

Efficiently encode images

Warning: This audit has moved into the [Improve image delivery](#)

(</docs/performance/insights/image-delivery>) insight as of Lighthouse 13. For more information see [What's new in Lighthouse 13](#) (</blog/lighthouse-13-0>).

The Opportunities section of your Lighthouse report lists all unoptimized images, with potential savings in [kibibytes \(KiB\)](https://en.wikipedia.org/wiki/Kibibyte) (<https://en.wikipedia.org/wiki/Kibibyte>). Optimize these images so that the page loads faster and consumes less data:

▲ Efficiently encode images

29.4 s ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

☒ Show 3rd-party resources (2)

URL		Size	Potential Savings
	/images/kitten2.jpg (cdn.glitch.com)	11,849 KB	5,824 KB
	/images/kitten.jpg (cdn.glitch.com)	559 KB	26 KB

How Lighthouse flags images as optimizable

Lighthouse collects all the JPEG or BMP images on the page, sets each image's compression level to 85, and then compares the original version with the compressed version. If the potential savings are 4KiB or greater, Lighthouse flags the image as optimizable.

How to optimize images

There are many steps you can take to optimize your images, including:

- [Using image CDNs](https://web.dev/articles/image-cdns) (<https://web.dev/articles/image-cdns>)
- [Compressing images](https://web.dev/articles/use-imagemin-to-compress-images) (<https://web.dev/articles/use-imagemin-to-compress-images>)

- [Replacing animated GIFs with video](https://web.dev/articles/replace-gifs-with-videos) (https://web.dev/articles/replace-gifs-with-videos)
- [Lazy loading images](https://web.dev/articles/codelab-use-lazysizes-to-lazyload-images) (https://web.dev/articles/codelab-use-lazysizes-to-lazyload-images)
- [Serving responsive images](https://web.dev/articles/serve-responsive-images) (https://web.dev/articles/serve-responsive-images)
- [Serving images with correct dimensions](https://web.dev/articles/serve-images-with-correct-dimensions) (https://web.dev/articles/serve-images-with-correct-dimensions)
- [Using WebP images](https://web.dev/articles/serve-images-webp) (https://web.dev/articles/serve-images-webp)

Optimize images using GUI tools

Another approach is to run your images through an optimizer that you install onto your computer and run as a GUI. For example, with [ImageOptim](https://imageoptim.com/mac) (https://imageoptim.com/mac) you drag and drop images into its UI, and then it automatically compresses the images without noticeably compromising quality. If you're running a small site and can handle manually optimizing all images, this option is probably good enough.

[Squoosh](https://squoosh.app/) (https://squoosh.app/) is another option. Squoosh is maintained by the Google Web DevRel team.

Stack-specific guidance

Drupal

Consider using [a module](https://www.drupal.org/project/project_module?%5B0%5D=&%5B1%5D=&%5B2%5D=im_vid_3%3A123&%5B3%5D=&%5B4%5D=sm_field_project_type%3Afull&%5B5%5D=&%5B6%5D=&text=image&solrsort=iss_project_release_usage+desc&op=Search)

(https://www.drupal.org/project/project_module?f%5B0%5D=&f%5B1%5D=&f%5B2%5D=im_vid_3%3A123&f%5B3%5D=&f%5B4%5D=sm_field_project_type%3Afull&f%5B5%5D=&f%5B6%5D=&text=image&solrsort=iss_project_release_usage+desc&op=Search)

that automatically optimizes and reduces the size of images uploaded through the site while retaining quality. Also, ensure you are using the native [Responsive Image Styles](https://www.drupal.org/documentation/modules/responsive_image)

(https://www.drupal.org/documentation/modules/responsive_image) provided from **Drupal** for all images rendered on the site.

Joomla

Consider using an [image optimization plugin](#)

(https://extensions.joomla.org/instant-search/?jed_live%5Bquery%5D=performance) that compresses your images while retaining quality.

Magento

Consider using a [third-party Magento extension that optimizes images](#)

(<https://marketplace.magento.com/catalogsearch/result/?q=optimize%20image>).

WordPress

Consider using an [image optimization WordPress plugin](#)

(<https://wordpress.org/plugins/search/optimize+images/>) that compresses your images while retaining quality.

Resources

- [Source code for **Efficiently encode images** audit](#)
(<https://github.com/GoogleChrome/lighthouse/blob/main/core/audits/byte-efficiency/uses-optimized-images.js>)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](#) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](#) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2019-05-02 UTC.