

# Installation via Docker image

Beschreibung:

Hier installation via Docker mit composer file.

Installation.

Die Verzeichnisse erstellen und die Composer Datei.

Im root Verzeichnis ein neues Verzeichnis erstellen.

```
mkdir -p /root/bookstack
```

Ins Verzeichnis wechseln und eine neue datei anlegen mit .env anlegen.

```
cd /root/bookstack  
nano /root/bookstack/.env
```

Inhalt

```
# Domain  
DOMAIN=wiki.example.com  
  
# BookStack Datenbank Variablen  
DB_HOST=db  
DB_DATABASE=bookstack  
DB_USERNAME=bookstack_user  
DB_PASSWORD=sicheres_passwort  
  
# BookStack App Variablen  
APP_URL=https://{DOMAIN}  
#der app key kann in der console mit folgendem Befehl generiert werden  
#echo "base64:$(openssl rand -base64 32)" Ausgabe :  
base64:0HxGQ72frdcVZ+XyZQ1Q0Kr6FkFGwJS/UeYT/NLtZTo=  
APP_KEY=base64:0HxGQ72frdcVZ+XyZQ1Q0Kr6FkFGwJS/UeYT/NLtZTo=
```

Nun eine angepasste php.ini für das Upload limit, hier 10 GB kann nach beliben geändert werden.  
In der Datei alles was 10G ist mit eigenem Wert überschreiben

```
nano /root/bookstack/php.ini
```

## Inhalt

```
upload_max_filesize = 10G
post_max_size = 10G
memory_limit = 512M
max_execution_time = 300
max_input_time = 300
```

## Nun die composer Datei

```
nano /root/bookstack/docker-compose.yml
```

Inhalt, achtung bei phpmyadmin die interne ip anpassen, für den fall das das bookstack auch public erreichbar ist

Denn phpmyadmin solls es ja nicht sein

```
version: '3.8'

services:
  app:
    image: linuxserver/bookstack:latest
    container_name: bookstack_app
    environment:
      - DB_HOST=${DB_HOST}
      - DB_DATABASE=${DB_DATABASE}
      - DB_USERNAME=${DB_USERNAME}
      - DB_PASSWORD=${DB_PASSWORD}
      - APP_URL=${APP_URL}
      - APP_KEY=${APP_KEY}
      - APP_LANG=de
    volumes:
      - ./bookstack_data:/config
      - ./php.ini:/etc/php7/conf.d/99-custom.ini # Mounte das php.ini-File
    depends_on:
      - db
    restart: always

  db:
    image: mariadb:10.6
```

```
container_name: bookstack_db
environment:
  - MYSQL_ROOT_PASSWORD=root_passwort
  - MYSQL_DATABASE=${DB_DATABASE}
  - MYSQL_USER=${DB_USERNAME}
  - MYSQL_PASSWORD=${DB_PASSWORD}
volumes:
  - ./bookstack_db:/var/lib/mysql
restart: always
```

```
caddy:
image: caddy:latest
container_name: bookstack_caddy
ports:
  - "80:80"
  - "443:443"
restart: always
environment:
  - CADDY_DOMAIN=${DOMAIN}
volumes:
  - ./caddy_data:/data
  - ./caddy_config:/config
command: caddy reverse-proxy --from ${DOMAIN} --to app:80
```

```
phpmyadmin:
image: phpmyadmin/phpmyadmin:latest
container_name: phpmyadmin
restart: always
ports:
  - "172.0.2.2:8080:80" # phpMyAdmin wird unter http://172.0.2.2:8080 erreichbar sein
environment:
  PMA_HOST: db # Der Hostname der Datenbank (muss mit dem Service-Namen übereinstimmen)
  PMA_USER: ${DB_USERNAME} # Standard-Benutzername (optional, da phpMyAdmin Login-Maske
hat)
  PMA_PASSWORD: ${DB_PASSWORD} # Standard-Passwort (optional)
  UPLOAD_LIMIT: 512M
depends_on:
  - db
```

Die container starten

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

Firewallregeln für Public nutzung hinzufügen.

ufw einrichten für docker

---

Version #18

Erstellt: 28 November 2023 12:49:37 von Admin

Zuletzt aktualisiert: 11 Februar 2025 13:13:57 von Admin