

Short Description

The WordPress rich content management system can utilize plugins, widgets, and themes.

Full Description

Supported tags and respective Dockerfile links

- [4.9.8-php5.6-apache](#) , [4.9-php5.6-apache](#) , [4-php5.6-apache](#) , [php5.6-apache](#) , [4.9.8-php5.6](#) , [4.9-php5.6](#) , [4-php5.6](#) , [php5.6](#) ([php5.6/apache/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php5.6/apache/Dockerfile>)
- [4.9.8-php5.6-fpm](#) , [4.9-php5.6-fpm](#) , [4-php5.6-fpm](#) , [php5.6-fpm](#) ([php5.6/fpm/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php5.6/fpm/Dockerfile>)
- [4.9.8-php5.6-fpm-alpine](#) , [4.9-php5.6-fpm-alpine](#) , [4-php5.6-fpm-alpine](#) , [php5.6-fpm-alpine](#) ([php5.6/fpm-alpine/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php5.6/fpm-alpine/Dockerfile>)
- [4.9.8-php7.0-apache](#) , [4.9-php7.0-apache](#) , [4-php7.0-apache](#) , [php7.0-apache](#) , [4.9.8-php7.0](#) , [4.9-php7.0](#) , [4-php7.0](#) , [php7.0](#) ([php7.0/apache/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.0/apache/Dockerfile>)
- [4.9.8-php7.0-fpm](#) , [4.9-php7.0-fpm](#) , [4-php7.0-fpm](#) , [php7.0-fpm](#) ([php7.0/fpm/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.0/fpm/Dockerfile>)
- [4.9.8-php7.0-fpm-alpine](#) , [4.9-php7.0-fpm-alpine](#) , [4-php7.0-fpm-alpine](#) , [php7.0-fpm-alpine](#) ([php7.0/fpm-alpine/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.0/fpm-alpine/Dockerfile>)
- [4.9.8-php7.1-apache](#) , [4.9-php7.1-apache](#) , [4-php7.1-apache](#) , [php7.1-apache](#) , [4.9.8-php7.1](#) , [4.9-php7.1](#) , [4-php7.1](#) , [php7.1](#) ([php7.1/apache/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.1/apache/Dockerfile>)
- [4.9.8-php7.1-fpm](#) , [4.9-php7.1-fpm](#) , [4-php7.1-fpm](#) , [php7.1-fpm](#) ([php7.1/fpm/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.1/fpm/Dockerfile>)
- [4.9.8-php7.1-fpm-alpine](#) , [4.9-php7.1-fpm-alpine](#) , [4-php7.1-fpm-alpine](#) , [php7.1-fpm-alpine](#) ([php7.1/fpm-alpine/Dockerfile](#)) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.1/fpm-alpine/Dockerfile>)
- [4.9.8-apache](#) , [4.9-apache](#) , [4-apache](#) , [apache](#) , [4.9.8](#) , [4.9](#) , [4](#) , [latest](#) , [4.9.8-php7.2-apache](#) , [4.9-php7.2-apache](#) , [4-php7.2-apache](#) , [php7.2-apache](#) , [4.9.8-php7.2](#) , [4.9-](#)

- [php7.2 . 4-php7.2 . php7.2 \(php7.2/apache/Dockerfile\)](https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.2/apache/Dockerfile) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.2/apache/Dockerfile>)
- [4.9.8-fpm . 4.9-fpm . 4-fpm . fpm . 4.9.8-php7.2-fpm . 4.9-php7.2-fpm . 4-php7.2-fpm . php7.2-fpm \(php7.2/fpm/Dockerfile\)](https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.2/fpm/Dockerfile) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.2/fpm/Dockerfile>)
- [4.9.8-fpm-alpine . 4.9-fpm-alpine . 4-fpm-alpine . fpm-alpine . 4.9.8-php7.2-fpm-alpine . 4.9-php7.2-fpm-alpine . 4-php7.2-fpm-alpine . php7.2-fpm-alpine \(php7.2/fpm-alpine/Dockerfile\)](https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.2/fpm-alpine/Dockerfile) (<https://github.com/docker-library/wordpress/blob/443eb3028423485fdf3943250db5609f3f37afbe/php7.2/fpm-alpine/Dockerfile>)
- [cli-2.0.1-php5.6 . cli-2.0-php5.6 . cli-2-php5.6 . cli-php5.6 \(php5.6/cli/Dockerfile\)](https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php5.6/cli/Dockerfile) (<https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php5.6/cli/Dockerfile>)
- [cli-2.0.1-php7.0 . cli-2.0-php7.0 . cli-2-php7.0 . cli-php7.0 \(php7.0/cli/Dockerfile\)](https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php7.0/cli/Dockerfile) (<https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php7.0/cli/Dockerfile>)
- [cli-2.0.1-php7.1 . cli-2.0-php7.1 . cli-2-php7.1 . cli-php7.1 \(php7.1/cli/Dockerfile\)](https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php7.1/cli/Dockerfile) (<https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php7.1/cli/Dockerfile>)
- [cli-2.0.1 . cli-2.0 . cli-2 . cli . cli-2.0.1-php7.2 . cli-2.0-php7.2 . cli-2-php7.2 . cli-php7.2 \(php7.2/cli/Dockerfile\)](https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php7.2/cli/Dockerfile) (<https://github.com/docker-library/wordpress/blob/08b057215dd2c8fbacdc39cee5b3b58d0f87af63/php7.2/cli/Dockerfile>)

Quick reference

- **Where to get help:**
[the Docker Community Forums](https://forums.docker.com/) (<https://forums.docker.com/>), [the Docker Community Slack](https://blog.docker.com/2016/11/introducing-docker-community-directory-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) (<https://stackoverflow.com/search?tab=newest&q=docker>)
- **Where to file issues:**
<https://github.com/docker-library/wordpress/issues> (<https://github.com/docker-library/wordpress/issues>)
- **Maintained by:**
[the Docker Community](https://github.com/docker-library/wordpress) (<https://github.com/docker-library/wordpress>)
- **Supported architectures:** (more info (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))
[amd64](https://hub.docker.com/r/amd64/wordpress/) (<https://hub.docker.com/r/amd64/wordpress/>), [arm32v5](https://hub.docker.com/r/arm32v5/wordpress/) (<https://hub.docker.com/r/arm32v5/wordpress/>), [arm32v6](https://hub.docker.com/r/arm32v6/wordpress/) (<https://hub.docker.com/r/arm32v6/wordpress/>), [arm32v7](https://hub.docker.com/r/arm32v7/wordpress/) (<https://hub.docker.com/r/arm32v7/wordpress/>), [arm64v8](https://hub.docker.com/r/arm64v8/wordpress/) (<https://hub.docker.com/r/arm64v8/wordpress/>), [i386](https://hub.docker.com/r/i386/wordpress/) (<https://hub.docker.com/r/i386/wordpress/>), [ppc64le](https://hub.docker.com/r/ppc64le/wordpress/) (<https://hub.docker.com/r/ppc64le/wordpress/>), [s390x](https://hub.docker.com/r/s390x/wordpress/) (<https://hub.docker.com/r/s390x/wordpress/>)
- **Published image artifact details:**
[repo-info](https://github.com/docker-library/wordpress) [repo's](https://github.com/docker-library/wordpress) [repos/wordpress/](https://github.com/docker-library/wordpress) [directory](https://github.com/docker-library/wordpress) (<https://github.com/docker-library/wordpress>)
[info/blob/master/repos/wordpress](https://github.com/docker-library/wordpress/blob/master/repos/wordpress)) ([history](https://github.com/docker-library/wordpress/info/commits/master/repos/wordpress) (<https://github.com/docker-library/wordpress/info/commits/master/repos/wordpress>))
 (image metadata, transfer size, etc)
- **Image updates:**
[official-images PRs with label library/wordpress](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fwordpress) (<https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fwordpress>)

[official-images repo's library/wordpress file \(https://github.com/docker-library/official-images/blob/master/library/wordpress\)](https://github.com/docker-library/official-images/blob/master/library/wordpress) (history (<https://github.com/docker-library/official-images/commits/master/library/wordpress>))

- **Source of this description:**

[docs repo's wordpress/ directory \(https://github.com/docker-library/docs/tree/master/wordpress\)](https://github.com/docker-library/docs/tree/master/wordpress) (history (<https://github.com/docker-library/docs/commits/master/wordpress>))

- **Supported Docker versions:**

[the latest release \(https://github.com/docker/docker-ce/releases/latest\)](https://github.com/docker/docker-ce/releases/latest) (down to 1.6 on a best-effort basis)

What is WordPress?

WordPress is a free and open source blogging tool and a content management system (CMS) based on PHP and MySQL, which runs on a web hosting service. Features include a plugin architecture and a template system. WordPress is used by more than 22.0% of the top 10 million websites as of August 2013. WordPress is the most popular blogging system in use on the Web, at more than 60 million websites. The most popular languages used are English, Spanish and Bahasa Indonesia.

[wikipedia.org/wiki/WordPress \(https://en.wikipedia.org/wiki/WordPress\)](https://en.wikipedia.org/wiki/WordPress)



How to use this image

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

The following environment variables are also honored for configuring your WordPress instance:

- `-e WORDPRESS_DB_HOST=...` (defaults to the IP and port of the linked `mysql` container)
- `-e WORDPRESS_DB_USER=...` (defaults to "root")
- `-e WORDPRESS_DB_PASSWORD=...` (defaults to the value of the `MYSQL_ROOT_PASSWORD` environment variable from the linked `mysql` container)
- `-e WORDPRESS_DB_NAME=...` (defaults to "wordpress")
- `-e WORDPRESS_TABLE_PREFIX=...` (defaults to "", only set this when you need to override the default table prefix in `wp-config.php`)
- `-e WORDPRESS_AUTH_KEY=...`, `-e WORDPRESS_SECURE_AUTH_KEY=...`, `-e WORDPRESS_LOGGED_IN_KEY=...`, `-e WORDPRESS_NONCE_KEY=...`, `-e WORDPRESS_AUTH_SALT=...`, `-e WORDPRESS_SECURE_AUTH_SALT=...`, `-e WORDPRESS_LOGGED_IN_SALT=...`, `-e WORDPRESS_NONCE_SALT=...` (default to unique random SHA1s)
- `-e WORDPRESS_DEBUG=1` (defaults to disabled, non-empty value will enable `WP_DEBUG` in `wp-config.php`)
- `-e WORDPRESS_CONFIG_EXTRA=...` (defaults to nothing, non-empty value will be embedded verbatim inside `wp-config.php` -- especially useful for applying extra configuration values this image does not

provide by default such as `WP_ALLOW_MULTISITE` ; see [docker-library/wordpress#142](https://github.com/docker-library/wordpress/pull/142) (<https://github.com/docker-library/wordpress/pull/142>) for more details)

If the `WORDPRESS_DB_NAME` specified does not already exist on the given MySQL server, it will be created automatically upon startup of the `wordpress` container, provided that the `WORDPRESS_DB_USER` specified has the necessary permissions to create it.

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-wordpress --link some-mysql:mysql -p 8080:80 -d wordpress
```

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

If you'd like to use an external database instead of a linked `mysql` container, specify the hostname and port with `WORDPRESS_DB_HOST` along with the password in `WORDPRESS_DB_PASSWORD` and the username in `WORDPRESS_DB_USER` (if it is something other than `root`):

```
$ docker run --name some-wordpress -e WORDPRESS_DB_HOST=10.1.2.3:3306 \
  -e WORDPRESS_DB_USER=... -e WORDPRESS_DB_PASSWORD=... -d wordpress
```

When running WordPress with TLS behind a reverse proxy such as NGINX which is responsible for doing TLS termination, be sure to set `X-Forwarded-Proto` appropriately (see ["Using a Reverse Proxy" in "Administration Over SSL" in upstream's documentation](#)

(https://codex.wordpress.org/Administration_Over_SSL#Using_a_Reverse_Proxy)). No additional environment variables or configuration should be necessary (this image automatically adds the noted

`HTTP_X_FORWARDED_PROTO` code to `wp-config.php` if any of the above-noted environment variables are specified).

If your database requires SSL, [WordPress ticket #28625](https://core.trac.wordpress.org/ticket/28625) (<https://core.trac.wordpress.org/ticket/28625>) has the relevant details regarding support for that with WordPress upstream. As a workaround, the ["Secure DB Connection" plugin](https://wordpress.org/plugins/secure-db-connection/) (<https://wordpress.org/plugins/secure-db-connection/>) can be extracted into the WordPress directory and the appropriate values described in the configuration of that plugin added in `wp-config.php`.

Docker Secrets

As an alternative to passing sensitive information via environment variables, `_FILE` may be appended to the previously listed environment variables, causing the initialization script to load the values for those variables from files present in the container. In particular, this can be used to load passwords from Docker secrets stored in `/run/secrets/<secret_name>` files. For example:

```
$ docker run --name some-wordpress -e WORDPRESS_DB_PASSWORD_FILE=/run/secrets/mysq
```

Currently, this is supported for `WORDPRESS_DB_HOST` , `WORDPRESS_DB_USER` , `WORDPRESS_DB_PASSWORD` , `WORDPRESS_DB_NAME` , `WORDPRESS_AUTH_KEY` , `WORDPRESS_SECURE_AUTH_KEY` , `WORDPRESS_LOGGED_IN_KEY` , `WORDPRESS_NONCE_KEY` , `WORDPRESS_AUTH_SALT` , `WORDPRESS_SECURE_AUTH_SALT` , `WORDPRESS_LOGGED_IN_SALT` , `WORDPRESS_NONCE_SALT` , `WORDPRESS_TABLE_PREFIX` , and `WORDPRESS_DEBUG` .

... via [docker 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 `wordpress` :

```
version: '3.1'
```

```
services:
```

```
  wordpress:
```

```
    image: wordpress
```

```
    restart: always
```

```
    ports:
```

```
      - 8080:80
```

```
    environment:
```

```
      WORDPRESS_DB_HOST: db
```

```
      WORDPRESS_DB_USER: exampleuser
```

```
      WORDPRESS_DB_PASSWORD: examplepass
```

```
      WORDPRESS_DB_NAME: exampledb
```

```
  db:
```

```
    image: mysql:5.7
```

```
    restart: always
```

```
    environment:
```

```
      MYSQL_DATABASE: exampledb
```

```
      MYSQL_USER: exampleuser
```

```
      MYSQL_PASSWORD: examplepass
```

```
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
```



Try in PWD

(<http://play-with-docker.com?stack=https://raw.githubusercontent.com/docker-library/docs/db214ae34137ab29c7574f5fbe01bc4eaea6da7e/wordpress/stack.yml>)

Run `docker stack deploy -c stack.yml wordpress` (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).

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 [wordpress Dockerfile](https://github.com/docker-library/wordpress/blob/618490d4bdff6c5774b84b717979bfe3d6ba8ad1/apache/Dockerfile#L5-L9) (<https://github.com/docker-library/wordpress/blob/618490d4bdff6c5774b84b717979bfe3d6ba8ad1/apache/Dockerfile#L5-L9>) 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 `wordpress` is updated.

Include pre-installed themes / plugins

Mount the volume containing your themes or plugins to the proper directory; and then apply them through the wp-admin webui. Ensure read/write/execute permissions are in place for the user.

- Themes go in a subdirectory in `/var/www/html/wp-content/themes/`
- Plugins go in a subdirectory in `/var/www/html/wp-content/plugins/`

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/).

When running WP-CLI via the `cli` variants of this image, it is important to note that they're based on Alpine, and have a default `USER` of Alpine's `www-data`, whose `UID` is `82` (compared to the Debian-based WordPress variants whose default effective `UID` is `33`), so when running `wordpress:cli` against an existing Debian-based WordPress install, something like `--user 33:33` is likely going to be necessary. See [docker-library/wordpress#256](https://github.com/docker-library/wordpress/issues/256) (<https://github.com/docker-library/wordpress/issues/256>) for more discussion around this.

Image Variants

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

`wordpress:<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.

`wordpress:<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).

`wordpress:cli`

This image variant does not contain WordPress itself, but instead contains [WP-CLI](https://wp-cli.org) (<https://wp-cli.org>).

The simplest way to use it with an existing WordPress container would be something similar to the following:

```
$ docker run -it --rm \
  --volumes-from some-wordpress \
  --network container:some-wordpress \
  wordpress:cli user list
```

Generally speaking, for WP-CLI to interact with a WordPress install, it needs access to the on-disk files of the WordPress install, and access to the database (and the easiest way to accomplish that such that `wp-config.php` does not require changes is to simply join the networking context of the existing and presumably working WordPress container, but there are many other ways to accomplish that which will be left as an exercise for the reader).

License

View [license information \(https://wordpress.org/about/license/\)](https://wordpress.org/about/license/) 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 `wordpress/` directory \(https://github.com/docker-library/repo-info/tree/master/repos/wordpress\)](https://github.com/docker-library/repo-info/tree/master/repos/wordpress).

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 wordpress
```