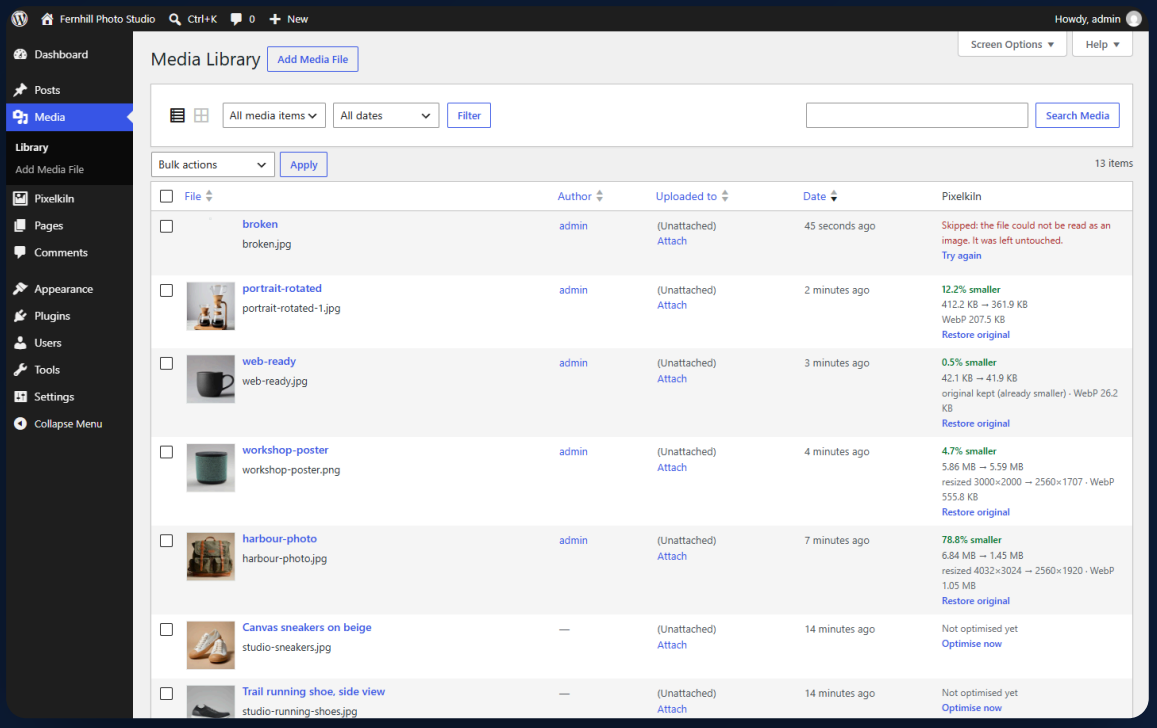
CODE CRAFT
Build. Launch. Grow.

WORDPRESS PLUGIN · USER GUIDE

Pixelkiln — WordPress Image Optimizer & WebP/AVIF Converter

Version 1.0.0 · Installation, licence activation, every setting explained, FAQ and troubleshooting.



coodecraft.com/market/product/wp-image-optimizer · support: coodecraft.com/contact

Pixelkiln user guide

Pixelkiln is a WordPress image optimizer that works entirely on your own server. It uses the GD or Imagick library that PHP already has to compress JPEG and PNG images, scale down huge uploads, and write WebP copies (and AVIF copies in Pro, where your server can encode AVIF). There is no cloud service, no account and no credit limit. Every original is copied to a backup folder before it is changed, so each image can be restored.

How much smaller your images get depends on the images. Large phone and camera photos usually shrink a lot, mostly because they are scaled down. Images already saved for the web may not shrink at all; Pixelkiln then keeps them exactly as they are.

---

1. Installation

1. In WordPress go to **Plugins** → **Add New** → **Upload Plugin**, choose the Pixelkiln .zip, click **Install Now**, then **Activate**. The Free edition can also be installed from Plugins → Add New by searching for Pixelkiln.
2. A **Pixelkiln** menu appears below Media, with **Settings** and **Licence** (Pro) or **Upgrade to Pro** (Free).
3. New uploads are optimised straight away with the defaults: quality 82, resize above 2560 × 2560 pixels, backups on, WebP copies on, WebP served by URL swap.

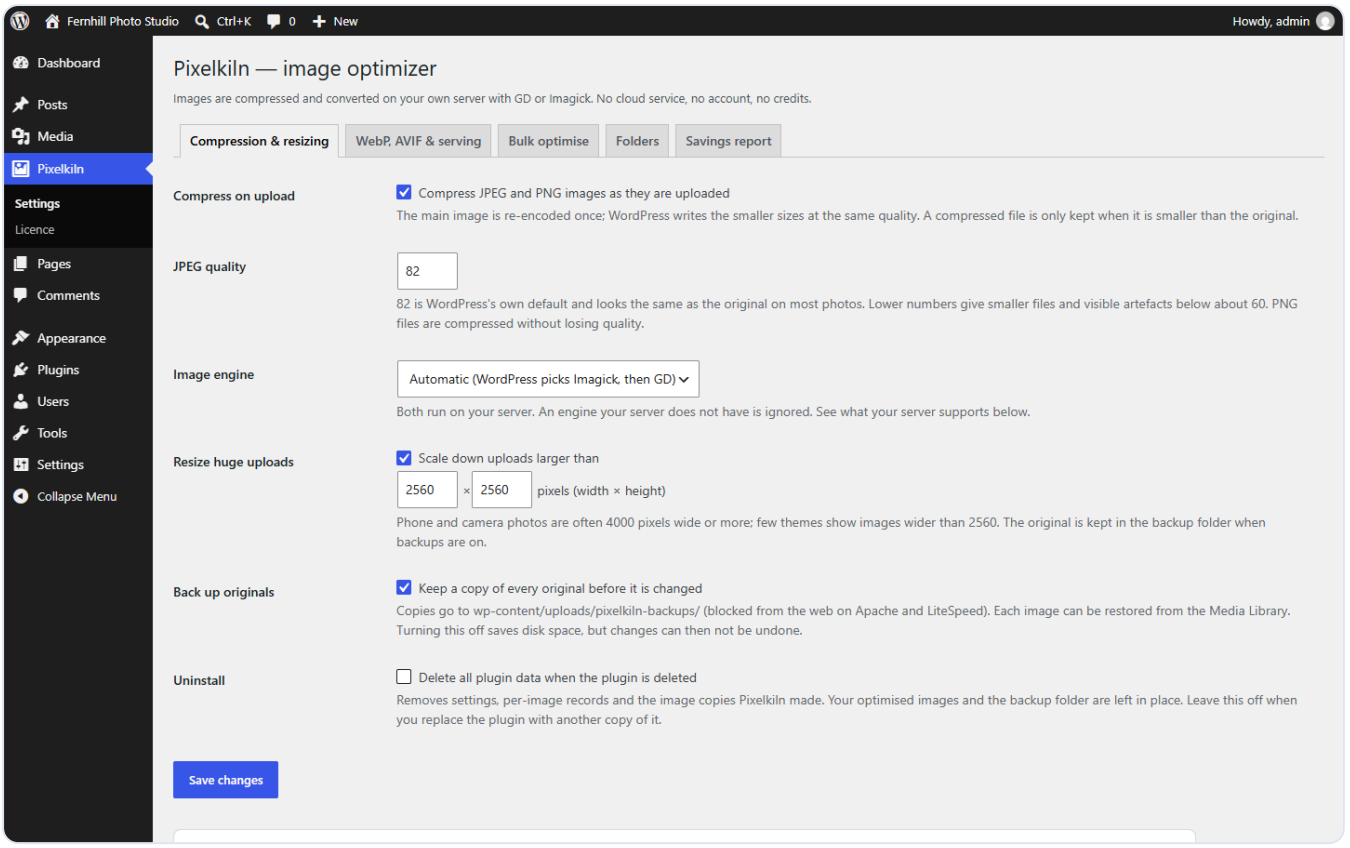
Activating Pro

1. Buy a plan on coodecraft.com. The licence key arrives by email.
2. Open **Pixelkiln** → **Licence**, paste the key and click **Activate licence**.
3. Monthly and yearly keys switch Pro off when they end and back on when you renew (click **I have paid — check again**). When Pro is off, Pixelkiln works like the Free edition: uploads are still compressed, resized, backed up and given WebP copies; `<picture>` and `.htaccess` serving fall back to the Free URL swap (the `.htaccess` block is removed); a running background job stops. Your Pro settings are kept.

---

2. Compression & resizing

Screenshot:

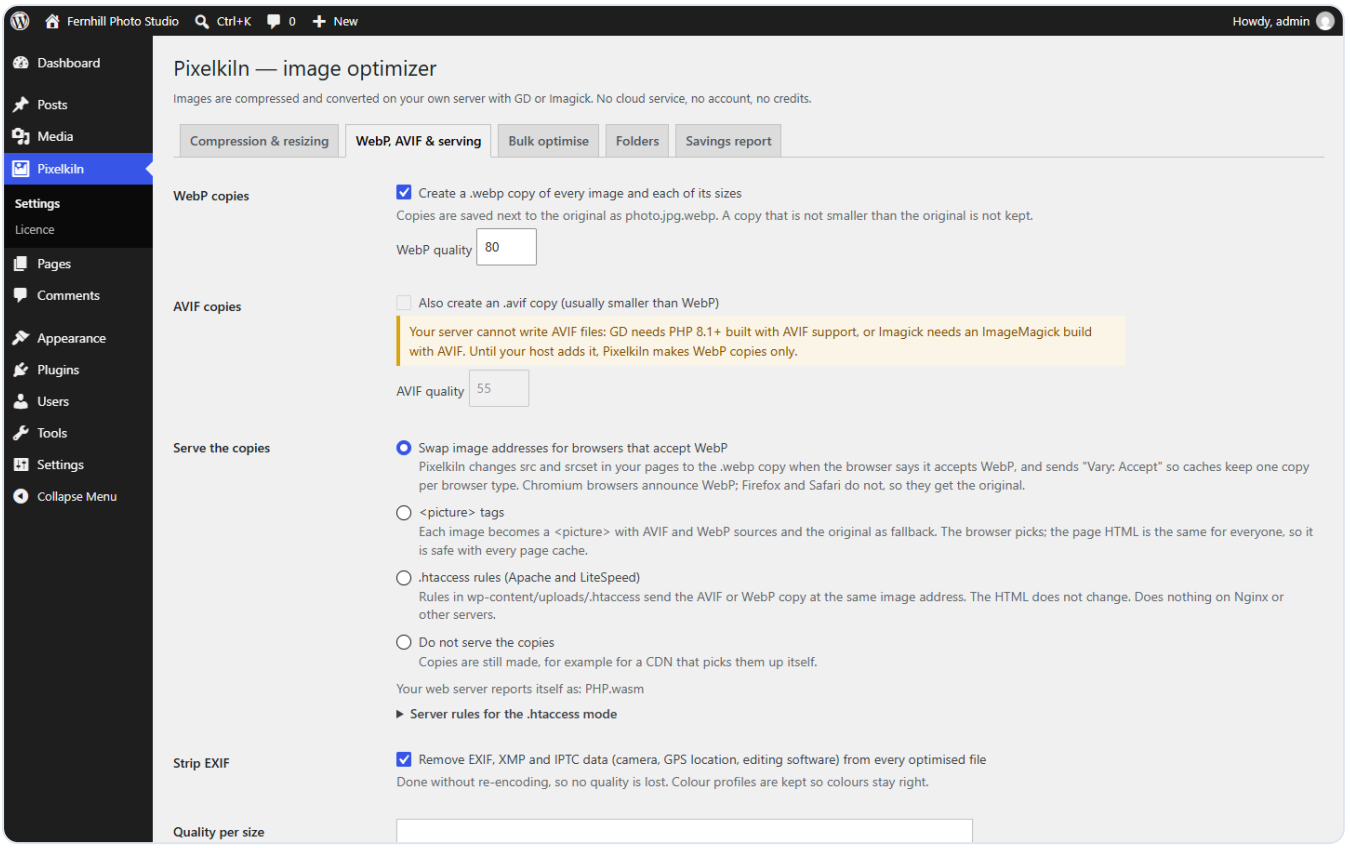


- **Compress on upload** — the main image is re-encoded once at your JPEG quality; WordPress writes the smaller sizes at the same quality. The new file is only kept when it is smaller than the original.
- **JPEG quality** — 82 is WordPress's own default. Lower numbers give smaller files; below about 60 artefacts become visible. PNG files are re-encoded without quality loss.
- **Image engine** — Automatic (WordPress's choice), GD or Imagick. An engine your server does not have is ignored.
- **Resize huge uploads** — uploads wider or taller than the limit are scaled down before WordPress makes its sizes. The EXIF rotation of phone photos is applied first, so portraits stay upright. WordPress's own 2560-pixel "-scaled" copy is not made while this is on.
- **Back up originals** — the upload is copied to `wp-content/uploads/pixelkiln-backups/` before anything changes. The folder has an `.htaccess` that blocks it from the web on Apache and LiteSpeed.
- **Delete all plugin data when the plugin is deleted** — off by default (see section 9).

The box **What your server supports** lists GD and Imagick, their versions, and whether each can write WebP and AVIF. Below it Pixelkiln reports the result of a real test: it writes a 1024 × 768 image as WebP and as AVIF. Some servers report AVIF support but fail on images of a realistic size; this test catches that.

3. WebP, AVIF & serving

Screenshot:

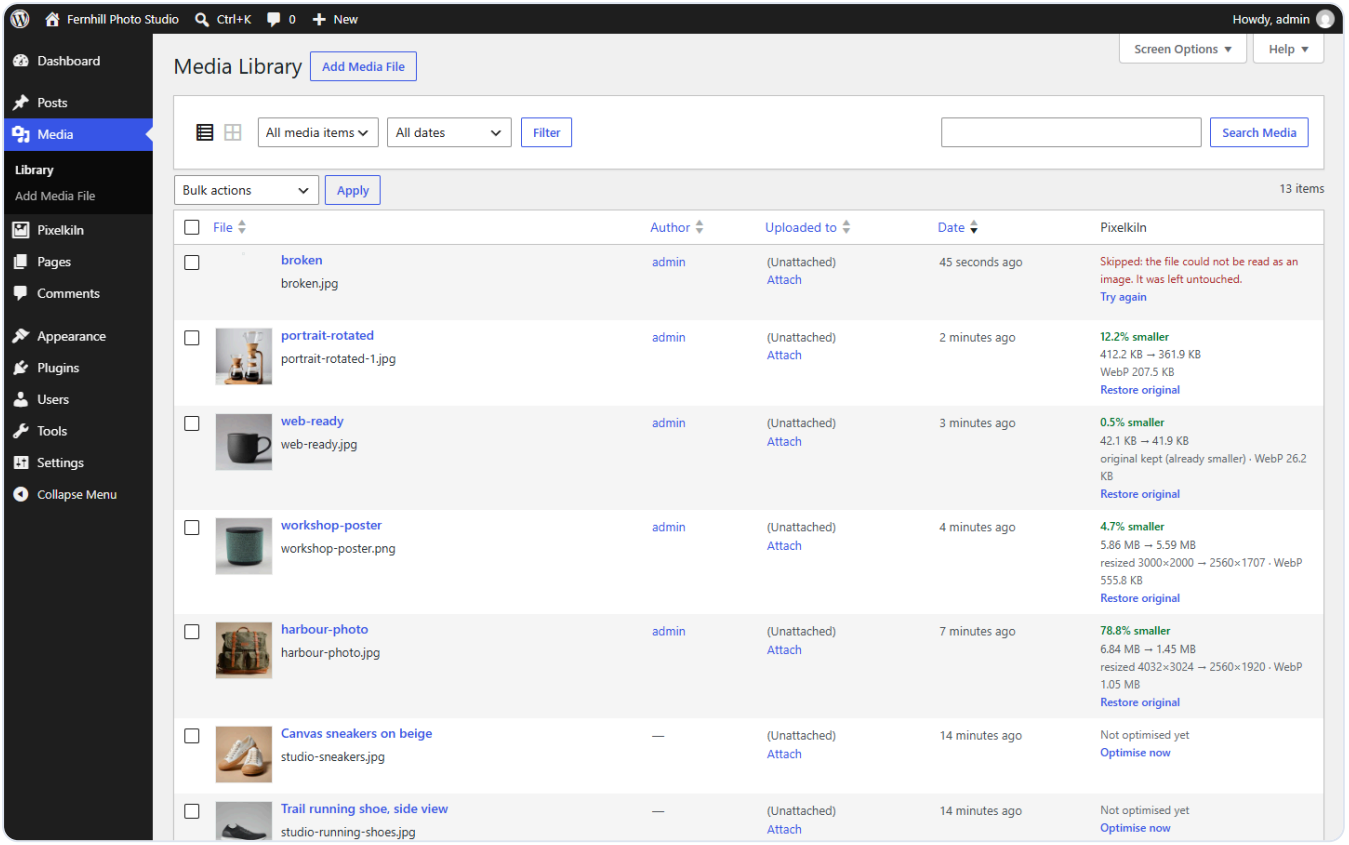


- **WebP copies** — a `.webp` copy of the main image and every size, saved next to it as `photo.jpg.webp`. A copy that is not smaller than the original is not kept.
- **AVIF copies (Pro)** — an `.avif` copy as well. This option is disabled with a plain message when the server test fails.
- **Serve the copies**
  - **Swap image addresses (Free)** — when the browser says in its page request that it accepts WebP, image `src` and `srcset` in your pages point to the `.webp` copies. Pixelkiln sends `Vary: Accept`, so caches keep a separate copy for those browsers. Chrome, Edge and other Chromium browsers announce WebP; Firefox and Safari do not announce it in page requests, so they get the original JPEG/PNG in this mode.
  - `<picture>` **tags (Pro)** — each image becomes a `<picture>` with AVIF and WebP sources and the original as fallback. Every browser picks the best format it supports, and the HTML is the same for every visitor, so this is safe with any page cache. Each source lists only the sizes that have a copy in that format.
  - `.htaccess` **rules (Pro)** — rules in `wp-content/uploads/.htaccess` send the AVIF or WebP copy at the original image address. This only works on Apache and LiteSpeed. The rules and an Nginx example are shown under **Server rules for the .htaccess mode**.
  - **Do not serve** — copies are still made, for example for a CDN that picks them up itself.
- **Strip EXIF (Pro)** — removes EXIF, XMP and IPTC data (camera, GPS location, software) and PNG text chunks without re-encoding. Colour profiles are kept. A file whose EXIF rotation is not "normal" is left alone, so it does not display sideways.

- **Quality per size (Pro)** — a JPEG quality for the main image and for each registered size. Empty fields use the main quality.

4. The Media Library column

Screenshot:



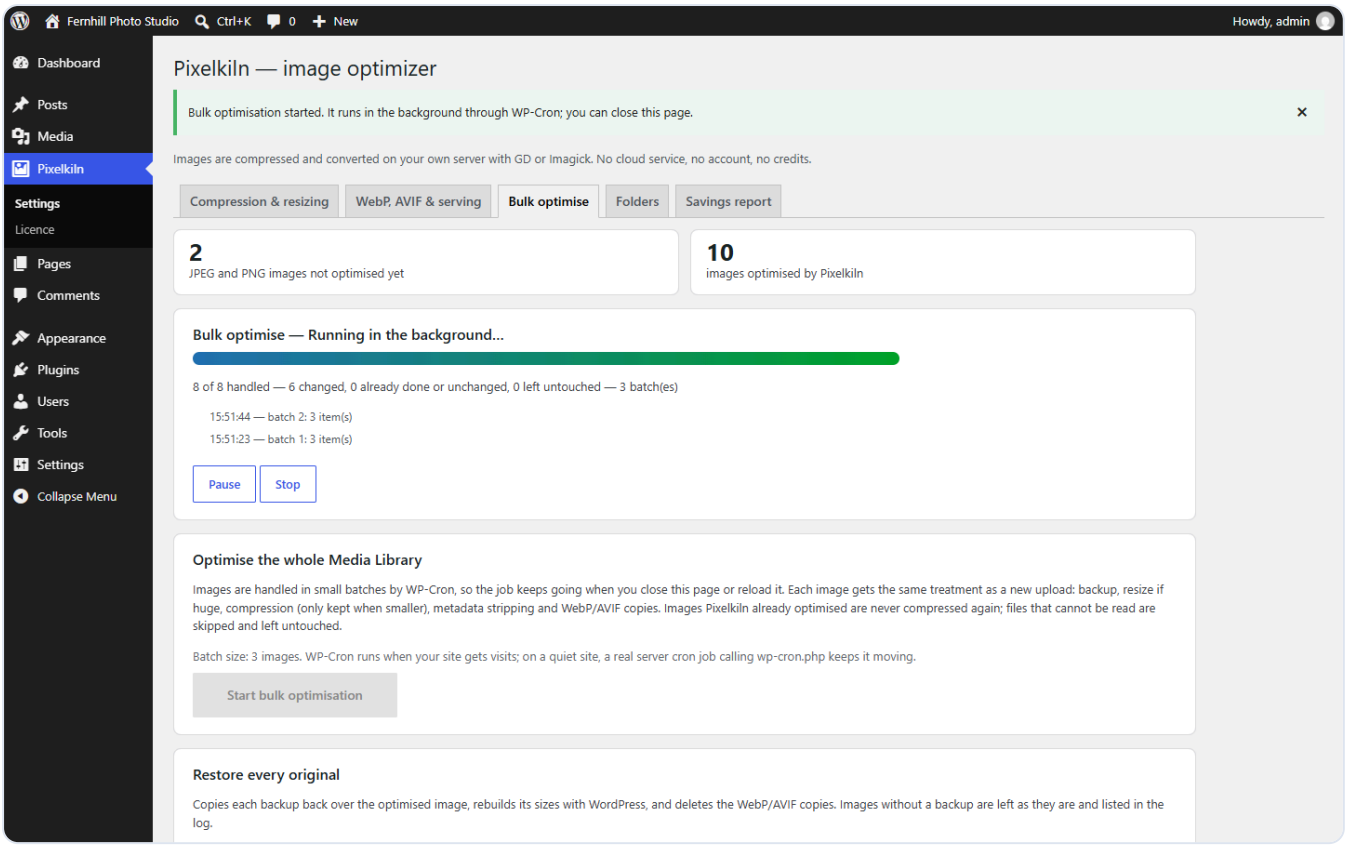
Open **Media → Library** in list view. The **Pixelkiln** column shows, for each JPEG and PNG:

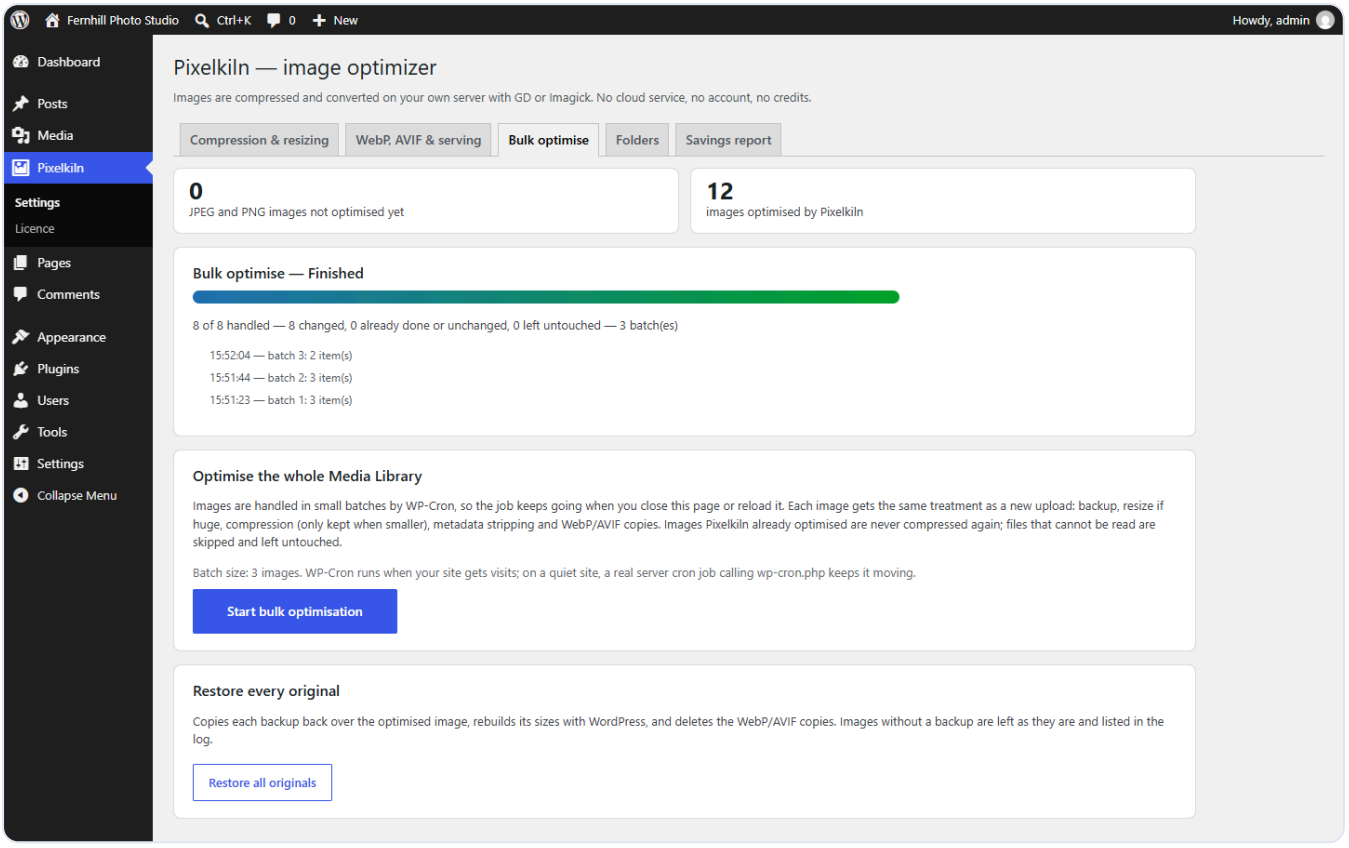
- bytes before and now (all sizes together) and the percentage saved;
- "resized W×H → W×H", or "original kept (already smaller)";
- the size of the WebP/AVIF copies;
- **Restore original** (when there is a backup) or **Optimise now** (images uploaded before Pixelkiln, or restored ones).

An image Pixelkiln already optimised is never compressed again. A file that cannot be read as an image is marked "Skipped" and left untouched.

5. Bulk optimise and restore all (Pro)

Screenshots: ,





1. Open **Pixelkiln → Settings → Bulk optimise**. The screen shows how many images are not optimised yet.
2. Click **Start bulk optimisation**. Images are handled in small batches by WP-Cron. Progress is saved after every batch, so you can reload the page or close it. **Pause**, **Resume** and **Stop** are on the same screen.

3. Each image gets the same treatment as a new upload: backup, resize if huge, compression (kept only when smaller), metadata stripping, WebP/AVIF copies.

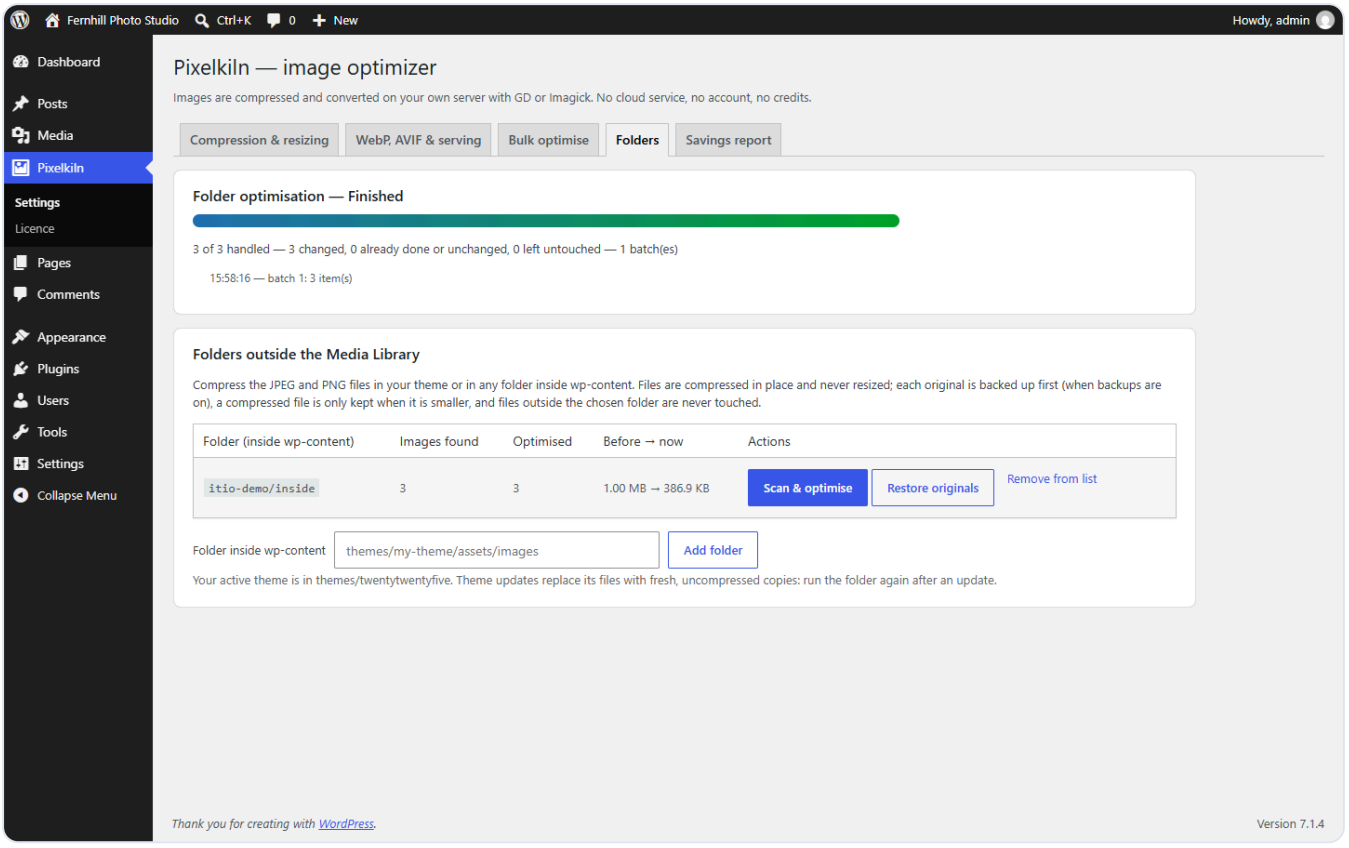
WP-Cron runs when your site gets visits. On a quiet site, a server cron job that calls `wp-cron.php` every minute keeps the job moving. A five-minute watchdog restarts a job whose next run went missing.

Restore all originals copies every backup back, rebuilds the sizes with WordPress and deletes the WebP/AVIF copies, also in background batches. Images without a backup are left as they are and counted in the log.

---

6. Folders (Pro)

Screenshot:

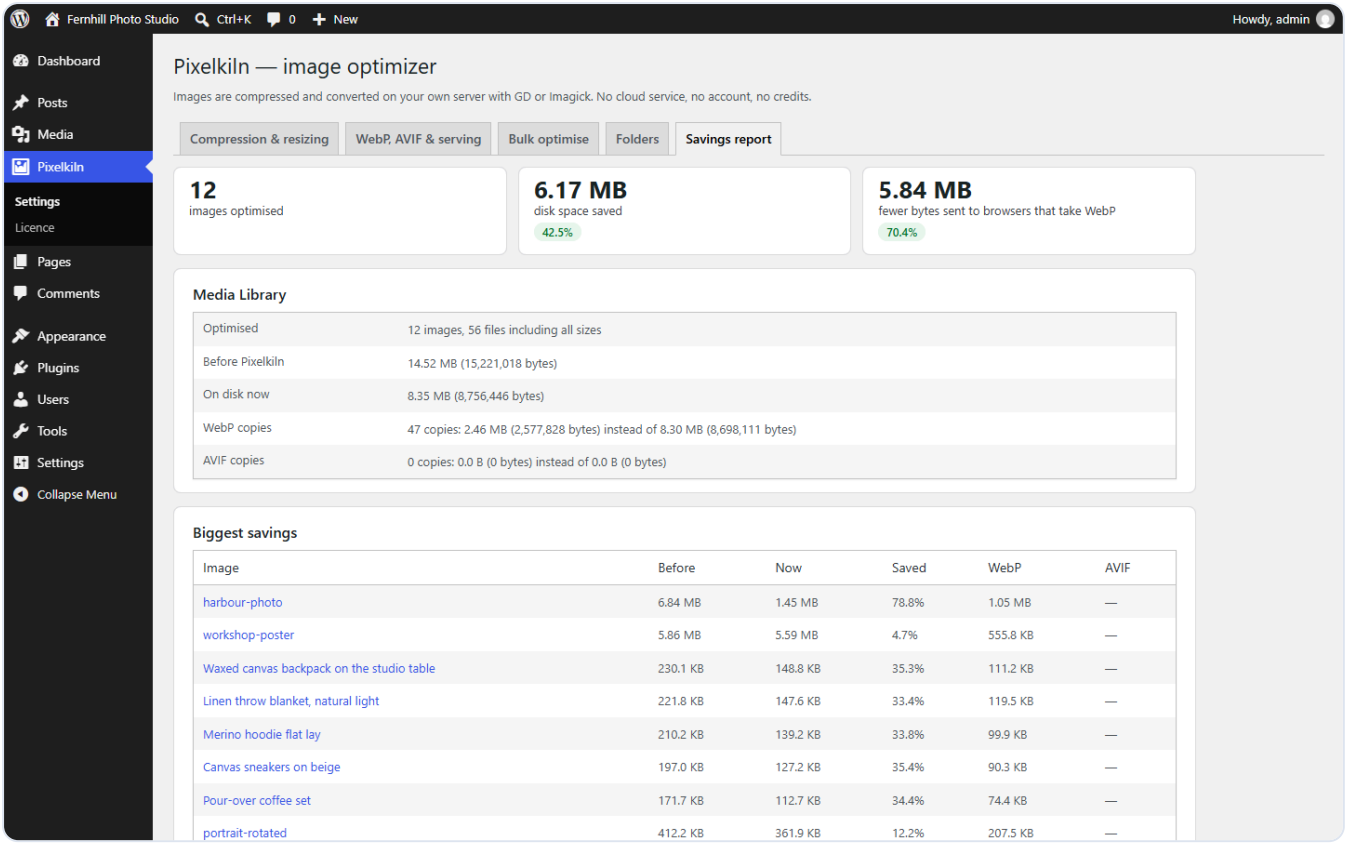


1. Open **Folders**. Type a folder inside wp-content (for example `themes/my-theme/assets/images`) and click **Add folder**. Your active theme's folder is suggested.
2. Click **Scan & optimise**. Pixelkiln lists the JPEG and PNG files inside that folder only and compresses them in place in the background. Files are never resized, because theme layouts may depend on their dimensions. Each original is backed up first.
3. **Restore originals** puts that folder's backups back. **Remove from list** forgets the folder; its files stay as they are.

Theme updates replace theme files with fresh copies: run the folder again after an update.

7. Savings report (Pro)

Screenshot:



The report is measured from the files on disk each time you open it. "Before" is each file's size before Pixelkiln changed it; "On disk now" is its size right now. WebP and AVIF lines compare the copies with the files they replace for browsers that accept them. The **Biggest savings** table lists the top 20 images, and the **Folders** table covers folder optimisation. New uploads show most of their saving on the main image, because WordPress already writes the smaller sizes at your quality.

8. Restoring

- One image: **Media** → **Library** (list view) → **Restore original**. The backup is copied back byte for byte, WordPress rebuilds the sizes, and the WebP/AVIF copies are deleted. If the original was larger than 2560 pixels, WordPress makes its own "-scaled" copy again, exactly as it would without Pixelkiln.
- Everything: **Bulk optimise** → **Restore all originals** (Pro).

9. Deactivating and deleting

- Deactivating clears Pixelkiln's scheduled jobs and removes its `.htaccess` block. Your images, copies and backups stay.
- Deleting without the delete-data setting keeps everything, so you can swap Free and Pro.

- Deleting with **Delete all plugin data** ticked removes the settings, the per-image records, the folder table and the WebP/AVIF copies. The optimised images stay as they are, and the backup folder `wp-content/uploads/pixelkiln-backups/` is kept, so no original is lost. Delete that folder yourself when you no longer need it.

---

10. FAQ

Does Pixelkiln upload my images anywhere? No. Everything runs on your server. Pro only contacts coodecraft.com to check its licence.

Why is AVIF greyed out? AVIF needs Pro and a server that can encode it (GD on PHP 8.1+ built with AVIF, or an Imagick build with AVIF). The settings screen tests this with a real image.

Does it support GIF, SVG, HEIC or WebP uploads? Not yet: Pixelkiln optimises JPEG and PNG.

Is there a CDN? No. Pixelkiln serves the copies from your own site.

Does it lazy-load images? WordPress already adds `loading="lazy"` to images itself; Pixelkiln keeps those attributes.

---

11. Troubleshooting

- **"Pixelkiln needs direct write access"** — your host asks WordPress for FTP details instead of writing files directly. Pixelkiln changes nothing until it can write directly; ask your host.
- **Images not smaller** — they were probably already optimised for the web. The column says "original kept (already smaller)".
- **WebP not shown in Firefox or Safari** — they do not announce WebP in page requests, so the Free URL swap gives them the original. Use `<picture>` tags (Pro).
- **Bulk job stuck at the same number** — WP-Cron only runs when the site gets visits. Open any page, or set up a server cron job for `wp-cron.php`.
- **Images look wrong after changing the theme's files** — use Folders → Restore originals for that folder.