

Installation & Konfiguration

Einige Einstellungen, die an der Docker Konfiguration vorgenommen werden können, um Bookstack anzupassen, wie Standard-Sprache, -Theme, Reverse Proxy. Eine vollständige Liste mit allen Umgebungsvariablen gibt es hier:

<https://github.com/BookStackApp/BookStack/blob/92922288dd55ce0f77acc83eea9068cad28dccd9/.env.example.complete#L38>

- [Installation](#)
- [Zeitzone einstellen](#)
- [Bookstack & DB nicht als root](#)
- [Standard-Design anpassen](#)
- [Bookstack hinter Reverse Proxy](#)
- [.env Konfigurationsdatei](#)
- [Ausgehende E-Mails](#)
- [LDAP/AD Synchronisation](#)
- [Cache & Session Konfiguration](#)

Installation

Bookstack kann auf verschiedenen Plattformen installiert werden, siehe hierzu: [Bookstack Docs Installation](#)

Hier wird folgend nur die Bereitstellung mittels Docker beschrieben, da diese Instanz ebenfalls mittels Docker gehostet wird.

Bookstack benötigt für die Verwaltung der Daten eine MariaDB als Backend. Somit müssen 2 Container bereitgestellt werden, zum einen ein MariaDB Container und zum anderen der eigentlich Bookstack Container. Sofern schon eine MariaDB Instanz zu Verfügung steht, kann diese eingebunden werden, anstatt einen neuen Container bereitzustellen.

Am einfachsten ist die Bereitstellung mittels Docker Compose. Dazu einfach folgende Compose Datei anpassen:

```
services:
  bookstack:
    image: lscr.io/linuxserver/bookstack
    container_name: bookstack
    environment:
      - APP_URL=http://Server-IP
      - DB_HOST=bookstack_db
      - DB_PORT=3306
      - DB_USER=bookstack
      - DB_PASS=Ein_langes_Passwort
      - DB_DATABASE=bookstackapp
    volumes:
      - ./bookstack_app_data:/config
    ports:
      - 80:80
    restart: unless-stopped
    depends_on:
      - bookstack_db
  bookstack_db:
    image: lscr.io/linuxserver/mariadb
    container_name: bookstack_db
    environment:
```

- MYSQL_ROOT_PASSWORD=Noch_ein_extra_langes_Passwort
- TZ=Europe/Berlin
- MYSQL_DATABASE=bookstackapp
- MYSQL_USER=bookstack
- MYSQL_PASSWORD=Ein_langes_Passwort

volumes:

- ./bookstack_db_data:/config

restart: unless-stopped

Schauen wir uns noch einige Einstellungen im Detail an.

- APP_URL=http://Server-IP

Die *APP_URL* gibt an, wie die Links innerhalb von Bookstack benannt werden sollen. Sollte Bookstack hinter einem Reverse Proxy agieren, ist hier die URL anzugeben, unter der Bookstack von außerhalb erreichbar sein soll. Auch wenn der Reverse Proxy mittels HTTP und unter einem anderen Namen auf den Bookstack Container zugreift, muss mittels der *APP_URL* die tatsächliche externe HTTPS URL angegeben werden. In diesem Beispiel wird die *Server-IP* eingegeben, dies ist nur für das lokale testen geeignet und ermöglicht es, dass auch von anderen Geräten innerhalb des selben Netzwerkes auf die URL zugegriffen werden kann. Wenn nur auf der Maschine getestet werden soll, die Bookstack bereitstellt, dann kann auch *localhost* angegeben werden.

Bereits vorhandene MariaDB verwenden

- DB_HOST=bookstack_db
- DB_PORT=3306
- DB_USER=bookstack
- DB_PASS=Ein_langes_Passwort
- DB_DATABASE=bookstackapp

In diesem Codeabschnitt wird die Verbindung zur MariaDB konfiguriert. Sofern kein bestehender MariaDB-Server verwendet werden soll, muss hier nur das Passwort geändert werden.

Vorsicht: Das Passwort für den Datenbank Benutzer *bookstack* muss an 2 Stellen angepasst werden. Einmal im gezeigten Code Ausschnitt und bei der Konfiguration des 2. Containers für die MariaDB.

Wenn ein bereits vorhandener MariaDB-Server verwendet werden soll, dann muss auf diesen zuerst die Datenbank *bookstackapp* angelegt werden sowie der Benutzer *bookstack*. Diesem Benutzer müssen dann alle Rechte für die neue Datenbank erteilt werden. Anschließend muss mindestens der *DB_Host* in dem oben gezeigten Codeabschnitt angepasst werden. Außerdem kann der

folgende Codeabschnitt aus der Konfiguration gelöscht werden:

```
depends_on:
  - bookstack_db
bookstack_db:
  image: lscr.io/linuxserver/mariadb
  container_name: bookstack_db
  environment:
    - MYSQL_ROOT_PASSWORD=Noch_ein_extra_langes_Passwort
    - TZ=Europe/Berlin
    - MYSQL_DATABASE=bookstackapp
    - MYSQL_USER=bookstack
    - MYSQL_PASSWORD=Ein_langes_Passwort
  volumes:
    - ./bookstack_db_data:/config
  restart: unless-stopped
```

Damit die Einstellungen und Daten auch nach dem neu erstellen eines Containers noch vorhanden sind, definiert die Konfiguration 2 Ordner als Volumes. In der Beispiel Konfiguration wird der aktuelle Ordner als Hauptordner verwendet, d. h. in diesem werden folgende Ordner erstellt: *bookstack_app_data* und *bookstack_db_data*. Es empfiehlt sich, hier absolute Pfade anzugeben wie: `/home/user/bookstack` anstelle des `./` (Punktes).

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegebenen Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

In wenigen Minuten sollte Bookstack unter *http://Server-IP* erreichbar sein.

Standard-Login:

- E-Mail: admin@admin.com
- Passwort: password

Zeitzone einstellen

Die Zeitzone kann zum einen im Docker Container selbst angepasst werden, sodass die Zeiten im Log des Containers stimmen sowie für die Bookstack App selbst, damit sie auch in Bookstack selbst passend angezeigt werden. Wenn die MariaDB ebenfalls als Container wie in der [Installation](#) beschrieben bereitgestellt wird, empfiehlt es sich auch dort die Zeitzone wie in Bookstack einzustellen.

Anpassung der Zeitzone für Bookstack

Um die Zeitzone anzupassen, sind einige Parameter unter *environment* anzupassen:

```
bookstack:
  environment:
    - TZ=Europe/Berlin
    - APP_TIMEZONE=Europe/Berlin
```

TZ setzt die Zeitzone im Container selbst, sodass die Logs im Container nun in der Zeitzone *Europe/Berlin* geschrieben werden. *APP_TIMEZONE* setzt die Zeitzone für Bookstack, sodass alle Zeiten in Bookstack ebenfalls die selbe Zeitzone verwenden. Die beiden Parameter können einfach zu den bereits vorhandenen *environment* Parametern in Docker Config für den Service *bookstack* hinzugefügt werden.

Anpassung der Zeitzone für MariaDB

Um die Zeitzone für den MariaDB Container anzupassen ist dem *environment* Bereich vom Service *bookstack_db* der Parameter *TZ* hinzufügen.

```
bookstack_db:
  environment:
    - TZ=Europe/Berlin
```

Zum Schluss nochmal `sudo docker-compose up -d` oder `sudo docker compose up -d` ausführen, um die Container neu erstellen zu lassen.

Bookstack & DB nicht als root

Im Standard werden die Container als Systemuser *root* ausgeführt. Dies gilt als Risiko, da ein Ausbruch aus dem Container dazu führt, dass der Angreifer mit *root* Rechten auf dem Host agieren kann. Um dies zu verhindern, können die Dienste in den Containern mit einem anderen Benutzer ausgeführt werden. Hierzu kann die Docker Config aus der [Installation](#) angepasst werden.

```
bookstack:
  environment:
    - PUID=1000
    - PGID=1000
```

Die beiden neuen Parameter (PUID und PGID) werden einfach unter die bereits vorhandenen Parameter in *environment* geschrieben. Sie haben zur Folge, dass die Dienste dann als User mit der ID 1000 und der Gruppen ID 1000 laufen, die IDs sind wichtig und müssen bei den Anpassungen der Berechtigungen berücksichtigt werden.

Durch die Anpassung des Dienst-Benutzers, kann dieser natürlich nicht mehr auf die bereits vorhandenen Dateien zugreifen. Deswegen müssen die Berechtigungen/Eigentümer der Verzeichnisse, die in der Docker Config angegeben sind, geändert werden.

Dazu kann der folgende Befehl angepasst und verwendet werden: `sudo chown -R 1000 ./bookstack_app_data`

Sofern auch die MariaDB als Container wie in der [Installation](#) beschrieben eingerichtet wurde, kann der Dienst-Benutzer auf die selbe Art angepasst werden.

```
bookstack_db:
  environment:
    - PUID=1000
    - PGID=1000
```

Nachdem die Config angepasst ist, kann der Container mit `sudo docker-compose up -d` neu erstellt werden, um die Änderungen anzuwenden.

Standard-Design anpassen

Bookstack erlaubt das Überschreiben einiger Standard-Einstellungen für die Seite. So kann z. B. standardmäßig der essentielle Dark Mode aktiviert, die Ansicht von Büchern und Regalen auf Liste und die Standard-Sprache (sofern nicht vom Browser erkannt) umgestellt werden. Es gibt noch viele weitere Einstellungen, die in der offiziellen Dokumentation nachzulesen sind:

<https://github.com/BookStackApp/BookStack/blob/92922288dd55ce0f77acc83eea9068cad28dccd9/.env.example.complete#L38>

Um die Einstellungen vorzunehmen, kann eine `.env` Datei angelegt und gepflegt werden oder es wird die bereits vorhandene Docker Config aus der [Installation](#) angepasst. Dazu werden einfach die gewünschten Parameter im Bereich `environment` hinzugefügt.

Im folgenden werden folgende Standard-Einstellungen vorgenommen:

- Essentieller, augenfreundlicher Dark Mode anstelle des Light Mode
- Sprache Deutsch, wenn nicht vom Browser erkannt
- Bücher Übersicht als Liste
- Regale Übersicht als Liste

```
bookstack:
  environment:
    - APP_LANG=de
    - APP_DEFAULT_DARK_MODE=true
    - APP_VIEWS_BOOKS=list
    - APP_VIEWS_BOOKSHELVES=list
```

Zum Schluss nochmal `sudo docker-compose up -d` oder `sudo docker compose up -d` ausführen, um die Container neu erstellen zu lassen. Anschließend erhalten Gäste die neuen Standards, registrierte Benutzer können diese einfach überschreiben, indem sie eigene Einstellungen für die Sichten in der Weboberfläche vornehmen.

Bookstack hinter Reverse Proxy

Bookstack hinter einem Reverse Proxy betreiben. Es müssen nicht zwingend Einstellungen an Bookstack vorgenommen werden, um es hinter einem Reverse Proxy zu betreiben. Im Reverse Proxy kann einfach die Konfiguration angepasst werden, sodass dieser Bookstack als Backend ansteuert und im Frontend anbietet. Um jedoch einen sicheren Betrieb sicherzustellen, sollten einige Anpassungen vorgenommen werden.

```
bookstack:
  environment:
    - APP_URL=https://beispiel.de
    - APP_PROXIES=*
    - LOG_FAILED_LOGIN_MESSAGE=true
```

.env Konfigurationsdatei

In der Konfigurationsdatei `.env` speichert Bookstack alle Standard- und Server-Einstellungen. Darunter auch die Einstellungen, die über die environment-Variablen in der Docker Compose Datei festgelegt wurden.

Die Datei `.env` wird automatisch erstellt, wenn der Bookstack Container einmal erzeugt und gestartet wird. Anschließend befindet sich im Ordner, der in der Compose Datei zum Speichern der Bookstack Konfiguration festgelegt ist (`./bookstack_app_data:/config`), der Ordner `www`.

Angenommen es würde zum Beispiel folgendes Volume für den Bookstack Container in der Compose Datei definiert.

```
services:
  bookstack:
    volumes:
      - /home/user/docker/bookstack_app_data:/config
```

Dann würde in diesem Fall die Konfigurationsdatei wie folgt abgelegt:

```
/home/user/docker/bookstack_app_data/www/.env
```

Die Datei kann so bearbeitet werden. Nach der Bearbeitung wird einfach der Container neugestartet und die Änderungen sind direkt angewendet.

In der Datei können zusätzliche Einstellungen vorgenommen werden. Eine vollständige Liste:

[Umfangreiches .env Beispiel](#)

Ausgehende E-Mails

Um automatisch Benachrichtigungen versenden zu können, wie z. B. bei Kommentaren oder beobachteten Seiten, ist ein SMTP-Server samt Postfach in Bookstack zu hinterlegen. Hierzu ist die `.env` Konfigurationsdatei anzupassen.

.env Konfigurationsdatei

In der Konfigurationsdatei `.env` speichert Bookstack alle Standard- und Server-Einstellungen. Darunter auch die Einstellungen, die über die environment-Variablen in der Docker Compose Datei festgelegt wurden.

Die Datei `.env` wird automatisch erstellt, wenn der Bookstack Container einmal erzeugt und gestartet wird. Anschließend befindet sich im Ordner, der in der Compose Datei zum Speichern der Bookstack Konfiguration festgelegt ist (`./bookstack_app_data:/config`), der Ordner `www`.

Angenommen es würde zum Beispiel folgendes Volume für den Bookstack Container in der Compose Datei definiert.

```
services:
  bookstack:
    volumes:
      - /home/user/docker/bookstack_app_data:/config
```

Dann würde in diesem Fall die Konfigurationsdatei wie folgt abgelegt:

```
/home/user/docker/bookstack_app_data/www/.env
```

Die Datei kann so bearbeitet werden. Nach der Bearbeitung wird einfach der Container neugestartet und die Änderungen sind direkt angewendet.

In der Datei können zusätzliche Einstellungen vorgenommen werden. Eine vollständige Liste: [Umfangreiches .env Beispiel](#)

In dieser Datei sind folgende Optionen zu ergänzen.

```
# Mail system to use
# Can be 'smtp' or 'sendmail'
MAIL_DRIVER=smtp
```

```
# Mail sender details
MAIL_FROM_NAME="Bookstack Postach"
MAIL_FROM=bookstack@beispiel.de

# SMTP mail options
# These settings can be checked using the "Send a Test Email"
# feature found in the "Settings > Maintenance" area of the system.
# For more detailed documentation on mail options, refer to:
# https://www.bookstackapp.com/docs/admin/email-webhooks/#email-configuration
MAIL_HOST=smtp.beispiel.de
MAIL_PORT=587
MAIL_USERNAME=bookstack@beispiel.de
MAIL_PASSWORD=SuperSicheresLangesPasswort
MAIL_ENCRYPTION=tls

# Folgende Option ist ein Sicherheitsrisiko
#MAIL_VERIFY_SSL=false
```

Danach den Container neustarten und fertig.

LDAP/AD Synchronisation

Statt der "normalen" Authentifizierung kann auch z. B. ein LDAP als Authentifizierung verwendet werden. Dann können sich die Benutzer im LDAP mit den gleichen Anmeldedaten in Bookstack anmelden und sogar die Gruppen können synchronisiert und für Rollenzuteilungen verwendet werden.

AD/LDAP Synchronisation: <https://www.bookstackapp.com/docs/admin/ldap-auth/>

Die in der Dokumentation beschriebenen Einstellungen über die .env Datei setzen, nicht die Environment Variablen in der Docker Compose verwenden.

.env Konfigurationsdatei

In der Konfigurationsdatei .env speichert Bookstack alle Standard- und Server-Einstellungen. Darunter auch die Einstellungen, die über die environment-Variablen in der Docker Compose Datei festgelegt wurden.

Die Datei .env wird automatisch erstellt, wenn der Bookstack Container einmal erzeugt und gestartet wird. Anschließend befindet sich im Ordner, der in der Compose Datei zum Speichern der Bookstack Konfiguration festgelegt ist (./bookstack_app_data:/config), der Ordner www.

Angenommen es würde zum Beispiel folgendes Volume für den Bookstack Container in der Compose Datei definiert.

```
services:
  bookstack:
    volumes:
      - /home/user/docker/bookstack_app_data:/config
```

Dann würde in diesem Fall die Konfigurationsdatei wie folgt abgelegt:

```
/home/user/docker/bookstack_app_data/www/.env
```

Die Datei kann so bearbeitet werden. Nach der Bearbeitung wird einfach der Container neugestartet und die Änderungen sind direkt angewendet.

In der Datei können zusätzliche Einstellungen vorgenommen werden. Eine vollständige Liste: [Umfangreiches .env Beispiel](#)

Cache & Session Konfiguration

Über .env Datei lassen sich der Cache und die Verarbeitung der Sessions (Sitzungen) anpassen.

.env Konfigurationsdatei

In der Konfigurationsdatei .env speichert Bookstack alle Standard- und Server-Einstellungen. Darunter auch die Einstellungen, die über die environment-Variablen in der Docker Compose Datei festgelegt wurden.

Die Datei .env wird automatisch erstellt, wenn der Bookstack Container einmal erzeugt und gestartet wird. Anschließend befindet sich im Ordner, der in der Compose Datei zum Speichern der Bookstack Konfiguration festgelegt ist (./bookstack_app_data:/config), der Ordner www.

Angenommen es würde zum Beispiel folgendes Volume für den Bookstack Container in der Compose Datei definiert.

```
services:
  bookstack:
    volumes:
      - /home/user/docker/bookstack_app_data:/config
```

Dann würde in diesem Fall die Konfigurationsdatei wie folgt abgelegt:

```
/home/user/docker/bookstack_app_data/www/.env
```

Die Datei kann so bearbeitet werden. Nach der Bearbeitung wird einfach der Container neugestartet und die Änderungen sind direkt angewendet.

In der Datei können zusätzliche Einstellungen vorgenommen werden. Eine vollständige Liste:

[Umfangreiches .env Beispiel](#)

Cache & Sitzungsspeicher

In der Standard Einstellung nutzt Bookstack das lokale Dateisystem als Cache. Darin sind neben dem Cache auch die Sitzungsdaten der Benutzer gespeichert. Um die Performance zu verbessern, können andere Systeme anstelle des lokalen Dateisystems verwendet werden, welche im folgenden vorgestellt werden.

Datenbank

Die am einfachsten einzurichtende Alternative ist das Speichern des Caches in der bereits für die Daten genutzten Datenbank. Dazu reicht es die folgenden beiden Zeilen in der `.env` Datei zu ergänzen und den Container neuzustarten.

```
CACHE_DRIVER=database
SESSION_DRIVER=database
```

Memcached

Um Memcached als Cache nutzen zu können, muss natürlich ein Memcached-Server vorhanden sein, der von Bookstack genutzt werden kann. Sobald ein solcher Server bereit steht, kann die `.env` Konfiguration angepasst werden.

```
CACHE_DRIVER=memcached
SESSION_DRIVER=memcached
MEMCACHED_SERVERS=127.0.0.1:11211:100
```

Neu ist nun die 3. Zeile, in der der Memcached Server zu definieren ist. Die localhost-IP ist dann durch die richtige IP oder den Hostname auszutauschen. Es können auch mehrere Server angegeben werden. Dabei spielt die Zahl hinter dem letzten Doppelpunkt eine entscheidende Rolle, die gibt die Gewichtung des Servers an.

Da Memcached nur Daten im Arbeitsspeicher zwischenspeichert, muss fast nichts konfiguriert werden, um einen Server bereitzustellen. Die folgende minimale Docker Compose Konfiguration reicht völlig aus, um einen Cache Server bereitzustellen.

```
services:
  memcached:
    image: memcached
    container_name: memcached
    restart: unless-stopped
```

Der Dienst *memcached* kann nun in eine beliebige andere Compose Konfiguration kopiert werden, um dort als Cache zu dienen. Dann nur noch bei den anderen Diensten den Servernamen hinterlegen und fertig.

Redis

Im Prinzip erfolgt die Konfiguration genauso wie bei Memcached. Nur das die Zahl nach dem letzten Doppelpunkt nun nicht die Gewichtung, sondern die Datenbank angibt.

```
CACHE_DRIVER=redis
SESSION_DRIVER=redis
REDIS_SERVERS=127.0.0.1:6379:0
```

Session Cookie konfigurieren

Die Cookies werden zum Verfolgen der Sitzungen in Bookstack genutzt. Hier werden 2 Anpassungsmöglichkeiten angeboten.

```
# Only send cookies over a HTTPS connection.
# Ensure you have BookStack served over HTTPS before enabling.
# Defaults to 'false'
SESSION_SECURE_COOKIE=true
# Set the name of the cookie used by BookStack to track sessions.
# Defaults to 'bookstack_session'
SESSION_COOKIE_NAME=custom_cookie_name
```

Session Timeouts

Im Standard wird ein Benutzer nach 2 Stunden Inaktivität automatisch ausgeloggt. Diese Zeitspanne kann ebenfalls in der .env angepasst werden.

```
# Session lifetime in minutes.
# Defaults to 120
SESSION_LIFETIME=240
```