

Short Description

Drupal is an open source content management platform powering millions of websites and applications.

Full Description

Supported tags and respective Dockerfile links

- [8.6.3-apache](#) , [8.6-apache](#) , [8-apache](#) , [apache](#) , [8.6.3](#) , [8.6](#) , [8](#) , [latest](#) ([8.6/apache/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/3f6716f12feaa1e3b3a78d2748941342073f4b32/8.6/apache/Dockerfile>)
- [8.6.3-fpm](#) , [8.6-fpm](#) , [8-fpm](#) , [fpm](#) ([8.6/fpm/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/3f6716f12feaa1e3b3a78d2748941342073f4b32/8.6/fpm/Dockerfile>)
- [8.6.3-fpm-alpine](#) , [8.6-fpm-alpine](#) , [8-fpm-alpine](#) , [fpm-alpine](#) ([8.6/fpm-alpine/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/3f6716f12feaa1e3b3a78d2748941342073f4b32/8.6/fpm-alpine/Dockerfile>)
- [8.5.8-apache](#) , [8.5-apache](#) , [8.5.8](#) , [8.5](#) ([8.5/apache/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/8a9569a2d2e9ded138454b34ae043afd0bdb2660/8.5/apache/Dockerfile>)
- [8.5.8-fpm](#) , [8.5-fpm](#) ([8.5/fpm/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/8a9569a2d2e9ded138454b34ae043afd0bdb2660/8.5/fpm/Dockerfile>)
- [8.5.8-fpm-alpine](#) , [8.5-fpm-alpine](#) ([8.5/fpm-alpine/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/8a9569a2d2e9ded138454b34ae043afd0bdb2660/8.5/fpm-alpine/Dockerfile>)
- [7.61-apache](#) , [7-apache](#) , [7.61](#) , [7](#) ([7/apache/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/3f6716f12feaa1e3b3a78d2748941342073f4b32/7/apache/Dockerfile>)
- [7.61-fpm](#) , [7-fpm](#) ([7/fpm/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/3f6716f12feaa1e3b3a78d2748941342073f4b32/7/fpm/Dockerfile>)
- [7.61-fpm-alpine](#) , [7-fpm-alpine](#) ([7/fpm-alpine/Dockerfile](#)) (<https://github.com/docker-library/drupal/blob/3f6716f12feaa1e3b3a78d2748941342073f4b32/7/fpm-alpine/Dockerfile>)

Quick reference

- **Where to get help:**
[the Docker Community Forums \(https://forums.docker.com/\)](#), [the Docker Community Slack \(https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/\)](#), or [Stack Overflow \(https://stackoverflow.com/search?tab=newest&q=docker\)](#)
- **Where to file issues:**
<https://github.com/docker-library/drupal/issues> (<https://github.com/docker-library/drupal/issues>)

- **Maintained by:**
the Docker Community (<https://github.com/docker-library/drupal>) (not the Drupal Community or the Drupal Security Team)
- **Supported architectures:** (more info (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))
[amd64](https://hub.docker.com/r/amd64/drupal/) (<https://hub.docker.com/r/amd64/drupal/>), [arm32v5](https://hub.docker.com/r/arm32v5/drupal/) (<https://hub.docker.com/r/arm32v5/drupal/>), [arm32v6](https://hub.docker.com/r/arm32v6/drupal/) (<https://hub.docker.com/r/arm32v6/drupal/>), [arm32v7](https://hub.docker.com/r/arm32v7/drupal/) (<https://hub.docker.com/r/arm32v7/drupal/>), [arm64v8](https://hub.docker.com/r/arm64v8/drupal/) (<https://hub.docker.com/r/arm64v8/drupal/>), [i386](https://hub.docker.com/r/i386/drupal/) (<https://hub.docker.com/r/i386/drupal/>), [ppc64le](https://hub.docker.com/r/ppc64le/drupal/) (<https://hub.docker.com/r/ppc64le/drupal/>), [s390x](https://hub.docker.com/r/s390x/drupal/) (<https://hub.docker.com/r/s390x/drupal/>)
- **Published image artifact details:**
[repo-info](https://github.com/docker-library/repo-info/blob/master/repos/drupal/) [repo's](https://github.com/docker-library/repo-info/blob/master/repos/drupal/) [repos/drupal/](https://github.com/docker-library/repo-info/blob/master/repos/drupal/) [directory](https://github.com/docker-library/repo-info/blob/master/repos/drupal/) (<https://github.com/docker-library/repo-info/blob/master/repos/drupal/>) (history (<https://github.com/docker-library/repo-info/commits/master/repos/drupal/>))
(image metadata, transfer size, etc)
- **Image updates:**
[official-images](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fdrupal) PRs with label [library/drupal](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fdrupal) (<https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fdrupal>)
[official-images](https://github.com/docker-library/official-images/blob/master/library/drupal) [repo's](https://github.com/docker-library/official-images/blob/master/library/drupal) [library/drupal](https://github.com/docker-library/official-images/blob/master/library/drupal) [file](https://github.com/docker-library/official-images/blob/master/library/drupal) (<https://github.com/docker-library/official-images/blob/master/library/drupal>) (history (<https://github.com/docker-library/official-images/commits/master/library/drupal>))
- **Source of this description:**
[docs](https://github.com/docker-library/docs/tree/master/drupal) [repo's](https://github.com/docker-library/docs/tree/master/drupal) [drupal/](https://github.com/docker-library/docs/tree/master/drupal) [directory](https://github.com/docker-library/docs/tree/master/drupal/) (<https://github.com/docker-library/docs/tree/master/drupal/>) (history (<https://github.com/docker-library/docs/commits/master/drupal/>))
- **Supported Docker versions:**
the latest release (<https://github.com/docker/docker-ce/releases/latest>) (down to 1.6 on a best-effort basis)

What is Drupal?

Drupal is a free and open-source content-management framework written in PHP and distributed under the GNU General Public License. It is used as a back-end framework for at least 2.1% of all Web sites worldwide ranging from personal blogs to corporate, political, and government sites including WhiteHouse.gov and data.gov.uk. It is also used for knowledge management and business collaboration.

[wikipedia.org/wiki/Drupal](https://en.wikipedia.org/wiki/Drupal) (<https://en.wikipedia.org/wiki/Drupal>)



How to use this image

The basic pattern for starting a `drupal` instance is:

```
$ docker run --name some-drupal -d drupal
```

If you'd like to be able to access the instance from the host without the container's IP, standard port mappings can be used:

```
$ docker run --name some-drupal -p 8080:80 -d drupal
```

Then, access it via `http://localhost:8080` or `http://host-ip:8080` in a browser.

There are multiple database types supported by this image, most easily used via standard container linking. In the default configuration, SQLite can be used to avoid a second container and write to flat-files. More detailed instructions for different (more production-ready) database types follow.

When first accessing the webserver provided by this image, it will go through a brief setup process. The details provided below are specifically for the "Set up database" step of that configuration process.

MySQL

```
$ docker run --name some-drupal --link some-mysql:mysql -d drupal
```

- Database type: MySQL, MariaDB, or equivalent
- Database name/username/password: <details for accessing your MySQL instance>
(`MYSQL_USER` , `MYSQL_PASSWORD` , `MYSQL_DATABASE` ; see environment variables in the description for [mysql](https://hub.docker.com/_/mysql/) (https://hub.docker.com/_/mysql/))
- ADVANCED OPTIONS; Database host: `mysql` (for using the `/etc/hosts` entry added by `--link` to access the linked container's MySQL instance)

PostgreSQL

```
$ docker run --name some-drupal --link some-postgres:postgres -d drupal
```

- Database type: PostgreSQL
- Database name/username/password: <details for accessing your PostgreSQL instance> (POSTGRES_USER , POSTGRES_PASSWORD ; see environment variables in the description for [postgres](https://hub.docker.com/_/postgres/) (https://hub.docker.com/_/postgres/))
- ADVANCED OPTIONS; Database host: postgres (for using the /etc/hosts entry added by --link to access the linked container's PostgreSQL instance)

Volumes

By default, this image does not include any volumes. There is a lot of good discussion on this topic in [docker-library/drupal#3](https://github.com/docker-library/drupal/issues/3) (<https://github.com/docker-library/drupal/issues/3>), which is definitely recommended reading.

There is consensus that /var/www/html/modules , /var/www/html/profiles , and /var/www/html/themes are things that generally ought to be volumes (and might have an explicit VOLUME declaration in a future update to this image), but handling of /var/www/html/sites is somewhat more complex, since the contents of that directory *do* need to be initialized with the contents from the image.

If using bind-mounts, one way to accomplish pre-seeding your local sites directory would be something like the following:

```
$ docker run --rm drupal tar -cC /var/www/html/sites . | tar -xC /path/on/host/sites
```

This can then be bind-mounted into a new container:

```
$ docker run --name some-drupal --link some-postgres:postgres -d \  
-v /path/on/host/modules:/var/www/html/modules \  
-v /path/on/host/profiles:/var/www/html/profiles \  
-v /path/on/host/sites:/var/www/html/sites \  
-v /path/on/host/themes:/var/www/html/themes \  
drupal
```

Another solution using Docker Volumes:

```
$ docker volume create drupal-sites  
$ docker run --rm -v drupal-sites:/temporary/sites drupal cp -aRT /var/www/html/sites /temporary/sites  
$ docker run --name some-drupal --link some-postgres:postgres -d \  
-v drupal-modules:/var/www/html/modules \  
-v drupal-profiles:/var/www/html/profiles \  
-v drupal-sites:/var/www/html/sites \  
-v drupal-themes:/var/www/html/themes \  
drupal
```

... via [docker stack deploy](https://docs.docker.com/engine/reference/commandline/stack_deploy/)

(https://docs.docker.com/engine/reference/commandline/stack_deploy/)

or [docker-compose](https://github.com/docker/compose) (<https://github.com/docker/compose>)

Example stack.yml for drupal :

```
# Drupal with PostgreSQL
#
# Access via "http://localhost:8080"
#   (or "http://$(docker-machine ip):8080" if using docker-machine)
#
# During initial Drupal setup,
# Database type: PostgreSQL
# Database name: postgres
# Database username: postgres
# Database password: example
# ADVANCED OPTIONS; Database host: postgres

version: '3.1'

services:

  drupal:
    image: drupal:8-apache
    ports:
      - 8080:80
    volumes:
      - /var/www/html/modules
      - /var/www/html/profiles
      - /var/www/html/themes
      # this takes advantage of the feature in Docker that a new anonymous
      # volume (which is what we're creating here) will be initialized with the
      # existing content of the image at the same location
      - /var/www/html/sites
    restart: always

  postgres:
    image: postgres:10
    environment:
      POSTGRES_PASSWORD: example
    restart: always
```



Try in PWD ([http://play-with-docker.com?stack=https://raw.githubusercontent.com/docker-](http://play-with-docker.com?stack=https://raw.githubusercontent.com/docker-library/docs/f81077b92e4522999836b8c5d098a103f568a431/drupal/stack.yml)

[library/docs/f81077b92e4522999836b8c5d098a103f568a431/drupal/stack.yml](http://play-with-docker.com?stack=https://raw.githubusercontent.com/docker-library/docs/f81077b92e4522999836b8c5d098a103f568a431/drupal/stack.yml))

Run `docker stack deploy -c stack.yml drupal` (or `docker-compose -f stack.yml up`), wait for it to initialize completely, and visit `http://swarm-ip:8080`, `http://localhost:8080`, or `http://host-ip:8080` (as appropriate). When installing select `postgres` as database with the following parameters: `dbname=postgres user=postgres pass=example hostname=postgres`

Adding additional libraries / extensions

This image does not provide any additional PHP extensions or other libraries, even if they are required by popular plugins. There are an infinite number of possible plugins, and they potentially require any extension PHP supports. Including every PHP extension that exists would dramatically increase the image size.

If you need additional PHP extensions, you'll need to create your own image FROM this one. The [documentation of the php image](https://github.com/docker-library/docs/blob/master/php/README.md#how-to-install-more-php-extensions) (<https://github.com/docker-library/docs/blob/master/php/README.md#how-to-install-more-php-extensions>) explains how to compile additional extensions. Additionally, the [drupal:7](#)

Dockerfile (<https://github.com/docker-library/drupal/blob/bee08efba505b740a14d68254d6e51af7ab2f3ea/7/Dockerfile#L6-9>) has an example of doing this.

The following Docker Hub features can help with the task of keeping your dependent images up-to-date:

- [Automated Builds](https://docs.docker.com/docker-hub/builds/) (<https://docs.docker.com/docker-hub/builds/>) let Docker Hub automatically build your Dockerfile each time you push changes to it.
- [Repository Links](https://docs.docker.com/docker-hub/builds/#repository-links) (<https://docs.docker.com/docker-hub/builds/#repository-links>) can ensure that your image is also rebuilt any time `drupal` is updated.

Running as an arbitrary user

See the "Running as an arbitrary user" section of the [php image documentation](https://hub.docker.com/_/php/) (https://hub.docker.com/_/php/).

Image Variants

The `drupal` images come in many flavors, each designed for a specific use case.

`drupal:<version>`

This is the defacto image. If you are unsure about what your needs are, you probably want to use this one. It is designed to be used both as a throw away container (mount your source code and start the container to start your app), as well as the base to build other images off of.

`drupal:<version>-alpine`

This image is based on the popular [Alpine Linux project](http://alpinelinux.org) (<http://alpinelinux.org>), available in the [alpine official image](https://hub.docker.com/_/alpine/) (https://hub.docker.com/_/alpine/). Alpine Linux is much smaller than most distribution base images (~5MB), and thus leads to much slimmer images in general.

This variant is highly recommended when final image size being as small as possible is desired. The main caveat to note is that it does use [musl libc](http://www.musl-libc.org) (<http://www.musl-libc.org>) instead of [glibc and friends](http://www.etalabs.net/compare_libcs.html) (http://www.etalabs.net/compare_libcs.html), so certain software might run into issues depending on the depth of their libc requirements. However, most software doesn't have an issue with this, so this variant is usually a very safe choice. See [this Hacker News comment thread](https://news.ycombinator.com/item?id=10782897) (<https://news.ycombinator.com/item?id=10782897>) for more discussion of the issues that might arise and some pro/con comparisons of using Alpine-based images.

To minimize image size, it's uncommon for additional related tools (such as `git` or `bash`) to be included in Alpine-based images. Using this image as a base, add the things you need in your own Dockerfile (see the [alpine image description](https://hub.docker.com/_/alpine/) (https://hub.docker.com/_/alpine/) for examples of how to install packages if you are unfamiliar).

License

View [license information](https://www.drupal.org/licensing/faq) (<https://www.drupal.org/licensing/faq>) for the software contained in this image.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).

Some additional license information which was able to be auto-detected might be found in [the repo-info repository's drupal/ directory](https://github.com/docker-library/repo-info/tree/master/repos/drupal) (<https://github.com/docker-library/repo-info/tree/master/repos/drupal>).

As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.

Docker Pull Command



```
docker pull drupal
```