Quickstart: Compose and WordPress

*Estimated reading time: 3 minutes*

You can use Docker Compose to easily run WordPress in an isolated environment built with Docker containers. This quick-start guide demonstrates how to use Compose to set up and run WordPress. Before starting, install [Compose installed](https://docs.docker.com/compose/install/).

Define the project

1. Create an empty project directory.

You can name the directory something easy for you to remember. This directory is the context for your application image. The directory should only contain resources to build that image.

This project directory contains a docker-compose.yml file which is complete in itself for a good starter wordpress project.

**Tip**: You can use either a .yml or .yaml extension for this file. They both work.

1. Change directories into your project directory.

For example, if you named your directory my\_wordpress:

cd my\_wordpress/

1. Create a docker-compose.yml file that starts your WordPress blog and a separate MySQL instance with a volume mount for data persistence:
2. version: '3'
3. services:
4. db:
5. image: mysql:5.7
6. volumes:
7. - db\_data:/var/lib/mysql
8. restart: always
9. environment:
10. MYSQL\_ROOT\_PASSWORD: somewordpress
11. MYSQL\_DATABASE: wordpress
12. MYSQL\_USER: wordpress
13. MYSQL\_PASSWORD: wordpress
14. wordpress:
15. depends\_on:
16. - db
17. image: wordpress:latest
18. ports:
19. - "8000:80"
20. restart: always
21. environment:
22. WORDPRESS\_DB\_HOST: db:3306
23. WORDPRESS\_DB\_USER: wordpress
24. WORDPRESS\_DB\_PASSWORD: wordpress
25. volumes:
26. db\_data:

**Notes:**

* The docker volume db\_data persists any updates made by Wordpress to the database. [Learn more about docker volumes](https://docs.docker.com/engine/admin/volumes/volumes/)
* WordPress Multisite works only on ports 80 and 443.

Build the project

Now, run docker-compose up -d from your project directory.

This runs [docker-compose up](https://docs.docker.com/compose/reference/up/) in detached mode, pulls the needed images, and starts the wordpress and database containers, as shown in the example below.

$ docker-compose up -d

Creating network "my\_wordpress\_default" with the default driver

Pulling db (mysql:5.7)...

5.7: Pulling from library/mysql

efd26ecc9548: Pull complete

a3ed95caeb02: Pull complete

...

Digest: sha256:34a0aca88e85f2efa5edff1cea77cf5d3147ad93545dbec99cfe705b03c520de

Status: Downloaded newer image for mysql:5.7

Pulling wordpress (wordpress:latest)...

latest: Pulling from library/wordpress

efd26ecc9548: Already exists

a3ed95caeb02: Pull complete

589a9d9a7c64: Pull complete

...

Digest: sha256:ed28506ae44d5def89075fd5c01456610cd6c64006addfe5210b8c675881aff6

Status: Downloaded newer image for wordpress:latest

Creating my\_wordpress\_db\_1

Creating my\_wordpress\_wordpress\_1

**Note**: WordPress Multisite works only on ports 80 and/or 443. If you get an error message about binding 0.0.0.0 to port 80 or 443 (depending on which one you specified), it is likely that the port you configured for WordPress is already in use by another service.

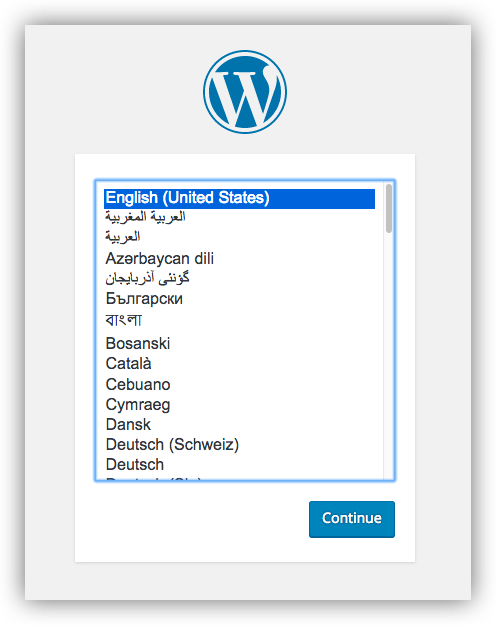
Bring up WordPress in a web browser

At this point, WordPress should be running on port 8000 of your Docker Host, and you can complete the “famous five-minute installation” as a WordPress administrator.

**Note**: The WordPress site is not immediately available on port 8000because the containers are still being initialized and may take a couple of minutes before the first load.

If you are using [Docker Machine](https://docs.docker.com/machine/), you can run the commanddocker-machine ip MACHINE\_VM to get the machine address, and then openhttp://MACHINE\_VM\_IP:8000 in a web browser.

If you are using Docker for Mac or Docker for Windows, you can usehttp://localhost as the IP address, and open http://localhost:8000 in a web browser.





Shutdown and cleanup

The command [docker-compose down](https://docs.docker.com/compose/reference/down/) removes the containers and default network, but preserves your Wordpress database.

The command docker-compose down --volumes removes the containers, default network, and the Wordpress database.