

AI-03936 Docker Portainer Cleanup

| | |
|----------------|------------------------|
| 👤 Owner | Ⓜ Rigel Arcayan |
| ☰ Tags | Project |
| 🕒 Created time | June 14, 2026 12:13 PM |

In GCP, I'm running an Ubuntu based docker lab and I want to start fresh. Is it possible to use Portainer to cleanup to a fresh start of my docker lab instead of CLI. Provide guidance

Yes, it is entirely possible to use Portainer to clean up your Docker environment and get back to a "fresh start" without touching the CLI. In fact, Portainer's visual interface makes it much easier to spot orphan volumes, dangling images, and leftover networks that you might otherwise miss.

Here is a step-by-step guide to wiping the slate clean using Portainer, along with a few important things to keep in mind.

⚠ Important Precautions Before You Start

- **Don't Delete Portainer itself:** If your Portainer instance is running as a container on this same Docker host (which is typical), **do not delete the Portainer container or its associated volume/networks**. If you do, you will lose access to the UI and will have to use the CLI to reinstall it.
- **Destructive Actions:** The steps below are permanent. Make sure you don't have any persistent data you actually want to keep.

Step-by-Step Cleanup Guide in Portainer

Log into your Portainer dashboard, select your local environment, and follow these steps in order.

1. Stop and Remove Stacks and Containers

To delete images or volumes, you must first stop and remove the containers using them.

- **For Stacks (Docker Compose):** 1. Click on **Stacks** in the left sidebar.
2. Select the stacks you want to remove (do not select Portainer's stack if it was deployed via one).
3. Click **Remove**. This will automatically stop and delete all containers associated with those stacks.
- **For Standalone Containers:**
 1. Click on **Containers** in the left sidebar.
 2. Check the boxes next to the containers you want to destroy. **Leave `portainer` or `portainer_agent` unchecked.**
 3. Click **Remove** at the top, and toggle the "**Automatically remove non-persistent volumes**" switch if you want to wipe their temporary volumes too.

2. Prune and Remove Images

Once the containers are gone, their images become "unused" and can be cleared out.

1. Click on **Images** in the left sidebar.
2. Look for the **Prune unused images** button at the top and click it. This is the safest and fastest way to wipe out all images that aren't actively running.
3. If you want to be absolutely thorough, you can manually select any remaining images (except the Portainer image) and click **Remove**.

3. Clear Out Volumes (The Data Wrecker)

Persistent data lives in volumes. If you want a truly "fresh" start, these need to go.

1. Click on **Volumes** in the left sidebar.
2. Look for the **Unused** badge next to the volumes.
3. Select all unused volumes (**make sure `portainer_data` is NOT selected**).
4. Click **Remove**.

4. Clean Up Custom Networks

Isolated environments often leave behind custom Docker networks.

1. Click on **Networks** in the left sidebar.
2. Select any custom networks you created for your lab. (Leave default networks like `bridge`, `host`, and `none` alone, as well as any network Portainer is using).
3. Click **Remove**.

The "Pro-Tip" Alternative: Portainer's Clean Up Feature

Portainer actually has a built-in automated background pruning feature if you want to automate this or ensure it stays clean.

Go to **Settings > Edge Compute / Environments** (depending on your Portainer version) and look for **Clean up unused resources**. However, for a one-time total reset, the manual steps above are usually preferred so you don't accidentally drop your Portainer data.

A Quick Sanity Check

Once you've done this, clicking on your **Dashboard** tab in Portainer should show `0` Stacks, `1` Container (Portainer itself), `1` Image (Portainer), and `1` Volume (`portainer_data`).

Are you running Portainer as a standalone container, or did you deploy it using Docker Swarm / a specific docker-compose file? I want to make sure we don't accidentally lock you out of your UI!

Will the cleanup procedure keep my Docker files in my `/home/dockerlab` folder with the various `.yaml` files? or it get wiped out as well?

No, it will **not** wipe them out. Your `.yaml` files and anything else in your `/home/dockerlab` folder will remain completely safe and untouched.

Here is why:

- **Portainer only controls the Docker daemon:** When you delete containers, images, volumes, and stacks through Portainer, you are only telling the Docker

engine to clear out the resources *it* manages internally (usually stored deep in `/var/lib/docker/`).

- **Portainer cannot delete host files:** Portainer does not have permission or a mechanism to reach out into your Ubuntu host's file system and delete arbitrary files in your `/home/` directory.

⚠️ One Important Exception: Bind Mounts

While your actual `docker-compose.yml` configuration files are safe, you need to check how your containers were storing their data.

If any of your stacks used **Bind Mounts** pointing inside that folder (for example, a line in your YAML file that looks like `- /home/dockerlab/wordpress/data:/var/www/html`), Portainer **cannot** delete that data folder either.

However, because Portainer can't delete it, it means any old configuration data or database files stored in those host folders **will persist**. If you deploy a new stack using those same paths, the new containers might pull in the old data instead of starting truly fresh.

How to ensure a 100% fresh start:

1. Complete the Portainer cleanup steps.
2. If you used bind mounts for data (not just for holding your `.yml` files), you *will* need to use the CLI or a file manager one time just to delete those specific data sub-folders (e.g., `/home/dockerlab/wordpress/data`), while leaving your `.yml` files alone.

Do you know if your stacks were using named Docker volumes, or were you mapping data directly to sub-folders inside `/home/dockerlab/` ?