DOCKER

# How to Stop and Remove Docker Containers

Jack Roper
03 Nov 2023 · 10 min read

Reviewed by: Paweł Piwosz

In this article, we will take a look at how to perform various useful command line operations in Docker, showing how to start, list, stop and remove containers, and the various options available.

We will cover:



## Prerequisites

If you don't already have Docker installed, you can grab it from the official website: Docker Download.

Once installed, to follow along with this article, pull the Ubuntu image from the Docker hub so we can use the image with the containers that we will manipulate using the command:

```
❯ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
445a6a12be2b: Pull complete
Digest: sha256:aabed3296a3d45cede1dc866a24476c4d7e093aa806263c27ddaadbdce3c1054
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

For other Docker commands, check out our Docker cheat sheet.

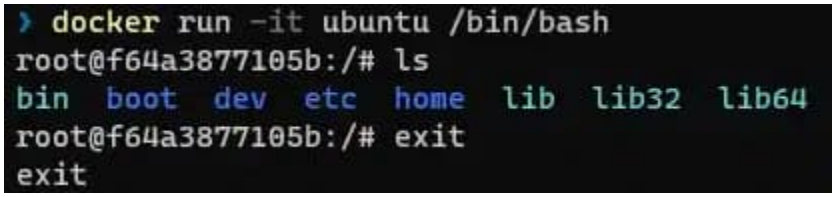
## How to start a Docker container

To run a simple container based on the Ubuntu image we just pulled, run the command below:

```
docker run -it ubuntu /bin/bash
```

After running the container, you'll be inside the container's shell (in this example, a Bash shell). You can interact with it just like a regular Linux system.

When you're done, you can exit the container by typing `exit`.




Let's take a look at the command options for `docker run`.

This command is used to start a container, and the syntax looks like this:

```
docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]
```

In our example, the IMAGE we used was `ubuntu` , the `-it` OPTIONS we specified make the container interactive and allocate a terminal. The COMMAND we supplied was the command to run inside the terminal of the running container.

You can find out more about the options available to the Docker run command over on the official Docker reference pages.

## How to list the running containers

To see a list of running containers, you can use the `docker ps` command. This command provides information about running containers, including their names, IDs, and status.

```
docker ps
```


Learn more: How to List Docker Containers

## How to stop a running Docker container

To stop a running container, you can use the `docker stop` command and provide either the container name or container ID. You can find these from the information supplied in the `docker ps` command.

For example, as shown above, the container ID of my running Ubuntu container is ccfac1f88d1b:

```
docker stop ccfac1f88d1b
```

Run `docker ps` again to check the container is stopped.

When you run `docker stop` the main process inside the container will receive `SIGTERM`, and after a grace period, `SIGKILL`. The first signal can be changed with the `STOPSIGNAL` instruction in the container's Dockerfile or the `--stop-signal` option to `docker run`.

The usage for the `docker stop` command:

```
docker stop [OPTIONS] CONTAINER [CONTAINER...]
```

You can also use the option `-s` to change the signal sent to the container, and `-t` to supply the number of seconds to wait before killing the container.


💡 **You might also like:**

- Common Infrastructure Challenges and How to Solve Them
- How to Create a CI/CD Pipeline with Docker
- 16 DevOps Best Practices to Follow


## How to force stop a Docker container

To forcefully stop a Docker container, you can use the `docker kill` command. The main process inside the container is sent `SIGKILL` signal (default), or the signal that is specified with the `--signal` option, which can terminate the process abruptly.

When you run this command, you will notice the container is terminated much more quickly than when running `docker stop` as the container is not gracefully terminated, and the container does not have a chance to perform any cleanup tasks.

The usage is similar to the `docker stop` command:

```
docker kill [OPTIONS] CONTAINER [CONTAINER...]
```

Be aware that **this can potentially lead to data corruption or unexpected behavior in the container**.

Because of this, `docker kill` should be used after command with caution, usually after trying `docker stop` first to allow the container to shut down cleanly and only resort to `docker kill` when the container is unresponsive or cannot be stopped by normal means.

## How to stop all running Docker containers

To stop all running docker containers with a single command rather than individually, you can use a for loop in the shell:

```
docker stop $(docker ps -q)
```

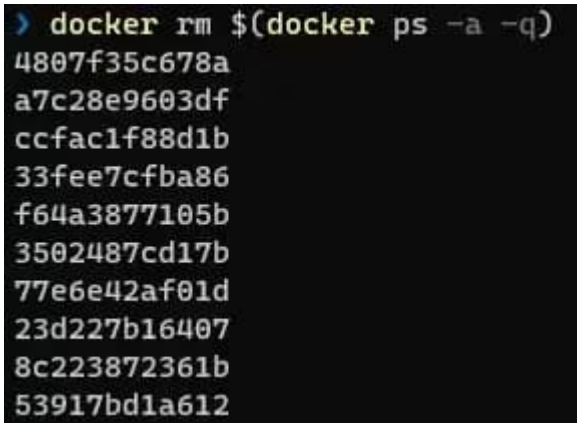
Breaking this down, the `docker ps -q` command lists all the running containers showing the containerID only. This is then passed to the `docker stop` command.

## How to remove the containers

If you want to remove the containers after stopping them, you can use the `docker rm` command in a similar manner:

```
docker rm $(docker ps -a -q)
```

The additional `-a` option in the `docker ps -a -q` command lists all containers, including stopped ones. After running the command, a list of the containerIDs that have been removed will be shown on the console:




Similar to the `docker kill` command, if required you can use the `docker rm` command with the `--kill` option to force the removal of a running container, using SIGKILL.

## How to stop and remove containers in Docker Compose

In Docker Compose, you can stop and remove containers associated with your Compose project using the `docker-compose down` command. You can execute the `docker-compose down` command from the directory where your `docker-compose.yml` file is located.

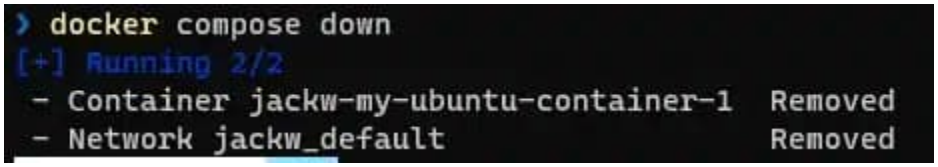
I have a docker-compose.yaml file to create a simple Ubuntu container:

```
version: '3'
services:
  my-ubuntu-container:
    image: ubuntu
    command: sleep infinity
```

I run `docker compose up` to create the container:

```
> docker-compose up
[+] Running 2/2
 - Network jackw_default                    Created
 - Container jackw-my-ubuntu-container-1    Created
Attaching to jackw-my-ubuntu-container-1
```

And then run `docker compose down` to remove it:




This command not only stops the containers but also removes:

- Containers for services defined in the Compose file.
- Networks defined in the networks section of the Compose file.
- The default network, if one is used.

Networks and volumes defined as external are never removed.

To remove named volumes declared in the “volumes” section of the Compose file and anonymous volumes attached to containers, you can use the `-v` option.

## Docker container stop vs. kill

Both commands will terminate containers, the main difference being that the `stop` command allows the container to terminate gracefully, and the `kill` command terminates it immediately.

You should generally prefer `docker container stop` when possible to ensure a clean shutdown of your containers but `docker container kill` can be useful when a container is not responding or needs to be stopped forcefully.

## Key points

Manipulating containers on the command line and knowing how to start, stop, kill, and remove containers on demand with the available options is a great way to start

learning and experimenting with Docker. Knowing the differences between the `stop` and `kill` commands as explained above is valuable, especially to avoid any issues when running in production!

We encourage you also to explore how Spacelift offers full flexibility when it comes to customizing your workflow. You have the possibility of bringing your own Docker image and using it as a runner to speed up the deployments that leverage third party tools. Spacelift's official runner image can be found here.



## Written by

**Jack Roper**

Jack Roper is a highly experienced IT professional with close to 20 years of experience, focused on cloud and DevOps technologies. He specializes in Terraform, Azure, Azure DevOps, and Kubernetes and holds multiple certifications from Microsoft, Amazon, and Hashicorp. Jack enjoys writing technical articles for well-regarded websites.

## Read also

DOCKER

12 min read

### How to Keep Docker Secrets Secure: Complete Guide

DOCKER

11 min read

## Docker Volumes – Guide with Examples

KUBERNETES

9 min read

## How to Delete Pods from a Kubernetes Node with Examples

spacelift

Get our newsletter

Subscribe