Setup & Configuration

We need to install the service through Portainer and configure any necessary settings.

- Preparation
- Installation
- Updating

Preparation

There are some things we need to do in preparation to install this service.

Volumes

Persistent Data

This is where the service will store its own application data and ensures we can quickly update the service image.

> Ensure your user has permissions to access the folder.

Environment

TZ

This is the current time zone formatted using the tz database.

> *For example:* America/Vancouver

PUID

This is the numeric ID of the user account on Debian. If you are unsure, open a terminal and run:

```
id -u
```

PGID

This is the numeric ID of the user account's group on Debian. If you are unsure, open a terminal and run:

```
id -g
```

DB_USER

This is the username to be used for logging into the database.

> *For example: bookstack_*admin

APP_URL

This is the URL you will be accessing the service through. This can be the local address or the domain name.

ALLOWED_IFRAME_SOURCES

This configures which websites can be embedded with your BookStack.

> *Recommended:* "https://*.draw.io https://*.youtube.com https://*.youtube-nocookie.com https://*.vimeo.com"

Passwords

> Keep these securely stored in a password manager, such as VaultWarden.

DB_PASS

This is the password that will be used for root access to the database.

> It is important to use secure, randomly generated password.

You can use a random alphanumeric string from a password manager, or open the terminal and run the command:

```
tr -dc 'A-Za-z0-9!"#$%&'\''()*+,-./:;<=>?@[\]^_`{|}~' </dev/urandom | head -c 32; echo
```

This pulls a random string from the 'urandom' device, removes unwanted characters and trim it to an appropriate length.

Installation

The service can be installed through the Portainer web interface.

> Learn about creating a new stack.

Docker Compose

Use the following code to install the service:

```
---
services:
  bookstack:
    image: lscr.io/linuxserver/bookstack
    container_name: bookstack
    environment:
      - PUID=${PUID}
      - PGID=${PGID}
      - TZ=${TZ}
      - APP_URL=${APP_URL}
      - DB_HOST=${DB_HOST}
      - DB_PORT=3306
      - DB_USER=${DB_USER}
      - DB_PASS=${DB_PASS}
      - DB_DATABASE=${DB_NAME}
      - ALLOWED_IFRAME_SOURCES=${ALLOWED_IFRAME_SOURCES}
      - ALLOWED_IFRAME_HOSTS=${ALLOWED_IFRAME_HOSTS}
    volumes:
      - /srv/bookstack/data:/config
    ports:
      - 6877:80
    restart: unless-stopped
    depends_on:
      - bookstack-db
```

```
  bookstack-db:
    image: lscr.io/linuxserver/mariadb
    container_name: bookstack-db
    environment:
      - PUID=${PUID}
      - PGID=${PGID}
      - TZ=${TZ}
      - MYSQL_ROOT_PASSWORD=${DB_PASS}
      - MYSQL_DATABASE=${DB_NAME}
      - MYSQL_USER=${DB_USER}
      - MYSQL_PASSWORD=${DB_PASS}
    volumes:
      - /srv/bookstack/db:/config
    restart: unless-stopped
```

Environment

Use the following environment to configure the service using the values prepared earlier:

```
TZ=
PUID=
PGID=
APP_URL=bookstack.example.com
DB_HOST=bookstack-db
DB_USER=bookstack_admin
DB_PASS=
DB_NAME=bookstackapp
ALLOWED_IFRAME_SOURCES="https://*.draw.io https://*.youtube.com https://*.youtube-nocookie.com
https://*.vimeo.com"
```

Updating

> Re-Deploy the Stack

This service has been optimized for running in Docker.

This allows you to re-deploy the stack through Portainer to download the latest updates.