



 Containers


 Images

 Volumes


 Dev Environments BETA

Extensions BETA




 Add Extensions


Containers







[Give feedback](#) 

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)


☐ Only show running containers


 Search




<input type="checkbox"/>	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	<div> heuristic_jang 59db8cc7c87d </div>	docker/getting-started:latest	Running	80:80 	16 minutes ag 	<div> </div>

Showing 1 items



RAM 1.58GB CPU 0.22%  Not connected to Hub

v4.13.1 

Getting Started

[Getting Started](#)

[Our Application](#)

[Updating our App](#)

[Sharing our App](#)

[Persisting our DB](#)

[Using Bind Mounts](#)

[Multi-Container Apps](#)

[Using Docker Compose](#)

[Image Building Best Practices](#)

[What Next?](#)

Getting Started

The command you just ran

Congratulations! You have started the container for this tutorial! Let's first explain the command that you just ran. In case you forgot, here's the command:

```
docker run -d -p 80:80 docker/getting-started
```

You'll notice a few flags being used. Here's some more info on them:

- `-d` - run the container in detached mode (in the background)
- `-p 80:80` - map port 80 of the host to port 80 in the container
- `docker/getting-started` - the image to use

Pro tip

You can combine single character flags to shorten the full command. As an example, the command above could be written as:

```
docker run -dp 80:80 docker/getting-started
```

The Docker Dashboard

Before going too far, we want to highlight the Docker Dashboard, which gives you a quick view of the containers running on your machine. It gives you quick access to container logs, lets you get a shell inside the container, and lets you easily manage container lifecycle (stop, remove, etc.).

Table of contents

[The command you just ran](#)

[The Docker Dashboard](#)

[What is a container?](#)

[What is a container image?](#)