y eliminar contenedores.
- Ver logs y estadísticas de rendimiento.
- Gestionar imágenes y volúmenes de Docker.
- Configurar redes y stacks con Docker Compose.

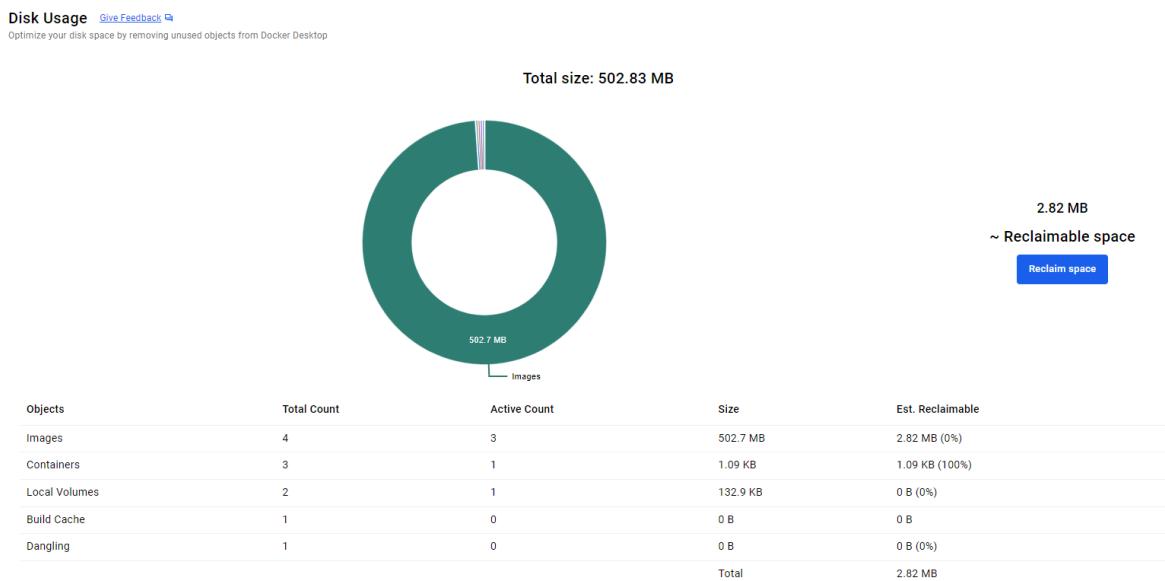
Herramientas de diagnóstico

Disk Usage (Uso de Disco)

- Muestra cuánto espacio ocupan las imágenes, contenedores y volúmenes.
- Ideal si Docker usa demasiado espacio en disco.

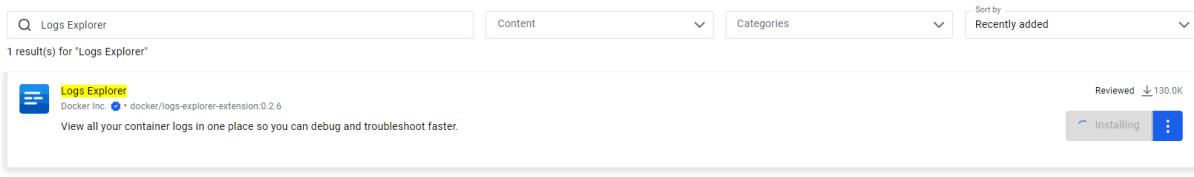


- Una vez instalada, vamos a "**Extensions**" > "**Disk Usage**".
- Revisamos el uso de espacio y liberamos lo que no necesitemos.



Logs Explorer (Explorador de Logs)

- Permite ver los logs de todos los contenedores sin usar la terminal.
- Muy útil para detectar errores sin ejecutar `docker logs`.



- Vamos a "**Extensions**" > "**Logs Explorer**".
- Seleccionamos el contenedor que queremos analizar y revisamos los logs en la interfaz.

Timestamp	Container	Message
! 2025-02-12 11:03:31	[portainer_portainer]	[90m2025/02/12 11:03AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/http/server.go:3...
! 2025-02-12 11:05:10	[portainer_portainer]	[90m2025/02/12 11:05AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:05:10	[portainer_portainer]	[90m2025/02/12 11:05AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:09:27	[portainer_portainer]	[90m2025/02/12 11:09AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:00	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:07	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:09	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:16	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:18	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:30	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:30	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:10:46	[portainer_portainer]	[90m2025/02/12 11:10AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:24:53	[portainer_portainer]	[90m2025/02/12 11:24AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:24:54	[portainer_portainer]	[90m2025/02/12 11:24AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:25:01	[portainer_portainer]	[90m2025/02/12 11:25AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:25:02	[portainer_portainer]	[90m2025/02/12 11:25AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:25:03	[portainer_portainer]	[90m2025/02/12 11:25AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...
! 2025-02-12 11:25:05	[portainer_portainer]	[90m2025/02/12 11:25AM [0m [32mINF [0m [1mgithub.com/portainer/portainer/api/jwt/jwt.go:174 [...