Install and Setup Bookstack, an Open Source Wiki

Bookstack is an incredibly powerful, open source, self hosted wiki solution. It's set apart from other wiki solutions by the way it allows you to organize content. Think of it as your digital library of documentation.

You start with Shelves (a larger overall categorization) of various documentation. Then on those shelves you have your books. Books are the next level down of categorizing things. Within books you have chapters, again a level to more closely gather like information. And within those chapters you have pages.   Pages are where the specific document lives.

Here's is an example, but certainly not the only way to use this organizational hierarchy.

Shelf 1: Networking -> Book 1: Networking Tools -> Chapter 1: IPv4 Tools - Page: Octets

We can repeat the entire line above except the page, and add a new page called "Network Mask".

Shelf 1: Networking -> Book 1: Networking Tools -> Chapter 1: IPv4 Tools - Page: Network Mask

And so on as we build out our wiki.

Here's the coolest part. You don't have to keep the Networking Tools book in just 1 shelf. You can have it on as many shelves as it matches. It's like having multiple copies of the book in different locations in your library.

Shelf 1: Networking

Shelf 2: VPNs

Shelf 3: Dynamic DNS

These could all be a home for a book called "Networking Tools". This guarantees a much higher percentage chance that you'll find the appropriate material when you go digging through topics that may be related, but not fit into quite the same top level category. OF course, you could take the topics of VPNs and Dynamic DNS and just as easily make them Books in the Networking shelf. There is no wrong way to organize things.

Installing and Running Bookstack

What You'll Need

- A server, either bare metal, VPS, or a VM or Container such as LXC, LXD, or Incus
- Docker and Docker Compose installed on that server
- (optional) A public IP address
- (optional) A Domain or Subdomain name for your installation
- (optional) a Reverse Proxy if running inside a LAN
- About 15 minutes of your time
  - * Optional items above are only if you want to expose your Wiki to the public internet.

Installation via a Simple Script

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

`wget -O install-docker.sh https://gitlab.com/bmcgonag/docker_installs/-/raw/main/install_docker_nproxyman.sh`

To download the script to your desired host.

Change the permissions to make the script executable:

`chmod +x ./install_docker.sh`

and then run the script with the command:

`./install_docker.sh`

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of sudo in front of them.

Install and Setup Bookstack in Docker using Docker Compose

First, let's create a folder structure which will support a solid backup strategy for our containers. We'll make a directory structure with aa top-level directory called docker. Inside that directory we'll then create a directory for each service or application we want to run. In this case that second level directory will be called "bookstack". We can make both directories with a single command, and this command gives the added benefit, that if either directory already exists, it will just use that directory.

> **mkdir -p docker/bookstack**
>
> From the man pages of the mkdir command:
>
> **-p, --parents**
>   no error if existing, make parent directories as needed

Next, we'll move into that directory with the command:

`cd docker/bookstack`

Inside the 'bookstack' directory we need to create one more directory called 'data'. This is where the container will house and persist the data and database for our docker container. This is a very important folder, and should be backed up regularly in order to preserve your bookstack data should anything catastrophic happen to the installation.

`mkdir data`

Now, let's create our file to tell Docker Compose how to bring up and configure our Bookstack Installation. I'm actually going to add two versions of the file below.

The first version will be for those wanting to use the built in Bookstack authentication system (login with username and password). The second will be for those who already have an Identity Provider setup like Authentik, Authelia, Keycloak, etc. and want to use OpenIDConnect (OIDC) to login to their Bookstack with a single sign on (SSO) method.

Create a new file in your 'docker/bookstack' folder, and call it 'compose.yaml' using this command:

`nano compose.yaml`

Now copy the yaml code from the version you need below, and paste it into your text editor. For the terminal you can paste by right-clicking with the mouse and selecting 'Paste', or you can use the CTRL + Shift + V hotkey combination on your keyboard (Linux / Windows), or CMD + V for MacOS.

> **compose.yaml to use the built in Login with Bookstack**

```yaml
services:
  bookstack:
    image: lscr.io/linuxserver/bookstack
    container_name: bookstack
    environment:
      - TZ=America/Chicago
      - PUID=1000 # check this value with the command 'id'
      - PGID=1000 # check this value with the command 'id'
      - APP_URL="https://books.yourgreatdomain.com"
      - DB_HOST=bookstack_db
      - DB_USER=bookstack
      - DB_PASS=a-different-long-strong-password-with-a-lot-of-numbers-and-letters
      - DB_DATABASE=bookstackapp
      - MAIL_DRIVER=smtp
      - MAIL_HOST=smtp.youremaildomain.org
      - MAIL_PORT=587
      - MAIL_USERNAME=you@youremaildomain.org
      - MAIL_PASSWORD=your-smtp-email-password
      - MAIL_ENCRYPTION=tls
      - MAIL_FROM=you@youremaildomain.org
    volumes:
      - ./data:/config
    ports:
      - 80:80
    restart: unless-stopped
    depends_on:
      - bookstack_db
  bookstack_db:
    image: lscr.io/linuxserver/mariadb
    container_name: bookstack_db
    environment:
      - PUID=1000
      - PGID=1000
      - MYSQL_ROOT_PASSWORD=a-long-strong-password-with-numb3rs-and-le77er5
      - TZ=America/Chicago
      - MYSQL_DATABASE=bookstackapp
      - MYSQL_USER=bookstack
      - MYSQL_PASSWORD=a-different-long-strong-password-with-a-lot-of-numbers-and-letters  # must match
the one in the above section
```

```yaml
    volumes:
      - ./data:/config
    restart: unless-stopped
```

Next we have the version to use if you have your own IdP to authenticate through. If you don't know what this means, you probably should use the first compose file above.

compose.yaml for those wanting OIDC for SSO

```yaml
services:
  bookstack:
    image: lscr.io/linuxserver/bookstack
    container_name: bookstack
    environment:
      - TZ=America/Chicago
      - PUID=1000 # check this value with the 'id' command
      - PGID=1000 # check this value with the 'id' command
      - APP_URL="https://docs.yourgreatdomain.org"
      - DB_HOST=bookstack_db
      - DB_USER=bookstack
      - DB_PASS=a-long-strong-password-with-lots-of-numbers-and-letters
      - DB_DATABASE=bookstackapp
      - MAIL_DRIVER=smtp
      - MAIL_HOST=smtp.yourgreatdomain.org
      - MAIL_PORT=587
      - MAIL_USERNAME=you@yourgreatdomain.com
      - MAIL_PASSWORD=your-long-strong-email-password-for-the-smtp-server
      - MAIL_ENCRYPTION=tls
      - MAIL_FROM=you@yourgreatdomain.org
      # Set OIDC to be the authentication method
      - AUTH_METHOD=oidc
      # Control if BookStack automatically initiates login via your OIDC system
      # if it's the only authentication method. Prevents the need for the
      # user to click the "Login with x" button on the login page.
      # Setting this to true enables auto-initiation.
      - AUTH_AUTO_INITIATE=false # feel free to change it to 'true'
```

```yaml
      # Set the display name to be shown on the login button.
      # (Login with <name>)
      - OIDC_NAME=Authentik # whatever you want the login button to say
      # Name of the claims(s) to use for the user's display name.
      # Can have multiple attributes listed, separated with a '|' in which
      # case those values will be joined with a space.
      # Example: OIDC_DISPLAY_NAME_CLAIMS=given_name|family_name
      - OIDC_DISPLAY_NAME_CLAIMS=name
      # OAuth Client ID to access the identity provider
      - OIDC_CLIENT_ID=the-client-id-from-your-auth-server
      # OAuth Client Secret to access the identity provider
      - OIDC_CLIENT_SECRET=the-auth-secret-from-your-auth-server-for-the-oidc-provider-you-created-ahead-of-time
      # Issuer URL
      # Must start with 'https://'
      - OIDC_ISSUER=https://auth.youroidcprovider.org/application/o/bookstack-oauth/
      # Enable auto-discovery of endpoints and token keys.
      # As per the standard, expects the service to serve a
      # `<issuer>/.well-known/openid-configuration` endpoint.
      - OIDC_ISSUER_DISCOVER=true
    volumes:
      - ./data:/config
    ports:
      - 80:80
    restart: unless-stopped
    depends_on:
      - bookstack_db
  bookstack_db:
    image: lscr.io/linuxserver/mariadb
    container_name: bookstack_db
    environment:
      - PUID=1000
      - PGID=1000
      - MYSQL_ROOT_PASSWORD=a-different-long-strong-password-with-lots-of-numbers-and-letters
      - TZ=America/Chicago
      - MYSQL_DATABASE=bookstackapp
      - MYSQL_USER=bookstack
      - MYSQL_PASSWORD=a-long-strong-password-with-lots-of-numbers-and-letters # must match the one in the section above
```

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

Regardless of the yaml you choose to setup, you need to make sure to update several of the environment variables, and potentially the ports that you'll access the service on.

You can modify the port on the left side of the port mapping in the yaml. The left side of the mapping is the port where the service is accessed on the host machine. 80 is the standard web server port, so I suggest if you are running this inside a LAN, you change the port to something non-standard, and then use a reverse proxy to access the service as needed from outside the LAN. A port like 8089 would be good for instance. As long as no other service is using port 8089 on the host machine, this should not cause an issue. If there is already a service using 8089, then you can use any valid port number you like which is not already in use.

Additionally, you'll need to change the passwords for the database connections. Use a very long, strong, password, and make sure the password entries for the variables MYSQL_PASSWORD and DB_PASS match exactly.

You'll need to update the email / smtp settings if you want your install to be able to send emails to you as well. If not, feel free to leave the placeholder info there.

Once you've made the necessary changes, save the file with CTRL + O, then press Enter to confirm. This is a good time to briefly go back through the file, check your spacing (as yaml is space dependent), and also make sure you have entered everything correctly.

Exit the nano text editor with CTRL + X.

Bring Up our Bookstack Install