

Missed DockerCon 2022? Watch now on-demand.



Search for great conte...

[Explore](#)[Pricing](#)[Sign In](#)[Register](#)[Explore](#)[Official Images](#)[adminer](#)

adminer



DOCKER OFFICIAL IMAGE



Database management in a single PHP file.

100M+[Linux](#)[PowerPC 64 LE](#)[IBM Z](#)[x86-64](#)[ARM](#)[ARM 64](#)[386](#)[Docker Official Image](#)

Copy and paste to pull this image

```
docker pull adminer
```

[View Available Tags](#)[Description](#)[Reviews](#)[Tags](#)

Quick reference

- **Maintained by:**
[Tim Düsterhus](#) (of the Docker Community)
- **Where to get help:**
[the Docker Community Forums](#), [the Docker Community Slack](#), or [Stack Overflow](#)

Supported tags and respective Dockerfile links

- [4.8.1-standalone](#) , [4-standalone](#) , [standalone](#) , [4.8.1](#) , [4](#) , [latest](#)
- [4.8.1-fastcgi](#) , [4-fastcgi](#) , [fastcgi](#)

Quick reference (cont.)

- **Where to file issues:**
<https://github.com/TimWolla/docker-adminer/issues>
- **Supported architectures:** ([more info](#))
[amd64](#) , [arm32v6](#) , [arm32v7](#) , [arm64v8](#) , [i386](#) , [ppc64le](#) , [s390x](#)
- **Published image artifact details:**
[repo-info](#) [repo's](#) [repos/adminer/](#) [directory](#) ([history](#))
(image metadata, transfer size, etc)
- **Image updates:**
[official-images](#) [repo's](#) [library/adminer](#) [label](#)
[official-images](#) [repo's](#) [library/adminer](#) [file](#) ([history](#))
- **Source of this description:**

[docs repo's](#) [adminer/](#) [directory \(history\)](#)

Adminer

What is Adminer?

Adminer (formerly phpMinAdmin) is a full-featured database management tool written in PHP. Conversely to phpMyAdmin, it consist of a single file ready to deploy to the target server. Adminer is available for MySQL, PostgreSQL, SQLite, MS SQL, Oracle, Firebird, SimpleDB, Elasticsearch and MongoDB.

adminer.org

How to use this image

Standalone

```
$ docker run --link some_database:db -p 8080:8080 adminer
```

Then you can hit `http://localhost:8080` or `http://host-ip:8080` in your browser.

FastCGI

If you are already running a FastCGI capable web server you might prefer running Adminer via FastCGI:

```
$ docker run --link some_database:db -p 9000:9000 adminer:fastcgi
```

Then point your web server to port 9000 of the container.

Note: This exposes the FastCGI socket to the Internet. Make sure to add proper firewall rules or use a private Docker network instead to prevent a direct access.

... via **docker stack deploy** or **docker-compose**

Example `stack.yml` for `adminer`:

```
# Use root/example as user/password credentials
```

```
version: '3.1'
```

```
services:
```

```
  adminer:
```

```
    image: adminer
```

```
    restart: always
```

```
    ports:
```

```
      - 8080:8080
```

```
  db:
```

```
    image: mysql:5.6
```

```
    restart: always
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: example
```



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

Loading plugins

This image bundles all official Adminer plugins. You can find the list of plugins on GitHub: <https://github.com/vrana/adminer/tree/master/plugins>.

To load plugins you can pass a list of filenames in `ADMINER_PLUGINS` :

```
$ docker run --link some_database:db -p 8080:8080 -e ADMINER_PLUGINS='tables-filter tinymce' adminer
```

If a plugin *requires* parameters to work correctly instead of adding the plugin to `ADMINER_PLUGINS` , you need to add a custom file to the container:

```
$ docker run --link some_database:db -p 8080:8080 -e ADMINER_PLUGINS='login-servers' adminer
```

Unable to load plugin file "login-servers", because it has required parameters: servers

Create a file `"/var/www/html/plugins-enabled/login-servers.php"` with the following contents to load the plugin:

```
<?php
require_once('plugins/login-servers.php');

/** Set supported servers
 * @param array array($domain) or array($domain => $description) or array($category => array())
 * @param string
 */
return new AdminerLoginServers(
    $servers = ???,
    $driver = 'server'
);
```

To load a custom plugin you can add PHP scripts that return the instance of the plugin object to `/var/www/html/plugins-enabled/` .

Choosing a design

The image bundles all the designs that are available in the source package of adminer. You can find the list of designs on GitHub:

<https://github.com/vrana/adminer/tree/master/designs>.

To use a bundled design you can pass its name in `ADMINER_DESIGN` :

To use a bundled design you can pass its name in `ADMINER_DESIGN` :

```
$ docker run --link some_database:db -p 8080:8080 -e ADMINER_DESIGN='nette' adminer
```

To use a custom design you can add a file called `/var/www/html/adminer.css` .

Usage with external server

You can specify the default host with the `ADMINER_DEFAULT_SERVER` environment variable. This is useful if you are connecting to an external server or a docker container named something other than the default `db` .

```
docker run -p 8080:8080 -e ADMINER_DEFAULT_SERVER=mysql adminer
```

Supported Drivers

While Adminer supports a wide range of database drivers this image only supports the following out of the box:

- MySQL
- PostgreSQL
- SQLite
- SimpleDB
- Elasticsearch

To add support for the other drivers you will need to install the following PHP extensions on top of this image:

- `pdo_dblib` (MS SQL)
- `oci8` (Oracle)

- `interbase` (Firebird)
- `mongodb` (MongoDB)

License

View [license information](#) 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 `adminer/` directory](#).

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.



Why Docker

[Overview](#)

[What is a Container](#)

Products

[Product Overview](#)

Product Offerings

[Docker Desktop](#)

[Docker Hub](#)

Features

[- ... - ...](#)

Developers

[Getting Started](#)

[Play with Docker](#)

[Community](#)

[Open Source](#)

[Docs](#)

[Hub Release Notes](#)

Company

[About Us](#)

[Resources](#)

[Blog](#)

[Customers](#)

[Partners](#)

[Newsroom](#)

[Events and Webinars](#)

Container Runtime

Developer Tools

Docker App

Kubernetes

Careers

Contact Us

© 2022 Docker Inc. All rights reserved | [Terms of Service](#) | [Subscription Service Agreement](#) | [Privacy](#) | [Legal](#)