

# BookStack

## Project Documentation Website

- Bookstack to Build
- Digital Ocean to Host
- Cloudflare to Protect
  
- [What is it?](#)
- [Deployment](#)
- [Secure Access](#)

# What is it?

BookStack is a self-hosted platform that allows you to store information in an organized and accessible way. There are several reasons I chose this, but mainly, it's ease of use and deployment. I was looking for something similar to Confluence from Atlassian, but was free and could also be made publicly available. This Wiki platform was perfect for that, and I've now started to use for essentially any and all of my note taking. I view it as my personal version of OneNote, but better, as I can choose to make "shelves" or "books" publicly available as I finish working on them and even let users collaborate and contribute.

# Deployment

Bookstack is a web application that is built on a LAMP stack. My deployment uses Linux, Apache, MySQL, and PHP to run services. Currently, I have this web application deployed on an Ubuntu Server.

## Digital Ocean

I utilized Digital Ocean, which is Cloud Infrastructure similar to AWS or Azure. I went with Digital Ocean purely because of the cheaper costs of resources.

Currently, I have a Ubuntu 22.04 LTS Server running on a Digital Ocean Droplet, comparable to an AWS EC2 instance. I used Squarespace to purchase my domain [austinhomelabs.com](https://austinhomelabs.com) and also to manage records, pointing it to my droplet public IPv4/IPv6 addresses.

## CertBot

To enable HTTPS and provide my domain a certificate, I used the open source software CertBot. This automatically generates a certificate using Let's Encrypt and renews the certificate every 60 days, which allows for secure and continuous access to my web application.

## BookStack Deployment

After obtaining the virtual private server, installing BookStack is relatively simple. Their documentation has detailed guides and also scripts to automatically install and deploy your website. Remaining configurations can be done either via SSH into the VPS or through the Web GUI itself as administrator.

# Secure Access

For users to securely access my website, I've integrated Google SSO services, which will automatically register you as a user through your Google account and give you "viewer" permissions that I've set.

I did this through adding a couple of lines of code to the .env file using Application IDs and Client IDs/Secrets I generated through Google Developers Console. More information on how this can be done can be found [here](#).