

Configuring File Uploads



- [Storage Options](#)
 - [Local](#)
 - [Local \(Secure\)](#)
 - [Local \(Secure - Restricted\)](#)
 - [S3](#)
 - [Non-Amazon, S3 Compatible Services](#)
 - [Separate Image and Attachment Storage](#)
 - [Migrating to "Secure" Images](#)
 - [Changing Upload Limits](#)
 - [BookStack](#)
 - [PHP](#)
 - [NGINX](#)
 - [Apache](#)
-

BookStack allows users to upload both images for content and files as attachments.

For information relating to security for file uploads please refer to the [Security Page](#).

Storage Options

Within BookStack there are a few different options for storing files:

- **[local](#)** (Default)
 - Files are stored on the server running BookStack.
 - Images are publicly accessible, served by your web-server.
 - Attachments are secured behind BookStack's permission control.
- **[local_secure](#)**
 - Same as local option but images are served by BookStack, with authentication (login required) on image requests.
 - Is only useful while the BookStack "Public Access" setting is disabled.

- More system resource intensive than the default "local" option thus could induce performance issues.
- Attachments are secured behind BookStack's permission control.
- **local_secure_restricted**
 - Same as local option but image access is controlled by user access permission to the item an image is uploaded to.
 - Is the most system resource intensive and could induce performance issues.
 - Has logical side-affects that can hinder ease-of-use.
 - Attachments are secured behind BookStack's permission control.
- **s3**
 - Store files externally on Amazon S3 (or [S3 compatible system](#)).
 - Images are made publicly accessible on upload.
 - Attachments are secured behind BookStack's permission control, as long as files are not exposed via other means.

For all options you can use the 'Enable higher security image uploads' in-app admin setting which appends a random string to each uploaded image name to make URL's hard to guess.

Local

This is the default storage mechanism in BookStack that stores uploads on the local filesystem. It can be forced by setting the following in your `.env` file:

```
STORAGE_TYPE=local
```

- Image uploads location: `<bookstack_install_dir>/public/uploads/images`.
- Attachment uploads location: `<bookstack_install_dir>/storage/uploads/files`.

Local (Secure)

This storage option stores uploads on the local filesystem but does so in a non-publicly exposed folder, where images are only served from if the user is logged-in to BookStack. The local secure option can be enabled by setting the following in your `.env` file:

```
STORAGE_TYPE=local_secure
```

After setting this option ensure you test system performance creating a page with many images and reload on that page multiple times to ensure your server can keep up with the multitude of image requests.

- Image uploads location: `<bookstack_install_dir>/storage/uploads/images`.
- Attachment uploads location: `<bookstack_install_dir>/storage/uploads/files`.

Refer to the [Migrating to “Secure” Images](#) for details about switching to this with existing file uploads.

Local (Secure - Restricted)

This option stores uploads on the local filesystem but controls access to image files based upon the user having permission to view the content that the image has been uploaded to.

Note: This option is relatively new to BookStack and currently considered somewhat experimental.

Due to the rather restrictive & granular permission control enforced by this option, various logical scenarios can be encountered that cause image visibility anomalies. For example if a page, with images uploaded to it, is copied to a new page with different visibility permissions, you could have users that are able to see that page but the images within may not load, since their visibility will remain controlled by the original source page. Another example is that deleting a page, where images were uploaded to, will prevent any user access to the related images.

This option can be enabled by setting the following in your `.env` file:

```
STORAGE_TYPE=local_secure_restricted
```

After setting this option ensure you test system performance creating a page with many images and reload on that page multiple times to ensure your server can keep up with the multitude of image requests.

- Image uploads location: `<bookstack_install_dir>/storage/uploads/images`.
- Attachment uploads location: `<bookstack_install_dir>/storage/uploads/files`.

Refer to the [Migrating to “Secure” Images](#) for details about switching to this with existing file uploads.

S3

The Amazon S3 option can be enabled by setting the following in your `.env` file:

```
STORAGE_TYPE=s3
STORAGE_S3_KEY=your-s3-key
STORAGE_S3_SECRET=your-s3-secret
STORAGE_S3_BUCKET=s3-bucket-name
STORAGE_S3_REGION=s3-bucket-region
```

For performance reasons uploaded images are made public upon upload to your S3 bucket and fetched directly by the end user when viewing an image on BookStack. Attachments are not made public and are instead fetched by BookStack upon request. Exact security will depend on the

configuration and policies of your bucket.

- Image uploads location: `<your_bucket>/uploads/images`.
- Attachment uploads location: `<your_bucket>/uploads/files`.

By default BookStack will generate a valid Amazon S3 URL for uploaded images. If you'd prefer to use a different URL, that you have pointed at your bucket, you can add the below option to your `.env` file which will be used as a base URL for all image uploads:

```
STORAGE_URL=https://images.example.com
```

Non-Amazon, S3 Compatible Services

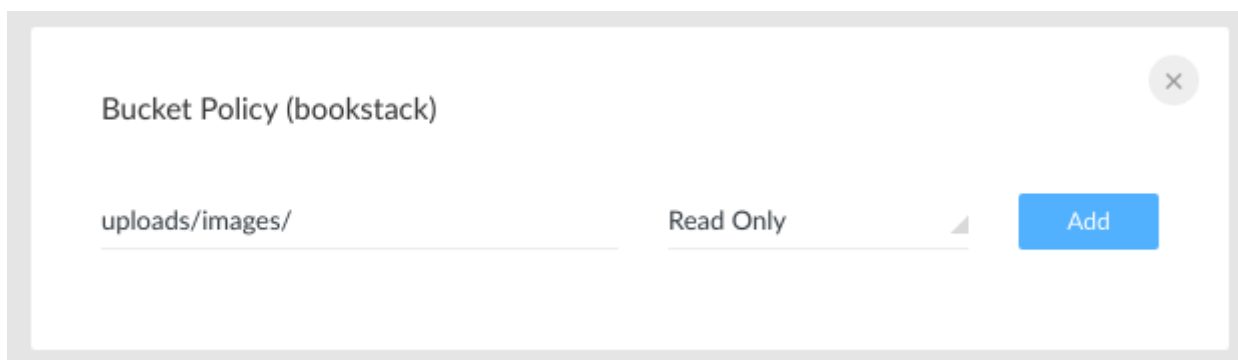
Via the `s3` connection BookStack does support S3-compatible services such as [Minio](#). Read the above S3 details to get an idea of general setup. For non-Amazon services the configuration, to be placed in the `.env` file, is a little different:

```
STORAGE_TYPE=s3
STORAGE_S3_KEY=your-service-key
STORAGE_S3_SECRET=your-service-secret-key
STORAGE_S3_BUCKET=your-service-bucket-name

STORAGE_S3_ENDPOINT=https://your-service-base-endpoint.com:8080
STORAGE_URL=https://your-service-base-endpoint.com:8080/your-service-bucket-name
```

Take note of how the first part of the `STORAGE_URL` path is the bucket name. This is important to ensure image URLs are set correctly.

BookStack's functionality to set image URL's as publicly accessible will likely not work for third-party services so you'll need to ensure files under the `<your_bucket>/uploads/images` path have policy or permissions to be publicly accessible. If using Minio you can add the following to the bucket policy:



Separate Image and Attachment Storage

If you'd prefer to store images and attachments via different storage options, you can use the below `.env` options to do so:

```
# Image storage system to use
# Defaults to the value of STORAGE_TYPE if unset.
# Accepts the same values as STORAGE_TYPE.
STORAGE_IMAGE_TYPE=local

# Attachment storage system to use
# Defaults to the value of STORAGE_TYPE if unset.
# Accepts the same values as STORAGE_TYPE although 'local' will be forced to 'local_secure'.
STORAGE_ATTACHMENT_TYPE=local_secure
```

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 completing this migration you must re-upload the 'Application Icon' and 'Application Logo' images found in `Settings > Customization`, 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.

```
# 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:

```
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:

```
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:

```
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:

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

📄 #5

📄 cocobook-sync 📄 2026-10-28 10:56:02 UTC

📄 cocobook-sync 📄 2026-10-29 03:58:09 UTC