



Sign in to your account (le__@g__.com) for your personalized experience.



Sign in with Google

Not you? [Sign in](#) or [create an account](#)

Top GUI for Docker



appfleet team

Follow

May 6 · 5 min read

Top GUI for Docker



 appfleet

Are you still monitoring your containers in tons of console windows or passionate about knowing dozens of terminal commands? There are a couple of nice Graphical User

Interfaces (GUIs) for Docker, that can make your life much simpler and increase your performance. Let's select which one will suit you best.

Portainer (Web app)

Open-source (Zlib license).

OS: Linux, Mac OS X, Windows.

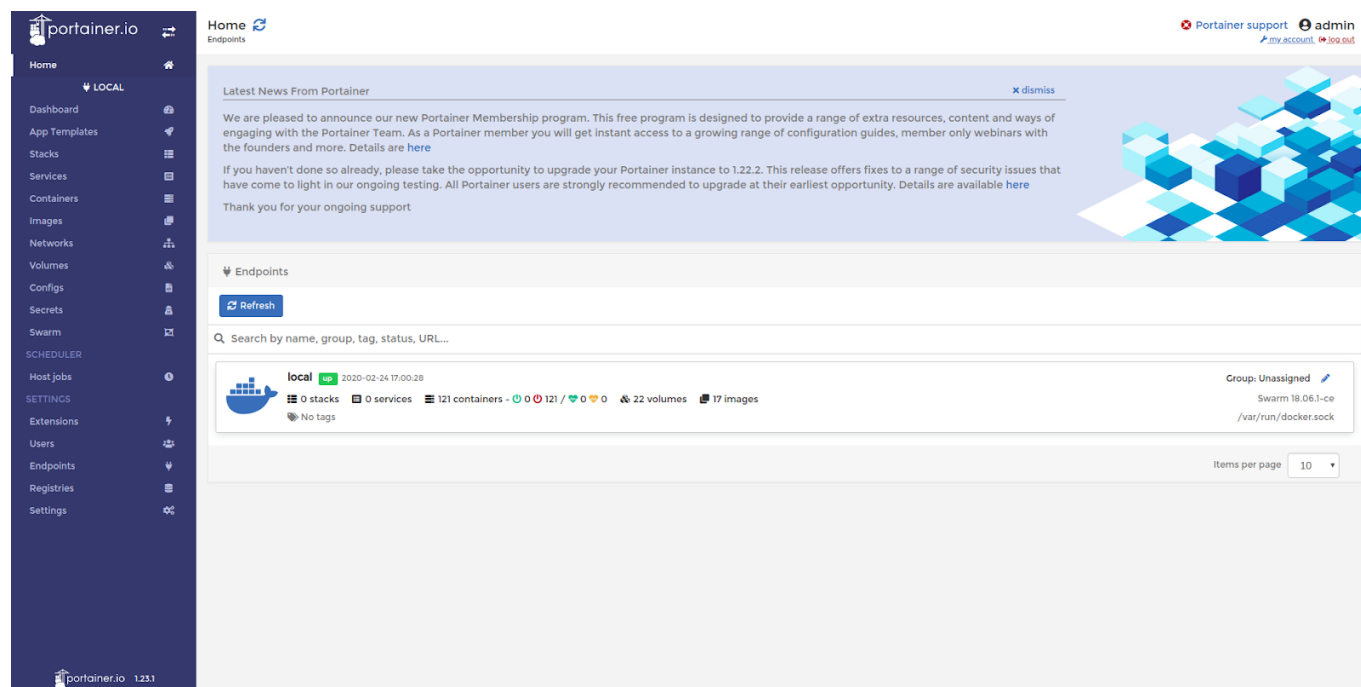
Portainer has full support for the following Docker versions:

- Docker 1.10 to the latest version
- Standalone Docker Swarm $\geq 1.2.3$ (*NOTE: Use of Standalone Docker Swarm is being discouraged since the introduction of built-in Swarm Mode in Docker. While older versions of Portainer had support for Standalone Docker Swarm, Portainer 1.17.0 and newer do not support it. However, the built-in Swarm Mode of Docker is fully supported.*)

Partial support for the following Docker versions (some features may not be available):

- Docker 1.9

You can test a live demo (*admin / tryportainer*).



Portainer can be easily installed with Docker on a standalone Linux/Windows server/swarm cluster. A full-featured platform allows you to work with different endpoints.

You can manage registries, networks, volumes, secrets, images, and containers. You are also able to save your configuration (you can find examples of alertmanager and Prometheus at the live demo), and configure a Docker Swarm and stacks. Portainer can check if the container is healthy.

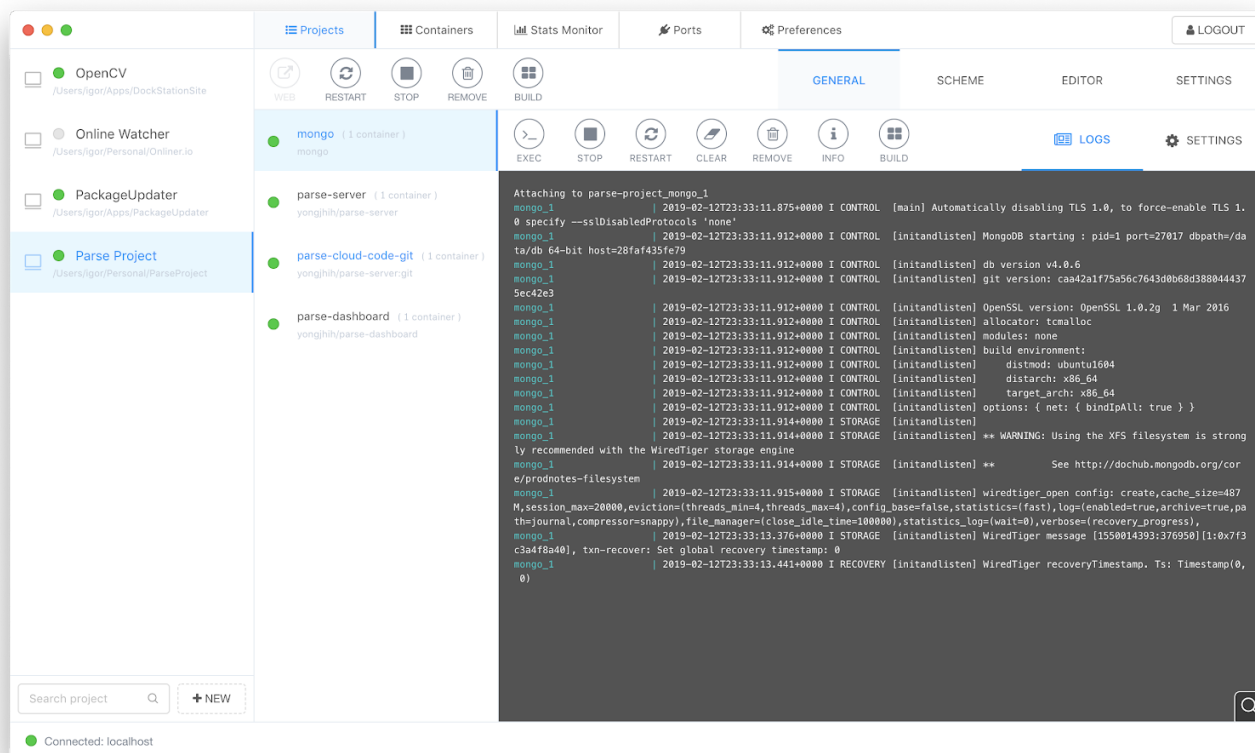
Apart from the basic operations you need to work with containers like *run*, *stop*, *resume*, *kill*, *remove*, etc., you can also *inspect containers*, *see logs*, *visualize basic stats*, *attach* and *open the console* for certain containers.

As a plus, you also get a **role-based access system** and the **ability to install extensions**.

Conclusion: a powerful GUI instrument that can be used for a team project with local or remote containers, Docker stack or Docker Swarm. However, Portainer may turn out to be too much for your generic needs. The interface may also be inconvenient especially if you use many projects at the same time.

DockStation (Desktop app)

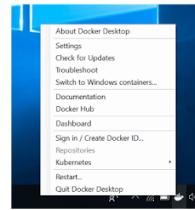
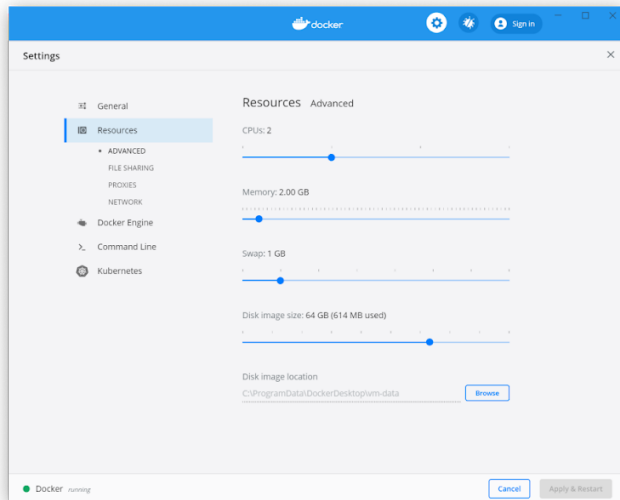
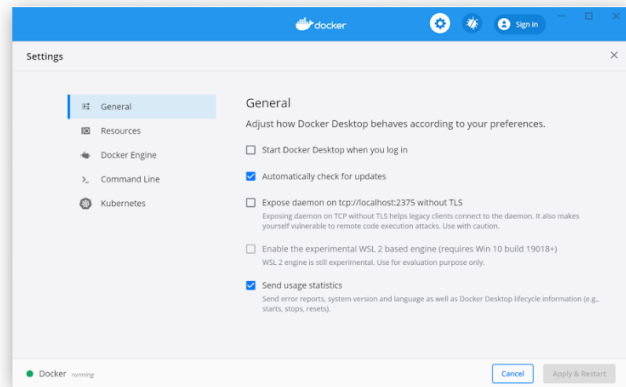
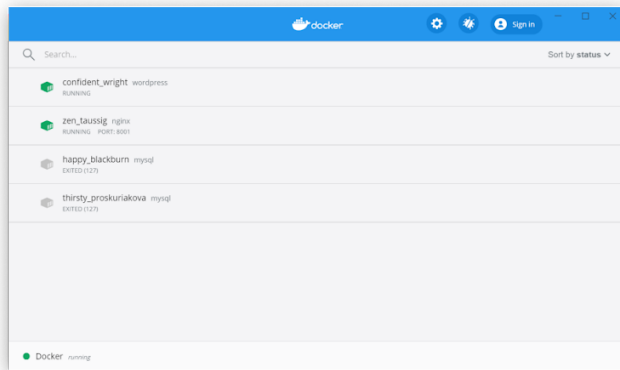
OS: Linux/Mac/Windows



DockStation is a free full-featured desktop app which allows you to work with `docker` and `docker-compose`. It can help generate a clean and native `docker-compose.yml` file which can be used even outside the application, using the native Docker Compose CLI commands. It also helps you to manage your containers and services (both remote and local), and monitor them (logs monitoring, searching logs, grouping, running tools and getting container info). There are additional tools available for common, multiple and single monitoring of container resources.

With DockStation, you can easily track CPU, Memory, Networks I/O, Blocks I/O usage and open ports. All work can be organized into projects, where you can check the status of each container, build a graphical scheme that visualizes each image in the project and relations between them. As an additional takeaway, DockStation works like charm with Docker Hub.

Docker Desktop (Desktop app)



Since Docker-toolbox (with Kitematic) is deprecated, all users are recommended to use Docker Desktop for Mac and Docker Desktop for Windows.

This tool gives you the ability to set resource limits for your Docker (memory, CPU, swap and disk image size, file sharing, proxies, and network), configure Docker engine, command line and Kubernetes (you can configure deployment to a Kubernetes from Docker Desktop).

Using dashboard, you can use not only basic container operations, but can also see logs, basic stats and inspect your container. All of this can be *called* through the contextual menu or from the indicator in the status bar.

Lazydocker (Terminal UI)

Open-source

OS: (Linux/OSX/Windows)

Requirements:

- Go version ≥ 1.8
- Docker ≥ 1.13 (API ≥ 1.25)
- Docker-Compose $\geq 1.23.2$ (optional)

The screenshot displays the Lazydocker application interface. It is divided into several sections:

- Status:** Shows 'lazydocker' as the active component.
- Services:** A list of running services with their CPU usage:

Service	CPU Usage
api	0.01%
mysql	0.13%
quote-fetcher	0.00%
redis4.0	0.31%
- Standalone Containers:** A list of containers with their status and CPU usage:

Container	Status	CPU Usage
migrate	exited (1)	0.00%
guard	running	0.00%
rails-console	running	0.00%
- Images:** A table showing local images and their sizes:

Image	Size
<none>	<none>
<none>	<none>
bumper	latest
<none>	<none>
- Volumes:** A list of local volumes with their IDs and sizes:

Volume	Size
local 68473898e49e0e4d2c5998f68c79	
local 90d4357f8a93810ee8f2a59671e7	
local b9a4a32a2d4d6366f9a9434aff8a	
local bc80a55695b8d780db6f54f31a	
- Config:** A section showing configuration details for a selected container (ID: a28c7b34cac8c23430c15a6c1a6738e998859057848457cf05):

Field	Value
Name	<none>
ID	sha256:a28c7b34cac8c23430c15a6c1a6738e998859057848457cf05
Tags	<none>:<none>
Size	144.02MB
Created	Sat, 29 Jun 2019 15:30:43 AEST

At the bottom, there is a navigation bar with keyboard shortcuts: PgUp/PgDn: scroll, esc/q: close, x: menu, ← → ↑ ↓: navigate. A 'Donate' link and 'unversioned' status are also visible.

Lazydocker can be used through both mouse and keyboard. For some elements, the contextual menu is available, where you can find all popular commands with shortcuts. The good news is that you not only have basic commands to manipulate containers, basic stats, logs, and inspections; you can use also graphical visualization of main metrics (by default CPU and memory usage) and 'top' of processes. Additionally you can also configure this section for almost all the metrics you want.

For the selected images, you can see commands from the Dockerfile that are executed when they run, and the inherited layers. Cleaning of unused containers, images, volumes (prune) is also provided, apart from the liberty of modifying available commands and adding new ones.

As a result, we have a minimalistic terminal interface that can be really helpful with several 'not so complicated' projects.

Docui (Terminal UI)

OS: Mac/Linux

Requirements:

- Go \geq 1.11.4~
- Docker Engine \geq 18.06.1
- Git



This console UI is made for convenient creation and configuration of new containers/services, where you can find a lot of keybindings for all necessary operations.

You can work with:

- Images (search/pull/remove, save/import/load, inspect/filter)
- Containers (create/remove, start/stop, export/commit, inspect/rename/filter, exec cmd)
- Volumes (create/remove, inspect/filter)
- Networks (remove, inspect/filter)

Conclusion

The above is not a full list but some of the most popular and convenient free GUIs for Docker. Which one to select — depends on your needs. If you need a really powerful instrument for a team with access management, works with Docker swarm, with Docker stack and can be deployed on a remote server — choose Portainer. If you require a powerful instrument, that works on several projects (possibly remote) with `docker-compose` and prefer a local desktop app - choose DockStation.

If your project is not that complex — you can choose among Lazydocker (if you want to mostly manage existing containers and services with console), Docui (if you mostly create simple images) or Docker Desktop (if you appreciate the desktop integrations and want to get a simple integration with Kubernetes).

Windows Docker

About Help Legal

Get the Medium app

