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, make sure you have 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.

2. Change into your project directory.

   For example, if you named your directory `my_wordpress` :

   ```
   cd my_wordpress/
   ```

3. Create a `docker-compose.yml` file that starts your `WordPress` blog and a separate `MySQL` instance with a volume mount for data persistence:

```
version: '3.3'

services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     environment:
       MYSQL_ROOT_PASSWORD: somewordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress

   wordpress:
     depends_on:
       - db
     image: wordpress:latest
     ports:
       - "8000:80"
     restart: always
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
       WORDPRESS_DB_NAME: wordpress
volumes:
    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 Docker 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.

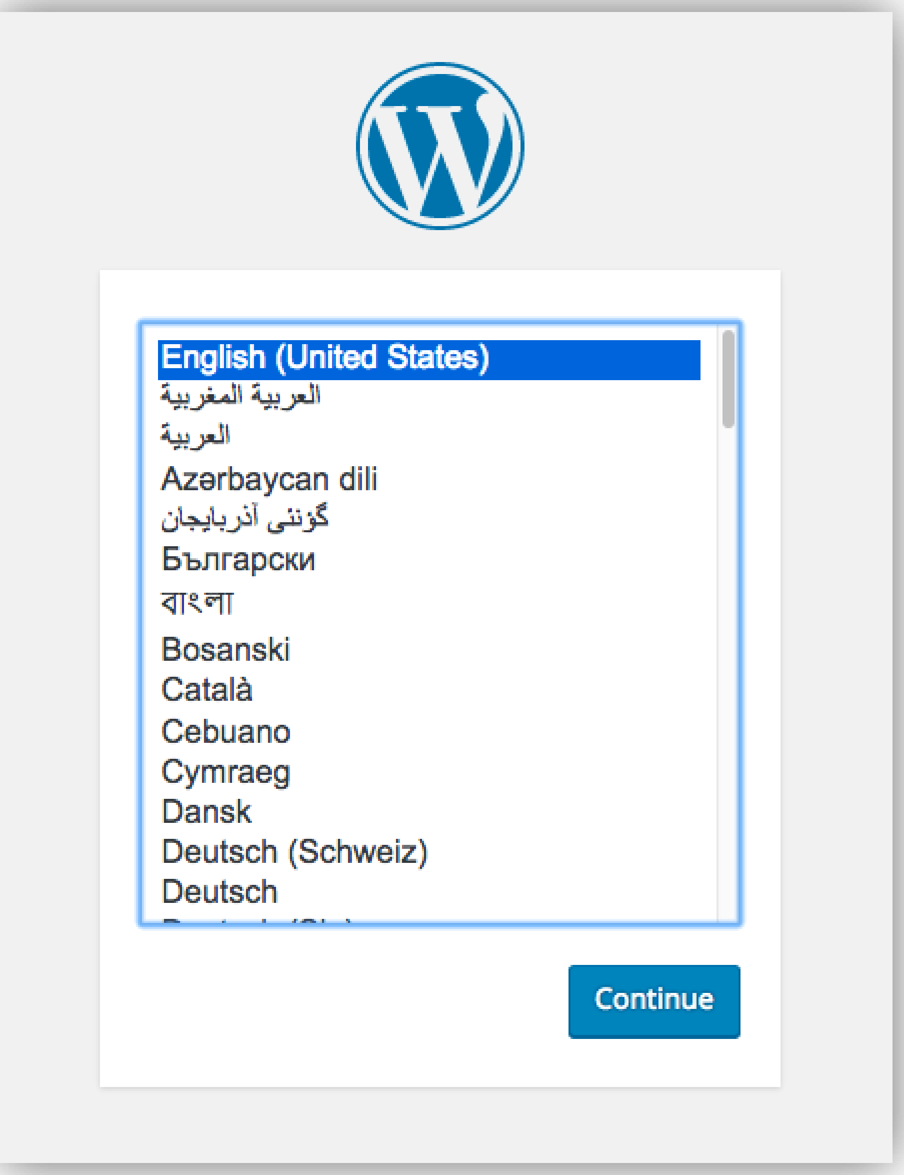
Bring up WordPress in a web browser

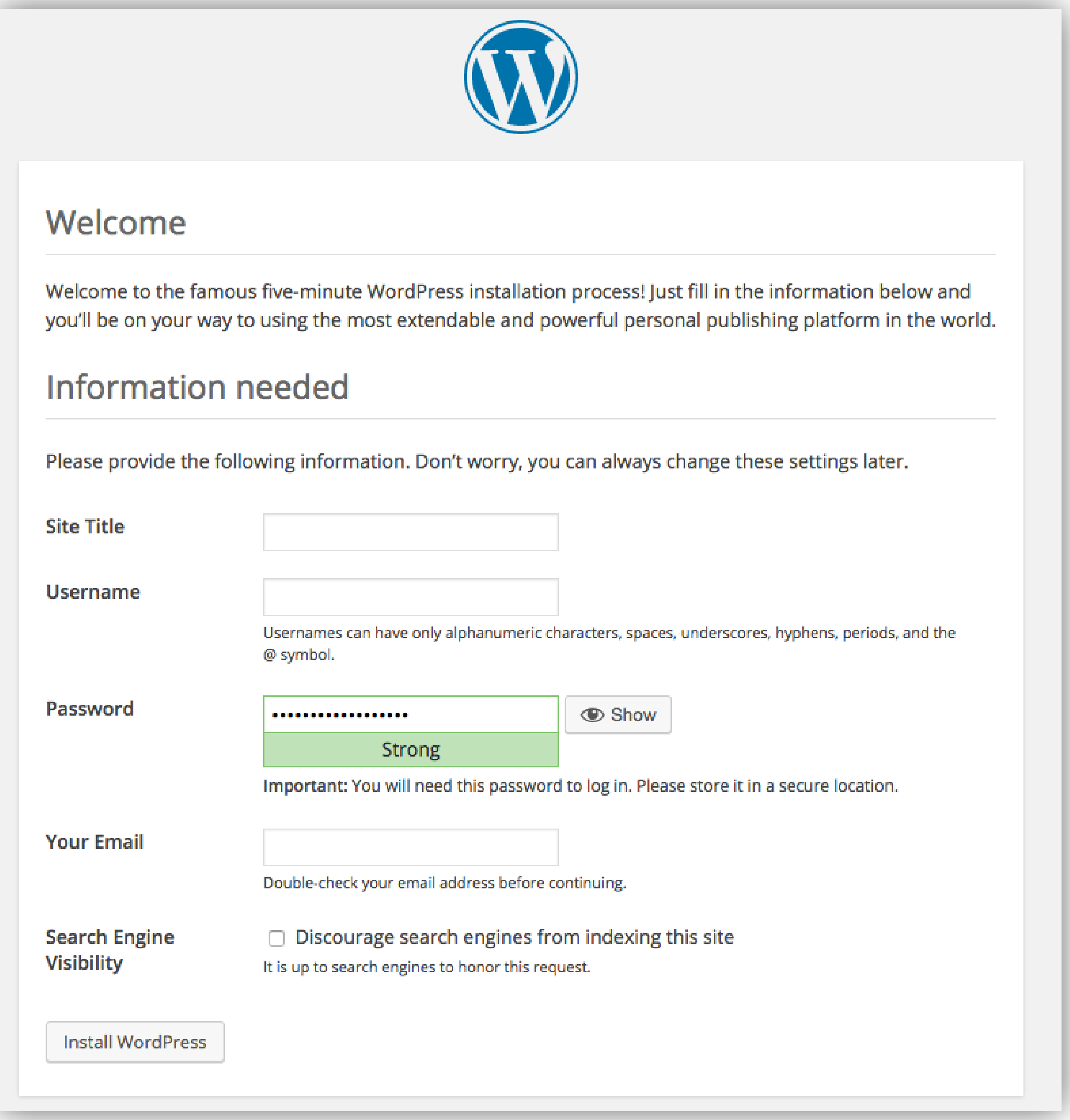
At this point, WordPress should be running on port `8000` of your Docker Host, and you can complete the “famous five-minute installation” as a WordPress administrator.

> **Note**: The WordPress site is not immediately available on port `8000` because 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 command `docker-machine ip MACHINE_VM` to get the machine address, and then open `http://MACHINE_VM_IP:8000` in a web browser.

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


English (United States)
العربية المغربية
العربية
Azərbaycan dili
گؤنئی آذربایجان
Български
বাংলা
Bosanski
Català
Cebuano
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch
Continue



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title	
Username	Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	•••••••••••••••••••• Strong **Important:** You will need this password to log in. Please store it in a secure location.
Your Email	Double-check your email address before continuing.
Search Engine Visibility	□ Discourage search engines from indexing this site It is up to search engines to honor this request.

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.

More Compose documentation

- User guide (https://docs.docker.com/compose/)
- Installing Compose (https://docs.docker.com/compose/install/)
- Getting Started (https://docs.docker.com/compose/gettingstarted/)
- Get started with Django (https://docs.docker.com/compose/django/)
- Get started with Rails (https://docs.docker.com/compose/rails/)
- Command line reference (https://docs.docker.com/compose/reference/)

- Compose file reference (https://docs.docker.com/compose/compose-file/)

documentation (https://docs.docker.com/glossary/?term=documentation), docs (https://docs.docker.com/glossary/?term=docs), docker (https://docs.docker.com/glossary/?term=docker), compose (https://docs.docker.com/glossary/?term=compose), orchestration (https://docs.docker.com/glossary/?term=orchestration), containers (https://docs.docker.com/glossary/?term=containers)