Install Bookstack Github

Link: https://github.com/linuxserver/docker-bookstack

git clone https://github.com/linuxserver/docker-bookstack.git

linuxserver.io

Blog Discord Discourse Fleet GitHub Open Collective

The LinuxServer.io team brings you another container release featuring:

- regular and timely application updates
- easy user mappings (PGID, PUID)
- custom base image with s6 overlay
- weekly base OS updates with common layers across the entire LinuxServer.io ecosystem to minimise space usage, down time and bandwidth
- regular security updates

Find us at:

- Blog - all the things you can do with our containers including How-To guides, opinions and much more!
- Discord - realtime support / chat with the community and the team.
- Discourse - post on our community forum.
- Fleet - an online web interface which displays all of our maintained images.
- GitHub - view the source for all of our repositories.
- Open Collective - please consider helping us by either donating or contributing to our budget

linuxserver/bookstack

Scarf.io pulls GitHub Stars GitHub Release GitHub Package Repository GitLab Container Registry Quay.io Docker Pulls Docker Stars Jenkins Build

Bookstack is a free and open source Wiki designed for creating beautiful documentation. Featuring a simple, but powerful WYSIWYG editor it allows for teams to create detailed and useful documentation with ease.

Powered by SQL and including a Markdown editor for those who prefer it, BookStack is geared towards making documentation more of a pleasure than a chore.

For more information on BookStack visit their website and check it out: https://www.bookstackapp.com

bookstack

Supported Architectures

We utilise the docker manifest for multi-platform awareness. More information is available from docker here and our announcement here.

Simply pulling `lscr.io/linuxserver/bookstack:latest` should retrieve the correct image for your arch, but you can also pull specific arch images via tags.

The architectures supported by this image are:

Architecture	Available	Tag
x86-64	✅	amd64-<version tag>
arm64	✅	arm64v8-<version tag>
armhf	✅	

Application Setup

The default username is admin@admin.com with the password of **password**, access the container at http://:6875.

This application is dependent on a MariaDB database, be it one you already have or a new one. If you do not already have one, we provide an image here https://github.com/linuxserver/docker-mariadb.

If you intend to use this application behind a subfolder reverse proxy, such as our SWAG container or Traefik you will need to make sure that the `APP_URL` environment variable is set to your external

domain, or it will not work.

Documentation for BookStack can be found at https://www.bookstackapp.com/docs/.

BookStack File & Directory Paths

This container ensures certain BookStack application files & folders, such as user file upload folders, are retained within the `/config` folder so that they are persistent & accessible when the `/config` container path is bound as a volume. There may be cases, when following the BookStack documentation, that you'll need to know how these files and folders are used relative to a non-container BookStack installation.

Below is a mapping of container `/config` paths to those relative within a BookStack install directory:

- **/config container path** => **BookStack relative path**
- `/config/www/.env` => `.env`
- `/config/log/bookstack/laravel.log` => `storage/logs/laravel.log`
- `/config/backups/` => `storage/backups/`
- `/config/www/files/` => `storage/uploads/files/`
- `/config/www/images/` => `storage/uploads/images/`
- `/config/www/themes/` => `themes/`
- `/config/www/uploads/` => `public/uploads/`

Changing APP_URL

If you change the APP_URL after initial install, you should run the following line from your host terminal to update the database URL entries:

```
docker exec -it bookstack php /app/www/artisan bookstack:update-url ${OLD_URL} ${NEW_URL}
```

Advanced Users (full control over the .env file)

If you wish to use the extra functionality of BookStack such as email, LDAP and so on you will need to set additional environment variables or make your own .env file with guidance from the BookStack documentation.

The container will copy an exemplary .env file to /config/www/.env on your host system for you to use.

Read-Only Operation

This image can be run with a read-only container filesystem. For details please read the docs.

Caveats

- `/tmp` must be mounted to tmpfs

Usage

To help you get started creating a container from this image you can either use docker-compose or the docker cli.

Note

Unless a parameter is flaged as 'optional', it is *mandatory* and a value must be provided.

docker-compose (recommended, click here for more info)

```
---
services:
  bookstack:
    image: lscr.io/linuxserver/bookstack:latest
    container_name: bookstack
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
      - APP_URL=
      - APP_KEY=
      - DB_HOST=
      - DB_PORT=3306
      - DB_USERNAME=
```

```
      - DB_PASSWORD=
      - DB_DATABASE=
      - QUEUE_CONNECTION= #optional
    volumes:
      - /path/to/bookstack/config:/config
    ports:
      - 6875:80
    restart: unless-stopped
```

docker cli (click here for more info)

```
docker run -d \
  --name=bookstack \
  -e PUID=1000 \
  -e PGID=1000 \
  -e TZ=Etc/UTC \
  -e APP_URL= \
  -e APP_KEY= \
  -e DB_HOST= \
  -e DB_PORT=3306 \
  -e DB_USERNAME= \
  -e DB_PASSWORD= \
  -e DB_DATABASE= \
  -e QUEUE_CONNECTION= `#optional` \
  -p 6875:80 \
  -v /path/to/bookstack/config:/config \
  --restart unless-stopped \
  lscr.io/linuxserver/bookstack:latest
```

Parameters

Containers are configured using parameters passed at runtime (such as those above). These parameters are separated by a colon and indicate `<external>:<internal>` respectively. For example, `-p 8080:80` would expose port `80` from inside the container to be accessible from the host's IP on port `8080` outside the container.

Parameter	Function
`-p 6875:80`	http/s web interface.
`-e PUID=1000`	for UserID - see below for explanation
`-e PGID=1000`	for GroupID - see below for explanation

Parameter	Function
`-e TZ=Etc/UTC`	specify a timezone to use, see this list.
`-e APP_URL=`	The protocol, IP/URL, and port that your application will be accessed on (ie. `http://192.168.1.1:6875` or `https://bookstack.mydomain.com`
`-e APP_KEY=`	Session encryption key. You will need to generate this with `docker run -it --rm --entrypoint /bin/bash lscr.io/linuxserver/bookstack:latest appkey`
`-e DB_HOST=`	The database instance hostname
`-e DB_PORT=3306`	Database port
`-e DB_USERNAME=`	Database user
`-e DB_PASSWORD=`	Database password (minimum 4 characters & non-alphanumeric passwords must be properly escaped.)
`-e DB_DATABASE=`	Database name
`-e QUEUE_CONNECTION=`	Set to `database` to enable async actions like sending email or triggering webhooks. See documentation.
`-v /config`	Persistent config files
`--read-only=true`	Run container with a read-only filesystem. Please read the docs.

Environment variables from files (Docker secrets)

You can set any environment variable from a file by using a special prepend `FILE__`.

As an example:

```
-e FILE__MYVAR=/run/secrets/mysecretvariable
```

Will set the environment variable `MYVAR` based on the contents of the `/run/secrets/mysecretvariable` file.

Umask for running applications

For all of our images we provide the ability to override the default umask settings for services started within the containers using the optional `-e UMASK=022` setting. Keep in mind umask is not chmod it subtracts from permissions based on it's value it does not add. Please read up

here before asking for support.

User / Group Identifiers

When using volumes (`-v` flags), permissions issues can arise between the host OS and the container, we avoid this issue by allowing you to specify the user `PUID` and group `PGID`.

Ensure any volume directories on the host are owned by the same user you specify and any permissions issues will vanish like magic.

In this instance `PUID=1000` and `PGID=1000`, to find yours use `id your_user` as below:

```
id your_user
```

Example output:

```
uid=1000(your_user) gid=1000(your_user) groups=1000(your_user)
```

Docker Mods

Docker Mods Docker Universal Mods

We publish various Docker Mods to enable additional functionality within the containers. The list of Mods available for this image (if any) as well as universal mods that can be applied to any one of our images can be accessed via the dynamic badges above.

Support Info

- Shell access whilst the container is running:
  ```
  docker exec -it bookstack /bin/bash
  ```
- To monitor the logs of the container in realtime:
  ```
  docker logs -f bookstack
  ```
- Container version number:

```
docker inspect -f '{{ index .Config.Labels "build_version" }}' bookstack
```

- Image version number:

```
docker inspect -f '{{ index .Config.Labels "build_version" }}' lscr.io/linuxserver/bookstac
```

Updating Info

Most of our images are static, versioned, and require an image update and container recreation to update the app inside. With some exceptions (noted in the relevant readme.md), we do not recommend or support updating apps inside the container. Please consult the Application Setup section above to see if it is recommended for the image.

Below are the instructions for updating containers:

Via Docker Compose

- Update images:
  - All images:

```
docker-compose pull
```

  - Single image:

```
docker-compose pull bookstack
```

- Update containers:
  - All containers:

```
docker-compose up -d
```

  - Single container:

```
docker-compose up -d bookstack
```

- You can also remove the old dangling images:

```
docker image prune
```

Via Docker Run

- Update the image:

```
docker pull lscr.io/linuxserver/bookstack:latest
```

- Stop the running container:
```
docker stop bookstack
```
- Delete the container:
```
docker rm bookstack
```
- Recreate a new container with the same docker run parameters as instructed above (if mapped correctly to a host folder, your `/config` folder and settings will be preserved)
- You can also remove the old dangling images:
```
docker image prune
```

Image Update Notifications - Diun (Docker Image Update Notifier)

Tip

We recommend Diun for update notifications. Other tools that automatically update containers unattended are not recommended or supported.

Building locally

If you want to make local modifications to these images for development purposes or just to customize the logic:

```
git clone https://github.com/linuxserver/docker-bookstack.git
cd docker-bookstack
docker build \
  --no-cache \
  --pull \
  -t lscr.io/linuxserver/bookstack:latest .
```

The ARM variants can be built on x86_64 hardware and vice versa using `lscr.io/linuxserver/qemu-static`

```
docker run --rm --privileged lscr.io/linuxserver/qemu-static --reset
```

Once registered you can define the dockerfile to use with `-f Dockerfile.aarch64`.

Versions

- **04.01.25:** - Add php-opcache.
- **17.12.24:** - Rebase to Alpine 3.21.
- **11.10.24:** - Default to environment config over .env file config.
- **06.09.24:** - Add php-exif for reading image EXIF data.
- **27.05.24:** - Rebase to Alpine 3.20. Existing users should update their nginx confs to avoid http2 deprecation warnings.
- **25.01.24:** - Existing users should update: site-confs/default.conf - Cleanup default site conf.
- **23.12.23:** - Rebase to Alpine 3.19 with php 8.3.
- **31.10.23:** - Further sanitize sed replace.
- **07.06.23:** - Add mariadb-client for bookstack-system-cli support.
- **25.05.23:** - Rebase to Alpine 3.18, deprecate armhf.
- **13.04.23:** - Move ssl.conf include to default.conf.
- **01.03.23:** - Add php iconv.
- **19.01.23:** - Rebase to alpine 3.17 with php8.1.
- **16.01.23:** - Wrap `.env` values in quotes.
- **05.01.23:** - Fix db password setting (sed escape `&`).
- **21.12.22:** - Update db info in .env file when env vars are updated.
- **10.10.22:** - Remove password escape logic which caused problems for a small subset of users.
- **20.08.22:** - Rebasing to alpine 3.15 with php8. Restructure nginx configs (see changes announcement).
- **14.03.22:** - Add symlinks for theme support.
- **11.07.21:** - Rebase to Alpine 3.14.
- **12.01.21:** - Remove unused requirement, as of release 0.31.0.
- **17.12.20:** - Make APP_URL var required (upstream changes).
- **17.09.20:** - Rebase to alpine 3.12. Fix APP_URL setting. Bump php post max and upload max filesizes to 100MB by default.
- **19.12.19:** - Rebasing to alpine 3.11.
- **26.07.19:** - Use old version of tidyhtml pending upstream fixes.
- **28.06.19:** - Rebasing to alpine 3.10.
- **14.06.19:** - Add wkhtmltopdf to image for PDF rendering.
- **20.04.19:** - Rebase to Alpine 3.9, add MySQL init logic.
- **22.03.19:** - Switching to new Base images, shift to arm32v7 tag.
- **20.01.19:** - Added php7-curl
- **04.11.18:** - Added php7-ldap
- **15.10.18:** - Changed functionality for advanced users
- **08.10.18:** - Advanced mode, symlink changes, sed fixing, docs updated, added some composer files
- **23.09.28:** - Updates pre-release
- **02.07.18:** - Initial Release.

Revision #1
Created 2025-05-23 22:20:28 UTC by Administrador
Updated 2025-05-23 22:24:22 UTC by Administrador