

# Installation

- [Installation via Ansible](#)
- [Installation via Docker image](#)

# Installation via Ansible

Vorraussetzungen.

Ein Debian Bullseye mit ssh Zugang per Schlüsseldatei.

Denn Bookstack installieren wir remote bequem mit Ansible.

Und auf unserem Remote Client / z.B. Laptop oder PC installieren wir Ansible.

Wie Ihr Ansible installiert und verwendet seht ihr in unserem Buch: Ansible.

Aber auch ohne Ansible Kenntnisse ist das hier installierbar.

Ansible und diese Ansible-Rollen installieren.

Vom Homeverzeichnis aus also:

```
cd ~
```

Nun diese Befehle absetzen

```
# install packages on ansible machine
sudo apt install python3-pip
pip install -U Jinja2
sudo pip install ansible
ansible-galaxy install supertarto.bookstack
ansible-galaxy install supertarto.apache
ansible-galaxy install supertarto.php
ansible-galaxy install supertarto.mariadb
```

Nun legen dazu eine neue Datei bookstack.yaml an.

Dies ist unser Playbook, sprich unser Installationscript.

Im Home Verzeichnis auf eurem Computer / Laptop einfach ein neues Verzeichnis namens Ansible erstellen

```
mkdir ~/ansible
```

Dann eine neue Datei anlegen

```
nano ~/ansible/boockstack.yaml
```

Folgender Inhalt einfügen:

```
# install packages an ansible machine
#ansible-galaxy install supertarto.bookstack
#ansible-galaxy install supertarto.apache
#ansible-galaxy install supertarto.php
#ansible-galaxy install supertarto.mariadb
#pip install -U Jinja2

#standard weblogin data :
#Username : admin@admin.com
#Pass : password

- hosts: all
  roles:
    - role: supertarto.apache
    - role: supertarto.php
    - role: supertarto.mariadb
    - role: supertarto.bookstack

  pre_tasks:
    - name: Update apt cache.
      apt:
        update_cache: true
        cache_valid_time: 600
        when: ansible_os_family == 'Debian'
        changed_when: false

  vars:
    php_packages:
      - php7.4
      - php7.4-mysql
      - php7.4-curl
      - php7.4-pdo
      - php7.4-xml
      - php7.4-mbstring
      - php7.4-gd
```

```
- php-tokenizer
- php7.4-tidy
#hier den tatsächlichen hostname aus der /etc/hostname angeben
bookstack_host: localhost
bookstack_db_name: bookstackdb
bookstack_db_user: bookstackuser
bookstack_db_password: <unser sicheres password für die db>
bookstack_lang: de
apache_create_vhosts: true
apache_mods_enabled:
  - rewrite

#wenn eine eigene vhosts angelegt werden soll
#apache_vhosts_filename: "bookstack.conf"
#wenn nur eintrag dann in die default vhosts
apache_vhosts_filename: "000-default.conf"
apache_vhost_config:
  - listen_ip: "*"
    listen_port: 80
    server_name: 192.168.178.209
    documentroot: "/var/www/Bookstack/public"
    serveradmin: admin@localhost
    custom_param: |
      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
      LogLevel warn
    directory:
      - path: "/var/www/Bookstack/public"
        config: |
          AllowOverride All
          Order deny,allow
          allow from all
          Options +FollowSymLinks
          RewriteEngine On
          RewriteCond %{REQUEST_FILENAME} !-d
          RewriteCond %{REQUEST_FILENAME} !-f
          RewriteRule ^ index.php [L]

mariadb_use_dump_script: false
mariadb_databases:
```

```
- name: "{{ bookstack_db_name }}"
```

```
mariadb_users:
```

```
- name: "{{ bookstack_db_user }}"
```

```
host: "{{ bookstack_host }}"
```

```
password: "{{ bookstack_db_password }}"
```

```
priv: "{{ bookstack_db_name }}.*:SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER,CREATE  
TEMPORARY TABLES,LOCK TABLES"
```

Nun noch folgenden Abschnitt in der Datei anpassen.

```
...  
bookstack_host: localhost  
bookstack_db_name: bookstackdb  
bookstack_db_user: bookstackuser  
bookstack_db_password: <unser sicheres password für die db>  
bookstack_lang: de  
...
```

Das password anpassen und die Sprache auf de umstellen.

Datei speichern fertig.

Nun noch eine inventory Datei anlegen.

In die wird der Server eingetragen auf den dann das BookStack installiert werden soll, sprich das ansible Script / Playbook ausgeführt werden soll.

```
nano ~/ansible/inventory.ini
```

Folgender Inhalt, der parameter ansible\_user= lehgt den Benutzer fest mit dem die Installation auf dem Server ausgeführt werden soll, hier root

```
192.168.178.210 ansible_user=root
```

Denn Ansible ist im Verzeichnis ~/.local/bin/ installiert

Nun kann das Playbook gestartet werden.

```
ansible-playbook bookstack.yml -i inventory.ini
```

Installation abgeschlossen.



# 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