

BookStack

- [Backup and Restore BookStack Instance](#)
- [Enabling HTTPS for the BookStack web interface](#)
- [Configuring File Upload](#)

Backup and Restore BookStack Instance

Backup and Restore

<https://www.bookstackapp.com/docs/admin/backup-restore/>

While BookStack does not currently have a built-in way to backup and restore content, it can usually be done via the command line with relative ease.

Please note the below commands are based on using Ubuntu. If you are using a different operating system you may have to alter these commands to suit.

Backup

There are two types of content you need to backup: Files and database records.

Database

The easiest way to backup the database is via `mysqldump`:

```
1
2
3
4
5
6
7
```

```
# Syntax
## Only specify the `-p` option if the user provided
has a password
mysqldump -u {mysql_user} -p {database_name}
> {output_file_name}

# Example
mysqldump -u benny bookstack >
bookstack.backup.sql
```

If you are using MySQL on Ubuntu, and are using the `root` MySQL user, you will likely have to run the command above with `sudo`:

```
1
```

```
sudo mysqldump -u root bookstack >  
bookstack.backup.sql
```

The resulting file (`bookstack.backup.sql` in the examples above) will contain all the data from the database you specified. Copy this file to somewhere safe, ideally on a different device.

Files

Below is a list of files and folders containing data you should back up. The paths are shown relative to the root BookStack folder.

- `.env` - File, contains important configuration information.
- `public/uploads` - Folder, contains any uploaded images.
- `storage/uploads` - Folder, contains uploaded page attachments.
- `themes/` - Folder, contains any configured [visual/logical themes](#).

Alternatively you could backup up your whole BookStack folder but only the above contain important instance-specific data by default.

The following command will create a compressed archive of the above folders and files:

```
1
```

```
tar -czvf bookstack-files-backup.tar.gz .env  
public/uploads storage/uploads
```

The resulting file (`bookstack-files-backup.tar.gz`) will contain all your file data. Copy this to a safe place, ideally on a different device.

Restore

If you are restoring from scratch follow the [installation](#) instructions first to get a new BookStack instance set-up but **do not run the `php artisan migrate` installation step when installing BookStack**. You may need to comment this command out if using an installer script.

If you are using a docker-container-based set-up, restore the database before running the BookStack container. An example of the process using a linuxserver.io-based docker-compose setup can be seen [in our video here](#).

Database

To restore the database you simply need to execute the sql in the output file from the `mysqldump` you performed above. To do this copy your database SQL backup file onto the BookStack or database host machine and run the following:

```
1
2
3
4
5
6
7
8
9
10
```

```
# Syntax
mysql -u {mysql_user} -p {database_name} <
{backup_file_name}
## Only specify the -p if the user provided has a
password

# Example
mysql -u benny -p bookstack <
bookstack.backup.sql

# If using the root user on Ubuntu you may
# have to run the above with root permissions via
sudo:
sudo mysql -u root bookstack <
bookstack.backup.sql
```

If you are restoring to a new version of BookStack you will have to run `php artisan migrate` after restore to perform any required updates to the database.

Files

To restore the files you simply need to copy them from the backup archive back to their original locations. If you created a compressed `bookstack-files-backup.tar.gz` archive as per the backup instructions above you can simply copy that file to your BookStack folder then run the following command:

```
1
```

```
tar -xvzf bookstack-files-backup.tar.gz
```

If you get errors during the above command it may be due to permissions. Change permissions so you can write to the restore locations.

After a backup of the files you should reset the permissions to ensure any write-required locations are writable by the server. The locations required for this can be found in the [installation](#)

[instructions.](#)

Configuration (.env File)

During a restore, you may end up merging various configuration options between your old and new instance `.env` files, to get things working for the new environment. For example, it's common to use the old `.env` settings for most things but use database settings from the `.env` file of a newly created instance.

One thing to be aware of is that you should use the `APP_KEY` value of the old `.env` file since this is used for various features like the encryption of multi-factor authentication credentials. Changing the `APP_KEY` may cause such features to break.

URL Changes

If you are restoring into an environment where BookStack will run on a different URL, there are a couple of things you'll need to do after restoring everything:

- Within the `.env` config file update the `APP_URL` value to exactly match your new base URL.
- Run the ["Update System URL" command](#) to update your database content to use your new URL.

If you migrated web-server configuration files, you may also need to tweak those to correctly use the new URL.

Enabling HTTPS for the BookStack web interface

<https://docs.sam.gy/books/bookstack/page/enabling-https-for-the-bookstack-web-interface>

When finished with this guide, will likely want to follow [Using Cert Bot to get a Valid SSL certificate](#)

The default BookStack installation script for Ubuntu 20.04/18.04/16.04 only configures the Apache2 server to use HTTP, not HTTPS. With plain HTTP becoming more and more obsolete each day, you'll likely want to use HTTPS instead in combination with a Let's Encrypt SSL certificate (or a self-signed cert if you're only accessing locally).

Move the original BookStack config file (so you can access it again if needed)

- `sudo mv /etc/apache2/sites-available/bookstack.conf /etc/apache2/sites-available/bookstack.conf.old`

Create a new BookStack Apache2 config file:

- `sudo nano /etc/apache2/sites-available/bookstack.conf`

Here is an example configuration - you will need to change the **ServerName** and **SSLCertificate** directives to match your requirements.

```

<VirtualHost *:80>
  ServerName YOUR-DOMAIN-HERE
  RewriteEngine On
  RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
</VirtualHost>

<VirtualHost *:443>
ServerName YOUR-DOMAIN-HERE
ServerAdmin webmaster@localhost
DocumentRoot /var/www/bookstack/public/

  SSLEngine on
  SSLCertificateFile /etc/letsencrypt/live/YOUR-DOMAIN-HERE/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/YOUR-DOMAIN-HERE/privkey.pem

  <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>

  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

```

Enable Apache2's SSL module

- `sudo a2enmod ssl`

Run Apache2's built in config tester to confirm there are no errors:

- `sudo apachectl configtest`
- Syntax OK

Edit BookStack's `.env` config file:

- `sudo nano /var/www/bookstack/.env`

Change the **APP_URL** parameter so that it uses the full domain name, making sure that it specifies **https://** not **http://**

```
# Application URL
# This must be the root URL that you want to host BookStack on.
# All URLs in BookStack will be generated using this value
# to ensure URLs generated are consistent and secure.
# If you change this in the future you may need to run a command
# to update stored URLs in the database. Command example:
# php artisan bookstack:update-url https://old.example.com https://new.example.com
APP_URL=https://YOUR-DOMAIN-HERE
```

Restart Apache2

- `sudo systemctl restart apache2`

Configuring File Upload

<https://www.bookstackapp.com/docs/admin/upload-config/#migrating-to-secure-images>

Migrating to “Secure” Images

Back-up your BookStack instance before attempting any migration

If you are migrating to the `STORAGE_TYPE=local_secure` or `STORAGE_TYPE=local_secure_restricted` options, with existing images, you will need to move all content from your previous image storage location (see above) to the `storage/uploads/images` folder within your BookStack instance.

Do not simply copy and leave content in the `public/uploads/images` as those images will still be publicly accessible. After doing this migration you may have to clean-up and re-upload any ‘App Icon’ images, found in settings, since these need to remain publicly accessible.

Changing Upload Limits

By default, a lot of server software has strict limits on upload sizes which causes errors when users upload new content. BookStack enforces its own limit but there may also be limits configured as part of PHP and your web sever software. If you run into problems with upload size limits follow the below details for BookStack, PHP and whichever web server you use.

BookStack

The upload limit in BookStack is configured through an option in your `.env` file. Find or add the follow option then configure to your requirements.

- 1
- 2
- 3

```
# File Upload Limit
# Maximum file size, in megabytes, that can be
uploaded to the system.
FILE_UPLOAD_SIZE_LIMIT=50
```

PHP

PHP has two main variables which effect upload limits. Find your `php.ini` file and look for the following variables:

- `post_max_size`
- `upload_max_filesize`

If the values of these variables are low increase them to something sensible that's not too high to cause issues. Unless you need something higher 10MB is a sensible value to enter for these values:

- 1
- 2

```
post_max_size = 10M
upload_max_filesize = 10M
```

If wanting to upload files over 128MB, you may also need to adjust your PHP memory limit like so:

- 1

```
memory_limit = 256M
```

After updating these values ensure you restart your webserver and also PHP if using PHP-FPM or something similar.

NGINX

By default NGINX has a limit of 1MB on file uploads. To change this you will need to set the `client_max_body_size` variable. You can do this either in the http block in your `nginx.conf` file or in the server block set up for BookStack. Here's an example of increasing the limit to 100MB in the http block:

1
2
3
4
5
6

```
http {  
    [#...  
        client_max_body_size 100m;  
        client_body_timeout 120s; # Default is 60,  
        May need to be increased for very large uploads  
    [#...  
}
```

As per the example above, If you are expecting upload very large files where upload times will exceed 60 seconds you will also need to add the `client_body_timeout` variable with a large value.

After updating you NGINX configuration don't forget to restart NGINX. You can test the configuration beforehand with `nginx -t`.

Apache

Apache does not have any built-in limits which you will need to change but something to note is that if you are using apache and mod_php with `.htaccess` files enabled you may be able to set the above PHP variables in your `.htaccess` file like so:

1
2

```
php_value upload_max_filesize 10M  
php_value post_max_size 10M
```