

Install your own BookStack instance

This is the way this instance of BookStack is installed.

This guide starts **from the point** where you already have a **Linux server** and **Docker** is installed. The “steps before that” will follow later — it’s really not hard (especially with AI help).

Overview: What’s being set up here?

In the end you’ll have:

- **BookStack** (wiki/docs system)
- **MariaDB** as the database
- **Caddy** as a reverse proxy with **automatic HTTPS** (Let’s Encrypt) ☐

Background: What is Caddy? ?

“**Caddy** is a modern web server (similar to *Nginx/Apache*), written in *Go*, that embraces “*security-by-default*” and, above all, makes **HTTPS** extremely convenient.

Core idea: “A web server that just works” ☐

Key features ☐

1. **Automatic HTTPS (TLS)**
 - Certificates are obtained and renewed automatically (typically via *Let’s Encrypt*).
2. **Simple configuration**
 - Via an easy-to-read **Caddyfile**.
3. **Reverse proxy & load balancing**

- Ideal for forwarding requests to services (e.g., Docker containers).

4. **Good defaults**

- Many sensible security standards are enabled by default.

5. **Modular extensibility**

- If you have special requirements, Caddy can be extended.

Typical use cases ☐☐

- Hosting websites (static/dynamic)
- Reverse proxy in front of an app (e.g., `/api` → backend)
- TLS termination
- Local dev setups with HTTPS

Step 1: Prepare directory & files ?

Create the directory:

- `/opt/bookstack`

Create **two files** inside it:

- `docker-compose.yml`
- `Caddyfile`

🗨️ **Important:** You'll need to adjust a few values in a moment — you can clearly see *where* in the YAML.

Step 2: Configure Docker Compose (`docker-compose.yml`) ?

Paste the following content (and adjust it where necessary):

```
services:
  mariadb:
    image: lscr.io/linuxserver/mariadb:latest
    container_name: bookstack-mariadb
    environment:
```

- PUID=1000
- PGID=1000
- TZ=Europe/Berlin
- MYSQL_ROOT_PASSWORD=PW_OF_MYSQL_ROOT
- MYSQL_DATABASE=bookstack
- MYSQL_USER=bookstack
- MYSQL_PASSWORD=PW_OF_MYSQL_DB

volumes:

- ./mariadb:/config

restart: unless-stopped

bookstack:

image: lscr.io/linuxserver/bookstack:latest

container_name: bookstack

depends_on:

- mariadb

environment:

- PUID=1000
- PGID=1000
- TZ=Europe/Berlin

- APP_URL=https://wiki.fabula.vision

Where will BookStack be accessible? (Your URL)

- APP_KEY=base64:...

To generate it, run: docker run -it --rm --entrypoint /bin/bash

lscr.io/linuxserver/bookstack:latest appkey

- APP_THEME=custom

'custom' makes it possible to use hacks; more on that here:

<https://www.bookstackapp.com/hacks/applying/>

- DB_HOST=mariadb
- DB_PORT=3306
- DB_DATABASE=bookstack
- DB_USERNAME=bookstack
- DB_PASSWORD=PW_OF_MYSQL_DB

(As above!)

- APP_DEFAULT_DARK_MODE=true

(Personal preference)

```
volumes:
  - ./bookstack:/config
restart: unless-stopped

caddy:
  image: caddy:latest
  container_name: caddy
  depends_on:
    - bookstack
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - ./Caddyfile:/etc/caddy/Caddyfile:ro
    - ./caddy/data:/data
    - ./caddy/config:/config
  restart: unless-stopped
```

Step 3: Configure Caddy (Caddyfile) ??

Paste the following and replace the domain:

```
wiki.fabula.vision {
  # Your URL, of course...
  encode zstd gzip

  # Reverse proxy to BookStack (container is named "bookstack", internal port 80)
  reverse_proxy bookstack:80

  # sensible headers (optional)
  header {
    # HSTS (only set this if you're sure HTTPS should remain enabled permanently)
    Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
    X-Content-Type-Options "nosniff"
    X-Frame-Options "SAMEORIGIN"
    Referrer-Policy "strict-origin-when-cross-origin"
```

```
}  
}
```

Step 4: Start the containers ??

1. Change into the directory:

- `cd /opt/bookstack/`

2. Start the stack:

- `docker compose up -d`

Updating later works like this:

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

3. Done ☐

Now quickly open the URL and change the default credentials!

- Default login: `admin@admin.com`
- Default password: `password`

Step 5: Cron job for BookStack ??

Create a cron job:

1. Open crontab:

- `crontab -e`

2. Add the following line (cron every minute):

- `* * * * * docker exec -i bookstack php /app/www/artisan schedule:run >/dev/null 2>&1`

Limitations & modules/hacks ?

There are three limitations that have stood out so far:

1. **Email sending**

- (e.g., “Forgot password”) doesn’t work for me, and my experiments haven’t produced a solution yet.
- I don’t strictly need it — but for example for comments (notifications) you can’t avoid it.
- More info:

<https://www.bookstackapp.com/docs/admin/email-webhooks/#email-configuration>

2. Extending via “hacks”

- BookStack can be extended via so-called “**hacks.**”

EDIT: I created [my own hacks](#) that are better!

- Two recommendations:
 1. **MathJax/TeX** (math formulas):
 - <https://www.bookstackapp.com/hacks/mathjax-tex/>
 2. **Mermaid Viewer** (diagrams):
 - <https://www.bookstackapp.com/hacks/mermaid-viewer/>

Important note about export ☐☐

- Mermaid (and formulas as well) **does not export cleanly** for me (e.g., PDF): code blocks appear instead of rendered content.

How do you install hacks as modules?

- We already did the first step: In the YAML it says

```
APP_THEME=custom
```

This loads modules from the following path:

- `/opt/bookstack/bookstack/www/themes/custom/modules`
- Then via SSH:
 1. Switch into the container:
 - `docker exec -it bookstack /bin/bash`
 2. Go to the web directory:
 - `cd /app/www/`
 3. Run the command that’s listed in the respective hack (example *Mermaid*):
 - On the hack page in the “Install as Module” section
<https://www.bookstackapp.com/hacks/mermaid-viewer/>

3. Shift+Enter — line break (CommonMark)

- Due to the decision to use the CommonMark standard, a simple line break is not rendered by default.
- Instead you need “two spaces at end of line + Enter” — that drove me crazy because many Markdown editors are used to *Shift+Enter*.
- My fix consists of two parts:

1. In the BookStack settings under

<https://example.com/settings/customization>

under “Custom HTML Head Content”, enter the following code:

```
<script>
  window.addEventListener('editor-markdown::setup', event => {
    event.detail.markdownIt.set({breaks: true});
  });
</script>
```

2. Create a `functions.php` under:

- `/opt/bookstack/bookstack/www/themes/custom/`

With the following content:

```
<?php

use BookStack\Theming\ThemeEvents;
use BookStack\Facades\Theme;

Theme::listen(ThemeEvents::COMMONMARK_ENVIRONMENT_CONFIGURE, function
($environment) {
    $environment->mergeConfig([
        'renderer' => [
            'soft_break' => "<br>",
        ]
    ]);

    return $environment;
});
```

Additional QoL fixes (under “Customization”) ??

Configure here:

- <https://yoururl.com/settings/customization>

Just paste each one under “Custom HTML Head Content”:

1. **Blockquotes: no thick bottom border / clean spacing**

- Cause: CSS (among other things, `<p>` with `margin-bottom`)
- Fix (same spacing top/bottom + no scrollbars):

```
<style>

/* 1) Last & first direct child element in the blockquote: margin */
.content-wrap blockquote > :last-child {
    margin-bottom: .3em;
}

.content-wrap blockquote > :first-child {
    margin-top: .3em;
}
```

```
/* 2) Blockquote: no scrollbars */
.content-wrap blockquote {
  overflow: visible;    /* default "no scrolling" */
  overflow-x: visible;
  overflow-y: visible;
}
</style>
```

2. Code blocks: soft wraps (no horizontal scrolling needed)

```
<script>
window.addEventListener('library-cm6:pre-init', event => {
  const detail = event.detail;
  const config = detail.editorViewConfig;
  const EditorView = detail.libEditorView;

  if (detail.usage === 'content-code-block') {
    config.extensions.push(EditorView.lineWrapping);
  }
});
</script>
```

3. Dark mode: lighter text (headings 70%, text 90%)

```
<style>
html.dark-mode .page-content h1, html.dark-mode .page-content h2, html.dark-mode
.page-content h3, html.dark-mode .page-content h4, html.dark-mode .page-content h5,
html.dark-mode .page-content h6 {
  color: hsl(0 0% 70%);
}
html.dark-mode body {
  color: hsl(0 0% 90%);
}
</style>
```

4. Text size: main content 110%, headings 90%

```
<style>
.page-content {
  zoom:1.1;
}
</style>
```

```
.page-content h1, .page-content h2, .page-content h3, .page-content h4, .page-  
content h5, .page-content h6 {  
  zoom:.9;  
}  
</style>
```

Revision #50

Created 2026-04-02 12:08:15 UTC by art10m

Updated 2026-04-19 06:58:24 UTC by art10m