HTTP Functions Ltd

BookStack Installation Guide
For RHEL 9 Systems

Written by: Dan Brown
Dated Created: 11th July 2024

Documentation Purpose

The purpose of this guide is to demonstrate the steps required to create a basic functional BookStack instance on a RHEL 9 based system. It specifically **does not** cover the following:

- System & network security hardening
- Further BookStack configuration & integration
- Onward maintenance
- Advanced MySQL use (clustering etc…)

These subjects are either already documented on the BookStack official website, or outside the scope of supporting for BookStack.

Considerations

This guide has been written from the experience of setting up a BookStack instance on a RHEL (Red Hat Enterprise Linux) 9.4 system with the following properties:

- RHEL 9.4 instance, using RHEL provided ISO as of 9th July 2024.
- Architecture: x86_64 (Virtual system ran via Proxmox).
- Installation Configuration: Server based environment used (No GUI), No security profiles selected, root account disabled, custom admin user created.
- System specs: 3 vCPUs, 2GB RAM, 32GB Disk space.

It's possible that the steps provided in this guidance may differ for environments with different properties, and some adapting of the guidance may be needed.

This guidance has been written by a someone with experience of Linux and RHEL based environments, although without any formal qualification, certification or deep professional insight into such systems.

This guidance has been written with the assumption you have direct network access to the machine that BookStack is being hosted upon, and that the host machine has network access for installing additional packages & software from the internet.

Many commands in this guidance will be prefixed with `sudo` to indicate where root level privileges are required. If you're already using the system command line as a root-level user, using `sudo` may not be required.

Installation Steps

The first thing to do is to gain command line access to the host system, either via direct interaction or via something like SSH. Almost all steps below will be performed via the command line.

***Note**: Any lines starting with a `#` character within command example sections below are commentary only and do not need to be ran as a command. Separate command lines shown will be separated by a blank space or a #-beginning comment line, otherwise they should be treated as a single line even if represented across multiple lines (wrapped due to document width).*

Installing System Requirements

First we need to install packages that are required to install or run BookStack. We'll start of by installing git (used to manage the BookStack instance code-base) and the Apache web server:

```
sudo dnf install git httpd
```

Next we'll install PHP. By default RHEL 9 provides PHP 8.0 but as of writing BookStack requires PHP 8.1 or higher. Instead we can use a Red Hat module to install & use PHP 8.2:

```
# Install base PHP 8.2 via module
sudo dnf module install php:8.2

# Install required PHP extensions
sudo dnf install php-mysqlnd php-ldap php-mbstring php-xml php-pdo php-gd
php-bcmath php-fpm
```

With PHP installed, we'll now install composer which is used to manage the PHP-based dependencies for BookStack:

```
# Download and run the composer installer
wget
https://raw.githubusercontent.com/composer/getcomposer.org/6a30e5ecca05f91a3
08c5fda2394d4a1770a2c55/web/installer -O - -q | php -- --quiet

# Move composer to make it globally command line accessible
sudo mv composer.phar /usr/bin/composer
```

You can test that composer has been installed correctly via running `composer -V`. Here is the example output as of the time of writing:

```
Composer version 2.7.7 2024-06-10 22:11:12
PHP version 8.2.13 (/usr/bin/php)
Run the "diagnose" command to get more detailed diagnostics output.
```

Lastly we need to install MySQL server to use as a database for BookStack. If you're intending to use an existing remote MySQL instance/cluster for BookStack, you won't need to perform this installation step.

```
sudo dnf install mysql-server
```

MySQL Setup

We'll need to create a user and database on the MySQL server for BookStack to use. If using an existing MySQL instance you may not need to perform all these steps, but you'll still likely want a BookStack specific user and database created on your instance. How and where that's done may depend on how your MySQL server is configured & managed.

```
# Start the locally installed MySQL server
sudo systemctl start mysqld.service

# Enable MySQL server to start at system boot
sudo systemctl enable mysqld.service
```

With that done, we'll access the MySQL server to create a "bookstack" named user and database, with as password of "bs49175". You'll likely want to generate a unique password instead.

```
# Access the locally installed MySQL server as root user
mysql -u root

# You should then gain MySQL command line access,
# reflected by a "mysql>" prompt.

# Create a new "bookstack" database
create database bookstack;

# Create a new "bookstack" user with a password.
# Replace "bs49175" in the command with your own password
# that you want to give this MySQL user.
# Note: For MySQL instances on a remote system you'd have to also change
#   "localhost" to instead reflect the connecting host/IP of
#   the BookStack host system.
create user bookstack@localhost identified by "bs49175";

# Grant the new user all privileges on the created database.
# Again, you may need to replace "localhost" in this command
# if you did so in the command above.
grant all privileges on bookstack.* to bookstack@localhost;

# Refresh database privileges
flush privileges;

# Quit the MySQL server
exit;
```

With that done you should have a running and configured MySQL server, ready to use with BookStack. If this is a new MySQL server system, Red Hat advise running the `mysql_secure_installation` command to go through some extra security hardening steps.

Initial BookStack Installation Setup

Now we'll create our BookStack instance on the host system. Our BookStack installation will be created at the path `/var/www/bookstack`.

```
# Navigate to the "/var/www" directory
cd /var/www

# Use git to clone BookStack release files into a new "bookstack" directory
sudo git clone https://github.com/BookStackApp/BookStack.git --branch
release --single-branch bookstack

# Navigate into our new "bookstack" install folder
cd /var/www/bookstack

# Tell git to ignore file permission changes
sudo git config core.fileMode false

# Install BookStack PHP dependancies via composer
sudo composer install --no-dev

# Create a new configuration file from the default template
sudo cp .env.example .env

# Generate a unique key in the application configuration
sudo php artisan key:generate --force
```

Now we'll need to perform some configuration in the application's `.env` file that we just created. Open this file in your preferred text editor:

```
# Example:
sudo nano .env
```

Then find the following options and update the values as follows:

- **`APP_URL`**: Update the value to match the base URL which you'd expect to access BookStack at after installation (configured domain, hostname or IP address). Ensure that this starts with `http://` (or `https://` if you're setting up HTTPS right away).
- **`DB_HOST`**: If using a locally install MySQL server as above then you can leave this as `localhost` otherwise this should be the hostname/IP of your MySQL server instance.
- **`DB_DATABASE`**: Set to `bookstack` (or the name of the database you created if different).
- **`DB_USER`**: Set to `bookstack` (or the name of the database user you created if different).
- **`DB_PASSWORD`**: Set to the password you set when creating your bookstack MySQL user.

As an example, on the test system I'm using to create this guidance these parts of my `.env` file are as follows:

```
APP_URL=http://192.168.4.40

# Database details
DB_HOST=localhost
DB_DATABASE=bookstack
DB_USERNAME=bookstack
DB_PASSWORD=bs49175
```

If any of those values contain spaces or hashes then ensure you wrap the value in quotes. Example:

```
DB_PASSWORD="abc#123"
```

With those details set, save the file and exit your text editor. You can then run the BookStack database migrations which will create the required database structure and content:

```
# This should still be ran from your /var/www/bookstack folder
sudo php artisan migrate
```

After confirming to continue, you should see the date-stamped migration names output as they are ran. Leave this to complete. If you get errors running this command, it's most likely due to BookStack not being able to connect or communicate with the database. Double check the database details in your `.env` file. If the MySQL server is remote, ensure that the networking is configured to allow BookStack to connect, and that the host for the bookstack user in the MySQL server is correct.

WebServer Configuration

With the application installed, we now need to configure the local web server to run and serve the installed BookStack application. We installed Apache earlier in the "Installing System Requirements" section for this purpose.

To start, create and open a new `/etc/httpd/conf.d/bookstack.conf` in your preferred text editor. Example:

```
sudo nano /etc/httpd/conf.d/bookstack.conf
```

Within this, add the following configuration content:

```
<VirtualHost *:80>
    ServerName docs.example.com
    DocumentRoot /var/www/bookstack/public/

    <Directory /var/www/bookstack/public/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
        <IfModule mod_rewrite.c>
            <IfModule mod_negotiation.c>
                Options -MultiViews -Indexes
            </IfModule>

            RewriteEngine On

            # Handle Authorization Header
            RewriteCond %{HTTP:Authorization} .
            RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

            # Redirect Trailing Slashes If Not A Folder...
            RewriteCond %{REQUEST_FILENAME} !-d
            RewriteCond %{REQUEST_URI} (.+)/$
            RewriteRule ^ %1 [L,R=301]

            # Handle Front Controller...
            RewriteCond %{REQUEST_FILENAME} !-d
            RewriteCond %{REQUEST_FILENAME} !-f
            RewriteRule ^ index.php [L]
        </IfModule>
    </Directory>
</VirtualHost>
```

Change the `docs.example.com` text on the second line to match the hostname/domain/IP that you intend to access BookStack on. Then save the file and quit your text editor.

Now we'll get the webserver running:

```
# Start the webserver and PHP services
sudo systemctl start httpd php-fpm

# Enable the services to start on boot
sudo systemctl enable httpd php-fpm
```

Firewall & File Permissions

With the web-server now hosting the application, we'll need to allow http traffic so it can be accessed over the web. We can allow this traffic through the firewall like so:

```
# Allow http traffic through the firewall
sudo firewall-cmd --add-service=http

# Make the firewall changes persistent
sudo firewall-cmd --runtime-to-permanent
```

Then we need to ensure that our BookStack installation files & folders are configured to allow web-server access where needed:

```
# Navigate into our BookStack installation folder
cd /var/www/bookstack

# Set the "apache" user as the owning group for the required folders
sudo chown -R :apache storage bootstrap/cache public/uploads

# Set group privileges for the required folders
sudo chmod -R g+rwx storage bootstrap/cache public/uploads
```

And we'll also need to tell SELinux that these folders are allowed read/write permission for web services:

```
# Tell SELinux about read/write web content
sudo semanage fcontext -a -t httpd_sys_rw_content_t
"/var/www/bookstack/storage(/.*)?"

sudo semanage fcontext -a -t httpd_sys_rw_content_t
"/var/www/bookstack/bootstrap/cache(/.*)?"

sudo semanage fcontext -a -t httpd_sys_rw_content_t
"/var/www/bookstack/public/uploads(/.*)?"

# Apply SELinux changes to files
sudo restorecon -R -v /var/www/bookstack
```

Initial Access

If all steps have been successfully followed, you should now be able to access your BookStack instance in the browser via the domain/IP you used, over http.

Upon access, you should be redirected to the login page. The default admin login email is “admin@admin.com” with a password of "`password`". After first login, you should change these details using the “My Account” area found via the profile menu in the header bar.

You should now have a working BookStack instance!

Useful Paths

Here are some useful paths to keep on reference when the above installation steps have been followed:

- **BookStack installation folder:** /var/www/bookstack
- **BookStack app log:** /var/www/bookstack/storage/logs/laravel.log
- **php.ini file:** /etc/php.ini
- **Apache BookStack config file:** /etc/httpd/conf.d/bookstack.conf
- **Apache (webserver) error log:** /var/log/httpd/error_log
- **Apache (webserver) access log:** /var/log/httpd/access_log

Post Installation Considerations

While not strictly required, some features in BookStack use email so it’s usually a good idea to configure email settings using the options at the bottom of your `/var/www/bookstack/.env` file. These settings are detailed within the BookStack documentation here, and can be tested via the “Settings > Maintenance > Send a Test Email” area of the interface BookStack.

After install you may want to review the BookStack security guidance to understand security options & considerations.

For some guidance within the BookStack documentation you may need to alter a `php.ini` file. When following the installation instructions above, your `php.ini` file will be located at `/etc/php.ini`. You’ll need to restart the php-fpm service (`sudo systemctl restart php-fpm.service`) after making any changes to this before they’ll become active when accessing the app through the browser.

When following the installation instructions above, BookStack will have been installed in a manner where the installation files & folders belong to the root user. This will mean most maintenance actions (like updating) in future will require root privileges. You may instead want to instead change the ownership of these files to belong to another user.

There is guidance for that in the BookStack documentation here.

If SELinux is active on the BookStack host, then it may play a role in blocking certain abilities/actions (For example, PDF exports when using a custom PDF generation engine). SELinux should be kept in mind, and the logs checked when encountering scenarios when certain things do not work in BookStack. Such logs can usually be observed via the "/var/log/audit/audit.log" log file.

This guidance will set up a HTTP only instance, and does not go into detail in using HTTPS since this can be achieved in many ways depending on environment. Where an instance is on the public web, HTTPS is typically most easily achieved by using something like certbot to add HTTPS and create a certificate for the locally installed Apache web server. However HTTPS is achieved, you will need to update the `APP_URL` value in the `/var/www/bookstack/.env` file to start with `https://` instead of `http://`. If you have existing content, you may also need to run the "Update System URL" command documented here to update URL references in the database.

The instructions in this guide makes use of PHP 8.2 provided as a module by Red Hat. If remaining on RHEL 9 for a while, and intending on keeping up with BookStack updates, it's possible that in the future BookStack will require a minimum version of PHP that's greater than what RHEL provides. In this event, you may need to use an alternate source for PHP (Like Remi's RPM repository for example).