

Speed Testing

Pagespeed's Business Impact

Pagespeed/Load Time directly influences conversions, time on page, bounce rate, reading time, Google search engine result rankings, user experience, and revenue. For E-Commerce sites improving Pagespeed increases sales, reduces the amount of abandoned carts, and increases the percentage of users who complete the checkout process.

Performance Optimization is one of the most important things you can do to improve a website, whether that be for improved user experience or increasing the amount of revenue generated by the business.

Web.dev Article

<https://web.dev/case-studies/vitals-business-impact>

Business impact analysis of Core Web Vitals improvements directly from Google.

Conductor Article

<https://www.conductor.com/academy/page-speed-resources/>

Another article with solid case studies of the improvements in many metrics that sites have seen after improving their Pagespeed.

Deloitte Report

<https://www.deloitte.com/ie/en/services/consulting/research/milliseconds-make-millions.html>

A thorough analysis of Pagespeed Improvement's impact on Business' revenue.

Metrics Explained

Escape Creative Article

<https://escapecreative.io/google-pagespeed-insights-explained/>

This page has a decent overview of the Metrics and what they mean.

Corelogix Article

<https://coralogix.com/guides/real-user-monitoring/core-web-vitals/>

Another useful article with somewhat in-depth explanations of various Core Web Vitals metrics and their definitions.

Google's Core Web Vitals Documentation

<https://web.dev/articles/vitals>

Core Web Vitals documentation and definitions straight from Google.

MultiDots Article

<https://www.multidots.com/blog/guide-to-measuring-improving-core-web-vitals-for-wordpress-site/>

Also has a decent breakdown of CWV metrics

Why Do Test Scores Change Every Time I Run A Speed Test?

There are many factors that will impact the score you see on these tests. These variables can range from your location related to the website, activity on your server, network routes and variability, and even the background activity of your local device. There will always be inter-test variance, across all speed testing providers, no two tests will be exactly the same regardless of the speed test used.

If your site is well optimized the range of values will be smaller and you'll get more consistent scores in the same range, but an unoptimized site is going to have wildly varying scores between tests. However, as aforementioned, there will always be variance between tests, both for mobile and desktop.

Note: Your true scores are best measured as an average of multiple page speed tests. I generally perform 3 tests before I average them together.

Mobile Pagespeed Scores Are The Only Scores That Matter

The reason mobile performance is all that matters is multi-fold.

Improving Mobile Scores Inherently Improves Desktop Scores.

The source of pagespeed issues (css, js, unoptimized images, etc) is not specific to either Mobile or Desktop devices, rather their impacts are universal and felt by both Desktop and Mobile users.

However, the impacts of each performance issue are exacerbated on mobile and the performance impact of these detriments is larger.

There are no "unique" optimizations that will only improve performance on mobile or only improve performance on Desktop. When you improve performance for mobile, you've improved performance across the board, including Desktop.

Mobile Has Greater Variability Between Test Scores

There is a greater range of variability in the scores. If you were only to rely on Desktop scores, you could easily end up being deceived that nothing else needs to be optimized.

It's quite easy to get a Desktop site to score under 1 second load time, and to get 90+ or 100/100 on Desktop is generally a walk in the park, but that's just Desktop.

Performance Issues Have A Greater Impact On Mobile Devices

Under a 1 second desktop load time can be anywhere from 2-6 seconds on mobile, potentially worse due to how much more powerful Desktop's are than mobile devices.

That is because network conditions may change (cell service or distance from a Wifi network access point), and device load is also in constant flux. Each additional kilobyte of data (especially images) has a far greater effect on a mobile device's performance more than it does a Desktop device.

A 60 KB image vs a 30 KB image may have no impact on the total Desktop performance scores out of 100, but the size difference certainly can and will make a noticeable impact on mobile. Desktop scores may show 0 CLS, but that can be very deceptive as mobile devices may have a large amount of CLS when accessing the same website.

Mobile Pagespeed Tests Are Taken Under Controlled Simulated Conditions

These simulated conditions are equivalent to the "worst case scenario" of an under powered device (Currently the simulated target is a Moto G Power).

Google's Pagespeed Weighting On Rankings Are Ranked By Mobile Pagespeed First, *Always*, Without Exception.

Whether the majority of your site's traffic is Desktop or Mobile is irrelevant insofar to how Google ranks your webpage. Google does not care about Desktop performance when they rank a website. 80%+ of traffic on the internet is from mobile devices, and Google calculates their rankings accordingly. Even if your website's users are browsing from a majority of Desktop devices, your mobile score is the only one that will be considered as a ranking factor.

Desktop Scores Are Irrelevant

Desktop scores have such little diagnostic value as to be essentially useless. In fact, their only diagnostic value is as a reverse metric. Poor Desktop scores are indicative of one thing and one thing only, a massive amount of performance issues that must be solved. Good Desktop performance scores indicate nothing as to how well a website is optimized.

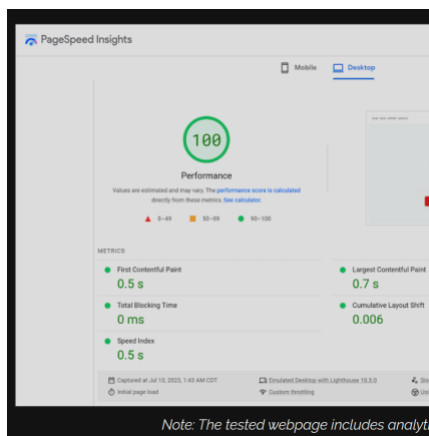
They have such little value that they can be summarily ignored, Desktop scores are irrelevant.

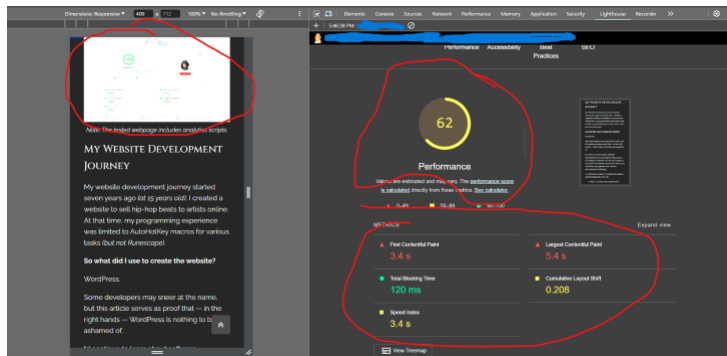
The vast majority of times a website screenshots with high scores in the 90s are generally displaying Desktop performance scores. Anyone lauding their Desktop performance scores are deceiving themselves as well as their customers, and the immediate reaction should be skepticism and to open your browser's console immediately with inspect to run a lighthouse mobile test on their website.

The amount of performance “experts” who claim good desktop performance means their site is well optimized is far, far too high.

Desktop Performance Vs Mobile Performance Example

Here is a great example of a site with good Desktop performance scores with poor mobile scores. Always verify what a site's mobile scores are when they claim to have rock solid performance, they're almost always referring to Desktop.





The Desktop version scores very well, with almost zero CLS and every metric displays as loading in under 1 second, and yet the Mobile Scores are a complete mismatch to the Desktop performance scores. Google will incorporate this site's poor Mobile Pagespeed scores as the performance metrics in their ranking score, with zero weight given to the Desktop score.

TTFB

TTFB Components

DNS

The DNS lookup time is the time taken to query and receive a response from a DNS server with the IP address of the domain. If the DNS response is not cached locally, this process can add significant time to the TTFB.

How to optimize the DNS portion of TTFB

Use a fast DNS provider with DNS caching such as Cloudflare. Even if you don't use them as a CDN, I highly recommend using them as your DNS provider.

Connect

Connection time includes the time it takes to perform the TCP handshake, which is essential for setting up a TCP connection before any actual data can be sent. This typically involves the sending of SYN and ACK packets between the client and server to acknowledge the connection.

How to optimize the Connect portion of TTFB

Follow the steps in the TCP optimization portion of the VPS section above.

SSL

The SSL/TLS handshake time is the duration taken to authenticate the server (and optionally the client), as well as to negotiate the encryption and the key exchange for the session. This process involves multiple steps of exchanging messages and can add considerable time to the

TTFB, especially in cases where a new session is established without using session resumption techniques.

How to optimize the SSL portion of TTFB

The SSL optimization section above has a multitude of resources on how to optimize SSL connection speed.

Sending

This component measures the time taken to send the first byte of the response from the server to the client after the server has finished processing the request. This includes the time it takes to send the initial data of the HTTP response header.

How to optimize the Sending portion of TTFB

Brotli/Gzip Compression

Utilize compression techniques such as GZIP or Brotli to compress the data sent from the server. This reduces the size of the data being transmitted, thus decreasing the time it takes for the first byte to reach the client. It's especially effective for text-based content like HTML, CSS, and JavaScript files.

Optimize Your Server Configuration

Ensure that your server is well-configured to handle requests efficiently. This includes tuning web server software settings (like Apache or Nginx) for better performance by adjusting worker processes, keep-alive settings, and client request handling.

Consider using a reverse proxy or a load balancer that can manage connections and distribute traffic effectively, minimizing delays in sending data.

Use HTTP/2 or HTTP/3

Implement HTTP/2 or HTTP/3, as both protocols provide improvements over HTTP/1.1 in reducing the overhead associated with sending data. HTTP/2 introduces multiplexing, which allows multiple requests and responses to be intermingled on a single connection, reducing the latency associated with setting up multiple TCP connections. HTTP/3 further improves the process by reducing connection establishment time and minimizing the impact of packet loss.

Minimize TLS Handshake Delays

- If using HTTPS (recommended), optimize SSL/TLS settings by choosing faster cipher suites that require less computational power and support techniques like TLS False Start

and Session Resumption. These methods can significantly reduce the time taken to establish a secure connection, hence speeding up the sending phase.

- Consider using a Content Delivery Network (CDN) with edge nodes that support modern TLS features and are closer to your users to decrease the distance data needs to travel.

(Additional SSL Optimization Strategies are in the SSL optimization section above)

Review and Optimize Application Code

Analyze and optimize the backend code of your application. Ensure that any server-side scripting (like PHP, Python, or Node.js) is optimized for performance. This includes reducing database query times, optimizing algorithms, and ensuring efficient data processing. Use server-side performance tools to monitor and identify bottlenecks in real-time.

This also encompasses all Wordpress optimizations below.

Content Delivery Network (CDN)

Deploying your content across a CDN can potentially reduce the sending time significantly. CDNs store copies of your data in multiple geographic locations, reducing the physical distance between the server and the client, thus minimizing the data travel time.

Database Optimization

Optimize database queries to ensure they execute swiftly and do not become a bottleneck. Efficient indexing, query optimization, and appropriate caching strategies can reduce the time it takes for the server to retrieve data and start sending it to the client.

(Database Optimization has a section below)

Waiting (Server Processing Time)

This is the time taken by the server to process the request and prepare a response. It includes the time taken to run server-side scripts, query databases, and load any necessary resources to generate the response.

How to optimize the Waiting portion of TTFB

Optimize Wordpress at the application level. This is where Wordpress optimization comes in. Inline javascript, how heavy your HTML is, inline CSS, etc. This is where your HTML DOM and Document size come into play.

Receiving

This final component accounts for the time it takes for the client to receive the first byte of data from the server.

How to optimize the Receiving portion of TTFB

1. Reduce Server Response Size

For the receiving portion, both inline and external CSS and JavaScript files matter because they can influence the size of the initial server response. While inline resources directly add to the HTML document size, external files contribute to the overall number of bytes the browser must handle after the initial HTML is received.

Reducing the overall size of these resources (both inline and external) can decrease the time it takes for the client to start receiving data from the server. Techniques such as minifying and compressing (using GZIP or Brotli) these resources are directly applicable here.

2. Streamline Initial Payload

Deliver only what is necessary for the initial render of the page. This might mean optimizing the HTML to include only essential inline CSS/JS and deferring everything else.

For external resources, adjust the timing of their fetching (through async and defer attributes in script tags) so that they do not clog the initial payload, potentially delaying the start of the data transmission.

Follow the Wordpress Optimization steps below

3. Use HTTP/2 or HTTP/3

Both HTTP/2 and HTTP/3 improve the efficiency of data transfer. HTTP/2's multiplexing feature allows multiple requests and responses between the client and server to be intermingled on a single connection, reducing the overhead caused by multiple connections. HTTP/3 further improves upon this by reducing connection and transport latency, utilizing QUIC, which can start sending data as soon as the connection is established, potentially decreasing the delay in receiving the first byte.

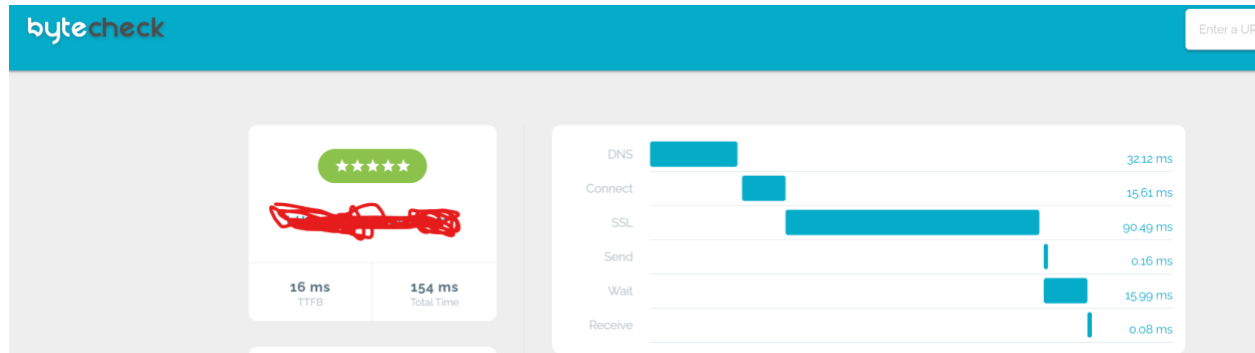
4. Efficient Use of TCP and Adjust TCP Windows

Optimize TCP settings, including the initial congestion window, which can affect how much data can be sent during the start of the connection. A larger window size allows more data to be "in flight" on the network, potentially reducing the time until the first byte is received. Ensure that the server's TCP stack is optimized for quick delivery, which includes considerations of TCP slow start, acknowledgments, and retransmissions.

TCP Optimization is a subsection of the VPS section

ByteCheck (Free)

<https://www.bytecheck.com/>



Byte Check provides a really nice breakdown of TTFB and breaks it down into its constituent components(DNS, Connect time, SSL, Send, Wait, Receive).

Byte Check is indicating that there is 16 MS of processing time from the origin server, which it's breaking out into a separate component it is labeling as the TTFB. On the majority of other speed tests, they will display the total value of 154 MS, which comprises the 6 components on the right, as the TTFB.

That means GTMetrix, Debug Bear, etc are going to display the TTFB as the total time before your files begin to serve to the user's browser. While Debug Bear and GTMetrix include small subsections on their waterfall charts for the initial html document request, Bytecheck is just way easier to interpret and read and is the best tool to measure TTFB.

Byte Check is a useful speed test to use as a complement alongside Debug Bear/GtMetrix/Pagespeed insights.

Note

PSA to anyone who uses Bytecheck: make sure the HTTP response code isn't a 403. Cloudflare likes to block Bytecheck, returning a 403, but Bytecheck doesn't make it immediately obvious unless you check the HTTP response code. So your results may look superb, but in reality that's because it's only loading a basic 403 error page.

Credit to /u/TTEH3 on Reddit.

What is a good TTFB?

Under 300 MS for Desktop and under 500 for mobile.

Contrary to what many guides online will tell you, **a TTFB higher than 300 ms on Desktop is unacceptable**. Many sites will tell you under 800 ms, which in kind words is...generous. That is a ridiculously, absurdly large TTFB and a value reserved for the lazy. For the sites I optimize, after optimization the TTFB is generally ~200 ms. I get mad at myself when I can't hit sub-300. On a good host I can hit 150-200 ms when the site is fully optimized depending on the way the site was built. When your site has a large TTFB, you can feel it. Reducing your TTFB score is critical for pagespeed performance.

Mobile TTFB

Note: Mobile TTFB will always be larger than Desktop TTFB. Under 300 MS TTFB is for Desktop scores, you want to shoot for ~400-500 MS maximum for mobile TTFB if possible. Since TTFB has 6 different components it can be a bit of a challenge to reduce TTFB time to a reasonable value, so be prepared for a bit of trial and error.

If your site is testing with a Desktop TTFB over 300 or a mobile TTFB over 500 MS, you likely have unimplemented optimization opportunities remaining.

You Do Not Need Paid Speed Testing Plans

Everything needed to optimize a site can be gleaned from free speed tests. Paid speed testing plans are completely unnecessary unless continuous monitoring is desired. Even then, there are 2 free self-hosted open source tools included at the bottom of this section for continuous monitoring.

Waterfall Charts

How to analyze a waterfall chart

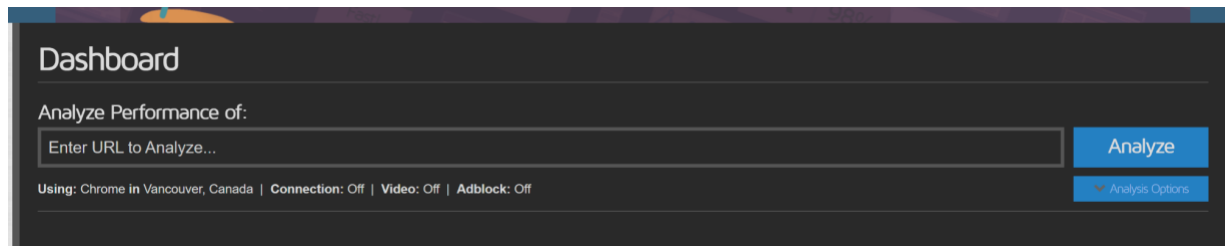
<https://www.keycdn.com/blog/waterfall-analysis>

Page Weight and reducing the amount of files that load.

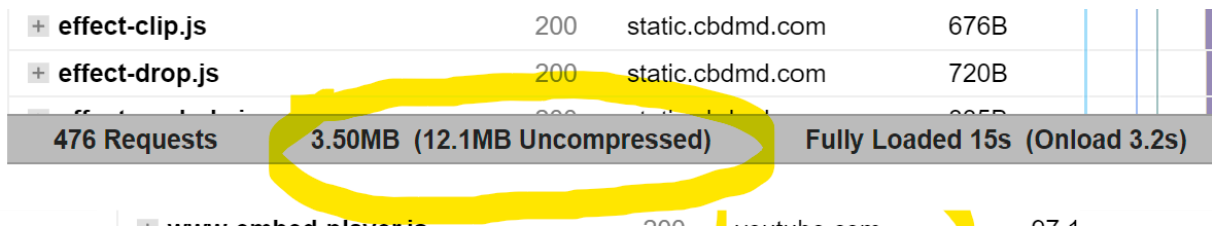
One of your primary goals when optimizing is to reduce the total weight(size in KB/MB) of a page to the absolute minimum possible.

For the following sections I am using a GTMetrix waterfall chart for the screenshots, however any waterfall chart will have the same information.

1. Input the url you want to speed test:



2. You can see the file weight of a page at the bottom of the waterfall chart on GTMetrix, another reason why it's one of the most useful tools you can use for speed tests.

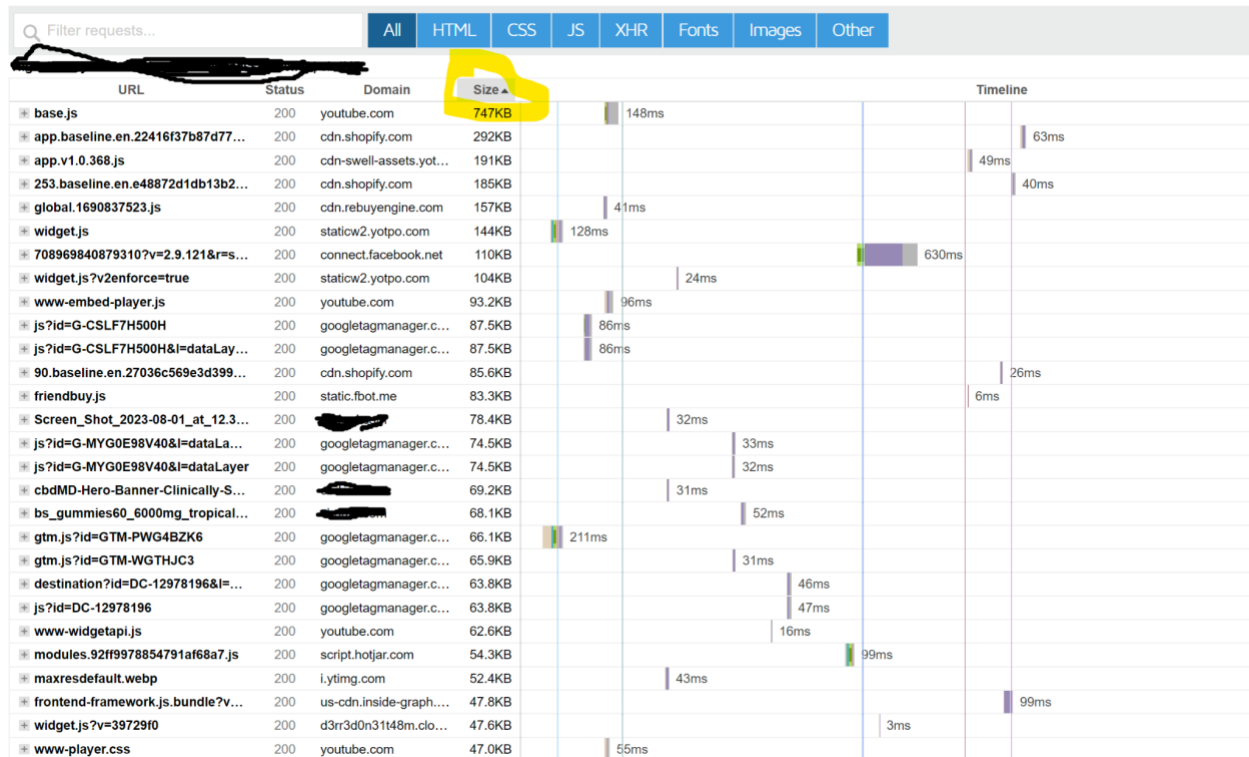


3. The other important metric you are trying to reduce is the number of requests, shown at the bottom of the GTMetrix waterfall.



Both total page weight and the number of requests need to be reduced to the minimum possible values you can achieve.

4. This is how to sort by file size:

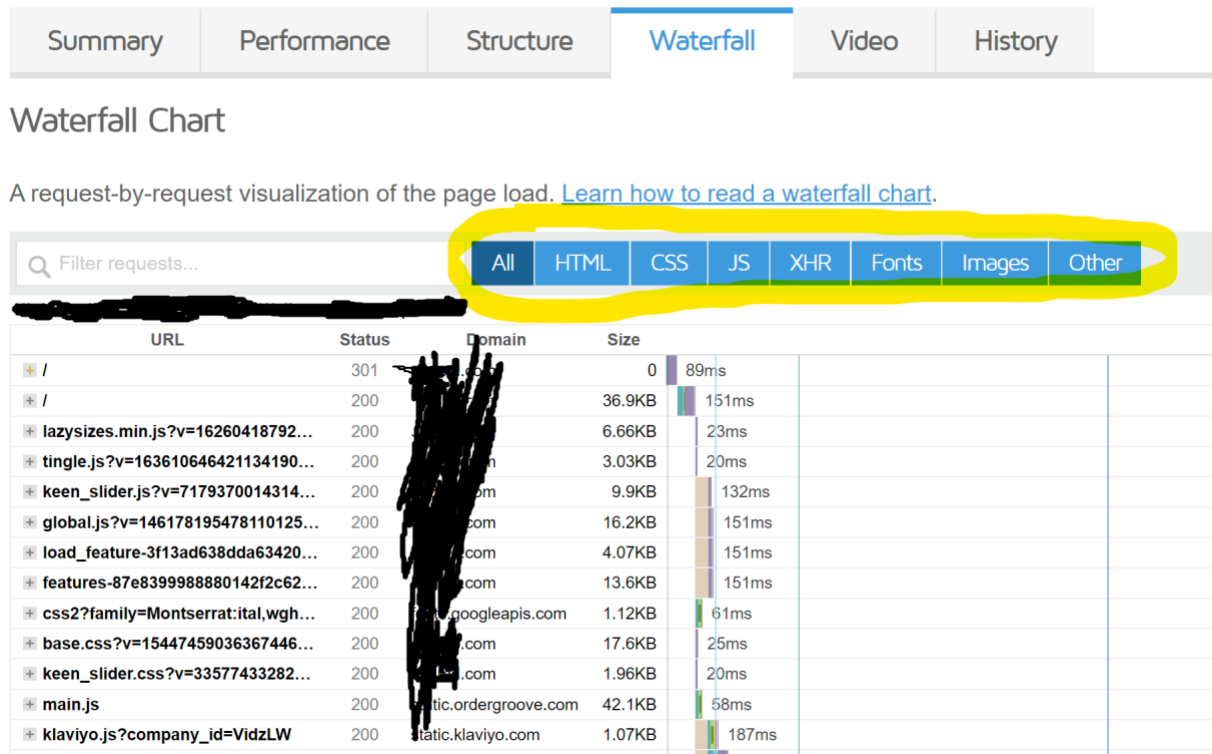


What is an ideal page weight?

An ideal page weight is 500 kb or less. My site's homepage running Elementor + Woocommerce is 168 kb before user interaction.

How to filter by File Type

This is how to filter by file type on GTMetrix. Debug Bear has similar filters:



Every additional request (even if they are incredibly small, even a .3 kb request) adds additional latency. Eliminate as many requests as you possibly can. Delaying javascript files will **completely negate** their pagespeed impact entirely, as the files will not download before user interaction. That is how my site's page weight tests at 250 kb.

The page will appear to load instantaneously, and the javascript will have no impact on the initial page load time. The user will experience no visible delay/performance impact when the delayed javascript is loaded.

How to identify which files need to be optimized

The prime targets for optimization are your Javascript and image files. CSS files can be taken care of in 1 shot with removed unused CSS (barring the need to add exclusions to the CSS removal feature of course).

Going based off the picture above, let's break it down file by file. Lazysizes.min.js cannot be delayed or deferred. That is because lazysizes is an optimization JS file that modifies image loading to improve page speed.

Tingle.js, while small (3 kb), is likely an extraneous file that does not need to load immediately. That should be delayed (and an exclusion added if needed). You want to reduce the amount of files loading as much as possible, even if they are small.

Keen_slider.js can safely be delayed. A slider requires user interaction to function, so unless the slider is set to auto-play between different photos and this feature is needed immediately after the slider loads, it can be delayed without issue. If the slider plugin is sensitive to delay and the file cannot be delayed, add an exclusion.

Global.js seems like something Klaviyo would load. If you expand the waterfall item, you can see where the file is coming from. A plugin, third party, etc. Based on that information, you can glean whether it's a good target for delaying.

Main.js and Klaviyo.min.js are both third party files. From Order Groove and Klaviyo respectively. Of the two, Order Groove is more likely to break something because it sounds like part of the e-commerce featureset. Klaviyo is for analytics and can be safely delayed. Order Groove's file would require testing to ensure it doesn't break any functionality.

I would recommend testing delaying each js file individually, one at a time just to be sure. You don't want to leave any performance on the table by not delaying them if possible. To reiterate, you want to reduce the amount of files loading on the initial page load as much as possible.

Which Speed Test To Use For Diagnosis

For Mobile, your best tool to test with will be [Debug Bear](#).

Debug Bear is superior to Pagespeed insights for Mobile scores and provides more actionable info on what you need to optimize, as you can access the waterfall chart to see load times for specific files, as well as a list of all CSS and JS files that are loading on a specific page that you are testing. You can run the test on individual URLs such as the homepage, an about page, service page or whatever you want to test on the site you are testing.

If your Mobile score is good on Debug Bear, it should be good enough for Google.

Chrome Dev Tools Network Tab

Chrome dev tools network tab can be accessed by right clicking on a page, clicking "inspect" and then clicking the network tab at the top of the panel. While the network tab in dev tools provides less detailed info than **Debug Bear**, you can access the network tab just by right clicking a page and clicking inspect. Useful for a quick analysis of a domain's request tree, but for a thorough analysis **Debug Bear** is recommended.

Mobile Page Speed Testing

Note: For mobile speed tests (the only ones that actually matter), Pagespeed Insights is the end all be all of the limbo bar, but their diagnostic utility (how you figure out what you need to optimize) is practically zero. You need to use third party tools to actually identify the issues.

Google Pagespeed Insights (Free)

<https://pagespeed.web.dev/>

You need to satisfy Google's speed tests for your SEO. Other sites may offer mobile pagespeed tests, and perhaps they have a waterfall chart for mobile, **but the end all be all for mobile tests is Google's Pagespeed Insights testing tool.**

While you are trying to give the user the best possible experience on mobile(as the majority of the site's traffic will be coming from mobile), the only company you need to satisfy for speed tests is Google **because they are the gatekeepers for SEO**. Bing refers less than **5%** of the traffic to your sites than Google does, and if your metrics satisfy Google, they will satisfy Bing. **You only care about optimizing for Google for SEO.**

Debug Bear (Free)

<https://www.debugbear.com/test/website-speed>

I use Debug Bear to diagnose my issues, but Pagespeed insights to gauge how much farther I have to go on my optimizations.

You are **always** aiming to optimize your mobile score on Pagespeed Insights. Pagespeed insights is the only metric used for Google's ranking for search results. Debug Bear is simply for diagnosis so you know what to fix.

I cannot stress this enough, the metrics Pagespeed Insights measures for mobile are the only ones that matter. Debug Bear scores only real value are as a gauge of your progress and what you need to fix.

If your mobile score improves on Debug Bear, your mobile performance scores on Pagespeed Insights will improve along with it.

Debug Bear Tools (Free)

<https://www.debugbear.com/tools>

Very useful collection of free pagespeed testing tools

Yellow Lab Tools (Free)

Online Version (Free)

<https://yellowlab.tools/>

Yellow Lab Tools has some unique metrics that can be useful in some instances to diagnose issues that may not be indicated by other scanners.

Docker Image (Free)

<https://github.com/YellowLabTools/YellowLabTools>

Docker Image for Yellow Lab Tools.

Bulk Pagespeed Speed Testing

Experte (Free)

<https://www.experte.com/pagespeed>

Bulk test multiple pages on a site at once.

Optimize for Pagespeed Insights lab data scores

Always optimize for **lab data**. Pretty much every other tutorial will tell you to focus on field data, the load times on real world user's devices. When you do a pagespeed scan, the scores that are immediately reported (regardless of the service used), are "lab data". When you improve lab test metrics, you inherently are improving the real world load time, the field data, for users. The lab data metrics are there for a reason. "Synthetic" pagespeed testing (lab data) are the only scores you can actually optimize for since that is what page speed tests generate. Real world metrics (Field Data) are collected over a 30 day period, which does not have any diagnostic utility.

Real world performance

The speed may seem artificially faster on speed tests (i.e. your scores have improved), however it may load faster than reported in the speed tests in the real world. Real world load times improvements are more important than Pagespeed Insights lab data results, but they are directly tied to the field data. Lab data improvements mean there are real world improvements, and that will be reflected when the next round of field data is collected over the next 30 days after optimizations have been implemented.

Page Speed tests lab results are mostly accurate (and satisfying Pagespeed Insights is absolutely necessary for SEO), but the real world measurement should be measured by your own perception and always gauged by your eyes. In some cases, a pagespeed test may tell you that your performance is good and rock solid, but loading it yourself and viewing the timing may actually show that it's slower (rarely).

The real world speed not matching a speed test result is very rare, but it does occur on occasion, so always check how fast the site loads on your own devices, instead of synthetic speed tests (pagespeed measurement tools).

If there is a mismatch between real world performance and your speed tests, something is wrong which you need to diagnose, and you may need to restore to a backup if you can't reverse the effects of changes you have made for any reason.

You need to improve both the real world load time **and** the pagespeed test results on pagespeed insights, but the lab data is always the best heuristic to use, and the mismatch is so unlikely that it should be considered an edge case.

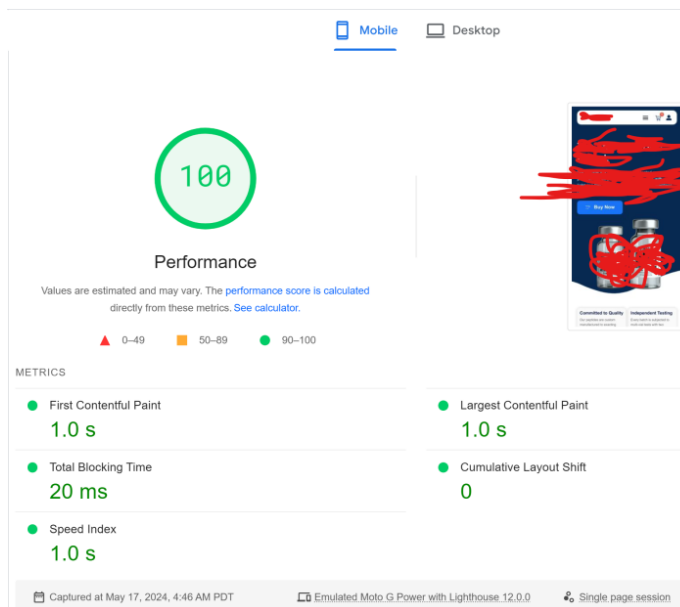
Not all speed tests are equal

Some speed testing providers will return wildly inaccurate results. Dotcom tools being one of them. To reiterate, **the only test results that actually matter are Google's. Everything else is just for analysis.**

What Is An Ideal Mobile Pagespeed Score On Pagespeed Insights?

Any site you are optimizing should ideally have a mobile score of 90+ once optimized. For the vast, *vast* majority of sites that is easily achievable. Anyone reading this guide should be able to achieve that on any normal site once the majority of the steps are implemented. Some dynamic sites require some additional special strategies which have yet to be added to the guide, but these are special cases and not the norm.

What does an ideal mobile score look like?



Don't settle for less than a score of 90! All it takes is a little effort and some trial and error. Once you implement everything in this guide, this will be easy to achieve and repeat, site after site. We're after all green.

Continuous Monitoring

Continuous Monitoring will measure your real world and lab data scores over time, which are useful metrics as the fluctuations do have diagnostic utility. These two tools are self-hosted, and are not SAAS tools/services. I recommend getting a separate VPS to host these tools so you don't cause unnecessary load on a server hosting live sites or staging sites. There are cheap VPS providers mentioned in the hosting section which are useful for this purpose.

Self-Hosted Tools

These can either be self-hosted on your local machine, or on a cheap VPS server.

Lighthouse CI (Free)

<https://github.com/GoogleChrome/lighthouse-ci>

Lighthouse CI is a suite of tools that make continuously running, saving, retrieving, and asserting against Lighthouse results as easy as possible.

Lighthouse CI requires a CI service, these are listed here, free options included:

<https://github.com/GoogleChrome/lighthouse-ci/blob/main/docs/introduction-to-ci.md>

Gitlab CI (Free)

https://docs.gitlab.com/ee/ci/quick_start/

Lighthouse CI Server (Free)

<https://github.com/GoogleChrome/lighthouse-ci/blob/main/docs/server.md>

Lighthouse CI Visual Frontend/Dashboard

SiteSpeed.io (Free)

<https://github.com/sitespeedio/sitespeed.io>

Another free Open Source continuous monitoring toolkit, visually presents more useful data than Lighthouse CI.

Available as either a Docker Image (cross-platform, including Windows support) or a NodeJS NPM package (Linux).

Competitor and Comparative Analysis

Lightest App (Free)

<https://lightest.app/>

Free competitor/comparison speed testing against multiple domains.

Debug Bear (Free)

<https://www.debugbear.com/docs/compare-pages>

Note 1: Requires a free account

Note 2: From my tests, the comparison speed tests statistics are mismatched, and have poorer scores than a regular Debug Bear speed test. Regular Debug Bear speed tests are more accurate.

WebPageTest (Free)

<https://www.webpagetest.org/video/>

Pretty janky interface, but offers comparative speed testing against multiple urls.

Inferences

Waterfall Chart Analysis

Performance Optimization requires inferences. You will never have every optimization opportunity identified by a Pagespeed Performance Test, regardless of if you are using the best in class performance test or not.

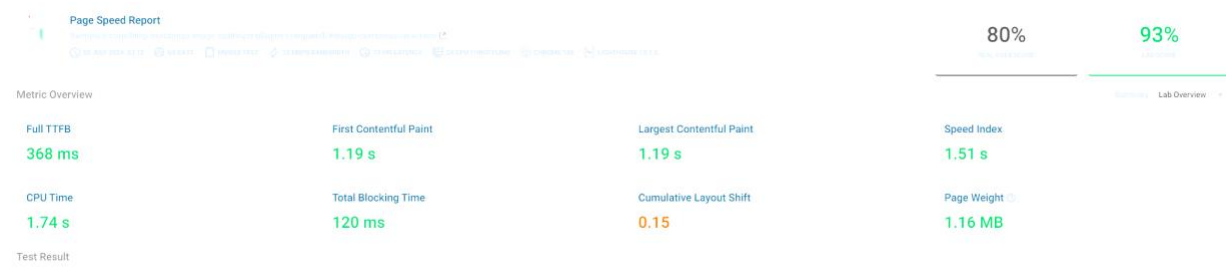
There is simply no way for an external tool to identify backend issues, and none of them will currently identify all frontend issues. Therefore, identifying issues based on information that is not explicitly identified in Pagespeed reports is required. This requires you to make these conclusions via interpretation of the Pagespeed Test Request Waterfall Chart.

The example below contains some of the inferences that can be made by reviewing a waterfall chart.

Even Well Optimized Sites May Have Lots Of Opportunities For Improvement

Themelsle Case Study

Some sites are sitting at the cusp of perfect performance, but have some low hanging fruit optimizations that would rocket their scores to 90-100 with just some simple tweaks. This is a great example of a site that is already ~85-90% of the way there to being fully optimized based on their pagespeed report.



TYPE	URL	PRIORITY	DURATION	SIZE	EXPAND	WATERFALL
blog/	themeisle.com		597 ms	86.0 KB		
fa-regular-400.woff2	themeisle.com	PRELOAD	180 ms	13.4 KB		
fa-solid-900.woff2	themeisle.com	PRELOAD	266 ms	77.7 KB		
themeisle/img/logo.png	themeisle.com	FP-HIGH PRELOAD	194 ms	9.05 KB		
joe-warnimont-headshot.jpeg	themeisle.com		178 ms	44.8 KB		
Best-WordPress-image-Optimizer-1024x512.p...	themeisle.com	FP-HIGH	220 ms	62.0 KB		
Long-coated-Dog-Grayscale-1024x683.jpg	themeisle.com	Low	330 ms	123 KB		
Mountains-near-water-painting-1024x736.jpg	themeisle.com	Low	380 ms	103 KB		
wp-mod-original.png	themeisle.com		218 ms	3.00 KB		
img/twitter.svg	themeisle.com	Low	210 ms	887 B		
/?primer=7c**&fvd=n7&v=3	use.typekit.net	PRELOAD	692 ms	40.7 KB		
/?primer=7c**&fvd=n4&v=3	use.typekit.net	PRELOAD	786 ms	39.5 KB		
banner-772x250.png?rev=2624...	ps.w.org	Low	725 ms	51.5 KB		
banner-772x250.png?rev=1730...	ps.w.org	Low	784 ms	42.3 KB		
banner-772x250.png?rev=2759...	ps.w.org	Low	813 ms	31.9 KB		
banner-772x250.jpg?rev=1582...	ps.w.org	Low	732 ms	21.2 KB		
9 Requests Hidden By Filters → Show All						

Request	
Method	GET
URL	https://themeisle.com/blog/wp-content/uploads/2021/09/joe-warnimont-headshot.jpeg

Render-blocking	Unknown
Request time	22 July 2024 at 23:12:18
Start time	510 ms
Protocol	h2 (TLS 1.3)
Connection	ID: 175 (Reused)
Priority	High (Initially: Low, changed after 1.09 s)
Request Index	5

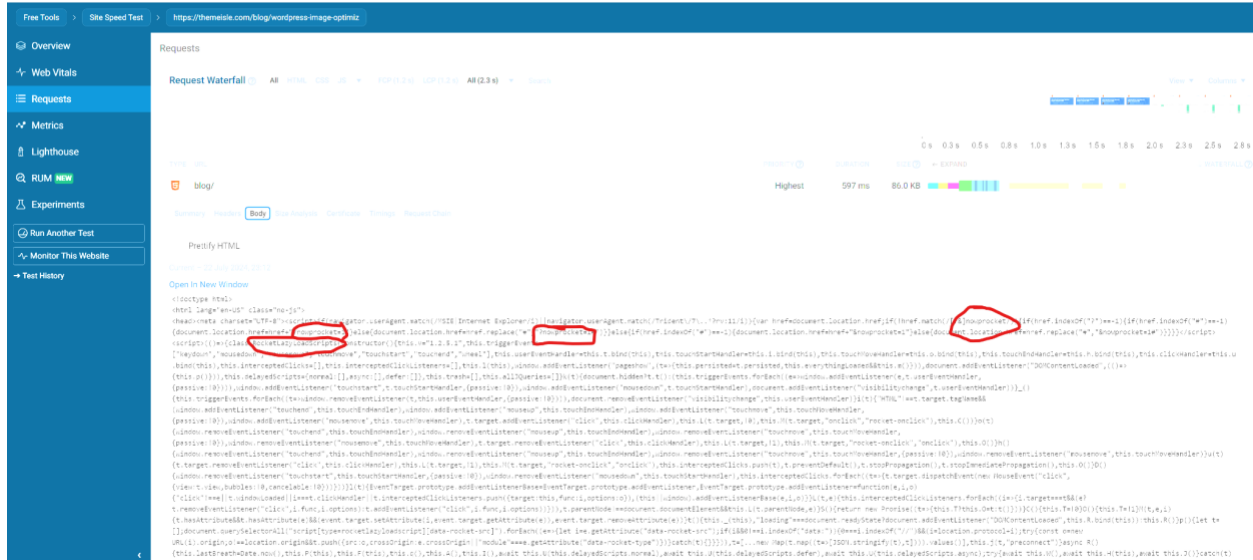
ⓘ Don't preload too many fonts

Only preload the 2-3 fonts used for most of the above-the-fold content to avoid font requests outcompeting more critical resources. Fonts that are not preloaded will be loaded later on when the browser naturally realizes they are needed.

TYPE	URL	PRIORITY	DURATION	SIZE
font	fa-regular-400.woff2 themeisle.com	PRELOAD	180 ms	13.4 KB
font	fa-solid-900.woff2 themeisle.com	PRELOAD	266 ms	77.7 KB
font	[?primer=7c**&vd=r7&n=3 use.typekit.net	PRELOAD	692 ms	40.7 KB
font	[?primer=7c**&vd=r3&n=3 use.typekit.net	PRELOAD	820 ms	40.0 KB
font	[?primer=7c**&vd=r4&n=3 use.typekit.net	PRELOAD	786 ms	39.5 KB

ⓘ Avoid render-blocking CSS @import chains

F fa-solid-900.woff2 themeisle.com	
Summary Headers Body Certificate Timings Request Chain	
Response Headers ⓘ	
HEADER	VALUE
Status	200 OK
accept-ranges	bytes
age	490623
alt-svc	h3=":443"; ma=86400
cache-control	max-age=31536000
cf-cache-status	HIT
cf-ray	8a79849fec4a6763-ATL
content-length	79444
content-type ⓘ	font/woff2
date	Tue, 23 Jul 2024 06:12:18 GMT
expires	Thu, 17 Jul 2025 13:28:50 GMT
last-modified	Wed, 08 May 2024 07:26:24 GMT
server	cloudflare



There are many, many conclusions that can be drawn from the above waterfall chart. Only 16 requests are displayed in the top screenshot, some sites can have hundred+ requests. None of these will be explicitly warned about in a Pagespeed report, you must make these conclusions through your own analysis of a report.

Icons

Font Awesome

The **Font Awesome Icon Library** is in use, and two separate woff2 font files for font awesome are loading, with weights of 400 and 900 (note the values in the file name). These files are preloaded, which forces them to the top of the request tree.

Their file sizes are 13.4 kb and 77.7 kb respectively, so the site is loading a total of 91.4 kb of just icon fonts. This is a big problem, and one that will have a sizable impact on a speed test. These two requests need to be eliminated if possible, and the icons need to be replaced with individual small svg files. Even on pages with 1 or 2 icons, many plugins and themes call the entire font library with the entirety of icons within (possibly thousands of icons), completely unused.

Sometimes font awesome has a toggle in plugins and themes, sometimes not.

And this is just **one conclusion** that can be drawn from the waterfall chart.

Individual Twitter SVG

The font awesome library is in use, and yet an individual Twitter icon SVG is also being loaded. This is completely unnecessary as the Font Awesome libraries already have a Twitter icon included. One of two things should be done, replace all icons with individual SVGs (best practice), or use the Font Awesome library's Twitter Icon (Preferable when Font Awesome cannot be removed) to eliminate an unnecessary request.

Unoptimized Images

Images Are Not Lazy Loaded

9 images are loading in the request tree, which right off the bat is the most noticeable issue. Firstly, only one of these images is actually in the viewport (the author photo). None of the photos that are out of the viewport are lazy loaded, meaning tons of unnecessary page weight is being downloaded on the initial render, which negatively impacts all metrics, but has the most significant impact on LCP and CLS scores.

Images Can Be Further Compressed

Secondly, many of the images are large and can be further compressed, preferably manually if possible.

Image Sizes

There are multiple image sizes that can be identified from the 9 image requests filenames: 772x250, 1024x683, 1024x512, and 1024x736.

This indicates a few things. The image sizes are wildly inconsistent, there are 4 separate image sizes being used on the page. The smaller the image size, the lower its file weight will be, reducing its performance impact.

It may be possible to resize some of these images while maintaining visual quality on mobile (specific resolutions/breakpoints can be targeted with SRC set/image resizing plugins) which will reduce their file size further.

There is an explicit warning in the report about image sizes, but even more information can be gleaned from just inspecting the filenames.

File Formats

Two File Formats for the 9 images that are loading can be identified in the waterfall, jpg and png. The file size of the images can potentially be further reduced beyond just resizing and compression by serving “next-gen” formats like AVIF and WebP to newer browsers while still serving the jpg and png files to unsupported older browsers as a fallback.

Bloated HTML Document

One issue with performance tests is it's threshold for the large HTML document warning is set too high. The appropriate threshold for the weight of a bloated HTML document should be 50 kb, anything greater than that should be considered large and has a noticeably detrimental effect on performance.

We can see based on the top request for the page “blog” (the root html of the page) that the HTML weight is 86.0 kb, which exceeds the 50 kb maximum threshold target. The HTML

document weight of a site should be reduced as much as possible for the greatest performance improvements.

Cache Plugin Used

WP Rocket

Because CSS files have been inlined using a remove unused CSS feature and all javascript has been delayed (likely) or inlined (unlikely), because there are no javascript files loading in the request tree, the optimization plugin cannot be identified by analyzing the URLs in the request tree.

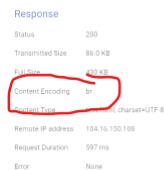
Instead, the HTML body must be viewed. As soon as the HTML body is opened, it is very clear that **WP Rocket** is in use since it is referenced multiple times ("**nowprocket**").

Although WP Rocket is not recommended as it's generally possible to obtain better results with Flyingpress, its implementation is functioning perfectly and no changes should be made unless problems arise when performing further optimizations. WP Rocket with all of its flaws is still one of the most performant caching plugins.

CDN

The site is using Cloudflare as the CDN, and there is a cache hit (meaning the file is served from Cache correctly) for some files (served from Cloudflare cache).

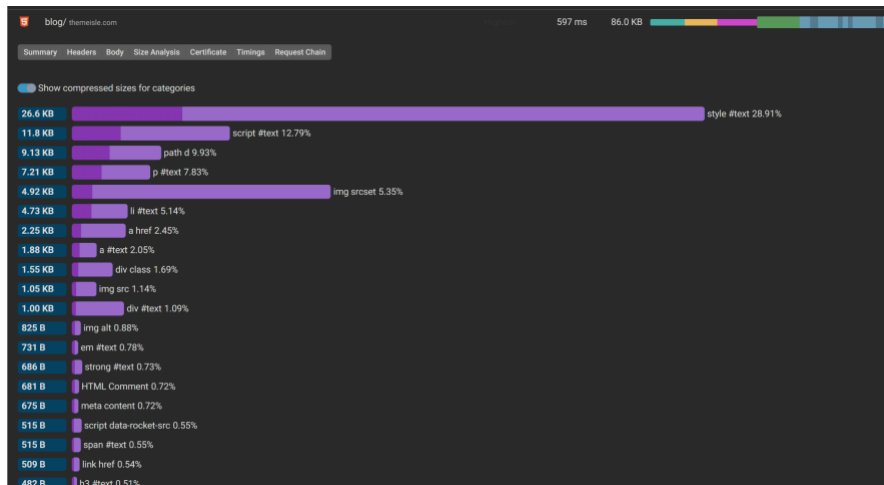
Brotli



Response	
Status	200
Transmitted Size	85.0 KB
Full Size	440.4 KB
Content-Encoding	br
Content-Type	text/javascript
Remote IP address	104.16.130.100
Request Duration	587 ms
Error	None

Brotli is in use.

Document Bloat Components



Debug Bear breaks down the components of the HTML document and displays their size. 28.91% of the HTML document weight comes from inlined CSS (style), 12.79% comes from inlined javascript (script), 9.93% is from inlined SVGs (path).

The inlined CSS styles have the greatest performance impact, 26.6 KB of inlined CSS styles is excessive.

11.1 KB of javascript also has a significant performance impact.

Inlining too many SVGs can also have a negative impact. Inlining should be used sparingly and only for critical resources.

You'll need to examine the sub-contents of these items and determine the source, then you can proceed to optimize it. This requires manually inspecting the HTML tree in Debug Bear.

External Third-Party Calls

There are 9 requests to files from the host domain.

At the bottom of the waterfall there are 6 requests to third party domains. 2 to use.typekit.net (fonts), and 4 to ps.w.org (images). The typekit calls are unavoidable and must be made unless the fonts are locally hostable, so if they are required, they should be set to preconnect to the third party domain (use.typekit.net), as well as preloading the resource. The preload attribute is set appropriately for the typekit requests, however the preconnect attribute is not and should be added.

The 4 calls to ps.w.org are images that should be replaced with locally hosted copies.

Quantity Of Requests Is Low

Very few resources are being called in the request tree (23 resources + the root HTML, 14 resources displayed in the non-expanded tree screenshotted). This is a good indicator of a well optimized site.

Anything under 40 is in the sweet spot.

Resource Fetch Priority

Two files have the Fetch Priority attribute set to “high”, logo.png and best-wordpress-image-optimizer.xxx.png. There are two more high priority requests to the use.typekit.net domain. All the other requests besides the root HTML document’s fetch priority is set to “low”. This fetch priority “low” attribute may be retained even when the file is set to preload, this does not negatively affect the preload functionality.

Too Many Fonts Are Preloaded

Too many font files are preloaded. The number of preloaded resources should be kept to a minimum and only used for critical resources.

Unused CSS Has Been Removed

There are no CSS files being loaded in the request tree. This is a good thing, and indicates that a Remove Unused CSS function has been activated in an optimization plugin, or that the needed code has been manually inlined. All unused CSS has been removed, and critical CSS inlined into the HTML document. CSS can be safely checked off the list and marked as optimized.

No Javascript Requests

There are zero requests for any javascript files. This is a very good thing as this indicates the site has delayed all of their javascript (in their case all of it could be safely delayed) or that it is a static website. There is zero javascript code that is being downloaded or executed during the initial render, completely negating any performance impact that javascript code would have on the initial page render.

Images Are Being Served Over HTTP/2

HTTP/3 is enabled on themeisle.com, however, some resources are still being served over HTTP/2. The image files are still being served over HTTP/2, which leaves some performance on the table.

Avoids Redirect Chains

There is no intermediary HTTP > HTTPS > [HTTPS://WWW](https://www.themeisle.com) redirect.

LCP (Logo) Is Preloaded

The LCP image is preloaded correctly.

Layout Shift

The CLS score (Layout Shift) indicates that height and width attributes have not been applied to images.

Low TTFB Indicates An Optimized, Low Load Database

Low TTFB indicates that the Database backend has been well optimized and is experiencing low load. High TTFB is usually indicative of an overwhelmed database, a VPS/Hosting with low RAM or weak CPU, or a poorly configured mysql implementation.

Low CPU Time Indicates Optimized Javascript And An Optimized Database

CPU evaluation time largely comes from javascript. Since all Critical Javascript has been inlined (no JS files loading in the request tree), the javascript inlined into the HTML document is minimal and well optimized. It is likely the database is well optimized, as a poorly optimized database can have significant performance impacts on both CPU time and TTFB.

No Iframes In Use

There are no third party calls for Iframes.

TLS 1.3 Is In Use

TLS 1.3 (SSL) is enabled on the domain, which is currently the newest and most performant version of TLS available.

Over Half Of The Current Page Weight Can Be Removed From The Initial Render

Simply implementing a functional lazy load solution would automatically halve the page weight. After implementing all of the above suggestions, the page weight could likely be reduced ~3x from 1.16 MB to ~400-450 kb. A 700 kb reduction, resulting in significant performance improvements.

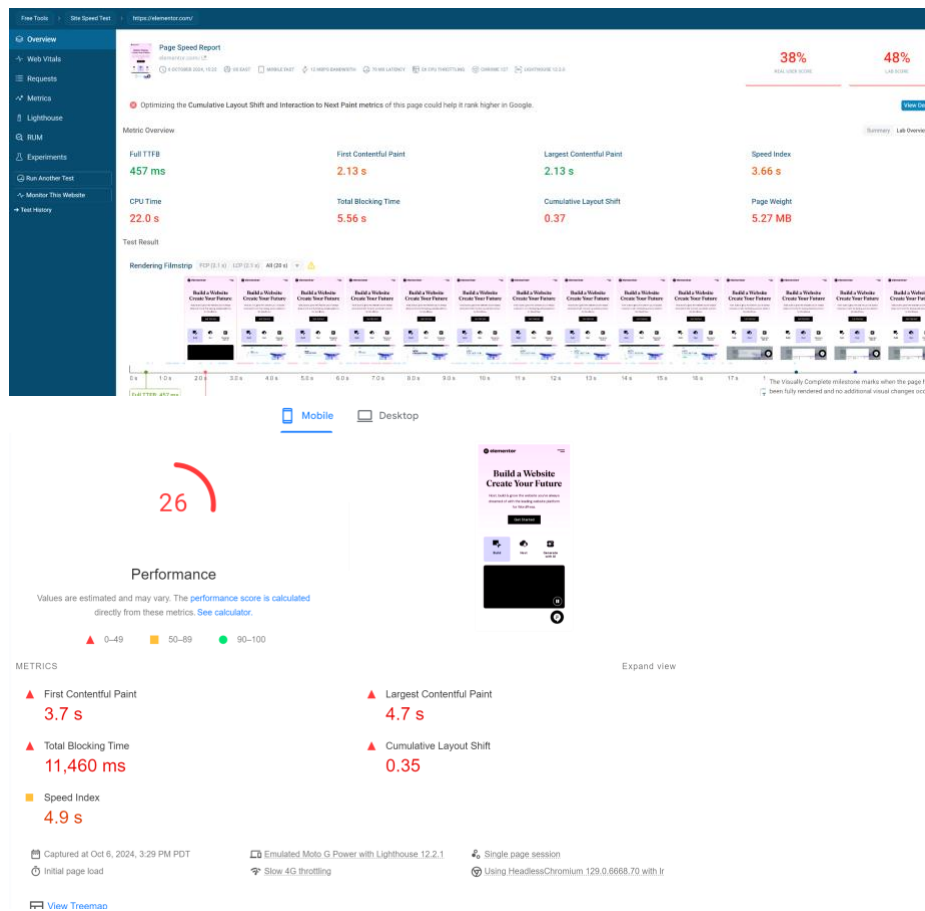
Case Study 2: Elementor.com

Comic Relief

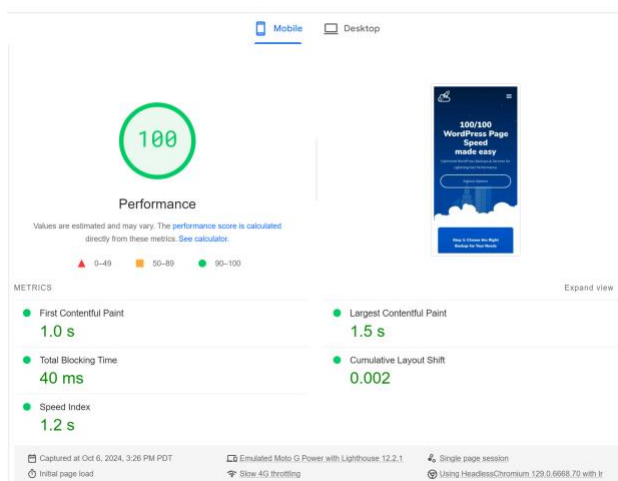
You've gotten this far into the guide, who's ready for some comic relief?

<https://www.debugbear.com/test/website-speed/jl9XSl2N/overview>

Here's some screenshots of pagespeed reports for Elementor.com itself.



And here's the Pagespeed report for a personally developed and optimized Pagespeed Optimization Services site (currently not live as it's still in development!) built with Elementor:



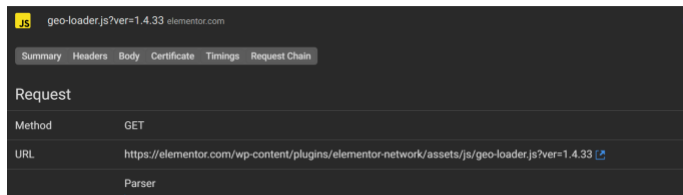
Strap in, this is going to be a long one. Going to do this in chunks, too many screenshots to do all at once.

	URL	Category	Severity	Score	Size	Time
JS	jquery.min.js?ver=3.7.1 elementor.com	BLOCKING		321 ms	29.9 KB	
JS	j.php?ar=740558&u=https%3A%2F%2Felementor.com%2F&f=1&m=1.5 dev.willabloblogspot.com	ACTIVE - PPARISH		102 ms	11.4 KB	
JS	geo-loader.js?ver=1.4.33 elementor.com	PASSER-BLOCKING	Medium	685 ms	1.27 KB	
JS	app.min.js?ver=1.1.6 elementor.com	PASSER-BLOCKING	Medium	693 ms	1.28 KB	
JS	js.cookie.min.js?ver=1.6 elementor.com	PASSER-BLOCKING	Medium	755 ms	1.00 KB	
JS	jquery.sticky.min.js?ver=3.23.3 elementor.com	PASSER-BLOCKING	Medium	688 ms	1.84 KB	
JS	jquery.smartmenus.min.js?ver=1.2.1 elementor.com	PASSER-BLOCKING	Medium	724 ms	7.22 KB	
JS	jquery.numerator.min.js?ver=0.2.1 elementor.com	PASSER-BLOCKING	Medium	696 ms	1.08 KB	
JS	elementor-forms-widget.js?ver=0.0.1 elementor.com	PASSER-BLOCKING	Medium	710 ms	1.03 KB	
JS	webpack.runtime.min.js?ver=3.23.4 elementor.com	PASSER-BLOCKING	Medium	713 ms	2.40 KB	
JS	frontend-modules.min.js?ver=3.23.4 elementor.com	PASSER-BLOCKING	Medium	706 ms	19.7 KB	
JS	waypoints.min.js?ver=4.0.2 elementor.com	PASSER-BLOCKING	Medium	742 ms	3.16 KB	
JS	core.min.js?ver=1.13.2 elementor.com	PASSER-BLOCKING	Medium	714 ms	7.24 KB	
JS	frontend.min.js?ver=3.23.4 elementor.com	PASSER-BLOCKING	Medium	728 ms	12.2 KB	
JS	newsletter-enrollment.min.js?ver=1.4.33 elementor.com	PASSER-BLOCKING	Medium	745 ms	1.57 KB	
JS	underscore.min.js?ver=1.13.4 elementor.com	PASSER-BLOCKING	Medium	719 ms	7.39 KB	
JS	wp-ull.min.js?ver=6.5.5 elementor.com	PASSER-BLOCKING	Medium	757 ms	1.05 KB	
JS	imagesloaded.min.js?ver=5.0.0 elementor.com	PASSER-BLOCKING	Medium	705 ms	2.10 KB	
JS	webpack-pro.runtime.min.js?ver=3.23.3 elementor.com	PASSER-BLOCKING	Medium	750 ms	2.97 KB	
JS	wp-polyfill-ieee754.min.js?ver=3.1.2 elementor.com	PASSER-BLOCKING	Medium	725 ms	2.75 KB	
JS	cloudflare-static/mirage2.min.js sjs.cloudflare.com	PASSER-BLOCKING	Medium	855 ms	12.2 KB	

<https://vwo.com/>

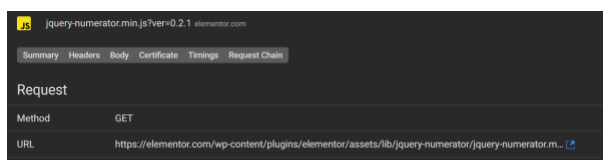
From the second item in the waterfall it's clear that they are using Visual Website Optimizer for customer analytics. To optimize VWO, the javascript should be delayed with a delay javascript feature in an optimization plugin.

Geo-Loader



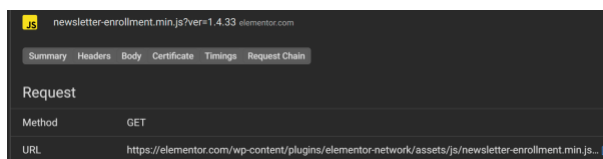
Elementor.com is using a self-developed plugin for Geolocation called “geo-loader”. We can infer that they are using this to identify where customers are located, and this data is fed into either an external analytics platform, or a custom developed one.

Jquery Numerator



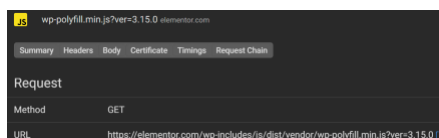
Jquery Numerator is loading, which is used to animate numbers. This js file can usually be delayed to negate its performance impact unless the animated element is above the fold and the animation is needed immediately without user interaction.

Newsletter Enrollment



They are using a custom developed newsletter enrollment plugin. This can almost certainly be delayed to negate the performance impact with no functionality affected as this is unneeded until the user interacts with the form linked to the functionality.

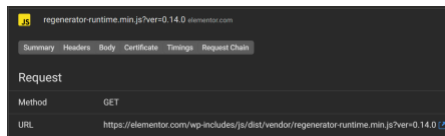
WP Polyfill



The WP Core file wp-polyfill.min.js is loading, which can safely be unloaded with Asset Cleanup/Perfmatters/a selective asset unloading plugin, or delayed. The tell that this is a file loaded from WP Core (Wordpress core) is that the file path contains “wp-includes”, which is where all WP Core files are stored.

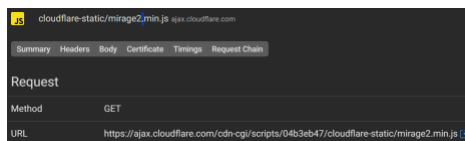
Polyfill is for fallback rendering for old unsupported browsers (internet explorer), which is unnecessary.

Regenerator Runtime



Another WP Core file which is unneeded in most cases. Elementor does not need this file for any functionality, and can be safely delayed or disabled. A few plugins depend on this file (rarely), so test your site after disabling/delaying it, but in the vast majority of cases it should be safely removable.

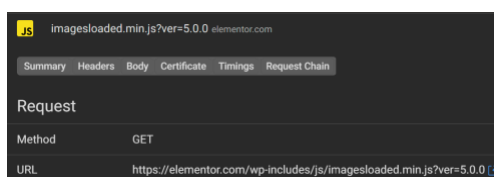
Cloudflare Mirage



There are a few things we can glean from this request. First is that Elementor.com is using Cloudflare as their CDN. Second is that it is using the [Mirage](#) Image Optimization feature. I recommend avoiding this as image optimization should be done locally on your webserver, and a CDNs feature like this is a band-aid, not an actual solution. If Mirage is truly desired, it should be used as a supplement *after* images have already been optimized manually or with an image optimization plugin. **It is not a replacement for image optimization.**

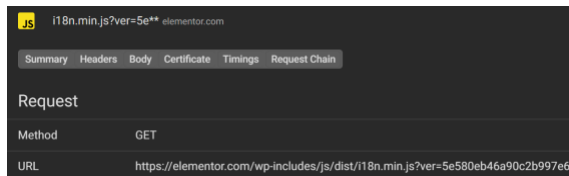
Elementor.com has made the mistake of relying entirely on Mirage for image optimization, which is why they have a lot of large, bloated images on the page.

Images Loaded JS



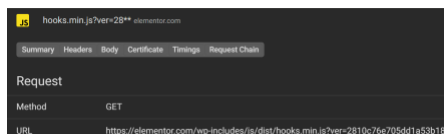
The imagesloaded javascript file is another WP Core javascript file which can usually be safely delayed. It is usually loaded by default. This file is sometimes unneeded in the initial render even when images are in the viewport/above the fold when the page loads. Sliders can potentially depend on this file however, so make sure to test delaying it or even unloading it on a staging site first before pushing the change to production.

i18n



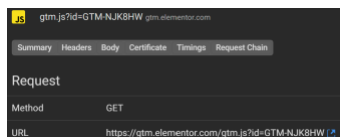
Another WP Core file which is loaded by default on every Wordpress site. This file is used for translation functionality. If your site is not multi-lingual, this file is extraneous and can be safely unloaded or delayed. **You will get an error in your browser console if this is unloaded or delayed**, but this error is ignorable and is not detrimental to performance and does not harm functionality.

Hooks.min.js



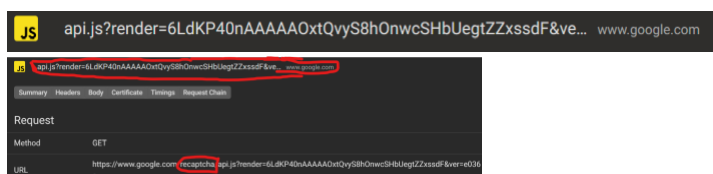
Another WP Core file which is loaded by default. Many plugins and themes depend on this js file for functionality, but most of them do not require it to load immediately, which means it can safely be delayed in most cases. I do not recommend unloading this file, and recommend delaying it instead.

Google Tag Manager



Elementor is using Google Tag Manager to implement some analytics tracking. This file is delayable which will mitigate its performance impact, but be aware delaying Tag Manager will fuzz your analytics a little bit, especially bounce rate statistics. This is a worthwhile tradeoff, because the real goal is increased traffic, conversions and sales. Data for data's sake ignores the actual goals of a website.

Captcha

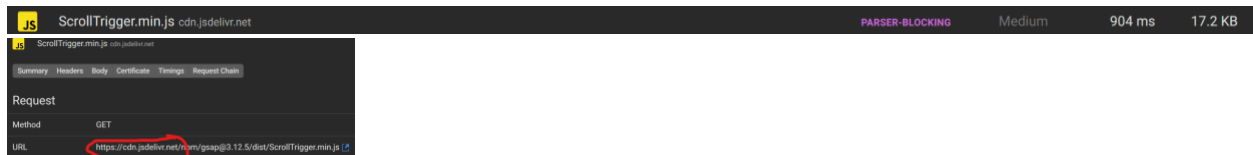


Without expanding this request it is not immediately obvious that this is for a captcha, even though the first screenshot references google.com on the right side. However, once expanded

we can see the word “recaptcha” in the url, which is a direct indicator that a captcha from Google is in use.

Captchas are extremely performance intensive, and add quite a bit of page weight. Captchas should be avoided entirely when possible and a different solution should be implemented. Since the site is already using Cloudflare, if a captcha is truly needed, Cloudflare’s captcha should be used instead. There are alternative solutions to captchas for forms and bots listed further down in the guide.

ScrollTrigger.min.js



We can make a few inferences based on this request. One is that the functionality is related to actions triggered by scrolling. Second is that this is a third party request to an external domain, in this case to a CDN.

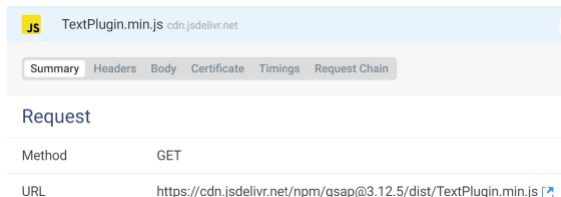
This javascript file is triply problematic. It adds 17.2 kb of page weight, which is already substantial for a javascript file and can cause a significant amount of additional processing time, even with the relatively small file weight.

Secondly, because it makes an external request to a separate domain, it introduces round trip network latency because of the HTTP request.

Third, all javascript files are parser blocking, which means the Blocking Time metric (also sometimes referred to as a TBT on some speed tests) is going to increase, which also hurts pagespeed scores.

This file should be removed entirely if possible, and if it is needed it should be delayed if doing so does not break the intended functionality.

TextPlugin.min.js



This javascript file is something related to text, potentially an animation or custom font insertion. This plugin is potentially safe to delay, but delaying it needs to be tested on staging. Additionally, this is served from a third party CDN domain, which has a negative impact on performance.

gsap.min.js

JS	dist/gsap.min.js	cdn.jsdelivr.net	PA
Summary Headers Body Certificate Timings Request Chain			
Request			
Method	GET		
URL	https://cdn.jsdelivr.net/npm/gsap@3.12.5/dist/gsap.min.js		

This is an animation related javascript file, delivered by a third party CDN. It is not served from the local domain (elementor.com). The third party HTTP call is poor practice, and is negatively affecting the site's performance.

If the animation is below the fold this should be safe to delay. If it is above the fold and delayed, the animation may not load at all, or won't play until the page has been interacted with in some way.

IvarHeadline-Bold.woff2

F	IvarHeadline-Bold.woff2	elementor.com	Highest	352 ms	34.0 KB
Summary Headers Body Certificate Timings Request Chain					
Request			Response		
Method	GET		Status	200	
URL	https://elementor.com/wp-content/uploads/2021/10/IvarHeadline-Bold.woff2		Transmitted Size	34.0 KB	
Parser	1 ms, 59.406 ms		Full Size	33.7 KB	

Ivar Headline is a custom font in WOFF2 format, and elementor.com is using the bold variant of the font. WOFF2 is best practice for font files as they are pre-compressed which reduces the font's file size.

K6z9mXg.woff2

F	IvarHeadline-Bold.woff2	elementor.com
F	**K6z9mXg.woff2	fonts.gstatic.com
Summary Headers Body Certificate Timings Request Chain		
Request		
Method	GET	
URL	https://fonts.gstatic.com/d/ansans/v15/rP2Yp2ywwg089Uit5-g4vIH9VoD8Cmc-000-k6z9mXg.woff2	

The actual font used is not discernable via the file name. We can see here that elementor.com is using a mix of locally hosted fonts as well Google Fonts called from the Google CDN from fonts.gstatic.com. All fonts should be locally hosted, either via manual upload or an automatic plugin solution like Yabe Webfont or OMGF (covered further down in the guide).

WebM Video Files

07/01-Build.webm	elementor.com	Low	1.00 s	498 KB
07/02-Host.webm	elementor.com	Low	1.61 s	724 KB
2024/07/03-AI.webm	elementor.com	Low	1.76 s	1.28 MB

These are the most problematic files on Elementor's website. In total, these 3 videos are 2.5 MB in size, which has a substantial negative impact on pagespeed performance. These files do not

appear above the fold on Desktop, and only one of them appears above the fold on mobile. This means on Desktop all 3 of these WebM files should be lazy loaded, and 2 of them should be lazy loaded on mobile. Elementor's site does not have any video lazy loading solution enabled.

The usage of the WebM format is good as the format compresses the image files and generally has better performance than other formats, however these videos can be manually compressed further to reduce the file size beyond just converting them to WebM (the video optimization section is further down).

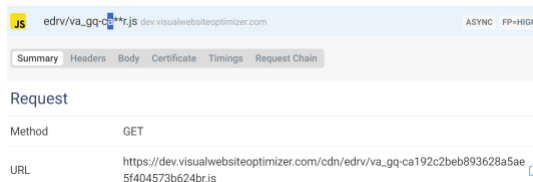
Visual Website Optimizer GIF



v.gif?cd=0&a=740558&d=elementor.com&u=D26471D462D2FFFF704E7D4424.	dev.visualwebsiteoptim...	Low	293 ms	144 B
---	---------------------------	-----	--------	-------

Visual Website Optimizer (the analytics tool mentioned further up in this case study) is loading a gif file from their domain, which is a third party HTTP call. Even though the file size is negligible (144 bytes), the external call will still have a noticeable negative performance impact.

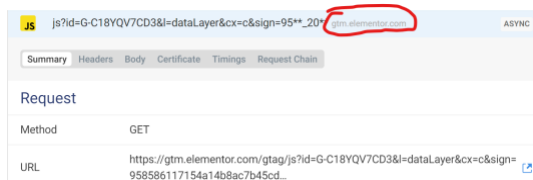
Visual Website Optimizer EDRV Script



JS	edrv/va_gq-ca192c2beb893628a5ae5f404573b624br.js	dev.visualwebsiteoptimizer.com	ASYNC	FP-HIGH
Summary Headers Body Certificate Timings Request Chain				
Request				
Method	GET			
URL	https://dev.visualwebsiteoptimizer.com/cdn/edrv/va_gq-ca192c2beb893628a5ae5f404573b624br.js			

Another external call for a script served from Visual Website Optimizer's CDN. This should be delayed to negate its performance impact.

Google Tag Manager



JS	js?id=G-C18YQV7CD3&l=dataLayer&cx=c&sign=95**_20*	gtm.elementor.com	ASYNC
Summary Headers Body Certificate Timings Request Chain			
Request			
Method	GET		
URL	https://gtm.elementor.com/gtag/js?id=G-C18YQV7CD3&l=dataLayer&cx=c&sign=958586117154a14b8ac7b45cd...		

Elementor's website is using an interesting tag manager implementation. Their tag manager script is being served locally (best practice), but instead of from the root local domain it is being served from a subdomain (gtm.elementor.com). The choice of serving it from a subdomain could have multiple reasons, to understand the logic behind the decision would require a peek into their backend configuration.

CookieLaw



JS	otSDKStub.js?did=31*	cdn.cookielaw.org	ASYNC	LOW	441 ms	7.61 KB
----	----------------------	-------------------	-------	-----	--------	---------

Elementor's site cookie banner is implemented with a service called "Cookie Law". This script is not served from the local domain, and is instead called from Cookie Law's CDN. It's a relatively small javascript file. The banner should be delayed if it's acceptable to not have it display until

user interaction (fine in most cases unless the site owner requests the file be excluded so it appears immediately).

Reddit Pixel



This is rare to encounter as not many sites use it, but Elementor.com is using a Reddit Analytics tracking pixel served from the redditstatic cdn domain. This file should be delayed to eliminate its performance impact, but like any other analytics script, delaying it will fuzz the bounce rate metrics.

Beacon.min.js



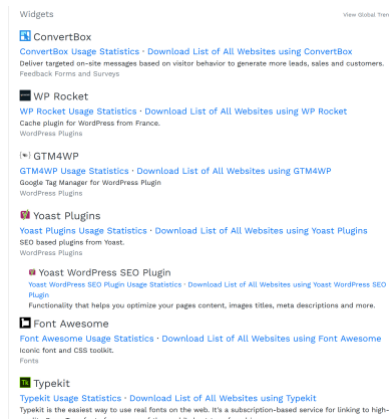
The beacon.min.js file is an analytics file used by Cloudflare. Unless Cloudflare Analytics are needed for your site, this should be disabled. As the file is loaded via the Cloudflare CDN itself, this cannot be delayed by a Wordpress javascript delay plugin as this is served at the network layer, and not via a plugin on the site. The only way to mitigate the performance impact is to disable Cloudflare's beacon analytics entirely.

To Be Continued

There are another ~110 files left in the waterfall for elementor.com left to add to this case study. I'll be continuing on with the analysis soon.

Beyond Waterfall Charts

BuiltWith



Some plugins and services (analytics for instance) cannot be identified from the waterfall tree. To identify those plugins, BuiltWith is a fantastic tool which allows plugin identification.

BuiltWith can identify common plugins explicitly. Here we can see Yoast, ConvertBox, Google Tag Manager for Wordpress and WP Rocket are directly identified. TypeKit is also in use.

Common Misconceptions

Pagespeed Lab Scores Don't Matter

A common misconception is that Google Pagespeed Insights lab scores do not matter. They are the metrics you are optimizing for, because any speed test can only measure “synthetic” lab data which is simulated under controlled conditions. They use a simulated version of the lowest common denominator phone with weak hardware, on a bandwidth limited connection. The tests are meant to simulate an average “worst case scenario” when a user loads a site.

These scores are very accurate within that context. Improvements in your lab scores via optimization will directly translate into improvements in real world performance.

It's Impossible To Create A Performant Site With A Page Builder

A common refrain is that you must avoid page builders entirely because they are bloated and slow. This is only a half truth. Any pagebuilder will be slow without optimization. Oxygen and Bricks are relatively well optimized out of the box, but if you put large images above the fold, your LCP and speed index are still going to score poorly. Every page builder can benefit from caching, database caching, css optimization, etc. Bricks and Oxygen included.

Once optimized, a well or even a poorly coded builder can easily achieve scores over 90 on a Mobile Pagespeed Insights test.

Server Level Page Caching Is A Replacement For Wordpress Caching Plugins

This is a very common misconception, and incorrect. Wordpress level html page caching with plugins is complementary to server level page html caching, and they should be used simultaneously.

Workflow and Interaction

Initial Request Handling

When a user requests a page for the first time, the request goes through the web server (e.g., NGINX, Apache). If the server-level cache (e.g., Varnish, NGINX FastCGI) doesn't have a cached version of the page, it forwards the request to the application (WordPress).

WordPress-Level Page Caching

The WordPress caching plugin (e.g., Flyingpress, W3 Total Cache) checks if it has a cached static HTML version of the requested page.

If a cached version exists, it serves the cached static HTML content. If not, WordPress dynamically generates the page, and the caching plugin saves a static HTML version for future requests.

Server-Level Caching

The generated or cached static HTML content from WordPress is then sent back to the web server.

The server-level cache can now cache this static HTML content for even faster delivery on subsequent requests.

Caching Layers Sequence

First Request

User request → Web server (server-level cache miss) → WordPress (WordPress-level cache miss) → WordPress dynamically generates page → WordPress caching plugin saves static HTML → Server receives static HTML → Server-level cache saves static HTML.

Subsequent Requests

User request → Web server (server-level cache hit) → Static HTML is served from server-level cache.

If server-level cache misses

User request → Web server (server-level cache miss) → WordPress (WordPress-level cache hit) → WordPress serves cached static HTML → Server receives and caches static HTML.

Summary of the Interaction

Server-Level Cache

Operates first, caching the entire response at the web server level for very fast delivery. It handles the request before it reaches WordPress.

WordPress-Level Cache

Acts as a secondary cache that handles requests if the server-level cache misses. It generates or serves static HTML from within the WordPress application.

Layering Caches creates a multi-tiered caching system

The server-level cache (e.g., NGINX, Varnish) serves as the first line of defense, caching and serving static HTML quickly without involving PHP or WordPress.

The WordPress-level cache (e.g., Flyingpress, W3 Total Cache) provides a secondary layer, ensuring that even if the server-level cache misses, a cached version within WordPress can be served without regenerating the page dynamically.

Why Page Caching Plugins Are Necessary

Optimized HTML Generation

WordPress caching plugins often do more than just caching. They also optimize the HTML output by removing unnecessary characters, comments, and whitespace, which reduces the overall size of the HTML document.

Granular Control

WordPress caching plugins allow for more granular control over what is cached and how. This can include specific rules for logged-in users, e-commerce pages, or other dynamic content that server-level caching might not handle as efficiently.

Reduced Backend Load

While server-level caching reduces the load on the web server, WordPress caching plugins reduce the load on the WordPress application itself. This means fewer database queries and PHP executions, leading to faster response times and better scores.

Improved TTFB (Time to First Byte)

WordPress caching plugins can improve the TTFB by serving static HTML content directly from the WordPress layer. This immediate response helps in achieving better PageSpeed scores.

Cache Warming

Some caching plugins provide a feature to pre-cache or "warm-up" pages, ensuring that visitors always get the fastest possible response.

Practical Example

Even with server-level caching enabled, consider the following improvements a WordPress caching plugin might bring

Before Plugin Installation

Server-level cache is enabled, but the initial HTML generation might still involve unnecessary database queries and unoptimized HTML output. PageSpeed scores are good but not optimal.

After Plugin Installation

The WordPress caching plugin starts serving a cleaner, optimized HTML version of the page.

Additional optimizations like minification, image compression, and reduced HTTP requests further enhance performance.

The combined effect results in lower page load times, improved TTFB, and higher PageSpeed scores.

The immediate improvement in PageSpeed scores after adding a WordPress caching plugin, even with server-level caching enabled, is primarily due to the additional optimizations and efficiency improvements the plugin brings to the HTML generation and overall resource handling. These enhancements go beyond simple caching and contribute significantly to better performance metrics.

A Bloated DOM Will Hurt Performance

This is another half truth. The whole truth is that a bloated DOM *in conjunction* with a large HTML document weight will significantly hurt your pagespeed. Your DOM can be large if your document weight is low without significant effects on performance.

Elementor for instance does add a lot of DOM elements which is unfortunate as it makes targeting elements with custom CSS more difficult, however once optimized the performance impact of the excessive DOM size is mitigated. The most significant factor regarding optimization is always page/asset weight.

Ideally a page's DOM size should be below 1000 elements to minimize performance impact. Once the DOM size increases beyond that you will see minimal drops in pagespeed. A page with 1500 or even 2000 can render quickly if the HTML document weight is low. If you exceed 2000 elements, then you will begin to encounter noticeable (but relatively small) drops in pagespeed scores, even with a smaller HTML document.

Pagebuilder Sites Cannot Handle High Server Load On High Traffic Sites

This is also a partial fallacy. An unoptimized pagebuilder created site will easily collapse under the strain of excessive traffic. The database server (MySQL or any other database variant (such as MariaDB)) will quickly be overwhelmed by an influx of database queries, even with object caching.

To mitigate the impact of these queries, ensure that all autoload options that can be set to "no" are configured accordingly. Optimizing your pagebuilders CSS, Javascript and any other files that can be optimized will reduce Database queries which will reduce the strain on your server.

An optimized Elementor site can easily handle 100k concurrent users on a mid-grade VPS, and that could potentially scale into millions of concurrent users.

Elementor can perform just as well as Gutenberg can when properly optimized.

Headers and Footers Must be Hardcoded In CSS instead of Your Builder For The Best Performance

While hardcoding Headers and Footers in CSS can have performance benefits, the impact is minimal at best when using the Remove Unused CSS feature with optimization plugins. Many Pagebuilders will generate bulky CSS files for many elements, including headers and footers, which can be mitigated with a simple toggle.

Caveat: Some Page Builders will receive greater benefits from hardcoding Header and Footer than others.

Pagespeed Only Affects SEO

While the primary motivation for improving page speed is often better SEO rankings, an often overlooked factor is customer retention. Increased visitor retention can lead to a substantial increase in revenue, in addition to simply increasing your total visitor count and reducing bounce rate. For instance, reducing your bounce rate from 60% to 30% can result in significantly higher revenue/conversions, more sales, and improvements in actions like contact form fill-outs.

The benefits of enhanced visitor engagement/retention extend beyond just attracting more visitors, the importance of retaining visitors cannot be overstated.

Pagespeed Scores Don't Matter

This is one of the most patently absurd refrains ever to be found when discussing pagespeed. Pagespeed scores absolutely matter. As discussed in the Speed Testing and Inferences sections, Pagespeed scores are a composite of various metrics which result in a singular composite score out of 100.

The total scores diagnostic value is a measure of progress towards the desired optimization metrics, but the score in and of itself does not relay useful diagnostic information. The sub-metrics which the composite Pagespeed score is based on (TTFB, INFP, INTP, CLS, Speed Index, LCP, FCP, Blocking Time (TBT)) are the metrics with utility, and have specific meanings explained above. These metrics are quantifiable, with clear indications of why they are relevant to a user's experience on the website.

When someone says that these metrics are unimportant, this indicates an underlying misunderstanding of what these metrics mean.

Included here once again are their definitions:

Escape Creative Article

<https://escapecreative.io/google-pagespeed-insights-explained/>

This page has a decent overview of the Metrics and what they mean.

Corelogix Article

<https://coralogix.com/guides/real-user-monitoring/core-web-vitals/>

Another useful article with somewhat in-depth explanations of various Core Web Vitals metrics and their definitions.

Google's Core Web Vitals Documentation

<https://web.dev/articles/vitals>

Core Web Vitals documentation and definitions straight from Google.

Security Plugins Like Wordfence Are A Requirement

Half true. Files should be scanned semi-regularly only on a staging site. Wordfence has a significant detrimental effect on site performance. A well optimized site can drop 10-15 points on a mobile Pagespeed test just by activating Wordfence.

Your staging server's performance is irrelevant to users and will have no impact on conversions.

Instead of using Wordfence's firewall functionality, a lightweight application level (Wordpress) firewall plugin is recommended (Ninja Firewall and BBQ firewall are relatively good options).

Instead of relying solely on Firewall plugins, it is recommended to have three layers of firewalls. One at the WAF level (Cloudflare for example). A server level firewall (such as the 8G/7G firewall for Apache/NGINX), and finally an application level firewall for Wordpress. Wordpress should be the absolute last line of defense, if bots are getting through the first two layers there's already a major problem. Additionally, adding in Patchstack is highly recommended.

Security

Security Measures Performance Impact

High CPU Load And Excessive RAM Pressure

High CPU Load and/or excessive RAM pressure will slow your site to a crawl, and in many instances may completely take down your server temporarily until an infection has been remediated (removed).

Malware's Detrimental Performance Impact

Sudden unexplained high CPU load or excessive RAM pressure may indicate a Malware infection. Security mitigations are critical to resolving issues related to Malware, however they will not increase the performance of a website, they will only prevent detrimental effects up to the existing baseline performance.

An example: A site is scoring in the 60s without additional CPU load and RAM load before a security incident such as a Malware infection or DDOS. Once the site starts to experience the attack, the average score jumps down to the 30s.

Remediating a malware infection will return the site's score back to baseline, i.e. the 60s in this hypothetical. Performance of the website **will not** improve from baseline performance sans the malware, only the negative effects of the malware infection/DDOS/other attack type will be mitigated.

Mitigations Performance Impact

In fact, **security solutions and mitigation implemented to remediate malware, infection protection methods such as firewalls and security plugins can and will have a detrimental effect on baseline performance.** However the tradeoff of preventing these kinds of attacks with a small performance hit may be worth it if your site is experiencing cyberattacks, especially regular cyberattacks.

Brute Force Attacks

Is It WP Article

<https://www.isitwp.com/stop-brute-force-attacks-wordpress-website/>

Basic article which has a few methods to mitigate brute force attacks

Brute Force attacks means that you receive an excessive amount of login attempts which can cause High CPU Load and saturate the servers RAM. Thousands of login attempts can slow

down your site and even cause it to crash. It can disrupt your user experience which means visitors will leave your site since it won't load fast enough by absorbing all of the server's resources.

Limit Login Attempts Reloaded (Free)

<https://wordpress.org/plugins/limit-login-attempts-reloaded/>

Limit Login Attempts Reloaded functions as a robust deterrent against brute force attacks. It restricts the number of login attempts allowed. This applies not only to the standard login method, but also to XMLRPC, Woocommerce, and custom login pages.

Disable XML-RPC

Why Should I Disable XML-RPC?

DDoS (Distributed Denial-of-Service) attacks are when a bad actor overloads a web server with internet traffic to stop other users from accessing your online websites and services. Attackers use the PingBack feature to send pingbacks to numerous sites, and with XML-RPC, giving the attacker an unlimited amount of IP addresses to distribute the DDoS attacks.

Brute Force Attacks

XML-RPC.php is used to test different combinations of usernames and passwords. This security hole allows them to automate the attack with a command that pushes thousands of combinations at once, bypassing the security of your website and gaining access to it.

How To Disable XML-RPC

Asset Cleanup (Free and Pro)

Asset Cleanup is a highly recommended multi-function plugin which largely focuses on selective CSS and JS asset disabling, as well as the ability to fully disable plugins from loading on selected pages.

Asset Cleanup also has functionality to disable XML-RPC.

Simple Disable XML-RPC (Free)

<https://wordpress.org/plugins/simple-disable-xml-rpc/>

Simple Disable XML-RPC's sole function is to disable XML-RPC, with a sizable amount of configurability. Very lightweight, the plugin is just 10 kb total.

Firewalls

Firewall Types

Firewalls will always have a negative performance impact on your site. Security vs performance is a trade-off that must be considered/evaluated when implementing various security measures.

Differences between WAFs and Network/Server Layer Firewalls

Firewalls Performance Impact

IJSTR Article

<https://www.ijstr.org/final-print/mar2017/Impact-Of-Firewall-On-Network-Performance.pdf>

A comprehensive analysis of the performance impact of Firewalls on website performance.

WAF Network Performance Impacts

Aisnet

<https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1007&context=wisp2020>

ResearchGate

https://www.researchgate.net/profile/Francis-Agbenyegah/publication/326260720_Impact_Of_Firewall_On_Network_Performance/links/5b420199458515f71cb221e8/Impact-Of-Firewall-On-Network-Performance.pdf

Fortinet Article

<https://www.fortinet.com/resources/cyberglossary/waf-vs-firewall#:~:text=A%20WAF%20protects%20web%20applications,external%20and%20internal%20network%20traffic.>

Performant Server Level Firewalls

7G NGINX Firewall (Free)

<https://perishablepress.com/7g-firewall-nginx/>

Jeff Starr's performant 7G firewall for NGINX. The 8G version for NGINX is currently in development.

8G Apache Firewall (Free)

<https://perishablepress.com/8g-firewall/>

Jeff Starr's 8G Apache Firewall.

Performant Web Application Firewalls (WAFs)

Web Application Firewall

A Web Application Firewall (WAF), usually from a CDN, works at the network layer to block malicious traffic before it reaches your webserver. Cloudflare has a solid WAF, Microsoft has Front Door for Azure, there are plenty of them out there. This allows much more granular control and regularly updated cloud definitions for threats as they emerge.

A WAF will also let you have some nifty features that allow you to blanket ban countries to restrict traffic which will reduce server load (if your site is being attacked regularly) and hacking attempts.

BunkerWeb (Free)

<https://github.com/bunkerity/bunkerweb>

Free Flexible and Configurable Server Level WAF.

NGINX App Protect WAF (Free)

<https://gigaom.com/report/high-performance-web-application-firewall-testing/>

In GigaOm's tests, NGINX App Protect was found to be more performant than the 3 Cloud WAFs they tested.

Configuration Guide

<https://docs.nginx.com/nginx-app-protect-waf/v4/configuration-guide/configuration/>

Cloudflare WAF (Free and Pro)

<https://www.cloudflare.com/application-services/products/waf/>

A performant WAF developed by Cloudflare. Like any firewall it has a negative performance impact, but Cloudflare's is relatively well optimized. Best combined with Cloudflare's Argo as well as other Cloudflare network and traffic performance optimization features.

Cloudflare Firewall Rules for Securing WordPress Websites

<https://gridpane.com/blog/cloudflare-firewall-rules-for-securing-wordpress-websites/>

PHP Errors

PHP errors can potentially have significant detrimental effects on performance.

Debug Log Manager (Free)

<https://wordpress.org/plugins/debug-log-manager/>

Debug Log Manager can enable multiple types of logging automatically which is a great way to catch PHP errors, which can relay a bunch of diagnostic information to both discover and remediate PHP errors. Most plugins and themes have PHP errors, even if they are minor, and every PHP error has the potential to have a negative performance impact.

If you have custom code, Debug Log Manager can help identify problems with the code you have written, and if you have PHP errors from plugins or themes it will give you clear information to report as a bug to the developer so they can implement a fix.

Query Monitor (Free)

<https://wordpress.org/plugins/query-monitor/>

Query Monitor reports PHP errors.

Security Plugins

WP Beginner Article

<https://www.wpbeginner.com/plugins/how-to-scan-your-wordpress-site-for-potentially-malicious-code/>

Very basic article which explains how to use a security scanner plugin

GOTMLS (Free)

<https://wordpress.org/plugins/gotmls/>

Lightweight anti-malware plugin

Plugins To Avoid

Wordfence

Do not use Wordfence on your live site. It's insanely bulky and very detrimental to performance. Wordfence is a good way to scan your site for malware, **but only use it on staging!** Firewall and scanner functionality can easily be obtained with other more performant plugins.

Two Factor Authentication

Two Factor (Free)

<https://wordpress.org/plugins/two-factor/>

Official 2 Factor login plugin from the Wordpress core dev team. Lightweight and great support for any issues.

Fluent Auth (Free)

<https://wordpress.org/plugins/fluent-security/>

Lightweight two factor authentication plugin

BBQ Firewall (Free)

<https://wordpress.org/plugins/block-bad-queries/>

Neat application level firewall from Jeff Starr. Recommended to layer this with the 8G server level firewall as well as a WAF.

Block Bad Bots

Banhammer (Free)

<https://wordpress.org/plugins/banhammer/>

Banhammer gives you full control over who and what may access your site.

Security Headers

Security Headers Scanner (Free)

<https://securityheaders.com/>

Your site should score an A on securityheaders.com. This is actually very easy to achieve!

Headers Security Advanced & HSTS WP (Free)

<https://wordpress.org/plugins/headers-security-advanced-hsts-wp/>

Headers Security Advanced & HSTS WP (Free)

<https://wordpress.org/plugins/headers-security-advanced-hsts-wp/>

The Headers Security Advanced & HSTS WP project implements HTTP response headers that your site can use to increase the security of your website.

The plug-in will automatically set up all Best Practices (you don't have to think about anything), these HTTP response headers can prevent modern browsers from running into easily predictable vulnerabilities.

The Headers Security Advanced & HSTS WP project wants to popularize and increase awareness and usage of these headers for all wordpress users.

Patchstack (Freemium)

<https://patchstack.com/>

Patchstack can apply virtual patches to vulnerable plugins before the plugin is even updated by the developers, which can prevent your site from being hacked. Patchstack is one of the best vulnerability analysis companies and are frequently the ones to find zero days. Highly recommended.

Patchstack has a free tier, but to get the really good, useful stuff you need one of the paid plans.

Hardenize (Free)

[Hardenize: Comprehensive web site configuration test](#)

Scan your site for common security issues.

No Unsafe Inline Scripts (Free)

<https://wordpress.org/plugins/no-unsafe-inline/>

Linux Server Hardening Guide

<https://ivansalloum.com/comprehensive-linux-server-hardening-guide/>

SOC Compliance for Wordpress

<https://www.kevinleary.net/blog/soc-compliance-wordpress-websites/>

Imunify360 (Paid)

<https://www.imunify360.com/>

A very comprehensive security suite. Does a lot of nifty things I haven't seen other security providers have. Janky early 2000s looking interface though.

Kevin Leary's Blog

<https://www.kevinleary.net/blog/securing-a-wordpress-website/>

Audit Log

CoreActivity Log (Free)

<https://wordpress.org/plugins/coreactivity/>

Free CoreActivity Log is a free plugin for monitoring and logging various activities. The plugin is highly modular, with events registered and controlled by multiple Components.

Currently, the plugin has 28 components with a total of 174 events, with direct integration with 12 popular plugins. Out of all the options for audit logging, this is the most comprehensive.

Stream (Free)

<https://wordpress.org/plugins/stream/>

The plugin records WordPress user and system action to the Stream logs. Every logged-in user action is displayed in an activity stream and organized for easy filtering by User, Role, Context, Action or IP address. Admins can highlight entries in the Stream log—such as suspicious user activity—to investigate what's happening in real time. Stream also allows you to configure email alerts and webhooks for integrations like Slack and IFTTT to notify you and your team when something has gone wrong.

Stream is entirely free with no pro plan

Aryo Activity Log (Free)

<https://wordpress.org/plugins/aryo-activity-log/>

Another activity monitoring plugin, also entirely free.

DecaLog (Free)

<https://wordpress.org/plugins/decalog/>

While DecaLog doesn't specifically log user events, it does have the wonderful function of logging various events on a site that are not logged by either Stream or Aryo, which gives you different insights into background processes, which makes it easier to catch malware in the act.

If you have trouble interpreting the results, I suggest screenshotting the logs and then uploading it to an AI bot to interpret them for you. Paid AI bots can be accessed for free via [Poe](#).

Automatically Update Server Software

Outdated Server Software

Outdated software such as old PHP versions, old MySQL versions, old NGINX versions, Old Apache versions, old varnish versions etc can potentially be big security issues. Patches for these pieces of software are regularly released, and it's highly recommended to regularly update the software installed on your server to mitigate security incidents.

Schedule System Cron Jobs (Free)

These updates can automated via Cron Jobs

Hive Velocity Article

<https://www.hivelocity.net/kb/how-to-schedule-system-updates-and-tasks-using-cron-jobs/>

System Cronjob Generators

Free System Cronjob Generators

Cronhub (Free)

<https://crontab.cronhub.io/>

Crontab Guru (Free)

<https://crontab.guru/>

Crontab Generator (Free)

<https://crontab-generator.org/>

File Change Monitoring

Website File Changes Monitor (Free)

<https://wordpress.org/plugins/website-file-changes-monitor/>

Get alerted via email of file changes on your WordPress sites to boost reliability & security

High CPU Load And Excessive RAM Pressure

Security Incidents

In addition to Malware Infections, Brute Force Attacks, DDOS's and other security incidents, high CPU usage and RAM pressure can be caused by other factors.

Excessive Database Load

A large amount of autoloaded options, excessive amount of revisions, excessive amounts of transients, bloated Database tables, slow queries, lack of Object Caching, lack of indexes on specific tables can all cause high CPU Load and Excessive RAM pressure.

Mitigations

Excessive Amount Of Wordpress Cronjobs

Disable Wordpress Cronjobs And Enable Server Cronjobs (Free)

WP Beginner Article

<https://www.wpbeginner.com/wp-tutorials/how-to-disable-wp-cron-in-wordpress-and-set-up-proper-cron-jobs/>

System Cronjob Generators

Free System Cronjob Generators

Cronhub (Free)

<https://crontab.cronhub.io/>

Crontab Guru (Free)

<https://crontab.guru/>

Crontab Generator (Free)

<https://crontab-generator.org/>

Cloud Server Autoscaling (Paid)

Many Cloud VPS providers and Serverless providers have an auto-scaling functionality which will automatically upgrade and downgrade your server resources based on server load. This service is on a usage basis and does not have a flat monthly cost. Auto-scaling is a necessity on high traffic sites.

Too Many WP-Admin Tabs Open Simultaneously

Having too many WP Admin Backend tabs open at the same time can cause excessive CPU load and RAM pressure, even with other mitigations applied. Modifying the Wordpress Heartbeat interval can mitigate some of the performance impact of numerous WP-Admin tabs in your browser.

Wordpress Heartbeat

WP Beginner Article

<https://www.wpbeginner.com/plugins/how-to-limit-heartbeat-api-in-wordpress/>

Very short explainer Wordpress's Heartbeat functionality.

Dynamic Frontend Wordpress Heartbeat Control (Free)

<https://wordpress.org/plugins/dynamic-front-end-heartbeat-control/>

The plugin offers an efficient approach to dynamically control the front-end heartbeat, automatically determining the optimal settings for your website. By analyzing user usage, WordPress site assets, and server environment, it empowers your WordPress performance to achieve its full potential.

Once activated, it immediately begins improving performance without any need for user intervention. It intelligently adjusts the heartbeat interval in real-time, adapting to the changing demands of your website. This dynamic control guarantees an optimized and efficient heartbeat system, resulting in improved overall performance and responsiveness.

Admin and Site Enhancements (Free)

<https://wordpress.org/plugins/admin-site-enhancements/>

Admin and Site Enhancements allows you to control the heartbeat frequency as well. Less flexible than Dynamic Frontend Heartbeat Control.

Tune MySQL Configuration Settings (Free)

Automatic Tuning

MySQL Tuner (Free)

<https://github.com/major/MySQLTuner-perl>

MySQLTuner is a script written in Perl that allows you to review a MySQL installation quickly and make adjustments to increase performance and stability. The current configuration variables and status data is retrieved and presented in a brief format along with some basic performance suggestions.

MySQLTuner supports ~300 indicators for MySQL/MariaDB/Percona Server in this la

Releem (Paid)

<https://releem.com/>

Releem will autotune your MySQL configuration on a VPS automatically and adjust the configuration based on both the current and average MySQL load on the server.

Reduce Autoloaded Options

SweepPress (Free)

<https://wordpress.org/plugins/sweeppress/>

Modify Autoload Settings or delete the autoload options

AAA Option Optimizer (Free)

<https://wordpress.org/plugins/aaa-option-optimizer/>

Another Autoload Option Optimization plugin

Object Caching

Docket Cache (Free)

<https://wordpress.org/plugins/docket-cache/>

Docket cache can be used when you cannot install Redis (some shared hosting plans don't have it for instance). Docket cache is an object cache, just like Redis is, except it functions at the Wordpress level as opposed to the server level. It can have better performance than Redis in many cases.

Redis (Free)

Dragonfly Redis Performance Optimization Guide

<https://www.dragonflydb.io/guides/redis-memory-and-performance-optimization>

Extensive article on optimizing Redis performance.

General Redis information for Wordpress

Redis is also a server level service, which you will need a Wordpress plugin to use as a connector to the server level redis service. Redis must be installed on your VPS/by your hosting provider for the plugin to work.

Redis Object Cache (Free)

<https://wordpress.org/plugins/redis-cache/>

Redis Object Cache developed by Till Krüss.

Poorly Coded Plugins And Themes

Poorly Coded PHP Scripts In Plugins And Themes

Many Themes And Plugins which are coded poorly have both a large amount of PHP scripts, as well as poorly coded PHP scripts which can overwhelm a server when it experiences even moderately high traffic.

Diagnosis

F12 Profiler (Free)

<https://wordpress.org/plugins/f12-profiler/>

F12 Profiler is a great plugin to diagnose plugin and theme performance impacts. It adds a section to the Wordpress admin bar which can display the load time impact of each.

It hasn't been updated in quite a while, but it still works perfectly from my testing.

Code Profiler (Free and Pro)

Code Profiler can be used to diagnose which plugins/theme is causing high CPU Load and Excessive RAM pressure

Poorly Configured Or Lack Of Page And/Or Server Caching

Poorly configured or lack of caching can severely impact CPU load and cause Excessive RAM pressure.

Very simple solution, simply enable server caching as well as page caching. This is covered ~halfway down in this guide.

Direct Link:

<https://docs.google.com/document/d/1ncQcxnD-CxDk4h01QYyrlOh1IEYDS-DV/edit#heading=h.3cqmetx>

Reinstall Wordpress Core (Free)




WP Beginner Article

<https://www.wpbeginner.com/wp-tutorials/how-to-uninstall-and-reinstall-wordpress/>

When all else fails, sometimes reinstalling Wordpress Core can resolve High CPU Load and Excessive RAM Pressure. This is very rarely needed.

Fix Redirect Chains

Web Design London | Digital Agency - WebsiteDeco

URL	Status	Domain	Size	Timeline
 /	301	websitedeco.com	253 B	438.4ms
 /	301	websitedeco.com	0	523ms
 /	200	websitedeco.com	18.3 KB	604.2ms

You should always rectify redirect chains. Your site should either be set to have a single https redirect to <https://yourdomain.com> or a single redirect directly to the <https://www.yourdomain.com> version.

You should never have an intermediary redirect between the unprotected SSLless http version to the https www version if you choose to use www.

E.G. http>https://>https://www. This extra hop will always introduce additional latency. Make sure that you eliminate it.

To reiterate, you should only have the bare http link which is immediately redirected to the correct https link in your waterfall chart on gtmetrix.

If it's www, it should be <http://yourdomain.com>>https://www.yourdomain.com. You should not have an intermediary redirect to <https://example.com>.

If your domain has no WWW link, it should be <http://yourdomain.com>> <https://yourdomain.com> with no intermediary redirect in the middle.

This can be done either through an htaccess/nginx rule, a redirection plugin or via your CDN. Cloudflare allows you to create a redirect rule that will always forward http links straight to www, skipping the intermediary redirect. Your SEO plugin will usually also have this option, but doing it via the wordpress side will introduce additional latency. You will want to implement this on the server level or via your CDN to achieve the lowest latency.

Plugin option and articles

<https://wordpress.org/plugins/safe-redirect-manager/>

<https://wordpress.org/plugins/redirection/>

Your SEO plugin almost certainly has an option for redirects as well. Plugin fixes may have slightly worse performance than implementing redirections via .htaccess or NGINX configuration files, but are easier to manage.

Manual identification of redirects and how to fix them

<https://kinsta.com/blog/remove-the-following-redirect-chain-if-possible/>

<https://prerender.io/blog/do-redirect-chains-hurt-seo/>

<https://tillison.co.uk/blog/how-to-fix-redirect-chains-and-loops-in-wordpress/>

<https://www.semrush.com/blog/301-redirect-htaccess/>

Server and Network Performance Monitoring

Activate New Relic only when diagnosing issues on your live Site Or Preferably On Staging

New Relic and Signoz have performance overhead and therefore will have negative performance impacts on your server. If possible, only activate New Relic or SigNoz when troubleshooting/logging/analyzing issues to avoid any performance impacts they may cause by leaving them active long term.

New Relic (Free and Pro)

New Relic Open Source Projects (Free)

<https://github.com/orgs/newrelic/repositories?type=all>

New Relic has free open source github projects, which basically function as a free community edition as well. A fantastic option for observability and performance statistics.

<https://docs.newrelic.com/docs/network-performance-monitoring/get-started/npm-introduction/>

New Relic is a server level application that monitors a variety of network traffic and statistics which provides a high level of observability into performance issues as well as troubleshooting. New Relic is an invaluable diagnostic tool that is highly recommended for any dev to integrate into their toolkit.

It monitors all aspects of your web stack performance.

Automation Script for Automatically Installing Open Source New Relic On Most Linux Distributions

https://docs.google.com/document/d/1wtlcwh4vjAEIr_71WioxlUCV1q3DtlzhQKYxDz6RcbY/

SigNoz (Free and Pro)

<https://signoz.io/>

<https://signoz.io/docs/install/docker/>

<https://github.com/signoz/signoz>

SigNoz has an Open Source community version which is self-hostable, which can be used entirely for free. A good alternative to New Relic. Slightly easier to install than New Relic.

Wordpress Optimizations

Should I use every optimization opportunity in this guide simultaneously?

Yes! Every feature listed here is complementary and can and should be used together simultaneously. Implement every feature that you can for the best performance benefit. Contrary to popular belief, there is no such thing as “too many” plugins. A plugin is simply a bundled snippet of code prepackaged and developed by a plugin developer. A plugin could just be a single simple PHP function. It's the quality and content of the code inside a plugin which is what counts.

If the plugin is lightweight and doesn't interfere with another, there is no harm running as many plugins as you'd like. A properly optimized site could run 90 plugins simultaneously without performance issues. Unoptimized plugins' negative effects are additive, so make sure to only choose quality, optimized plugins to avoid slowing down your site.

On the other hand, you should only have debug and analysis plugins active when they are needed, as they will cause a performance hit and slow your site down if left active. These should preferably only be enabled on a staging site to avoid slow downs for users.

Even when debug plugins are well optimized, there are a lot of hits to the database, so only use them on staging or enable them on your live site for as short a time as necessary to get the information you need.

Use as few plugins as necessary to achieve whatever design you'd like, but there is simply no such thing as too many optimization plugins(or other plugins in general if they are well optimized). Optimization plugins are intended to improve your performance, not hurt it. You will not bog your site down running all of the optimization stack simultaneously. Implementing all of these techniques together is how to achieve optimal scores.

Best practices

Activate plugins you are testing on staging.

Do not activate any plugins on your live site that you are not certain that you want. Always test them in staging or on a local wordpress development server. This cannot be stressed enough. Some plugins introduce lag that cannot be completely removed even by cleaning the database tables/disabling autoload options or deleting the options entirely.

Take Regular Backups

Create backups often when you are working on the Wordpress side. You don't want to lose the work you've put in due to an easily avoidable mistake which causes you to restore. It's best to create them after every change to prevent loss of progress.

Sometimes a plugin or change will create a critical error(relatively rare), lock you out of a specific section of the admin backend, prevent the admin backend from loading entirely or other weird behavior. When this occurs, the only solution is to restore to a working backup.

I've gained some performance back by removing the database entries that some plugins have created, but there have been times you cannot reclaim all of the lost performance, and you will have to revert to a backup.

Always test your new plugins in a non-live staging environment before you commit to them and implement them on your live site.

Choose Quality Plugins

If there are multiple plugins that all suit your needs for a certain task, pick the most optimized one, because it will run the best. If there are 3 plugins, one is 300 kb, 700kb, and 1.2MB, the 300 kb plugin is the one that will perform best most of the time as a general rule(although this is not always the case). Always compare the file sizes of the plugins, but the only real way to tell is to test them one at a time. Comparing file sizes to get a likely ballpark for the pagespeed impact

is a good rule of thumb however. Plugins with a smaller file size will usually perform better, so it's a good heuristic to judge a plugin's likely impact.

All of the plugins listed in this guide are performance optimized. Any of the plugins for specific features(charts, SEO, tables, custom fields, backups, user management, woocommerce, forms, LMS's, support plugins, etc) that I've included have been intentionally selected as they have optimized, performant code. If you build your site with only plugins listed in this guide, your page weight will be low even prior to optimization.

Disable Unneeded Database Autoload Options

After you install a plugin, make sure to inspect your database with Advanced Database Cleaner to check how many tables and plugin options were added. Many plugins will create too many options set to autoload, which can seriously slow your site down. Most options can generally be set to autoload no without impacting functionality, **but that is not always the case.**

Disabling autoload could have severe negative impacts if you disable the wrong option, in some extreme cases you can even be entirely locked out of the backend and trigger critical plugin errors. Disabling autoload options for performance is always a good idea for performance, but be careful which ones you disable and always test your changes on staging first!

ALWAYS test disabling autoloading options for active plugins on staging before pushing the changes live. **There's trial and error involved, make sure you have backups before changing autoload settings on the live site!**

Remove Unneeded Database Entries

If a plugin adds too many database tables and options and you want to remove the plugin, you have to delete the database entries with Advanced Database Cleaner or another database cleanup plugin in the database section after deleting the plugin as most plugins leave all of their options and tables in the database when deactivating/deleting them. Many plugins set options to autoload even if they do not need to be autoloaded, so make sure to disable as many autoloaded options as possible(without breaking functionality).

Automatic plugin and theme updates

Keep themes and plugins up to date.

Keep your themes and plugins up to date for security and performance reasons. Devs generally will frequently issue performance, feature and security updates for their plugins, and they should ideally always be kept up to date.

Only set some of your plugins that do not have critical site functionality to auto-update.

You wouldn't want to auto-update a pagebuilder for instance. Some plugin updates can completely break certain functionality by introducing new bugs, so it's a good idea to have backups before updating in case you need to revert. Test on staging when updating plugins. In very rare cases even a non-critical plugin can break your site, so as always, take regular backups!

Theme updates could break your site as well, especially since if you're following the advice in this guide, you should be disabling everything humanly possible in your theme and work exclusively with some sort of page builder or custom code your site if that's your preference.

WP-Config Memory limit constant

You can paste it into your wp-config file just above the line **“/* That's all, stop editing! Happy publishing. */”** just make sure to modify the value to match whatever you set in your PHP settings.

E.g:

```
define( 'WP_MEMORY_LIMIT', '256M' );
```

Note: The memory limit is how much RAM is allowed to be allocated to a single PHP process. You should not have any plugins or scripts taking up that much RAM. If you do, I recommend you switch plugins. The default recommendation on most guides is 256M. If you must raise the value above 256M, do not set the value to exceed 512M. 512M should be the cap.

Disable XML-RPC

Disable XML-RPC. There are a variety of ways to disable it. I use Asset Cleanup to do so, but there are many plugin options. This will enhance your site's security and also prevent some ddos attacks. You can just delete the file, XML-RPC is not used by 99% of wordpress sites, you only want XML-RPC active if it's necessary for a plugin to function.

Diagnosis and Analysis

Database Tools

Database Query Analysis

Query Monitor (Free)

<https://wordpress.org/plugins/query-monitor/>

Helps to debug database queries, PHP errors, hooks and actions, block editor blocks, enqueued scripts and stylesheets, HTTP API calls, and more

Add-Ons (Free)

<https://querymonitor.com/help/add-on-plugins/>

Collection of Query Monitor Addons to extend Query Monitor's functionality.

Fluent Query Logger (Free)

<https://wordpress.org/plugins/fluent-query-logger/>

Query Logger with Selective Logging Functionality.

WP Database Snapshots (Free)

<https://wordpress.org/plugins/wpvivid-snapshot-database/>

WPvivid Database Snapshots allows you to quickly create snapshots of all tables in your WordPress database.

Making a database snapshot is much faster than making a database backup. If you need a quicker rollback solution for safely testing WordPress database-related changes, use a snapshot.

Enable Database Tools (Free)

<https://wordpress.org/plugins/enable-database-tools>

Does **one thing**: Enables you to optimize and repair InnoDB and MyISAM tables using WordPress built-in database tools.

Database Administration

Database Admin (Free)

<https://github.com/griouslad/database-admin>

Adminer replacement, directly from within the Wordpress admin dashboard!

WP phpMyAdmin (Free)

<https://wordpress.org/plugins/wp-phpmyadmin-extension/>

PHP My Admin from within the Wordpress admin dashboard.

Performance Profiling

Code Profiler (Free and Pro)

<https://wordpress.org/plugins/code-profiler/>

<https://code-profiler.com/>

Code Profiler helps you measure the performance of your plugins and themes at the PHP level and to quickly find any potential problem in your WordPress installation. Useful for troubleshooting and identifying which of the plugins you are using have poor performance.

It can point to the specific lines of code and plugins which are triggering performance issues. Very useful if you are creating your own plugin, want to modify someone else's to fix the issue yourself, identify a plugin which is detrimental to performance or want to report the identified issue to a plugin developer so it can be fixed.

F12 Profiler (Free)

<https://wordpress.org/plugins/f12-profiler/>

A code profiling plugin, not as granular or detailed as Code Profiler, but has some nice features that Code Profiler doesn't have, such as a frontend admin menu bar which displays profiling information for the current page you are viewing on the frontend.

AIO Performance Profiler (Free)

<https://wordpress.org/plugins/all-in-one-performance-accelerator/>

Note: I don't recommend activating this on a live site, activate it on staging. It's useful for diagnosis, but it slowed the site I was working down a lot even once deactivated, I never dug into why. I suggest you use this plugin only on a staging site.

LAPS Performance Profiling (Free)

<https://github.com/Rarst/laps>

Lightweight performance profiler. It provides a visual summary in the admin toolbar that is quick and easy to inspect.

Fix Plugin Conflicts

Check Conflicts (Free)

<https://wordpress.org/plugins/check-conflicts/>

The plugin allows you to disable/enable plugins and/or activate a default theme for checking conflict between them only for your IP; other users won't see any changes during the tests.

Plugin Detective (Free)

<https://wordpress.org/plugins/plugin-detective/>

Helps diagnose plugin conflicts in a step-by-step manner.

WP Tools (Free)

<https://wordpress.org/plugins/wptools/>

The WP Tools plugin (wp-tools) features a comprehensive dashboard showcasing server statistics, including a chart of errors, page load time also by page and boasts over 47 tools designed to simplify your WordPress management tasks.

System Dashboard (Free)

<https://wordpress.org/plugins/system-dashboard/>

This plugin provides a central dashboard to monitor various WordPress components, processes and data, including server hardware, software and resource usage. Pairs well with Query Monitor

Filters and Hooks

WP Hooks Finder (Free)

<https://wordpress.org/plugins/wp-hooks-finder/>

If you look at any WordPress page or post, they appear through the combination of action & filter hooks. Sometimes useful for debugging and plugin development.

Per Page List Of Plugins In Use

Plugin List (Free)

<https://wordpress.org/plugins/plugins-list/>

The plugin inserts an XHTML list into any post/page through a shortcode. If you're into customization, you can specify a format argument and indicate the exact output you are after.

There's also an option to display inactive plugins as well.

- A simple template system allows you to format how you'd like the plugin information to be shown
- Template tags are available for automatically linked items as well as simple text
- Choose from a number of pieces of plugin data to be output
- Display inactive plugins as well as active plugins if you wish
- Automatically limit long descriptions to specific lengths, and even remove emojis from the output
- Output is cached to provide a super-quick response
- A separate shortcode allows you to display how many plugins you have!

Gutenberg Block Usage

Find My Blocks (Free)

<https://wordpress.org/plugins/find-my-blocks/>

Fast search for block locations

Save your time by scanning your entire site (including posts, pages and custom post types) to locate instances of WordPress Blocks.

- Find & replace blocks safely with quick edit and preview links.
- WordPress Full Site Editing support. Block Templates & Template parts will be searched.
- Performance settings for low (shared hosting) to high-end hosting.
- 100% free & open source.

Issues Tracker (Free)

<https://wordpress.org/plugins/issues-tracker/>

Issues Tracker allows you to view and search WordPress Debug logs, receive security advice, track 404 errors, and view your server settings.

Inspect and Block HTTP Requests (Free)

<https://wordpress.org/plugins/inspect-http-requests/>

Monitor all the HTTP Requests being made via WP HTTP Methods i.e. wp_remote_get, wp_remote_post. Block any request by just a click of a button.

Track how much time a request like updating core/plugin/theme taking (may be useful for bandwidth consumption analysis)

Transients Manager (Free)

<https://wordpress.org/plugins/transients-manager/>

Transients manager allows you to:

1. Toolbar button to suspend transient writes
2. View all transients in the database
3. Edit the name, expiration, and value of any transient
4. Delete any transient
5. Search transients by name
6. Bulk actions to delete: all, expired, unexpired, or persistent transients

Performance Testing From Within The Admin Dashboard

The speed testing section above is one of the major resources you will use to diagnose performance issues. Waterfall charts are critical to diagnosing which files need to be delayed, disabled, compressed, lazy loaded or need loading attributes such as defer.

Vibes (Free)

<https://wordpress.org/plugins/vibes/>

A plugin from the PerfOps one suite, free lighthouse testing directly from the admin dashboard, very useful.

WP Speed Of Light (Free)

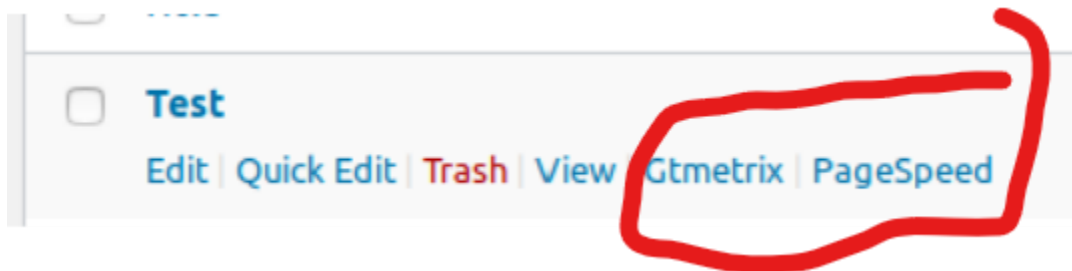
<https://wordpress.org/plugins/wp-speed-of-light/>

WP Speed Of Light is a free optimization plugin with integrated pagespeed testing. There is a pro version, but it is completely unnecessary if this is used as an optimization/caching plugin in addition to the speed testing functionality. All of the pro features are available via free plugins in other sections.

WPO Checker (Free)

<https://wordpress.org/plugins/wpo-checker/>

Adds links to external pagespeed tools on the page/post list.



Has not been updated in a while, activate on staging to ensure there are no conflicts.

Speed Guard (Free)

<https://wordpress.org/plugins/speedguard/>

Bulk Pagespeed Insights Testing tool. Add urls to test in the backend, and it will run bulk pagespeed tests on all of them.

Caveats: the functionality for displaying the LCP score of each tested page on the plugin settings page is broken, and all of the pagespeed reports need to be opened individually (there is no overview) to see the scores.

Admin Bar Tools (Free)

<https://wordpress.org/plugins/admin-bar-tools/>

Adds a Pagespeed Insights link to the admin bar to launch a test for any page viewed on the frontend.

SEO Booster (Paid)

<https://wordpress.org/plugins/seo-booster/>

The paid version of SEO Booster has Pagespeed Insights Testing functionality.

WP Performance Tester (Free)

<https://wordpress.org/plugins/wppercformancetester/>

WPPERformanceTester was written as a tool to benchmark WordPress in the WordPress Hosting Performance Benchmarks (2015) by Review Signal. Current benchmarks are on WPHostingBenchmarks.com. It was designed to test the server's performance by stressing PHP, MySQL and running \$wpdb queries.

WPPERformanceTester performs the following tests

- Math – 100,000 math function tests
- String Manipulation – 100,000 string manipulation tests
- Loops – 1,000,000 loop iterations
- Conditionals – 1,000,000 conditional logic checks
- MySQL (connect, select, version, aes_encrypt) – basic mysql functions and 5,000,000 AES_ENCRYPT() iterations
- \$wpdb – 250 insert, select, update and delete operations through \$wpdb
- It also allows you to see how your server's performance stacks up against our industry benchmark. Our industry benchmark is the average of all submitted test results.

Page Size Analysis

GreenerWP (Free)

<https://wordpress.org/plugins/greenerwp/>

Bulk page size scanner. Not very detailed info, but provides a quick overview of individual page weight across multiple individual pages.

Logging

Mail Logging

WP Mail Logging (Free)

<https://wordpress.org/plugins/wp-mail-logging/>

Use this plugin to log all outgoing emails from your WordPress site. If there are any errors when sending the email from your site, our email logs will catch that error and display it to you.

Debug Logging

Debug Log Manager (Free)

<https://wordpress.org/plugins/debug-log-manager/>

- Parse the debug.log file and view distinct errors and when they last occurred
- Quickly find and filter more specific errors for your debugging work.
- Make error details easier to read by identifying error source
- Easily view files where PHP errors occurred. This includes WordPress core, plugin and theme files.
- Enable auto-refresh to automatically load new log entries.
- Easily clear the debug.log file to save disk space and more easily observe newly occurring errors on your site.
- Show an indicator on the admin bar when error logging is enabled.
- Add a dashboard widget showing the latest errors logged.
- Use `error_log()` to output error info into your debug log

Bugfu (Free)

<https://wordpress.org/plugins/bugfu-console-debugger/>

Log/Debug the PHP code in your Theme/Plugin in your Browser Console (no extension needed).

Debug This (Free)

<https://wordpress.org/plugins/debug-this/>

Provides a ton of information about your WordPress installation, all from the front-end admin bar.

- *The current WP_Query object*
- Blog info and options
- Embed providers
- Files in rendered HTML (CSS, images, JavaScript)
- Filters and actions
- Images sizes
- Globals and constants
- Menus
- Object cache stats

- Plugins, must-use plugins, and dropins
- Rewrite rules
- Queries
- PHP and server information
- Post attachments
- Post types
- Scripts and styles enqueued
- Shortcodes
- Sidebars and widgets
- Taxonomies and terms
- Themes
- Users
- WP cron schedules and jobs
- WP debug log

Debug Bar (Free)

<https://wordpress.org/plugins/debug-bar/>

Adds a debug menu to the admin bar that shows query, cache, and other helpful debugging information.

Action and Audit Logging

CoreActivity Log (Free)

<https://wordpress.org/plugins/coreactivity/>

Free CoreActivity Log is a free plugin for monitoring and logging various activities. The plugin is highly modular, with events registered and controlled by multiple Components.

Currently, the plugin has 28 components with a total of 174 events, with direct integration with 12 popular plugins. Out of all the options for audit logging, this is the most comprehensive.

Log Dash (Free)

<https://wordpress.org/plugins/logdash-activity-log/>

Lightweight activity log allows you to effortlessly monitor and track all actions on your WordPress site, including user logins, content changes, plugin updates, and more. Its user-

friendly dashboard gives you instant access to critical information, making it easy to identify and resolve issues quickly.

Stream (Free)

<https://wordpress.org/plugins/stream/>

The plugin records WordPress user and system action to the Stream logs. Every logged-in user action is displayed in an activity stream and organized for easy filtering by User, Role, Context, Action or IP address. Admins can highlight entries in the Stream log—such as suspicious user activity—to investigate what’s happening in real time. Stream also allows you to configure email alerts and webhooks for integrations like Slack and IFTTT to notify you and your team when something has gone wrong.

Stream is entirely free with no pro plan

Aryo Activity Log (Free)

<https://wordpress.org/plugins/aryo-activity-log/>

Another activity monitoring plugin, also entirely free.

DecaLog (Free)

<https://wordpress.org/plugins/decalog/>

While DecaLog doesn’t specifically log user events, it logs various events on a site that are not logged by either Stream or Aryo, which gives you different insights into background processes, which makes it easier to catch malware, and do deeper dives into performance issues.

If you have trouble interpreting the results, I suggest screenshotting the logs and then uploading it to an AI bot to interpret them for you. Paid AI bots can be accessed for free via [Poe](#).

Simple History (Free)

<https://wordpress.org/plugins/simple-history/>

Audit log for Wordpress, useful for diagnosing changes made by users and has detailed logging, and direct integration with multiple plugins.

Ajax Event Logging

Amuga Ajax Log (Free)

<https://wordpress.org/plugins/amuga-ajax-log/>

Logs Ajax calls. Has not been updated in quite a long time, use this only on staging for diagnostic purposes.

Cron Logging

Cron Logger (Free)

<https://wordpress.org/plugins/cron-logger/>

This plugin will help you identify which Cronjobs (if any) are problematic.

Performance and Wordpress Analysis

Query Monitor (Free)

<https://wordpress.org/plugins/query-monitor/>

Helps to debug and identify slow database queries, PHP errors, hooks and actions, block editor blocks, enqueued scripts and stylesheets, HTTP API calls, and more. A critical tool in the toolbelt for diagnosing performance problems.

PerfOps One Plugins (Free)

A suite of free and open source WordPress plugins dedicated to observability and operations performance.

OPcache Manager (Free)

<https://wordpress.org/plugins/opcache-manager/>

Full featured OPcache management and analytics reporting tool. OPcache Manager allows you to supervise and optimize OPcache operations on a WordPress site.

Sessions (Free)

<https://wordpress.org/plugins/sessions/>

Powerful sessions manager for WordPress with multi-criteria sessions limiter and full analytics reporting. Sessions relies on the standard WordPress sessions manager and adds extra features and controls.

Traffic (Free)

<https://wordpress.org/plugins/traffic/>

Full featured analytics reporting tool that brings accurate oversight about all inbound and outbound API calls made to/from a WordPress site.

Vibes (Free)

<https://wordpress.org/plugins/vibes/>

User experience and browsing performance monitoring solution that analyzes perceived performance from a users' viewpoint.

Keys Master (Free)

<https://wordpress.org/plugins/keys-master/>

Robust application passwords manager for WordPress with role-based usage control and full analytics reporting. Keys Master relies on the standard WordPress "application password" core feature and adds extra features and controls.

oEmbed Manager (Free)

<https://wordpress.org/plugins/oembed-manager/>

Management tool which helps to fine-tune and conditionally activate WordPress core oEmbed producer/consumer. oEmbed Manager can help to achieve GDPR compliance.

DecaLog (Free)

<https://wordpress.org/plugins/decalog/>

De-facto standard stack for WordPress observability, DecaLog provides reliable and powerful logging, monitoring and tracing features for WordPress core, PHP, database, plugins and themes.

APCu Manager (Free)

<https://wordpress.org/plugins/apcu-manager/>

Full featured APCu management and analytics reporting tool. APCu Manager allows you to supervise and optimize APCu operations on a WordPress site.

IP Locator (Free)

<https://wordpress.org/plugins/ip-locator/>

Country and language detection service for WordPress. Fast, reliable, and plug & play. IP Locator provides conditional customization and detection API for WordPress core and plugins needing it.

Device Detector (Free)

<https://wordpress.org/plugins/device-detector/>

Detection and management tool that detects all devices accessing a WordPress site. Device Detector provides comprehensive analytics reports, CSS customization and detection API for WordPress core and plugins which need it.

MailArchiver (Free)

<https://wordpress.org/plugins/mailarchiver/>

Automatic outbound mails archiving tool which perfectly works regardless of the mail service provider and the technology used: API, SMTP, IMAP or native PHP

Apache Status & Info (Free)

<https://wordpress.org/plugins/htaccess-server-info-server-status/>

Tool to activate and set up urls for Apache server-status and server-info modules. In addition, Apache Status & Info allows you to check Apache configuration and settings, within WPadmin.

Image Optimization

The majority of options included in this section are free

Manual Image compression

Always compress multiple times(as many as possible) until you get to the point where compressing further unacceptably degrades quality. I've done 10-20 different compression passes on an image before without noticeable degradation in quality. Run TinyJPG and TinyPNG on images as many times as possible before switching to another compressor.

Once you compress an image with an alternative service, you can run it through TinyPNG and TinyJPG again for further compression runs!

As an example: TinyPNG > TinyPNG > TinyPNG > TinyPNG > TinyJPG > TinyJPG > TinyPNG > TinyPNG > 11zon > TinyPNG > 11zon > TinyJPG > TinyPNG

Just keep compressing it and bouncing it around services to compress it further until you can't compress the image on any service further or quality degrades beyond an acceptable level.

Even once a service exhausts its ability to compress further after multiple runs, using another service to generate an even more compressed image will sometimes allow the previous compression site to compress further. Converting an image to another format will always allow you to use compression services to compress them even further once they've been converted.

TinyPNG (Free)

<https://tinypng.com/>

Great optimization site. Tinypng is unique in that you can do multiple compression passes. E.g. Compress the image once, download it, then immediately drag it back to tinypng to compress it further. TinyPNG can usually do at least 3 passes. Always do multiple passes with TinyPNG! Once TinyPNG can't reduce the size further, run the image through TinyJPG

TinyJPG (Free)

<https://tinyjpg.com/>

Incredibly similar to tinypng, but uses a slightly different algorithm. Run images through TinyJPG after TinyPNG (or vice-versa)

Compress-or-die (Free)

<https://compress-or-die.com/>

Batch Compress (Free)

<https://batchcompress.com/en>

Batch compress converts PNGs to JPG and will eliminate any transparency effects on the compressed png. Only use batchcompress if you do not need png transparency.

Squeeze Pics (Free)

<https://www.squeeze.pics/>

Batch compress images

11zon (Free)

<https://imagecompressor.11zon.com/>

Free Convert (Free)

<https://www.freeconvert.com/>

Note

I recommend bouncing the images you are compressing through multiple services, compressing the image multiple times.

It's best to start with a high quality picture that has a large file size, because when you run it through a compressor multiple times it degrades the quality with each pass.

You can get your images to under 100 kb through this process, even if the image starts at 4-9 MB. It's certainly a hassle to do this, but it does work. The quality degrades on each compression run, so it's best to start with an ultra crisp high quality image so the degradation isn't noticeable. Once you start getting to the low hundreds check the image quality after each compression run to ensure it still looks acceptable.

Some compression services are better than others and will produce better quality images at equivalent file sizes.

When you notice that the image quality degrades past the point where you find it acceptable(photo becomes too grainy), use the compressed image with the lowest file weight (most compressed version) of the image that is still visually high quality.

Compressed images should ideally be under 100 kb if possible. The lower the file size, the better. A **300-400 kb file weight difference could be the difference from a score in the low 80s vs a score in the high 90s (e.g. 490 kb original file weight vs 90 kb for the compressed image).**

Image file weight can definitely make that big of a difference in your pagespeed scores, so compress your images as much as possible!

Image Compression Plugins

Image Quality Control (Free)

<https://wordpress.org/plugins/still-be-image-quality-control/>

Neat little plugin, very customizable, unlimited free usage.

Bulk Image Optimizer (Free)

<https://wordpress.org/plugins/bulk-image-resizer/>

Compresses images, resizing, and converts to WebP. All completely free, zero image size limits, all done on the host server, no third party services.

Smush (Free and Premium)

<https://wordpress.org/plugins/wp-smushit/>

One of the best image optimization plugins. Its free tier is extremely generous(unlimited images under 5 MB), and has a full suite of image optimization features. Smush Pro helps due to allowing image optimization to run in the background, which is necessary if you have a large amount of images on your site(if you choose Smush)

PhastPress (Free)

<https://wordpress.org/plugins/phastpress>

Image optimization:

- Phast optimizes images using PNG quantization (pngquant) and JPEG recoding (libjpeg-turbo).
- Phast inlines small images (< 512 bytes) in the page.
- Phast converts JPEG images into WebP for supporting browsers.
- Phast enables native lazy loading to speed up page loading and save bandwidth.

Flying Images (Free)

<https://wordpress.org/plugins/nazy-load/>

Free multi-function image optimization plugin with on-the-fly image compression. Free unlimited usage.

ShortPixel (Freemium)

<https://wordpress.org/plugins/shortpixel-image-optimiser/>

Themelsle Article

<https://themeisle.com/blog/wordpress-image-optimizer-plugins-compared/#image-compression-winner>

ShortPixel is the best image optimization plugin according to Themelsle's comparison tests.

	Optimole	Imagify	Smush	EWWW	ShortPixel
Total Score	25	10	6	9	11
Average Image Compression	90%	42%	27%	21%	59%
Average Mobile PageSpeed Score	82.4	83	83.6	81.3	90.6
Average Desktop PageSpeed Score	97.3	97.3	97	96.8	96.9
Average Mobile LCP (seconds)	4.62	4.19	4.34	4.75	3.31
Average Desktop LCP (seconds)	0.92	1.13	1.17	1.2	1.22

Mobile is the optimization target, and ShortPixel wins by a mile. While ShortPixel's Image Compression is apparently inferior to Optimole's, ShortPixel has better overall results and results in a greater performance improvement over Optimole.

Resmush it (Free)

<https://wordpress.org/plugins/resmushit-image-optimizer/>

“ReSmush.it Image Optimizer allows you to use free Image optimization based on the [reSmush.it API](#). reSmush.it provides image size reduction based on several advanced algorithms. The API accepts JPG, PNG and GIF files up to 5MB.”

CompressX (Free)

<https://wordpress.org/plugins/compressx/>

Free image compression and conversion, fully local.

WP Vivid Image Optimization (freemium)

<https://wordpress.org/plugins/wpvivid-imgoptim/>

Fully featured optimization plugin, the majority of optimization features enabled for free. 2000 images/month, any image size under 5 MB. Highly recommend checking this out and comparing with the other options.

WP-Optimize (Free)

<https://wordpress.org/plugins/wp-optimize/>

Free Bulk Image Compression

Reduce Colors

Reducing the amount of colors used in an image will reduce its file size. It is possible to remove some without harming image quality, it will take some trial and error to find what's within an acceptable threshold. The amount of colors that can be removed without damaging image quality will vary image to image.

Online PNG Tools (Free)

Decrease PNG Color Count

<https://onlinepngtools.com/decrease-png-color-count>

Reduce WebP Color Count

<https://onlinetools.com/webp/reduce-webp-colors>

Reduce JPG Colors

<https://onlinejpgtools.com/reduce-jpg-colors>

COD (Compress-Or-Die) (Free)

Installable Compression Software

Installable compression software on your host device (Windows, Mac, Linux)

LibVips (Free)

<https://github.com/libvips/libvips>

LibVips is an image compression and optimization library with a multitude of features, which extends beyond just simple image compression, allowing you to compress/optimize your images deeply and granularly.

Official GUI

Nip2 (Free)

<https://github.com/libvips/nip2>

WSRV (Free)

<https://wsrv.nl/>

<https://github.com/weserv/images>

wsrv.nl is an image cache & resize service. Their servers resize your image, cache it worldwide, and display it. Uses the LibVips library, extends it with more features.

ImgProxy (Free)

<https://github.com/imgproxy/imgproxy>

Uses the LibVips library, it has a multitude of image optimization features.

Documentation

<https://docs.imgproxy.net/>

Lazy Load Images and Videos

Lazyload will prevent images that are below the fold(out of the user's viewport when the site loads) from loading until the picture enters the user's viewport(i.e. they can see it on screen), which reduces the weight of pages that contain images, which will significantly speed up their load time. This is crucial if you have a very image heavy website.

Exclude Above The Fold Images From Lazy Load

Above the fold images(images that are loaded within the viewport without scrolling) should be excluded from Lazy Load. Only Lazy Load images below the fold. Most plugins can auto-detect image locations, but you may need to add manual exclusions. Perfmatters does a good job (which is what I use) of auto-excluding images it can detect, but it isn't perfect.

Every lazy load plugin is functionally the same as far as excluding images. There will be a text box where you can paste the image filename to exclude it from Lazy Loading, just check the settings.

Auto-sizes for Lazy-loaded Images (Free)

<https://wordpress.org/plugins/auto-sizes/>

This plugin implements the HTML spec for adding sizes="auto" to lazy-loaded images. For background, see: <https://github.com/whatwg/html/issues/4654>

Flying Images (Free)

<https://wordpress.org/plugins/nazy-load/>

Has a lazyload feature, both browser native and javascript lazy load.

Rocket Lazyload (Free)

<https://wordpress.org/plugins/rocket-lazy-load/>

A3 Lazyload (Free)

<https://wordpress.org/plugins/a3-lazy-load/>

Paid Optimization Plugins

All paid caching plugins have the ability to lazy load images. Many, many image optimization plugins have it as well.

Image Converters

Converter for Media (Free)

<https://wordpress.org/plugins/webp-converter-for-media/>

Auto Converts images to WebP.

Batch Image Format Converter

XnConvert (Free)

<https://www.xnview.com/en/xnconvert/>

XnConvert is a fast, powerful and free cross-platform batch image converter. You can convert and compress your images, photos and pictures easily, and apply over 80 actions (like resize, crop, color adjustments, filter, and more, all common picture and graphics formats are supported (JPEG, TIFF, PNG, GIF, WebP, PSD, JPEG2000, JPEG-XL, OpenEXR, camera RAW, HEIC, PDF, DNG, CR2). You can save and re-use your presets for another batch image conversion.

Select the right image size

Once you have uploaded the file to Wordpress, Wordpress will automatically create various sizes of your images. If you choose one of the smaller sizes that it creates (medium, medium-large etc), if you are using a page builder such as Elementor, you can select the smaller size (if appropriate) and the file size will be reduced further.

You can also use an adaptive/responsive images plugin to automatically resize them.

Make sure the image is of acceptable quality before you put it live on your site.

Flying Images (Free)

<https://wordpress.org/plugins/nazy-load/>

Flying Images for image compression, WebP and lazyload(free):

You can use Flying Images to compress and optimize my images.

It automatically serves them from an image based cdn, and auto-converts to the WEBP format on the fly for supported browsers. Do not install Flying Images if you are using WP-Rockets lazyload instead of Flying Images lazyload, it will load the media files from your site instead of the statically cdn. **If you want to use Flying Images/the statically cdn, Lazy Load must be disabled in other plugins.** You should only be using one plugin for Lazy Loading functionality.

Fix Image Related CLS Issues

Add missing width and height attributes to your images

Adding missing height and width attributes to your images will help reduce the Content Layout Shift (CLS) page speed metric.

Perfmatters (Paid)

<https://perfmatters.io/docs/missing-width-height-images/>

Flyingpress (Paid)

<https://docs.flyingpress.com/en/article/add-missing-width-and-height-i1n2w9/>

Adding missing height and width dimensions to your images will significantly reduce Content Layout Shift(CLS) on Page Speed scores.

Optimize More! Images (Free)

<https://wordpress.org/plugins/optimize-more-images/>

Adds missing height and width attributes to improve CLS scores.

Specify Missing Image Dimensions (Free)

<https://wordpress.org/plugins/specify-missing-image-dimensions/>

Free plugin, the only feature is adding missing image dimensions

JCH Optimize (Free)

<https://wordpress.org/plugins/jch-optimize/>

Multiple function optimization plugin that can add missing height and width attributes

Phastpress (Free)

<https://wordpress.org/plugins/phastpress/>

Multi-function optimization plugin, can also add missing height and width attributes

However, you need to add image exclusions in the exclusion box to all the images that load in the above the fold content so that they are not lazy loaded in WP-Rockets settings if you use their lazy-load.

You do not want to lazyload the files that are above the fold, as that will artificially slow down their load times because they will load after the lazyload js file. This will negatively impact your performance if you do not add the exclusions, so don't forget! As long as you don't forget to add exclusions, lazyload will significantly boost your site's performance.

Image Resizing

Imsanity (Completely free)

<https://wordpress.org/plugins/imsanity/>

Features:

1. Automatically scales large image uploads to a more “sane” size
2. Bulk resize feature to resize existing images
3. Selectively resize images directly in the Media Library (List View)
4. Allows configuration of max width/height and JPG quality
5. Optionally converts BMP and PNG files to JPG for more savings
6. Once enabled, Imsanity requires no actions on the part of the user
7. Uses WordPress built-in image scaling functions

Bulk Image Optimizer (Completely Free)

<https://wordpress.org/plugins/bulk-image-resizer/>

Compresses images, resizing, and converts to WebP. All completely free, zero image size limits, all done on the host server, no third party services.

Simple Image Sizes (Free)

<https://wordpress.org/plugins/simple-image-sizes/>

This plugin allows create custom image sizes for your site. Override your theme sizes directly on the media option page.

Better Image Sizes (Completely Free)

<https://wordpress.org/plugins/better-image-sizes/>

This plugin allows you to:

- disable auto generated image sizes
- dynamically generate only needed image sizes for only needed images
- delete generated image sizes from this plugin individually or all together
- specify different focal point that will be in the focus while creating cropped versions of the image
- automatically detect focal point by face detection algorithm
- disable big image size threshold, so image bigger than 2560x2560px will be not be scaled down

ImgProxy (Free)

<https://github.com/imgproxy/imgproxy>

Imgproxy is a fast and secure standalone server for resizing, processing, and converting images. The guiding principles behind imgproxy are security, speed, and simplicity.

Adaptive Images

Adaptive Images (Free)

<https://wordpress.org/plugins/adaptive-images/>

Free adaptive image generation plugin

ShortPixel Adaptive Images (Freemium)

<https://wordpress.org/plugins/shortpixel-adaptive-images/>

Enable Responsive Images for Gutenberg (Free)

<https://wordpress.org/plugins/enable-responsive-image/>

Flying Images (Free)

<https://wordpress.org/plugins/nazy-load/>

Free Image Optimization plugin with nice features such as free image CDN, on the fly image compression, on the fly webP conversion, and adaptive images.

Preload Images

Lazyload, Preload and More (Free)

<https://wordpress.org/plugins/lazyload-preload-and-more/>

Can add the Preload attribute as well as the loading eager attribute.

Preload is more powerful than Fetch Priority and takes precedence over the Fetch Priority attribute. If you need an image to download as soon as possible, use Preload over Fetch Priority.

Preload Featured Images (Free)

<https://wordpress.org/plugins/preload-featured-images/>

Automatically preloads featured images

Pre-Party (Free)

<https://wordpress.org/plugins/pre-party-browser-hints/>

Pre-Party can add a variety of loading attributes, including the preload attribute.

Perfmatters (Paid)

<https://perfmatters.io/docs/preload/>

Perfmatters has an integrated Preload function for images

Image Fetch Priority

Custom Image Attributes (Free)

<https://wordpress.org/plugins/custom-image-attributes/>

Another plugin to add image attributes(including fetch priority).

Image Prioritizer (Free)

<https://wordpress.org/plugins/image-prioritizer/>

This plugin optimizes the loading of images which are the LCP (Largest Contentful Paint) element, including both img elements and elements with CSS background images (where there is a style attribute with a background-image property). Different breakpoints in a theme's responsive design may result in differing elements being the LCP element. The LCP element for each breakpoint is captured so that high-fetchpriority preload links with media queries are added which prioritize loading the LCP image specific to the viewport of the visitor.

Fetch Priority Plugin (Free)

<https://wordpress.org/plugins/fetchpriority/>

Fetch Priority Plugin made by the Wordpress Performance Team. Automatically applies the fetch priority attribute to a single image determined automatically. There are no settings and nothing to configure.

If you need granular control over the Fetch Priority Attribute, I recommend purchasing Perfmatters.

Perfmatters (Paid)

<https://perfmatters.io>

<https://perfmatters.io/docs/fetch-priority/>

Perfmatters gives you explicit control over which images have the fetch priority attribute

SpeedSize (Paid)

<https://wordpress.org/plugins/speedsize-ai-image-optimizer/>

Interesting unique features. Could be good, haven't had a chance to test it yet.

Image Decoding

Wordpress Image Async

<https://kyberaut.cz/en/clanky/the-unintended-consequences-of-decodingasync-on-your-wordpress-sites-lcp/>

Inline Image Base64 (Free)

<https://wordpress.org/plugins/inline-image-base64/>

With Inline Image Base64 you can inline specific images directly into the HTML as base64.

Note: Only use this with extremely small images that are less than 20-30 kb(ideally under10 kb). Keep the file size of the inlined images even lower than that if possible.

Thumbnail Regenerations

Regenerate Thumbnails (Free)

<https://wordpress.org/plugins/regenerate-thumbnails/>

Regenerate Thumbnails allows you to regenerate all thumbnail sizes for one or more images that have been uploaded to your Media Library.

WebP and AVIF

CompressX (Free)

<https://wordpress.org/plugins/compressx/>

WebP and AVIF conversion plugin from the WPVivid team. Completely free, and the compression and conversion is done locally on your site.

QuickWebP (Free)

<https://wordpress.org/plugins/quickwebp/>

Completely free WebP converter plugin. Fully converts to WebP, no fallback.

PhastPress (Free)

<https://wordpress.org/plugins/phastpress>

Image optimization:

- Phast optimizes images using PNG quantization (pngquant) and JPEG recoding (libjpeg-turbo).
- Phast inlines small images (< 512 bytes) in the page.
- Phast converts JPEG images into WebP for supporting browsers.
- Phast enables native lazy loading to speed up page loading and save bandwidth.

Next-Gen Image Format Converters

WebP Express (Free)

<https://wordpress.org/plugins/webp-express/>

Serve auto-generated WebP images instead of jpeg/png to browsers that support WebP.

WebP Express Plus (Paid)

<https://wordpress.org/plugins/webp-express-plus/>

Addon for WebP Express which adds additional features. Requires the WebP Express plugin to be active.

AVIF Express (Free)

<https://wordpress.org/plugins/avif-express/>

On upload, images are converted to AVIF format. AVIF Images are a new format and not supported on all browsers. Fallback images are in WebP format.

Remote Image Thumbnails (Free)

<https://wordpress.org/plugins/wp-remote-thumbnail/>

WP Remote Thumbnail is a very lightweight WordPress plugin that comes with zero configuration and one-click setup. Using this plugin, you can easily set any external image as the featured image of any post/page or even custom post types.

Note: Make sure to use together with a resource hints plugin (pre-party) to add preload and preconnect attributes to the third party domain.

Dynamic Image Caching

Local Storage with Blobs

IndexedDB

Implementation Steps

Open a Database: Use JavaScript to open an IndexedDB database. Create an object store for your images if it doesn't already exist.

Fetch and Store Images: When you fetch an image from the remote gallery, write it to the IndexedDB store. You might want to key it based on a unique identifier, such as a URL or a custom ID.

Check Cache First: Before fetching an image, check if it exists in your IndexedDB cache. If it does, load it from there.

Update Strategy: Decide on a strategy for updating cached images. For real-time photos, you might need a periodic check to see if newer versions of the images are available.

Image Filters

Easy Image Filters (Free)

<https://wordpress.org/plugins/easy-image-filters/>

Free Lightweight Plugin to add image filters

Free High Quality Images

Instant Images (Free)

<https://wordpress.org/plugins/instant-images/>

Free high quality images

Image Watermarking

Image Watermark (Free)

<https://wordpress.org/plugins/image-watermark/>

Ultra Lightweight Image Watermarking Plugin.

Icon Optimization

Free SVG icons

You can download free SVGs from here:

FontAwesome (Free)

<https://fontawesome.com/download>

FlatIcon (Free)

<https://www.flaticon.com/>

IconFinder (Free)

<https://www.iconfinder.com/>

Compress SVGs

Use both Vecta.io and SVGOMG together to reduce the size of your SVG icons as much as possible.

Vecta.io (Free)

<https://vecta.io>

SVG Compressor

SVGOMG (Free)

<https://jakearchibald.github.io/svgomg/>

Another SVG Optimizer. I recommend using it together with Vecta.io

Video Optimization

Lazy Load Iframes

Lazy Embed (Free)

<https://wordpress.org/plugins/lazy-embed/>

The Lazy Embed plugin defers the loading of any resource required for playing the video until the video is requested to be played. It does this by adding a srcdoc attribute to the iframe which shows in place of the normal iframe content.

Other plugins

Many plugins aside from Lazy Embed can lazy load video iframes. Perfmatters, Flyingpress, and a wide variety of other free plugins can do so as well.

Video Compression

Handbrake (Free)

<https://handbrake.fr/>

An open source installable program which can compress your videos.

Convert videos to WebM to further compress video file size

WebM video files are ~30% smaller than MP4. Handbrake allows you to convert your videos to WebM format. WebM is the video equivalent of WebP images.

Free Convert Video Compressor (Free)

<https://www.freeconvert.com/video-compressor>

10web.io Video Optimization article

<https://10web.io/blog/optimizing-videos-on-your-wordpress-site/>

Great article with further optimization tips for sites with videos.

Video Player Plugins

Presto Player (Free)

<https://wordpress.org/plugins/presto-player/>

Lightweight Video Player. HTML5, Youtube and Vimeo video support.

Super Video Player (Free)

<https://wordpress.org/plugins/super-video-player>

Super Video Player is a lightweight, fully customizable self-hosted video player plugin that supports MP4/OGG as well as multiple caption tracks or subtitle files.

HTML5 Video Player (Free)

<https://wordpress.org/plugins/videojs-html5-player/>

Another Lightweight Video Player.

Video Pack (Free)

<https://wordpress.org/plugins/video-embed-thumbnail-generator/>

Lightweight Video Player, local hosting.

Lottie Animation Compressor

Lightweight Lottie Animation Plugin

AM Lottie Player (Free)

<https://wordpress.org/plugins/am-lottieplayer/>

The most complete free Lottie Player yet.

AM LottiePlayer is easy to use, lightweight, and gives you total control over how to implement crisp, vectorized animations to your website.

Lottie Optimization

Lottiemizer (Free)

<https://www.lottiemizer.com/>

Compresses your Lottie animations

Lottie JSON Optimizer (Free)

<https://lottiefiles.com/tools/lottie-json-to-optimized-lottie-json>

Optimize your Lottie JSON files.

dotLottie (Free)

<https://dotlottie.io/>

dotLottie is an open-source file format that aggregates one or more Lottie files and their associated resources into a single file. They are ZIP archives compressed with the Deflate compression method and carry the file extension of ".lottie".

How To Make Interactive Lottie Animations

Youtube Tutorial (Free)

<https://www.youtube.com/watch?v=-bGTLnGbZo>

Tutorial on how to make Lottie's interactive

Gifs

Free Convert (Free)

<https://www.freeconvert.com/gif-compressor>

Ezgif (Free)

<https://ezgif.com/optimize>

Another gif compressor

Gifcompressor (Free)

<https://gifcompressor.com/>

Gif Compressor

Xconvert (Free)

<https://www.xconvert.com/compress-gif>

ILoveIMG (Free)

<https://www.iloveimg.com/compress-image/compress-gif>

Veed.io (Free)

<https://www.veed.io/tools/video-compressor/gif-compressor>

Animate.ly (Free)

<https://animately.co/tools/compress-gif>

Optimize Embeds

Embed Optimizer (Free)

<https://wordpress.org/plugins/embed-optimizer/>

This plugin's purpose is to optimize the performance of embeds in WordPress, such as YouTube videos, TikToks, and so on. Initially this is achieved by lazy-loading them only when they come into view.

This improves performance because embeds are generally very resource-intensive and so lazy-loading them ensures that they don't compete with resources when the page is loading. Other optimizations are planned for the future.

Themes

Pagebuilder users

Your theme is slowing you down

If you are using a Page Builder, **themes are dead weight**. They add absolutely nothing to the design since all of the css and js necessary for a page is generated by the page builder itself. Astra for example will insert hundreds of kb of useless unnecessary code into your html, css and js when you don't need a single line of it. And gutting a heavy theme is problematic in itself. Even when all of the theme's functionality is unused, disabling theme generated css and js files will *still* break your site's design and functionality.

The theme system is a core part of Wordpress and cannot be disabled, so you cannot be truly "themeless". You should select the most optimized theme that loads the least code from the outset, to minimize the impact of the theme system as much as possible. You will still need to remove unused css and delay js with a plugin, but your results will be much better.

If you're using a builder, build the page from scratch or with imported templates directly into the builder. For Elementor for example those would be "template kits" or stuff from the template library.

Never use theme addons, or any feature provided by your theme, and disable settings that would impact performance from the settings panel if possible. All of that adds bloat that will load on the frontend. Plugin addons installed via theme (generally happens with paid themes) are fine if they are well optimized, but theme specific features should be avoided.

My recommendation is to use either Hello Elementor or Picostrap. These two load very little code in the frontend. My personal preference is Hello Elementor, the entire plugin zip is 3x smaller than Picostrap and loads very little code. Picostrap is definitely a fantastic option however, and both are incredibly fast. I recommend trying both of them to see which fits your use case best. I haven't used Picostrap since v4 (it's currently on v5), so I'm sure its performance has improved even further.

Performance optimized themes

Hello Elementor (Free)

<https://wordpress.org/themes/hello-elementor/>

Hello Elementor is a very, very light barebones theme.

Picostrap (Free)

<https://picostrap.com/>

Picostrap is a very lightweight theme. If you are just starting to build your site, I highly recommend Picostrap.

Themes to avoid

Astra (Free and Pro)

I can no longer recommend Astra. Avoid Astra for new builds if your site is not already built on it. Astra adds an insanely large amount of css and javascript which nukes pagespeed performance. It adds over 300 kb of js/css on a beaver builder site I've optimized. Delaying the js will also break functionality(header, menus etc). Do not use it whenever possible.

Optimizing Astra

If you must use Astra, here are some steps that can be taken to optimize Astra performance.

You do not need the pro version, it will make your site slower.

These are the optimal settings for Astra that you should enable with Asset Cleanup in my experience. YMMV, test thoroughly and re-enable files if necessary:

The screenshot displays the Asset Cleanup plugin interface for two files from the Astra theme. The top section is for a CSS file: `wp-content/themes/astra/assets/css/minified/style.min.css`. It is currently set to 'Preload (if kept loaded)? Yes, async' and '* applies site-wide'. The location is '<HEAD> * initial'. Below this, there are three toggle buttons: 'Unload on this page' (which is currently turned on), 'Unload site-wide * everywhere', and 'Unload on All Pages of "page" post type'. There is also a checkbox for 'Unload it for URLs with request URI matching this RegEx(es):'. The bottom section is for a JavaScript file: `wp-content/themes/astra/assets/js/minified/style.min.js`. It is set to 'Preload (if kept loaded)? No (default)' and '* applies site-wide'. The location is '<BODY> * initial'. Similar to the CSS file, it has three toggle buttons: 'Unload on this page' (turned on), 'Unload site-wide * everywhere', and 'Unload on All Pages of "page" post type * bulk unload'. There is also a checkbox for 'Unload it for URLs with request URI matching this RegEx(es):'. At the bottom, there is a section for 'Set everywhere * site-wide' with three radio buttons: 'Keep rule' (selected), 'Remove rule', and 'not here (exception)'. There is also a checkbox for 'on this page' which is checked.

Source: `wp-content/themes/astra/assets/css/minified/style.min.css` → Preload (if kept loaded)? **Yes, async** * applies site-wide ?

Version: 3.0.2 / Location: `<HEAD> * initial` * applies site-wide / File Size: 77.05 KB

☒ Unload on this page ☐ Unload site-wide * everywhere ☐ Unload on All Pages of "page" post type

☐ Unload it for URLs with request URI matching this RegEx(es): ?

Inline styling associated with the handle: **Show / Hide +**

If kept loaded, make the browser download the file `on any screen size (default)` ?

No notes have been added about this stylesheet file (e.g. why you unloaded it or decided to keep it loaded) → **Add Note**

Handle: **astra-theme-js** * JavaScript (js)

Source: `wp-content/themes/astra/assets/js/minified/style.min.js` → Preload (if kept loaded)? **No (default)** * applies site-wide ?

Version: 3.0.2 / Location: `<BODY> * initial` * applies site-wide / File Size: 10.35 KB

☒ Unload on this page ☐ Unload site-wide * everywhere ☐ Unload on All Pages of "page" post type * bulk unload

☐ Unload it for URLs with request URI matching this RegEx(es): ?

Inline JavaScript code associated with the handle: **Show / Hide +**

If kept loaded, set the following attributes:

async → ☒ on this page * locked by site-wide rule **Set everywhere * site-wide** ☒ Keep rule ☐ Remove rule ☐ not here (exception)

Handle: **woocommerce-general** * Stylesheet (.css) ⓘ *This file has its initial location changed.*

Source: </wp-content/themes/astra/assets/css/minified/compatibility/woocommerce/woocommerce.min.css>

→ Preload (if kept loaded)? **Yes, async** * applies site-wide ⓘ

Version: 3.0.2 / Location: <BODY> ∨ * applies site-wide / File Size: 103.11 KB

Unload on this page

Unload site-wide * everywhere

Unload on All Pages of "page" post type

Unload it for URLs with request URI matching this RegEx(es): ⓘ

Inline styling associated with the handle: Show / Hide +

If kept loaded, make the browser download the file on any screen size (default) ∨ ⓘ

No notes have been added about this stylesheet file (e.g. why you unloaded it or decided to keep it loaded) → 📝 Add Note

Handle: **woocommerce-layout** * Stylesheet (.css) ⓘ *This file has its initial location changed.*

Source: </wp-content/themes/astra/assets/css/minified/compatibility/woocommerce/woocommerce-layout.min.css>

→ Preload (if kept loaded)? **Yes, async** * applies site-wide ⓘ

Version: 3.0.2 / Location: <BODY> ∨ * applies site-wide / File Size: 17.12 KB

Unload on this page

Unload site-wide * everywhere

Unload on All Pages of "page" post type

Unload it for URLs with request URI matching this RegEx(es): ⓘ

Handle: **woocommerce-smallscreen** * Stylesheet (.css)

Source: </wp-content/themes/astra/assets/css/minified/compatibility/woocommerce/woocommerce-smallscreen.min.css> 📺 only screen and (max-width: 921px)

→ Preload (if kept loaded)? **No (default)** ∨ * applies site-wide ⓘ

Version: 3.0.2 / Location: <HEAD> * initial ∨ * applies site-wide / File Size: 5.84 KB

This stylesheet file is unloaded site-wide (everywhere).

● Keep site-wide rule Remove site-wide rule

All other unload rules (e.g. per page, RegEx) are overwritten by this site-wide rule.

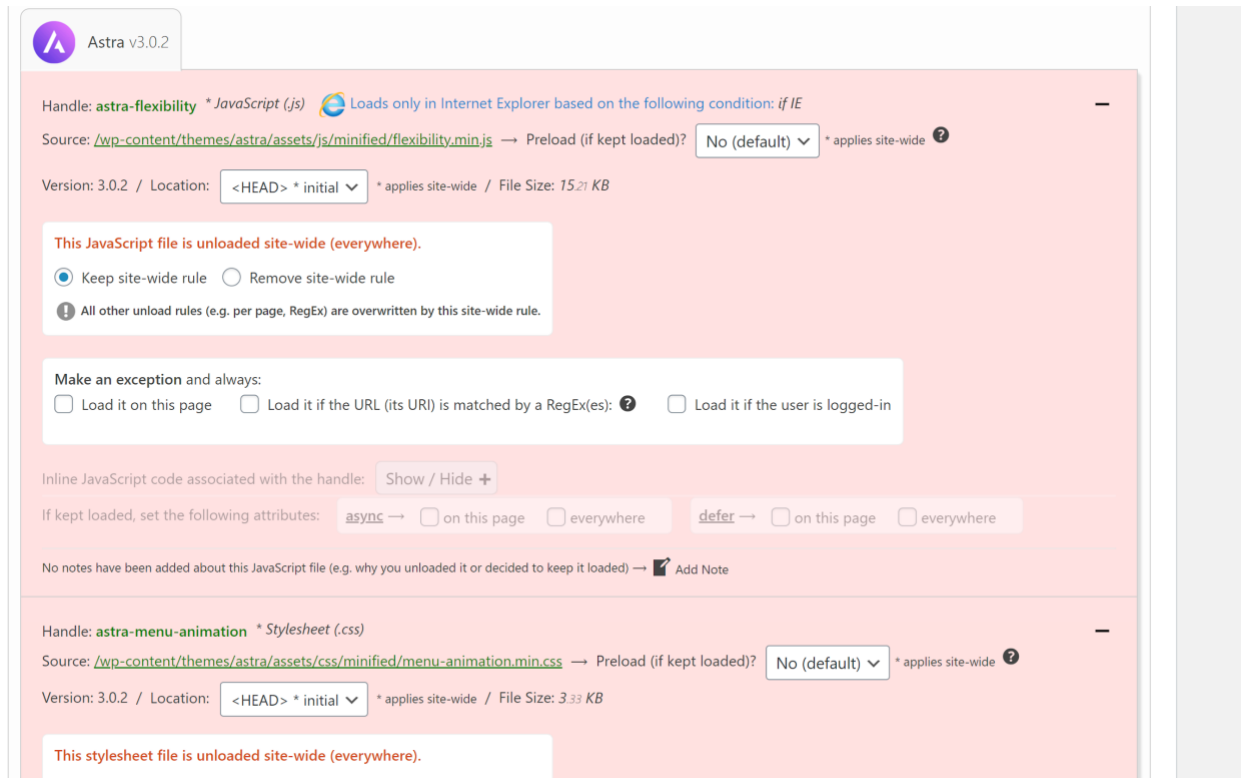
Make an exception and always:

Load it on this page

Load it if the URL (its URI) is matched by a RegEx(es): ⓘ

Load it if the user is logged-in

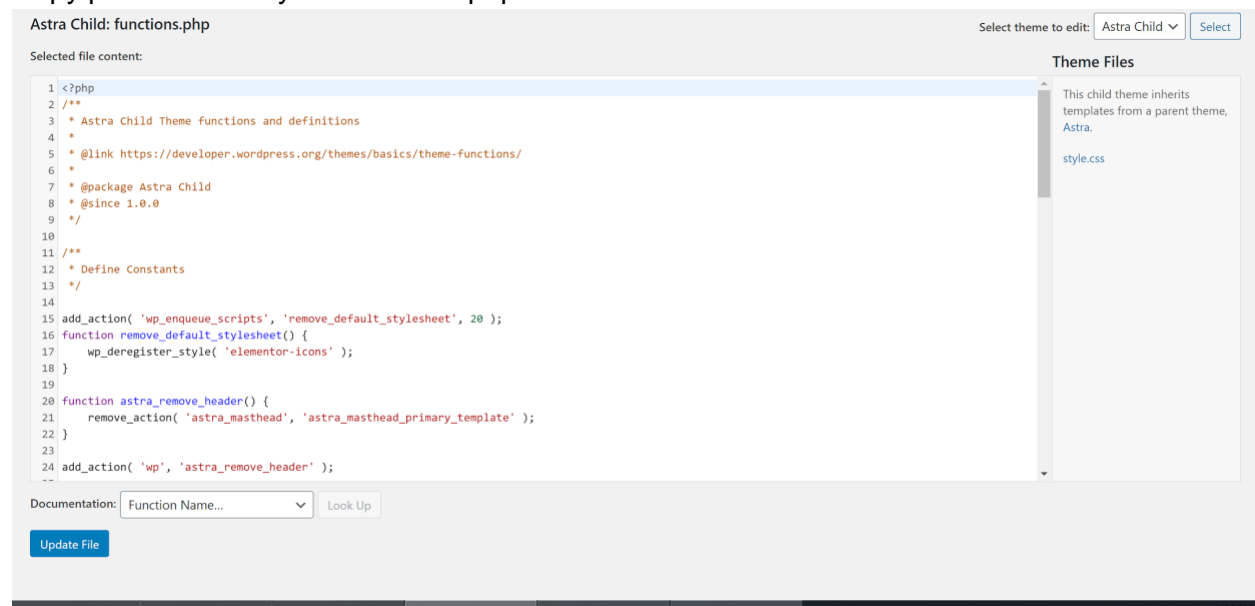
No notes have been added about this stylesheet file (e.g. why you unloaded it or decided to keep it loaded) → 📝 Add Note



You should also download the astra child theme:

<https://wpastra.com/child-theme-generator/>

Copy paste this into your functions.php in theme editor:



Put these lines right under where it says “define constants”

```
function astra_remove_header() {  
    remove_action( 'astra_masthead', 'astra_masthead_primary_template' );  
}
```

```
add_action( 'wp', 'astra_remove_header' );
```

```
add_filter( 'astra_schema_enabled', '__return_false' );
```

```
add_filter( 'astra_meta_box_options', 'default_disable_options' );
```

```
/**
```

```
 * Default disable the Meta Options
```

```
 *
```

```
 * @param array $meta_option Page Meta.
```

```
 * @return array
```

```
 */
```

```
function default_disable_options( $meta_option ) {
```

```
    $meta_option['ast-main-header-display'] = array(  
        'default' => 'disabled',  
        'sanitize' => 'FILTER_DEFAULT',  
    );
```

```
    $meta_option['footer-sml-layout'] = array(  
        'default' => 'disabled',  
        'sanitize' => 'FILTER_DEFAULT',  
    );
```

```
    $meta_option['footer-adv-display'] = array(  
        'default' => 'disabled',  
        'sanitize' => 'FILTER_DEFAULT',  
    );
```

```
    $meta_option['site-post-title'] = array(  
        'default' => 'disabled',  
        'sanitize' => 'FILTER_DEFAULT',  
    );
```

```
    $meta_option['site-sidebar-layout'] = array(  
        'default' => 'disabled',  
        'sanitize' => 'FILTER_DEFAULT',  
    );
```

```
    $meta_option['site-content-layout'] = array(  
        'default' => 'disabled',  
        'sanitize' => 'FILTER_DEFAULT',  
    );
```

```
    $meta_option['ast-featured-img'] = array(  
        'default' => 'disabled',  
        'sanitize' => 'FILTER_DEFAULT',  
    );
```

```
    $meta_option['ast-featured-img'] = array(  
        'default' => 'disabled',  
    );
```



```
'sanitize' => 'FILTER_DEFAULT',
);
$meta_option['ast-breadcrumbs-content'] = array(
    'default' => 'disabled',
    'sanitize' => 'FILTER_DEFAULT',
);

return $meta_option;
}

add_filter( 'astra_amp_support', '__return_false' );
```

Performance Optimized Page Builders

Oxygen Builder (Paid)

<https://oxygenbuilder.com/>

Oxygen Builder is a well optimized page builder that allows you a lot of flexibility, while also producing clean, optimized code that is pre-optimized for page speed. Far less unnecessary div tags, too many div tags leads to a bloated DOM. A bloated DOM will increase the amount of time to parse your HTML, which increases your TTFB.

Bricks Builder (Paid)

<https://bricksbuilder.io/>

Bricks is another well optimized page builder that outputs clean code.

Live Canvas (Paid)

<https://livecanvas.com/>

LiveCanvas: the only page builder that outputs super clean, Bootstrap 5 - flavored HTML5 code, which you can always edit and refine via a built-in live code editor.

Live Canvas is extremely lightweight and creates pre-optimized code. Highly recommend considering Live Canvas as an option.

Live Canvas is best paired with Picostrap, a performance optimized theme made by the same developers:

Picostrap (Free)

<https://picostrap.com/>

Performance optimized theme created by the Live Canvas dev team.

Elementor (Freemium)

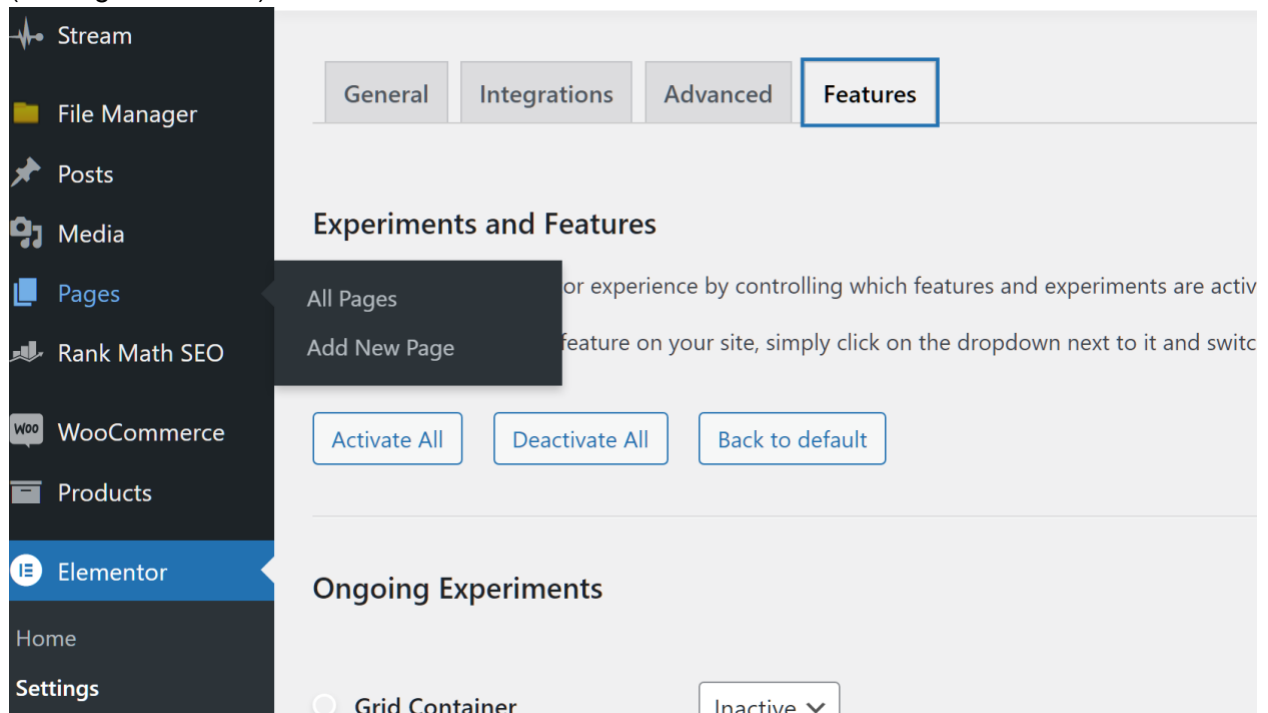
<https://elementor.com/>

Contrary to popular belief, **Once Optimized**, Elementor is very performant. Unoptimized Elementor on the other hand has very poor performance. While Elementor is very bloated pre-optimization, it is one of the lightest weight builders post optimization, and it is very easy to optimize.

Plugins to Remove Unused CSS and delay Javascript have built in exclusions for Elementor (except Flyingpress, no built in exclusions with a checkbox), which means optimizing Elementor core and Elementor Pro CSS and JS assets is a breeze. Debloat and Perfmatters do have a way to apply automatic exclusions when Delay All JS functionality is used.

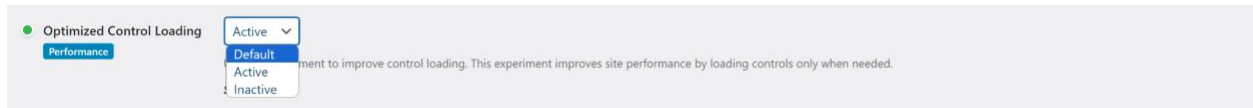
Performance Experiments

Elementor has many built in performance enhancement features that are in the features tab (Settings>Features)



Many of these are not activated by default, especially if you have a pre-existing site when they were added in an update. That means many Elementor sites are leaving free performance on the table which can be improved by simply changing the experiment from inactive to active in the dropdown!

For example, Elementor just added the optimized control loading recently, which can have a big performance improvement for large sites. This feature is not enabled by default, and must be manually enabled.



Note: It is fine to activate Elementor Lazy Loading functionality on top of another optimization plugin. Personally, I haven't found the native Elementor lazy loading to work all that well, so I still default to the lazy loading options from performance optimization plugins. However pairing it with Elementor's native functionality will not hurt performance.

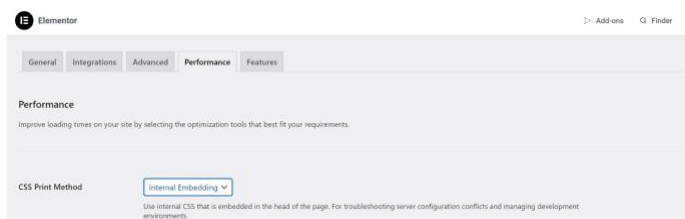
Other than that, I would recommend you run a pagespeed test to see your baseline scores before you enable each performance feature, and then another test after the feature is enabled to compare.

Limit use of Containers, Sections and Columns

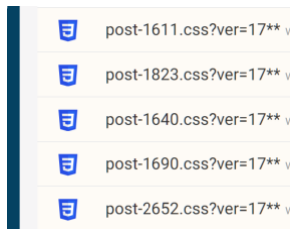
Every additional Container/Section/Column adds additional DOM elements and HTML weight to your page. While containers/sections/columns are sometimes easier to use in design, they negatively affect pagespeed. Containers(the replacement for sections and columns) are very flexible and multiple elements can be positioned inside them. If desired, use the grid format containers to mimic columns.

CSS Print Method

Internal Embedding

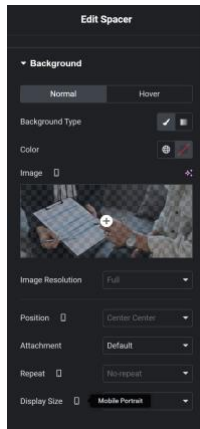


Enabling the Internal Embedding CSS Print Method in Elementor settings will inline some of the CSS that a Remove Unused CSS feature will miss. It is recommended that this be enabled only once all CSS optimizations have been applied.



If you see post-xxxx.css files in your request tree, changing the CSS print method will eliminate those requests by inlining them into the HTML document. If too much CSS is inlined this can actually have the reverse of the intended effect and hurt performance, so make sure to run Pagespeed tests before enabling the feature and once it has been enabled to check to see whether its impact is beneficial. Remove Unused CSS features will generally not target these files, so inlining them is required to eliminate those requests.

Do not set background images on unconventional elements



Background images set on unconventional elements **cannot be Lazy Loaded with optimization plugins as they will not be detected by their algorithms.** If an image is set to a background image on an element such as a spacer, Elementor's background image lazy load feature will catch **some** of them, but not all, which will negatively affect your pagespeed. Even Elementor's own Lazy Load will miss them for some reason.

You should only set images on widgets that contain a section that has an image picker OR as a regular background image **(not through the advanced section when editing an element)**.

If your images cannot be lazy loaded, it significantly negatively affects your pagespeed. **Always check in the network tab of inspect to see what files are loading when optimizing a site to see what files are loading. That applies to any site, whether it's built with Elementor or not.**

Inline SVG Icons Into The Root HTML Document (Free)

<https://github.com/jazir555/Inline-SVGs-uploaded-to-Elementor-Icon-Widget>

This is my personally developed plugin to inline SVG icons into the root HTML document for the **Elementor icon widget** (currently the only supported widget). This plugin is still in development, and will be expanded to cover all widgets/elements with locally hosted SVG icon capabilities.

Current functionality works as intended, and adds a toggle to the Advanced settings tab for the Icon widget, which can be activated on a per widget basis **(it is not global, which allows granular implementation)**.

Further subsequent updates to the plugin **(after the functionality has been expanded to all Elementor icon based widgets)** will include an options page in the Admin Dashboard which allows local hosting of any arbitrary SVG file, locally hosted or remote from any third party domain to eliminate any and all HTTP requests.

Note: This plugin will eventually be added to the Wordpress.org Plugin Repository once the Elementor functionality has been fully built out.

Elementor Panel Editor

Flexible Elementor Panel (Free)

<https://wordpress.org/plugins/flexible-elementor-panel/>

Very lightweight plugin which improves the Editor panel. Can safely be selectively deactivated in the frontend.

- Draggable Elementor panel and save position
- Move to right side and switch in RTL mode
- Vertical collapsible Elementor panel
- Resize Elementor panel and save size
- Multiple dark skin for Elementor editor
- Collapsible all widgets category
- Sortable widgets category and save position
- Add exit button in the footer of panel editor
- Option to close the first tab in Accordion widget
- Add more options for responsive mode

Editor Performance Optimization

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

Element Manager allows you to disable all native Elementor widgets.

Editor Cleanup for Elementor (Free)

<https://wordpress.org/plugins/editor-cleanup-for-elementor/>

Editor Cleanup For Elementor is an add-on of Freesoul Deactivate Plugins to clean up the editor of Elementor.

- It will not only clean up the assets of other plugins, their PHP code will not run either.
- The Elementor editor will be faster and without conflicts with other plugins.
- Both Freesoul Deactivate Plugins and Elementor must be installed and active, in another case this plugin will not run.

Restricted Content

QuentnWP (Free)

<https://wordpress.org/plugins/quentn-wp/>

Free content restriction plugin with Elementor integration.

Dynamic Visibility for Elementor (Free)

<https://wordpress.org/plugins/dynamic-visibility-for-elementor/>

Dynamic Visibility extension allows you to hide widgets, columns, containers, sections, or pages.

It is particularly useful when you are building something that you don't want to show everyone.

- Choose an OR/AND condition.
- Limit visibility for specific custom fields or pages.
- Set a date (from-to), day of the week, or an hour and decide when each element will be visible.
- Limit visibility for specific user roles, user meta, IP, referral, or users.
- Limit visibility for WooCommerce products
- Set a fallback text (e.g.: 'Coming soon') for hidden elements that will be displayed in place of the element.

- You can choose to hide the element via CSS or remove it from the DOM.

Advanced Queries (Free)

<https://wordpress.org/plugins/advanced-post-queries/>

An Elementor PRO extension that allows you to easily dynamically query your posts and portfolio widget results without custom query IDs.

Note: Requires Elementor Pro

Extra CSS Classes For Widgets

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

Fundamento (Free)

<https://wordpress.org/plugins/fundamento/>

Allows you to inject extra CSS classes into Elementor's widgets.

Clickable Columns and Sections

Make Sections & Columns Clickable (Free)

<https://wordpress.org/plugins/make-section-column-clickable-elementor/>

Ultra-lightweight plugin to make sections and columns clickable.

Elementor Free

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

Custom CSS

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

RRdevs (Free)

<https://wordpress.org/plugins/rrdevs-for-elementor/>

Plugin for Elementor free that adds custom css

Ultra Addons lite (Free)

<https://wordpress.org/plugins/ultraaddons-elementor-lite/>

Free addon pack with Custom CSS functionality.

Xpro Addons (Free)

<https://wordpress.org/plugins/xpro-elementor-addons/>

Another plugin for Elementor Free that enables custom CSS functionality.

Mini Cart for Woocommerce

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

Ultra Addons Lite (Free)

<https://wordpress.org/plugins/ultraaddons-elementor-lite/>

Includes a mini cart widget

Live Chat (Free)

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, Floating Buttons etc). Floating buttons in Elementor Pro have live chat functionality.

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

Simple Chat Plugin (Free)

<https://wordpress.org/plugins/simple-chat-button/>

Free ultra lightweight (no javascript!) Whatsapp live chat plugin

Indigital Live Chat (Free)

<https://wordpress.org/plugins/indigital-web-push-notifications/>

Ultra-lightweight Free Whatsapp Chat plugin.

Theme Builder

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

Xpro Theme Builder (Free)

<https://wordpress.org/plugins/xpro-theme-builder/>

Create custom layouts for post types, blog archives, WooCommerce products & categories, search pages, 404 pages, and more with Elementor Free. You can personalize your WooCommerce store experience by changing the default layout to match any style.

- Header
- Footer
- Singular Layouts
- Single Post
- Single Page
- Archive(Blog) Page
- 404 Page
- Search Page
- WooCommerce Single Product Page
- WooCommerce Cart Page
- WooCommerce Checkout Page
- WooCommerce Account Page
- Custom Post Single (CPT)
- Archive Layouts
- Blog Post Archive
- Category Archives
- Tag Archives
- Author Archive
- Custom Post Archive (CPT)
- Dynamic & Global Tags
- ACF Dynamic Tags
- ACF Group Tags
- Post Dynamic Tags
- Woo Dynamic Tags
- Term & Taxonomy Tags
- User & Site Tags
- Custom PHP Tag
- One-Click Template Importer

Dynamic Tags

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

Royal Addons (Free)

<https://wordpress.org/plugins/royal-elementor-addons/>

Adds dynamic tags functionality to Elementor free. It's more than that, a full dynamic website builder for Elementor

Use Dynamic Tags to create Websites Driven by Dynamic Content. Create Custom Post Types and Custom Taxonomies. Style and customize Custom Fields, Custom Post Type & Custom Taxonomies Templates. Add Dynamic Product Data even for Woocommerce Archive Grid and Single Products Pages and much more.

Supported Advanced Custom Fields: Text, Textarea, Number, Range, Email, URL, Password, Select, Checkbox, Radio Button, True / False, Gallery, Image, Repeater, File, Flexible Content, Clone, WYSIWYG Editor, oEmbed, Date Picker, Relationship, Taxonomy, Time Picker, Color Picker, Post Object, Accordion, Tab, Google Maps.

Dynamic Visibility

Dynamic Visibility for Elementor (Free)

<https://wordpress.org/plugins/dynamic-visibility-for-elementor/>

Dynamic Visibility extension allows you to hide widgets, columns, containers, sections, or pages.

It is particularly useful when you are building something that you don't want to show everyone.

- Choose an OR/AND condition.
- Limit visibility for specific custom fields or pages.
- Set a date (from-to), day of the week, or an hour and decide when each element will be visible.
- Limit visibility for specific user roles, user meta, IP, referral, or users.
- Limit visibility for WooCommerce products
- Set a fallback text (e.g.: 'Coming soon') for hidden elements that will be displayed in place of the element.
- You can choose to hide the element via CSS or remove it from the DOM.

Sticky Header for Elementor

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

JetSticky For Elementor (Free)

<https://wordpress.org/plugins/jetsticky-for-elementor/>

Adds sticky functionality to the free version of Elementor.

Document Embedding

Document Embedder (Free)

<https://wordpress.org/plugins/media-player-addons-for-elementor/>

Animations

Pro Elements (Free)

<https://proelements.org/>

<https://github.com/proelements/proelements>

Enables all Elementor Pro functionality (header/footer builder, dynamic tags, custom css, etc).

If used, none of the plugins needed to enable individual Elementor Pro features listed are necessary.

UiCore Animate (Free)

<https://wordpress.org/plugins/uicore-animate/>

Lightweight free additional animations for Elementor.

Embeds

All Embed (Free)

<https://wordpress.org/plugins/all-embed-addons-for-elementor/>

All Embed Addon For Elementor is a collection of audio, video, social media widgets for Elementor. It has 20 plus widgets, enables functionality to embed any content.

Elementor Addon Packs

Xpro Addons (Free)

<https://wordpress.org/plugins/xpro-elementor-addons/>

Another plugin for Elementor Free has a bunch of usually premium widgets and enables custom CSS functionality for Elementor free.

Superb Addons (Free)

<https://wordpress.org/plugins/superb-blocks/>

Neat little addon pack with editor enhancements, new widgets and free templates.

LA-Studio Element Kit (Free)

<https://wordpress.org/plugins/lastudio-element-kit/>

Free widget kit and templates.

Image Hover Effects (Free)

<https://wordpress.org/plugins/image-hover-effects-addon-for-elementor/>

Ultra-lightweight image hover effects plugin.

Custom Icons (Free)

<https://wordpress.org/plugins/custom-elementor-icons/>

Free custom icon manager with additional icons

Hash Elements (Free)

<https://wordpress.org/plugins/hash-elements/>

Free set of addon elements.

Header & Footer Builder (Free)

<https://wordpress.org/plugins/header-footer-elementor/>

Lets you build your header and footer with the free version of Elementor (no pro required).
Lightweight.

Template Kits **(Free)**

SKT Templates (Free)

<https://wordpress.org/plugins/skt-templates/>

Free templates.

Rife Extensions (Free)

<https://wordpress.org/plugins/rife-elementor-extensions/>

Large set of free templates. Lightweight.

Templately (Free and Pro)

<https://wordpress.org/plugins/templately/>

Lightweight free template plugin

Envato Elements (Free and Pro)

<https://wordpress.org/plugins/envato-elements/>

Prebuilt Template kits from Envato. Both free and pro templates. Lightweight templates, but like anything else they still require optimization.

Optimized widgets for Elementor (Free)

<https://wordpress.org/plugins/optimized-widgets-for-elementor/>

Reduces the DOM size by removing one of the default nested containers in each widget.

Included widgets are: Headline, Text, Image.

Since it is removing the elementor-widget-container div it can potentially break your theme or other plugins/scripts that require the div to be there. Please use with caution and test your page to see if everything still works. Recommended to enable this on staging prior to activating this plugin on live.

Import Figma Designs

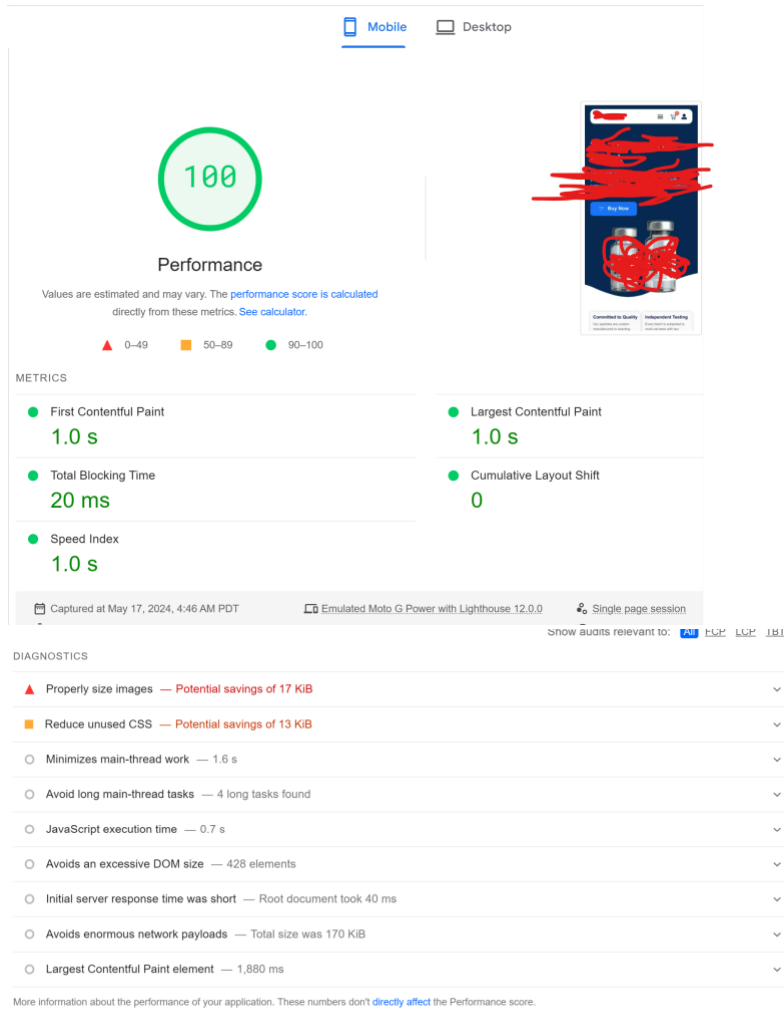
UIChemmy (Free and Pro)

<https://wordpress.org/plugins/uichemmy/>

Allows you to import 10 designs monthly on the free plan. To use the free plan to import unlimited templates, just make a new temp site on instawp.com and register another account.

Elementor Optimization Target Pagespeed Score

Elementor can easily be optimized for mobile pagespeed scores over 90.



This site is in the **top 5%** of website page speed rankings. This result was trivial to accomplish.

Plugins used:

1. **Advanced Database Cleaner Pro** (Substitute AAA Option Optimizer and Advanced Database Cleaner Free for the free equivalent)
2. **Index WP MySQL For Speed** (free)
3. **Asset Cleanup Pro** (Asset Cleanup Pro has alternatives, including free plugin options listed in the Selectively Disable Plugins Section). Use the free version if you don't want to spring for pro to disable unnecessary WP Core files from loading, polyfill as an example.

If using Asset Cleanup Free, you will need a substitute plugin for selectively disabling plugins such as **Freesoul Plugin Deactivator** or **WP Plugin Manager**. Scroll down to the “Selectively Disable Plugins” section for more info. **Asset Cleanup Free** can still selectively disable css and js assets.

4. **Perfmatters (Paid)**
5. **Disable Gutenberg (free)**
6. **Docket Cache (free)**
7. **Pre-Party (free)**
8. **Flyingpress (paid)**. There are other free plugins for caching that would satisfy the caching plugin requirement as well on the <https://wordpress.org/plugins> repository (I recommend **WP Speed Of Light** for a great free option as a substitute).
9. **Flying Pages (free)**: Preloads inner pages HTML for faster load times (**make sure to delay flying-pages.min.js in your javascript delay plugin to negate the performance impact**).
10. **Inspect HTTP Requests (free)**
11. **Disable Wordpress and Woocommerce Bloat (Free)**
12. **Yabe Webfont (Free)**
13. **Theme: Hello Elementor (Free)**.

Free alternatives needed to achieve the same functionality as **Perfmatters** are:

1. **Asset Cleanup (Free version)**
2. **Debloat (Free)**
3. **Pre-Party (Free)**
4. **Remove Global Styles (Free)**
5. **Disable Wordpress and Woocommerce Bloat (Free)**
6. **CAOS for Local Analytics (Free)**
7. **Freesoul Plugin Deactivator (Free)**
8. **Fluent Snippets (Free)**
9. **A3 Lazyload or any plugin with Lazy Load functionality (Free)**
10. **Dynamic Heartbeat Control (Free)**
11. **CDN Enabler (if desired for CDN url rewrites) (Free)**
12. **Yabe Webfont (for local hosting fonts) (Free)**
13. **Disable Google Fonts (Completely disable Google Fonts if desired) (Free)**
14. **WPS Hide Login (Free)**
15. **Lazy Load Anything (Free)**

Note: Because it takes 15 free plugins to fully replicate Perfmatters functionality I recommend springing for Perfmatters, but the hodgepodge of free plugins combined works just as well if money is an issue.

I also recommend using Yabe Webfont instead of Perfmatters or OMGF for locally hosting Google Fonts.

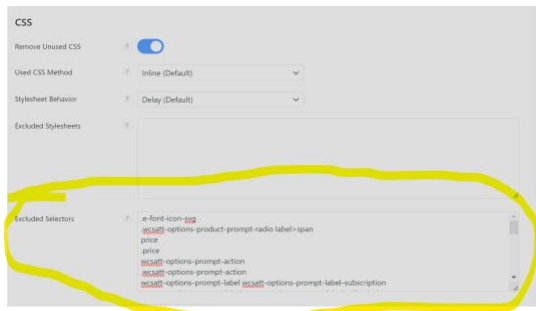
Hosting: [VPSDime](https://vpsdime.com).

Javascript files delayed with perfmatters (feel free to use any plugin with Javascript Delay, including Debloat):

<https://pastebin.com/m94cSYku>

CSS Selectors excluded from Remove Unused CSS:

<https://pastebin.com/j4QeL89d>



You can drop the selector exclusions into the exclusions box in Debloat as well, after checking the toggle to remove unused css from all files.

You will have to implement selective plugin disabling, as well as selective asset disabling (there are sections dedicated to that further down in the guide). Don't forget to disable unnecessary WP Core files. You will need to activate the setting to allow you to disable WP Core files via a checkbox in the Asset Cleanup settings panel.

An Elementor site can be optimized to 90+ in 20 minutes if done right. Easy peezy.

This is replicable on *any* Elementor site! Once you have your configuration down on the first site, you can export your settings and import them on any Elementor site. Some features will load additional files (Elementor animations for instance), so depending on if you are using animations on site vs another you may have to add a few more files to the delay list or exclusions list.

Note

You will have to manually configure Asset Cleanup selective asset disabling, selective plugin disabling, delay javascript and remove unused css exclusions. Minor configuration is necessary for caching plugins.

For Object Caching, once Docket Cache is installed you need to go into Docket Cache's settings and manually enable object caching, then go into the configuration settings tab and enable the additional optimization features you would like to use. Pre-Party must also be

manually configured. The database autoload option settings need to be manually configured via one of the Database Optimization plugins.

The other 5 plugins have no settings to configure, just activate them and you're good to go. If you disable Google Fonts, you will have to upload local fonts (assuming you want custom fonts), which you can download via the link in the fonts section. Don't forget to subset them to reduce the font file weights!

If anything breaks from copy pasting my Delay JS and Remove Unused CSS lists, you can remove the files from the javascript delay list one by one (or a few at a time) until you identify the file that must be excluded from delay, then readd the files to the list that are safe to delay.

Why am I getting errors in the console after delaying the files in your list?

I am intentionally delaying the i18n.min.js javascript file which is required for translations. Wordpress pitches a fit in the console when this file is delayed, but there are no functionality impacts to the site unless you are using translations. You may remove this file from the delay list if you wish to suppress the console errors, or you can fully unload it with Asset Cleanup.

Files that may need to be removed from the delay list depending on your configuration:

wp-util.min.js
underscore.min.js
i18n.min.js
hooks.min.js

Builders to Avoid

WPBakery

Just...don't. It's hard to find a more problematic plugin than WPBakery. Avoid it.

Beaver Builder

Beaver Builder is poorly optimized. While its javascript can largely be safely delayed without issue, basic features such as Beaver Builder's native shape dividers will massively bloat the HTML document. Additionally, themes generate a ton of unnecessary unused CSS for beaver builder which is both inlined into the HTML document as well as files, adding tons of dead code which just adds to your page weight, slowing your site down.

+ m0LndiN8row?rel=0	200	youtube.com	28.1
+ Desktop_34.jpg	200	cbdmd.com	95.2
+ SkyHigh-page-link__desktop.png	200	cbdmd.com	8.40
+ logo-pain.png	200	cbdmd.com	1.77
+ stress-icon.png	200	cbdmd.com	1.58
+ wellness-icon.png	200	cbdmd.com	1.87
+ sleep-icon.png	200	cbdmd.com	1.46
+ montserrat-v15-latin-800.woff2	200	static.cbdmd.com	19.0
+ montserrat-v15-latin-300.woff2	200	static.cbdmd.com	18.5
+ www-player.css	200	youtube.com	48.7
+ www-embed-player.js	200	youtube.com	97.1
+ base.js	200	youtube.com	581
+ fetch-polyfill.js	200	youtube.com	2.72
+ KFOmCnqEu92Fr1Mu4mxK.woff2	200	fonts.gstatic.com	15.0
+ KFOICnqEu92Fr1MmEU9fBBc4.woff2	200	fonts.gstatic.com	15.2
+ angular.js	200	cdn.searchspring.net	8.99

Youtube, fonts.gstatic.com, and cdn.searchspring.net under the domain column in the waterfall chart are the third party domains listed in this screenshot.

You want to host as many of the files from your plugins locally on your site as possible. Photos, videos, anything that you can. CAOS for Google Analytics for example downloads the files needed for Google Analytics locally and served by your site instead of downloading the files from Google's servers.

On that note, normal Youtube implementations specifically will absolutely murder your site speed (Use WP Youtube Lyte instead listed below). If your client is demanding you use Youtube and you can't escape using it, your site speed is going to be impacted regardless of the player used, however some of that weight can be mitigated using optimized Youtube players.

There will be a short delay for the video player to appear if you lazyload the video(it will load/appear as the user scrolls down), and possibly some CLS since the player is so big(especially on mobile connections), since it takes a bit to download once it enters the viewport, but it's better than slowing down the initial page load time. Many youtube plugins now have placeholder images which will prevent the entire video from downloading until the user interacts with the player's iframe via click or tap.

The load times for mobile users (especially with video) will be exacerbated by slow internet speeds, so if your client's sites operate in rural areas, they are going to experience an even more severe delay than populated areas such as big cities.

Delay the WP Youtube Lyte JS for the best results!

Locally Host Third-Party Scripts And Styles

Articles

Daan.dev Article

<https://daan.dev/blog/how-to/host-js-locally-crontab/>

WPFY Article

<https://wpfy.org/host-any-external-file-locally-in-wordpress/>

GDPR Cache Scripts And Styles (Free)

<https://wordpress.org/plugins/gdpr-cache-scripts-styles/>

Locally cache external third-party CSS and JS scripts and styles.

Flyingpress (Paid)

<https://flyingpress.com/>

No documentation yet, but this is a recently added feature to Flyingpress which will locally host all third party CSS and JS files.

Locally Host Google Fonts

Manually Locally Download Google Font Files

Google Fonts Web Helper (Free)

<https://gwfh.mranftl.com/fonts>

Fonts are heavy files that add a lot of page weight, and when you use Google Fonts from Google's CDN, the user has to make requests to their external server, which has a big performance cost. If you are using a fast VPS or managed host you will definitely want to serve those fonts locally.

Every single external call to a third-party domain will introduce additional lag. Minimize third-party calls **as much as possible**. Fonts are a big opportunity to reduce lag, so this is a key filetype to self-host.

Check your GTMetrix waterfall to identify what fonts you are using. Many plugins load fonts that you aren't even using, so you might want to disable the Google Fonts feature of the plugin entirely and replace them with your downloaded WOFF2 files (which have the most browser compatibility, and the smallest file sizes).

If you are using Elementor, it has a section in the settings to upload custom fonts. When you are using an element, use the setting to select your font, it should be listed at the top of the list once you upload them. Other builders should also have this option, you'll have to check their documentation to have to find out how to use it on your builder if you aren't using Elementor

Automatically Locally Host Font Files

Yabe Webfont (Free)

<https://wordpress.org/plugins/yabe-webfont/>

Best plugin for locally hosting Google Fonts.

OMGF (Free)

<https://wordpress.org/plugins/host-webfonts-local/>

OMGF will also locally host your Google Webfonts, which removes external calls for your fonts. I recommend using Yabe instead.

Flyingpress (Paid)

<https://docs.flyingpress.com/en/article/optimize-google-fonts-1lotnj4/>

Flyingpress can also locally host Google Fonts. **Cannot serve manually specified local fonts, use OMGF or Yabe Webfont if you want to manually upload local fonts.**

Resource Hints

When you cannot host third party files locally, you should use resource hints to improve connection speed and performance when calling a file from a third party domain.

Pre-Party (Free)

<https://wordpress.org/plugins/pre-party-browser-hints/>

This plugin allows users to automatically and easily embed resource hints to improve page load time.

Resource hints included: DNS prefetch, prerender, preconnect, prefetch, and preload are all supported.

Combine with HTTP/2 Server Push (Free)

HTTP/2 Server Push is more powerful and has a performance edge over resource hints.

HTTP/2 and resource hints are complementary and should be used together. However, the HTTP/2 Rel="preload" is more powerful than the Rel="preload" resource hint because with HTTP/2 Server Push the server pushes the resource before the user's browser requests it (before the HTML document is served).

Resource hints on the other hand are placed within your HTML document, which means the user's browser has to make a request for the resource to initiate the transfer first. Resource hints have performance benefits and should be used when possible, but Server Push has additional performance benefits over resource hints and should be used instead when possible.

Extended Explanation and Compare/Contrast

https://docs.google.com/document/d/1ZHGvCrXyKxAFLwVqj-Y1nRtzlYmU_vur0G1Ro8TDEkk/edit?usp=sharing

Fonts

Best Practices

- Use WOFF/WOFF2, not TTF/other font formats as WOFF2 has the best compression.
- Consolidate font families, weights, icons.

- Disable font icon libraries (woff2, ttf, etc) and use individual SVG files instead. Font Icon libraries load the entire icon set instead of only the icons you need, which is a lot of unnecessary page weight.
- Host fonts locally (instead of Google fonts from fonts.gstatic.com).
- Once hosted locally, preload important font files loading above the fold.
- If your use case demands that you need to load fonts from an external domain (such as Google Fonts from fonts.gstatic.com), make sure to set up preconnect (pre-party resource hints, perf-matters and other plugins have this feature) to the external domain.

Locally Host Google Fonts

Manually Locally Download Google Font Files

Google Fonts Web Helper (Free)

<https://gwfh.mranftl.com/fonts>

Fonts are heavy files that add a lot of page weight, and when you use Google Fonts from Google's CDN, the user has to make requests to their external server, which has a big performance cost. If you are using a fast VPS or managed host you will definitely want to serve those fonts locally.

Every single external call to a third-party domain will introduce additional lag. Minimize third-party calls **as much as possible**. Fonts are a big opportunity to reduce lag, so this is a key file type to self-host.

Check your GTMetrix waterfall to identify what fonts you are using. Many plugins load fonts that you aren't even using, so you might want to disable the Google Fonts feature of the plugin entirely and replace them with your downloaded WOFF2 files (which have the most browser compatibility, and the smallest file sizes).

If you are using Elementor, it has a section in the settings to upload custom fonts. When you are using an element, use the setting to select your font, it should be listed at the top of the list once you upload them. Other builders should also have this option, you'll have to check their documentation to have to find out how to use it on your builder if you aren't using Elementor.

Yabe Web Font (Free)

<https://wordpress.org/plugins/yabe-webfont/>

Yabe is a very flexible local font hosting plugin.

- **GDPR / DSGVO friendly:** You can import/self-host Google Fonts files from your server through your WordPress admin page. Fewer reasons to worry about GDPR / DSGVO compliance.
- **Google Fonts and Adobe Fonts ready:** Start using any available fonts on Google Fonts and Adobe Fonts.
- **Custom Fonts:** A dedicated custom fonts manager for WordPress. The central place to manage all your fonts and typography.
- **Lightweight and blazingly fast:** The font management system uses WordPress REST API and a modern JavaScript framework for an instant, responsive user experience. The front page cache ensures fast loading and reduces database queries.
- **Variable Fonts support:** The format allows a single font file to contain multiple stylistic variations.

OMGF (Free)

<https://wordpress.org/plugins/host-webfonts-local/>

OMGF will locally host your Google Webfonts, which removes external calls for your fonts.

Flyingpress (Paid)

<https://docs.flyingpress.com/en/article/optimize-google-fonts-1lotnj4/>

Flyingpress can also locally host Google Fonts. **Cannot serve manually specified local fonts, use OMGF or Yabe Webfont if you want to manually upload local fonts.**

Font Subsetting

Use a font subsetter to strip unused characters out of your font files, to reduce their file size.

Everything Fonts Subsetter (Free)

<https://everythingfonts.com/subsetter>

Aspose Subsetter (Free)

<https://products.aspose.app/font/generator>

Typst Subsetter (Free)

<https://github.com/typst/subsetter>

Glyphhanger (Free)

<https://www.stefanjudis.com/notes/glyphhanger-a-tool-subset-and-optimize-fonts/>

<https://github.com/zachleat/glyphhanger>

Subfont (Free)

<https://github.com/Munter/subfont>

Font Squirrel (Free)

<https://www.fontsquirrel.com/tools/webfont-generator>

Granular Performance optimized Web Font Generator. This can significantly reduce the size of your locally hosted fonts. I highly recommended it.

Font Squirrel can be paired with any locally hosted font, it is complementary to the other plugins listed here. You may use them together.

Very similar to subsetting. I recommend using both.

Font Display (Free)

How Font Display Properties Work

<https://css-tricks.com/almanac/properties/f/font-display/>

WP FOFT Loader (Free)

<https://wordpress.org/plugins/wp-foft-loader/>

This plugin implements and automates Zach Leatherman's "Critical FOFT with preload, with a polyfill fallback emulating font-display: optional" to optimize and speed up web font loading and improve UX by minimizing Flash of Invisible Text, Flash of Unstyled Text, and DOM Reflow.

Article

<https://github.com/zachleat/web-font-loading-recipes#the-compromise-critical-foft-with-preload-with-a-polyfill-fallback-emulating-font-display-optional>

Asset Cleanup (Free)

<https://wordpress.org/plugins/wp-asset-clean-up/>

Can apply Font Display properties to both Locally Hosted Fonts, or Google Fonts.

OMGF (Free and Pro)

<https://wordpress.org/plugins/host-webfonts-local/>

Font Style Matcher (Free)

<https://meowni.ca/font-style-matcher/>

If you're using a web font, you're bound to see a flash of unstyled text (or FOUC), between the initial render of your websafe font and the webfont that you've chosen. This usually results in a jarring shift in layout, due to sizing discrepancies between the two fonts. To minimize this discrepancy, you can try to match the fallback font and the intended webfont's x-heights and widths. This tool helps you do exactly that.

Article

<https://blog.hubspot.com/website/web-safe-html-css-fonts>

FontForge (Free)

<https://fontforge.org/en-US/>

Remove unnecessary glyphs to decrease font weight (Free)

FontForge is free and open-source software that will allow you to dig into font files. By removing unnecessary glyphs, we can safely reduce our font sizes by 80-90%!

Optimize Font Glyphs (Free)

Simplifying the design of the glyphs in the font can also reduce file size. This involves removing unnecessary points on the glyph outlines without compromising the design.

Remove Font Features to reduce file size

Adjusting font settings like hinting, kerning pairs, and ligatures can be a way to optimize the file size of your fonts, but these adjustments often involve trade-offs in terms of text appearance and functionality.

Remove Font Hinting (Free)

Hinting is a technique used to improve the legibility of fonts at small sizes on digital displays. It adjusts the font's outline to align better with the pixel grid.

Impact of Removal: Removing hinting can reduce file size, but it may make the font look worse on low-resolution screens where the font rendering relies on hinting for clarity.

Kerning Pairs (Free)

Kerning involves adjusting the spacing between specific pairs of characters to make the spacing appear more uniform. This is particularly important for certain character combinations where the default spacing looks awkward.

Impact of Removal: Removing or reducing the number of kerning pairs can decrease the font file size but might lead to uneven character spacing that can affect readability and aesthetics.

Ligatures (Free)

Ligatures are single glyphs that represent a combination of two or more characters, commonly used for character pairs like 'fi', 'fl', etc.

Impact of Removal: Omitting ligatures can reduce the file size and simplify the font but can degrade the typographic quality of the text, especially in print or high-resolution displays.

Database Optimization

Always take backups before implementing any database optimizations

Disable Unnecessary Auto-loaded Database Options

Disabling unnecessary autoloaded database options is a great way to boost performance by removing overhead and dead weight. Many plugin options do not need to be autoloaded and can safely be set to autoload no.

HOWEVER, setting certain autoload options to no for many plugins can break your site. **Always** take a database backup before changing these settings. You should generally be able to just reset the option autoload yes if some functionality breaks, but as a best practice it is always recommended to take a database backup before changing anything, just in case. It is possible to be locked out of the WP-Admin backend entirely due to a critical error caused by disabling an autoload option in rare cases.

SweepPress (Free)

<https://wordpress.org/plugins/sweeppress/>

Autoload Option Manager and Database Sweeper with 43 separate items cleaned. Very useful.

AAA Option Optimizer (Free)

<https://wordpress.org/plugins/aaa-option-optimizer/>

This plugin tracks which of the autoloaded options are used on a page, and stores that data at the end of page render. It keeps an array of options that it has seen as being used. On the

admin page, it compares all the autoloaded options to the array of stored options, and shows the autoloaded options that have not been used as you were browsing the site.

If you've been to every page on your site, or you've kept the plugin around for a week or so, this means that those options probably don't need to be autoloaded.

Autoload Checker (Free)

<https://wordpress.org/plugins/autoload-checker/>

Autoload Checker helps you monitor the total size of the autoloaded data in the wp_options table of your database. Autoloaded data is data that is loaded on every page load, regardless of whether it is needed or not. This can affect the performance and speed of your site, especially if you have a lot of plugins installed.

Autoload Checker does not remove any autoloaded data from your database, it only checks the status and displays it in your WordPress dashboard. You can see the total size of the autoloaded data, as well as the top autoloaded data entries sorted by size. This can help you identify which plugins or themes are adding the most autoloaded data to your database, and decide if you need to optimize them or not.

Database Cleaner (Free)

<https://wordpress.org/plugins/database-cleaner/>

Another free option for managing database options autoload status.

Advanced Database Cleaner (Freemium)

<https://wordpress.org/plugins/advanced-database-cleaner/>

Neat database management plugin.

Database Cleaners

SweepPress (Free)

<https://wordpress.org/plugins/sweeppress/>

Database Sweeper with 43 separate items cleaned. Very useful.


WP-Sweep (Free)

<https://wordpress.org/plugins/wp-sweep/>

Advanced Database Cleaner (Freemium)

Free version:

<https://wordpress.org/plugins/advanced-database-cleaner/>



Advanced DB Cleaner PRO '3.1.6'
By Younes JFR. | Need help? [Contact me](#)

General clean-up

Tables

Options

Cron jobs

Overview & settings

License

Bulk actions ▼

Apply

14 items

<input type="checkbox"/>	Elements to clean	Count	View	Scheduled ?	Keep last ?
<input type="checkbox"/>	Revisions	87			0 days Edit
<input type="checkbox"/>	Auto drafts	3			0 days Edit
<input type="checkbox"/>	Trashed posts	11			0 days Edit
<input type="checkbox"/>	Pending comments	0			0 days Edit
<input type="checkbox"/>	Spam comments	0			0 days Edit
<input type="checkbox"/>	Trashed comments	0			0 days Edit
<input type="checkbox"/>	Pingbacks	0			0 days Edit
<input type="checkbox"/>	Trackbacks	0			0 days Edit
<input type="checkbox"/>	Orphaned post meta	4			N/A
<input type="checkbox"/>	Orphaned comment meta	0			N/A

108 Element(s) can be cleaned!

0 Cleanup schedule(s) set

Add new schedule

General clean-up

Tables

Options

Cron jobs

Overview & settings

License

Some of your options are not categorized yet! Please click on the button below to categorize them!

Scan options

All (795) | Uncategorized (19) | Orphans (11) | Plugins options (617) | Themes options (6) | WP options (142)

Search for

Autoload ▼

All "belongs to" ▼

Filter


Items per page 50

Show

Bulk actions ▼

Apply

795 items << < 1 of 16 > >>



Advanced DB Cleaner PRO '3.1.6'

By Younes JFR. | Need help? [Contact me](#)

General clean-up

Tables

Options

Cron jobs

Overview & settings

License

Some of your tables are not categorized yet! Please click on the button below to categorize them!

Scan tables

All (79)

Uncategorized (1)

Orphans (0)

Plugins tables (64)

Themes tables (0)

WP tables (14)

Search for

All tables

All "belongs to"

Filter

Items per page

50

Show

Bulk actions

Apply

79 items

<<

<

1

>

>>

<input type="checkbox"/>	Table name	Rows	Size	Belongs to
<input type="checkbox"/>	wp_actionscheduler_actions	723	496 KB	woocommerce
<input type="checkbox"/>	wp_actionscheduler_claims	1	32 KB	woocommerce
<input type="checkbox"/>	wp_actionscheduler_groups	8	32 KB	woocommerce
<input type="checkbox"/>	wp_actionscheduler_logs	1563	288 KB	woocommerce
<input type="checkbox"/>	wp_advanced_notification_triggers	2	16 KB	woocommerce-advanced-notifications

Your database is optimized!

0 optimize schedule(s) set

Paid Pro version

<https://sigmaplugin.com/downloads/wordpress-advanced-database-cleaner/>.

It's important to go through and delete every unneeded option you can, unneeded database options build up over time as you remove plugins and if you change themes, and can slow your site without you realizing it. If you install a plugin, then uninstall it, always delete the leftover tables and options with advanced database cleaner pro or another database cleanup plugin.

Some plugins create a ton of database tables and options, you should avoid those (Sumo Plugins for example). You want as low a database hit as possible, so fewer queries is better. **This is important for Object Caching Performance (Redis, Docket Cache, Memcached, etc).**

Better optimized plugins will make fewer calls to the database. Also set anything that shouldn't be autoloading set to "autoload no" via advanced database cleaner or another plugin.

Many plugins will autoload unnecessary data on every page load, so disabling the autoload should boost your site speed. Just make sure to bug test your site to ensure the entries you've disabled do not cause issues with the plugin's features.

Database Query Analysis and Diagnosis

Query Monitor (Free)

<https://wordpress.org/plugins/query-monitor/>

Use Query Monitor to debug and identify plugins that have slow database queries.

It's one of the best tools you can use to diagnose performance issues. Once you have identified which plugins have slow queries, you can look for alternative plugins that have the same functionality, but that are leaner and faster.

Add-Ons (Free)

<https://querymonitor.com/help/add-on-plugins/>

Collection of Query Monitor Addons to extend Query Monitor's functionality.

Fluent Query Logger (Free)

<https://wordpress.org/plugins/fluent-query-logger/>

Query Logger with Selective Logging Functionality.

Database indexing

You should not use Index MySQL for Speed and Scalability Pro together. Pick one or the other

Index WP MySQL For Speed (Free)

<https://wordpress.org/plugins/index-wp-mysql-for-speed/>

(Unnecessary if you use Scalability Pro)

Index WP Users For Speed (Free)

<https://wordpress.org/plugins/index-wp-users-for-speed/>

This plugin speeds up the handling of your WordPress registered users, especially when your site has many thousands of them.

This plugin puts in separate indexes that Index MySQL for Speed does not index. I would not recommend using this plugin alongside Scalability Pro. If you get Scalability Pro, don't use either of the other Database Indexing plugins above.

Do not leave Index WP Users For Speed active after adding the indexes, it has a lot of unnecessary database calls on every page load when active which will likely have a negative impact on your site's pagespeed. Once the indexes have been added it is safe to deactivate the plugin.

Scalability Pro (Paid)

<https://www.wpintense.com/product/scalability-pro/>

Helpful on pretty much any site, but especially so for woocommerce stores. If your site uses Woocommerce, I would highly suggest you pick Scalability Pro over Index Mysql For Speed if you can afford it, as it adds indexes to Woocommerce tables (sadly Index MySQL For Speed does not).

Meta Optimizer (Free)

<https://wordpress.org/plugins/meta-optimizer/>

This plugin optimizes WordPress meta data storage by saving all meta data for each post, comment, user, or term in a single row with separate columns for each meta key. This reduces the number of rows and improves the query performance and data export. Convert myISAM tables to InnoDB.

Templ Optimizer (Free)

<https://wordpress.org/plugins/templ-optimizer/>

Automatically Convert myISAM tables to InnoDB tables which are more performant. The vast majority of Wordpress sites are already using InnoDB tables, but this plugin is useful for any legacy sites still on myISAM.

Transients Manager (Free)

<https://wordpress.org/plugins/transients-manager/>

Manage Wordpress Transients stored in the database.

Delete Expired Transients (Free)

<https://wordpress.org/plugins/delete-expired-transients/>

Delete old, expired transients from the WordPress options table (wp_options), to prevent them from bloating your database and slowing down your website.

Disable WP Rest API (Free)

<https://wordpress.org/plugins/disable-wp-rest-api/>

Disables The Rest API.

Analytics

Delaying Analytics Javascript Will Affect Your Metrics

I recommend delaying all analytics scripts to mitigate their performance impact. Analytics platforms work via javascript, which is incredibly performance heavy and causes a large hit to pagespeed. Deferring javascript execution only slightly mitigates it, to eliminate the performance impact entirely the javascript must be delayed from downloading until user interaction.

Delaying your analytics javascript will fuzz your data. A trade off for maintaining performance. Your collected data will not be 100% accurate if the javascript is delayed, and your traffic count may be higher than the data collected indicates. What's important is conversions, not the statistics. Increased conversions for fuzzier data is a no brainer.

Self-hosted Local Analytics

Matomo Analytics (Freemium)

<https://wordpress.org/plugins/matomo/>

Matomo is a great analytics plugin which is open source and free. Out of the box without addons it has equivalent functionality to Google Analytics. You are able to locally host Matomo analytics and do not have to share them with a third party service, which is a critical necessity for GDPR. The analytics are also going to be more accurate than Google analytics.

I recommend choosing Matomo over Google Analytics, although you can use both simultaneously if desired.

Optimization

Delay Matomo.js in your javascript delay plugin to mitigate its performance impact

Koko Analytics (Free)

<https://wordpress.org/plugins/koko-analytics/>

Lightweight local hosted analytics. Very basic statistics. I'd recommend going with Matomo over Koko Analytics just to get more in depth data, but they're both fine local hosted solutions.

Lazy Load Analytics

True Lazy Analytics (Free)

<https://wordpress.org/plugins/true-lazy-analytics/>

This plugin enables lazy loading for Google Analytics, Facebook Pixel, Hotjar, Yandex Metrica (Yandex Metrika) and Liveinternet counter.

Lazy Load Clarity (Free)

<https://wordpress.org/plugins/lazy-load-clarity/>

Lazy Load the Microsoft Clarity tracking script

Heatmaps and Clickmaps

Microsoft Clarity (Free)

<https://clarity.microsoft.com/>

<https://wordpress.org/plugins/microsoft-clarity/>

Microsoft Clarity is a drop-in replacement for Hotjar for collecting click/scroll analytics(Heatmaps, scrollmaps, click maps, rage clicks etc) as well as Session recording and a few other neat features, **completely free!**

Microsoft Clarity is better and has more features than Aurora Heatmaps, but you need to create a (free) account. No strings attached, it really is totally free.

Lazy Load Microsoft Clarity (Free)

<https://wordpress.org/plugins/lazy-load-clarity/>

The Lazy Load Clarity plugin is a simple plugin that allows you to load your Microsoft Clarity script without affecting your website page speed.

The script will load only after an initial interaction by the user, thus reducing Clarity's page speed impact to zero.

Aurora Heatmap (Free)

<https://wordpress.org/plugins/aurora-heatmap/>

Neat little free plugin, heatmaps and clickmaps, all collected and stored locally on your webserver.

Session Recording

Microsoft Clarity (Free)

<https://clarity.microsoft.com/>

<https://wordpress.org/plugins/microsoft-clarity/>

<https://wordpress.org/plugins/lazy-load-clarity/>

A/B Testing

Mojito (Free)

<https://github.com/mint-metrics/mojito>

<https://github.com/mint-metrics/mojito-js-delivery>

Mojito is an incredibly powerful Open Source A/B testing toolkit with a JS file weight of just 5 KB. It's the fastest, most performant way to run javascript based A/B testing. It can even be hooked into Google Optimize to see the data output in Google's web interface.

NOTE

Mojito may appear to be dead at first glance due to the inactivity in the top level repo, but its dependency repos (and therefore the project) are still under active development!

Do not delay Mojito's JS File

Note: Do not delay Mojito's JS file. It needs to load without a delay to correctly determine which page variant to display.

Google Analytics

How to mitigate Google Analytics Performance impact

Delay Analytics Javascript

Javascript Files To Delay List (you can copy paste this list):

gtag.js
gtag.min.js
analytics.min.js
google-analytics.com
ga
/gtag/js
gtag(
/gtm.js
/gtm-
gtm.js
gtag
googletagmanager.com

Locally Host Google Analytics

In addition to delaying the analytics javascript files, you should also locally host them. Below are a couple plugins which allow you to locally host the analytics files, but many other plugins have that feature as well.

Rankmath Self-Hosted Analytics (Paid)

2 Self-Hosted Analytics JS File

This option hosts the GA script locally and the script is updated once every 12 hours. This helps to boost up the speed of a website as an external JS file from Google is no longer requested.

You can enable this option from [Rank Math > General Settings > Analytics](#) section.



Self-Hosted Analytics JS File

Rankmath has cookieless tracking and hosting Google Analytics locally as options in the pro version. You should always host the analytics file locally.

Its cookieless tracking option means you can completely remove or ignore any cookie related GDPR compliance plugins, since you are no longer tracking users with cookies!

CAOS (Free)

<https://wordpress.org/plugins/host-analyticsjs-local/>

The screenshot shows the 'Basic Settings' tab of the CAOS (Free) plugin. At the top, there is a field to 'Enter your Tracking ID, e.g. UA-1234567-89 (v3 API) or G-123ABC789 (v4 API)'. Below this, there are several settings: 'Track logged in Administrators?' with an unchecked checkbox and a warning; 'Allow tracking...' with radio buttons for 'Always (default)', 'When cookie is set', 'When cookie is NOT set', and 'When cookie has a value', with a note about GDPR compliance; 'Snippet type' with a dropdown menu set to 'Asynchronous' and a note about performance; 'Anonymize IP?' with an unchecked checkbox and a note about legal requirements; and 'Position of tracking-code' with radio buttons for 'Header (default)', 'Footer' (which is selected), and 'Add manually', with a note about loading the snippet. At the bottom, there are 'Save Changes' and 'Update analytics.js' buttons.

The screenshot shows the 'Advanced Settings' tab of the CAOS (Free) plugin. It has a note: '* Manual update required after saving changes.' There are two main settings: 'Enable Compatibility Mode' with a dropdown menu set to 'None (default)' and a detailed note about using another Google Analytics plugin; and 'Which file to download? *' with a dropdown menu set to 'Analytics.js (default)' and a note about the recommended file and API versions. At the bottom, there is a link to 'Need help choosing?'.

If you use any tracking analytics or any tags(you should!), use CAOS analytics to host gtag.js locally to negate the performance impact they have on your site.

The cookie value will be unique to what the laws are for the areas your customers are in, so be aware of local privacy laws.

Google Tag Manager

<https://tagmanager.google.com/#/home>

Use Google Tag Manager to add third party tracking tags. Never use additional tracking plugins. Once you've set up all your tags with Tag Manager, use CAOS to download gtag.js. All your tracking tags should function via that one file, which should reduce requests and speed up your load times. Many sites are very laggy because of bad tracking tag implementations.

GTM Server Side (Free)

<https://wordpress.org/plugins/gtm-server-side/>

Google Tag Manager can be implemented server side, which eliminates its js library and improves performance.

Stape Article

<https://stape.io/blog/what-is-google-tag-manager-server-side-tracking>

GTM Kit (Free)

<https://wordpress.org/plugins/gtm-kit/>

Free Google Tag Manager Local Hosting plugin, able to load Tag Manager scripts with a delay to eliminate Tag Manager's performance impact.

Optimization

See the above section for optimizing google analytics. You can delay the same JS files. Just copy paste the list into your javascript delay plugin's script list.

Klaviyo (Paid)

Delay Klaviyo Javascript

Klaviyo does not have a lazy load solution, however Klaviyo's Javascript files can be delayed using any plugin with a JS Delay feature, including the ones listed below.

Other Analytics Platforms

Other analytics providers/platforms performance impact can be negated by simply delaying their javascript files with the Delay JS feature of any optimization plugin.

Delay Javascript

Same thing, all analytics plugins use javascript to calculate the analytics. Delay the analytics files/the analytics domains to prevent them from loading until user interaction to mitigate their performance impact.

HTTP Server Push

Smashing Magazine Article

<https://www.smashingmagazine.com/2017/04/guide-http2-server-push/>

HTTP Push Content (Free)

<https://wordpress.org/plugins/http2-push-content/>

- Push / Pre-load all JS files in site with one simple option
- Push / Pre-load all the CSS files in your website
- Push / Pre-load other resources throughout the site or based on the page types
- Load CSS Asynchronous or Remove any CSS file throughout the site, or there is a conditional selector that you can apply
- Async / Defer / Remove any JS file throughout the site or based on the WordPress page type
- You can create mobile device specific rule to push, pre-load, remove, async js or css, this works based on the device user agent detection
- You can create desktop device specific rule to push, pre-load, remove, async js or css, this works based on the device user agent detection
- Conditionally remove JS and CSS from various WooCommerce pages like checkout page, cart page, product page, product category page
- You can push pre-load resource specific on some pages by their ID
- Remove CSS, JS specific to page or post by their ID

Remove Inline CSS and JS Via HTTP Resource ID

Optimize More! (Free)

<https://wordpress.org/plugins/optimize-more/>

Has the ability to remove inline CSS styles and inline javascript files selectively on a per page basis.

Resource Hints

Pre-Party Browser Hints (Free)

<https://wordpress.org/plugins/pre-party-browser-hints/>

This plugin allows users to automatically and easily embed resource hints to improve page load time.

Resource hints included: DNS prefetch, prerender, preconnect, prefetch, and preload are all supported.

Cleanup Unused Custom Post Types

Custom Post Type Cleanup (Free)

<https://wordpress.org/plugins/custom-post-type-cleanup/>

Cleanup unused Custom Post Types to improve Database Performance.

Reduce HTTP Requests

Inspect HTTP Requests (Free)

<https://wordpress.org/plugins/inspect-http-requests/>

Monitor all the HTTP Requests being made via WP HTTP Methods i.e. wp_remote_get, wp_remote_post. Block any request by just a click of a button.
Also tracks how much time a request like updating core/plugin/theme takes.

HTTP Requests Manager (Free)

<https://wordpress.org/plugins/http-requests-manager/>

Prevent WP HTTP requests from slowing down your WordPress website and admin interface. The HTTP Requests Manager plugin will log all WP HTTP requests with time taken to complete for each request. If there are multiple requests per page they will be color grouped.

Helps to prevent website slowdown by:

- Sets the request timeout period to 2 seconds. Where default is 5.
- Limits the number of requests per page by 3. Default is unlimited.
- Limit WP HTTP request if page load time is longer than 3 seconds. Default is unlimited.
- Option to block all external requests or allow only requests to wordpress.org for plugin, theme and core updates.

Passive Listeners

Helper Lite For Pagespeed (Free)

<https://wordpress.org/plugins/helper-lite-for-pagespeed/>

Remove the Passive Listeners warning on Pagespeed tests, improves scores. **Has not been updated in over a year, test on staging before activating this plugin on a live site!**

Optimize More! (Free)

<https://wordpress.org/plugins/optimize-more/>

Function to remove passive listeners warning

Reduce Admin Ajax Usage

Diagnosing high admin-ajax usage:

<https://kinsta.com/blog/admin-ajax-php/>

<https://koerge.com/learn/wordpress-admin-ajax-php-error/>

Ajax Usage Logging

Amuga Ajax Log (Free)

<https://wordpress.org/plugins/amuga-ajax-log/>

Amuga Ajax Log tracks and logs actions that hit admin-ajax

- Requested action name
- Possible function or method name
- Suspected location
- Page that triggered the hit

Note: Only leave active when diagnosing, deactivate after use to eliminate its performance impact.

Heartbeat Control

Dynamic Frontend Heartbeat Control (Free)

<https://wordpress.org/plugins/dynamic-front-end-heartbeat-control/>

The WordPress Heartbeat API provides real-time communication between the server and the browser when you are logged into your WordPress admin panel. It uses the file /wp-admin/admin-ajax.php to run AJAX calls from the browser. By default, AJAX requests are sent every 15 seconds on post edit pages, and every 60 seconds on the dashboard.

Almost every paid optimization plugin has a function to control the Wordpress heartbeat as well.

Cache Ajax Requests

Cookie Cache Busting to cache Dynamic Ajax Content

<https://wpshout.com/crash-course-ajax-wordpress/>

<https://wpshout.com/using-cookies-in-wordpress-part-ii-cache-busting-with-ajax/>

Swift Performance

<https://swiftperformance.io/>

<https://swiftperformance.io/blog/how-to-cache-ajax-get-requests-with-swift-performance-pro/>

<https://docs.swiftperformance.io/knowledgebase/cache-ajax-requests/>

Swift Performance Lite (Free)

<https://wordpress.org/plugins/swift-performance-lite/>

The Lite version is capable of caching Ajax requests. This is probably the easiest way to cache Ajax requests.

Cloudflare Super Page Cache (Free)

<https://wordpress.org/plugins/wp-cloudflare-page-cache/>

In addition to Varnish (or Swift Performance if you use it), Cloudflare Super Page Cache can cache Ajax get requests (not post requests sadly)

<https://imgur.com/xWmKHmL>

Uncheck the box in “do not cache dynamic content” titled “Ajax Requests”

Cloudflare Limitations

If the Ajax requests are coming from a plugin that uses Ajax Nonces, the page/ajax will not be able to be cached(when using Cloudflare Super Page Cache*).

See the developer’s explanation here:

<https://wordpress.org/support/topic/how-to-correctly-cache-or-bypass-ajax-requests/>

Control WP-Cron Jobs

Advanced Cron Scheduler (Free)

<https://wordpress.org/plugins/migrate-wp-cron-to-action-scheduler/>

Lightweight Advanced Cronjob Manager

WP Control (Free)

<https://wordpress.org/plugins/wp-control/>

WP Control enables you to view and control what’s happening in the WP-Cron system.

Cron Logging (Free)

<https://wordpress.org/plugins/cron-logger/>

This plugin will help you identify which Cronjobs (if any) are problematic.

Create Server/System Cron Jobs And Disable WP Cron (Free)

<https://wpspeedmatters.com/external-cron-jobs-in-wordpress/>

Disable Pingbacks (Free)

<https://wordpress.org/plugins/no-self-ping/>

A pingback is an automated comment that gets created when another blog links to you. There can also be self-pingbacks which are created when you link to an article within your own blog.

Some people really like that WordPress sends pings from your own site to your own site when you write posts; it gives them a trail of related posts.

Some people do not like this behavior; it clutters up their comments.

This plugin disables intra-blog pinging.

Disable Bloat

Unbloater (Free)

<https://wordpress.org/plugins/unbloater/>

Some unique debloat features that other plugins don't have, including:

1. Remove DNS prefetch to s.w.org
2. Load comment script only when needed
3. Remove recent comments inline style
4. Prevent auto-linking URLs in comments
5. Normalize favicon
6. Normalize login logo title
7. Normalize login logo URL
8. Disable login language selector

WP Control (Free)

<https://wordpress.org/plugins/wpcontrol/>

Free debloat toolkit. A few options for disabling features not found in the other plugins in this section.

Disable Bloat for WordPress & WooCommerce (Free)

<https://wordpress.org/plugins/disable-dashboard-for-woocommerce/>

Great plugin with a lot of Debloat features for both Woocommerce and Wordpress. Any paid "pro" options can be achieved with other free plugins. **There is no reason to purchase the pro version of the plugin. The pro version is completely unnecessary.**

Disable Comments

Turn Off Comments (Free)

<https://wordpress.org/plugins/turn-off-comments/>

Disable Comments Sitewide

Optimization Toolkits

These are optimization toolkits with many additional features to other plugins included in this guide. **All of the listed plugins are adjuncts to everything else**, they are not the end all be all of optimization.

Advanced Settings (Free)

<https://wordpress.org/plugins/advanced-settings/>

Toolkit with a lot of features. Adjunct for other optimization plugins.

- Remove type="text/javascript" attribute from <script> tag
- Track and list enqueued scripts/styles
- Merge and include removed scripts/styles
- Load merged removed scripts in footer
- Load merged removed styles

Post types

- Manage/create/edit
- Add supports: title, editor, author, thumbnail, excerpt, trackbacks, custom fields, comments, revisions, page attributes, etc.
- Configure: hierarchical, has_archive, query_var, show_in_menu, show_ui, publicly_queryable, public, etc.
- Taxonomies: category, post_tag

HTML Code

- Fix incorrect Facebook thumbnails including OG metas
- Hide top admin menu
- Automatically add a FavIcon (whenever there is a favicon.ico or favicon.png file in the template folder)
- Add a description meta tag using the blog description (SEO)
- Add description and keywords meta tags in each posts (SEO)

- Remove header WordPress generator meta tag
- Remove header WLW Manifest meta tag (Windows Live Writer link)
- Remove header RSD (Weblog Client Link) meta tag
- Remove header shortlink meta tag
- Configure site title to use just the wp_title() function (better for hardcode programming)
- Limit the excerpt length
- Add a read more link after excerpt
- Remove wptexturize filter
- Remove Trackbacks and Pingbacks from Comment Count
- Insert author bio in each post
- Allow HTML in user profile
- Compress all HTML code
- Remove HTML comments (it's don't remove conditional IE comments like: <!--[if IE]>)
- Add Google Analytics code
- Add FeedBurner code

System

- Hide the WordPress update message in the Dashboard
- Add dashboard logo
- Unregister default WordPress widgets
- Disable widget system
- Disable comment system
- Disable Posts Auto Saving
- Automatically generate the Post Thumbnail (from the first image in post)
- Set JPEG quality
- Resize image at upload to max size
- Display total number of executed SQL queries and page loading time
- Fix post type pagination

Scripts

- Remove unnecessary jQuery migrate script (jquery-migrate.min.js)
- Include jQuery Google CDN instead local script (version 1.11.0)
- Remove type="text/javascript" attribute from tag
- Track enqueued scripts
- Merge and include removed scripts
- Load merged removed scripts in footer

Styles

- Track enqueued styles
- Merge and include removed styles
- Filters/Hooks
- Disable wp filters/hooks

Falcon (Free)

<https://wordpress.org/plugins/falcon/>

Neat little toolkit, very basic features, but a few unique ones that other kits don't have (like disabling the admin email nag notification)

WP Tools (Free)

<https://wordpress.org/plugins/wptools/>

The WP Tools plugin (wp-tools) features a comprehensive dashboard showcasing server statistics, including a chart of errors, page load time also by page and boasts over 47 tools designed to simplify your WordPress management tasks. It serves as a versatile Swiss army knife for your WordPress needs.

Disable Everything (Free and Pro)

<https://wordpress.org/plugins/disable-everything/>

Another toolkit with a few unique features, another adjunct.

MagicBox (Free)

<https://wordpress.org/plugins/magicbox/>

The Swiss Army Knife all-in-one optimization and functionality toolkit.

- **SMTP**
- **Contact Buttons**
- **Form Builder**
- **Bulk Mail**
- **Lazy Load**
- **Database Repair**
- **Cache Manager**
- **Preloader**
- **Popups**
- **Live Search**
- **Fancybox**
- **Custom CSS**
- **Cookie Policy Options**
- **Login Logs**
- **PHP Information**
- **Error Logs**

- **Post Authentication**
- **Disqus Comment**
- **Comment Manager**
- **Backup Manager**
- **Image Upload Optimiser**
- **Bulk Image Resize**
- **Keyword Suggestion**
- **Url Replacement**
- **Site Redirects**
- **Sitemap**
- **Seo Image**
- **Error Redirects**
- **Error Page Manager**
- **Duplicate Manager**
- **Custom Redirects**
- **Visit Limitation**
- **Login Url**
- **Google Authenticator**
- **Captcha**
- **Tracking Codes**
- **User.ini Editor**
- **Htaccess Editor**
- **Code Insert Manager**
- **Gutenberg Editor modifications**
- **White Label**
- **User Switch**
- **Maintenance Mode**
- **Clone Manager**
- **Custom Login**
- **Custom Dashboard**
- **Admin Menu Manager**
- **User Permissions**
- **Robot Txt Editor**

WPMasterToolKit (Free)

<https://wordpress.org/plugins/wpmastertoolkit/>

Great toolkit with a bunch of functionality. A few unique features.

Global Styles

Disable Global Styles (Free)

<https://wordpress.org/plugins/disable-global-style/>

Disables the global styles inlined by WordPress since WP 5.9. It also removes the duotone SVGs added after the opening body tag. Perfmatters has this feature as well.

Perfmatters (Paid)

Features: Selective disabling of plugins, CSS and JS.

- Delay Javascript
- Remove Unused CSS
- Preload and Preconnect
- Code insertion into the Header and Footer
- Remove Global Styles
- Remove default Wordpress bloat (emojis, RSS, shortlinks, etc)
- Defer Javascript
- Image and Iframe/Video Lazyloading
- Disable Google Fonts
- CDN url rewrite
- Custom Login URL
- Woocommerce Disable Scripts on Non-Woocommerce Pages
- JS and CSS Minification
- Heartbeat Control

Free plugins to replicate Perfmatters functionality:

- 1) **Asset Cleanup**
- 2) **Debloat** (For JS Deferral, JS Delay, Minification and Remove Unused CSS)
- 3) **Pre-Party**
- 4) **Remove Global Styles**
- 5) **Disable Wordpress and Woocommerce bloat**
- 6) **CAOS for local analytics**
- 7) **WP Headers and Footers for Code Insertion**
- 8) **Freesoul Plugin Deactivator**
- 9) **Image and Iframe Lazy Load plugin (A3 lazy load, Optimize More!, a million options)**
- 10) **Advanced Database Cleaner free or Database Cleaner**
- 11) **Disable Google Fonts (if desired)**
- 12) **CDN Enabler**
- 13) **WPS Hide Login**
- 14) **Dynamic Front-End Heartbeat Control**

Preload Inner Pages

Flying Pages (Free)

<https://wordpress.org/plugins/flying-pages/>

Flying Pages by WP Speed Matters Settings

Settings

Compatibility

FAQ

Optimize More!

Delay to start preloading

0 second (recommended) ▼

Delay to start preloading links in the viewport after browser becomes idle (hovering links doesn't have any delay)

Max requests per second

Unlimited ▼

Maximum requests browser should send to the server per second

Preload only on mouse hover

☐

Start preloading links only on mouse hover

Mouse hover delay

50 ms ▼

Delay to start preloading links on mouse hover

Ignore keywords

/wp-admin
/wp-login.php
/cart
/checkout
add-to-cart

The list of keywords that should be ignored from preloading

Disable for logged in admins

☒

Helps to reduce server load since most of the cache plugins do not serve cached pages when log

In my opinion this is the best inner page preloading plugin. Flying Pages not only preloads links that are hovered over, but **any link that enters the viewport**. This is great because when any link is clicked, the pages will already be preloaded.

This plugin will preload inner pages. Inner Pages are the pages that are loaded after the initial page the user starts on. For example, they load the homepage first, and there are links to other pages in the header and footer menu.

Those page's HTML is preloaded with Flying Pages so that when these links are clicked, their HTML has already been downloaded, resulting in faster loading time for the inner page that was clicked.

Note: Make sure to delay the js file "flying-pages.min.js" in Perfmatters(or whatever plugin you are using to delay javascript) settings panel. WP Meteor, while it does delay everything, does not seem to touch this file, so it will have to be delayed with an additional plugin.

Quick Links (Free)

<https://wordpress.org/plugins/quicklink/>

Quicklink for WordPress attempts to make navigation to subsequent pages load faster. Embedded with the plugin is a javascript library, which detects links in the viewport, waits until the browser is idle and prefetches the URLs for these links. The library also tries to detect if the user is on a slow connection or on a data plan.

I recommend delaying the Quick Links javascript file, the same as you would if you were using Flying Pages instead.

Speculative Loading (Free)

<https://wordpress.org/plugins/speculation-rules/>

By default, the plugin is configured to prerender WordPress frontend URLs when the user hovers over a relevant link. This can be customized via the "Speculative Loading" section under Settings > Reading.

Selectively deactivate plugins and CSS/JS

Use Asset Cleanup, Perfmatters, Freesoul Plugin deactivator, or any other plugin with this functionality. I recommend using Asset Cleanup Pro as it has the most configuration options, and it can also disable unnecessary WP Core files in addition to selective plugin deactivation.

When To Selectively Disable Plugins

If a plugin is only used on the frontend or backend, you should unload them where they are not needed. A file manager plugin for instance only needs to run in the backend, so it should be disabled on the frontend.

Asset Cleanup Pro (Paid)

<https://www.gabelivan.com/items/wp-asset-cleanup-pro/>

The paid version of asset cleanup has a very configurable plugin disabler.

Perfmatters (Paid)

<https://perfmatters.io>

FreeSoul Plugin Deactivator (Free)

<https://wordpress.org/plugins/freesoul-deactivate-plugins/>

Selectively deactivate plugins per page/post type on both the frontend and backend.

WP Asset Manager (Free)

<https://wordpress.org/plugins/gonzales/>

Neat free asset and plugin control manager. Full featured, can selectively deactivate css/js/plugins.

Plugin Organizer (Free)

<https://wordpress.org/plugins/plugin-organizer/>

Plugin Organizer will let you selectively deactivate plugins, as well as let you change the order in which they load.

Plugin Load Filter (Free)

<https://wordpress.org/plugins/plugin-load-filter/>

Another selective plugin disabling plugin

WP Plugin Manager (Free)

<https://wordpress.org/plugins/wp-plugin-manager/>

Selectively deactivate css and js files

Selectively disabling assets (css and js files) is more selective than disabling a plugin wholesale. Sometimes you need a plugin to load on a page, but you don't need every file the plugin loads by default. Asset Cleanup, Perfmatters

Asset Cleanup (Free and Pro)

<https://wordpress.org/plugins/wp-asset-clean-up/>

<https://www.gabelivan.com/items/wp-asset-cleanup-pro/>

Asset Cleanup Disabling CSS and JS files in the Asset Cleanup Metabox on pages:

Asset CleanUp Pro: CSS & JavaScript Manager

page post type.

Page URL: <https://www.lonecbd.com/about/>

Please select the styles & scripts that are **NOT NEEDED** from the list below. Not sure which ones to unload? 🤔 Use "Test Mode" (to make the changes apply only to you), while you are going through the trial & error process. "Load in on this page (make an exception)" will take effect when a bulk unload rule is used. Otherwise, the asset will load anyway unless you select it for unload.

WordPress CSS & JavaScript core files are hidden as requested in the plugin's settings. They are meant to be managed by experienced developers in special situations.

Assets List Layout: * any new change will take effect after you use the "Update" button

Grouped by location (themes, plugins, core & external)

→ Total enqueued files (including core files): 128

Contract All Groups Expand All Groups

From Plugins (.css & .js) → Total files: 72

M Clever Mega Menu for Elementor v1.1.2 / 1 files

Unload on this page
[Check All](#) | [Uncheck All](#)

Handle: **cmm4e-mega-menu** * JavaScript (.js)

Source: [/wp-content/plugins/clever-mega-menu-for-elementor/assets/frontend/js/cmm4e.min.js](#)

→ Preload (if kept loaded)? No (default) * applies site-wide

You should disable all unnecessary js and css files from plugins that are loading on a page if you do not want to completely disable them with the plugin manager that you can without breaking the page.

Handle: **devowl-wp-real-utills-helper** * JavaScript (.js)

There are JS "children" files depending on this file. By unloading it, the following will also be unloaded: [real-media-library.min.js](#) → ☐ Ignore dependency rule and keep the "children" loaded

Source: [/wp-content/plugins/real-media-library/vendor/devowl-wp-real-utills/dist/helper.js](#) → Preload (if kept loaded)? No (default) * applies site-wide

"Child" of one "parent" JS file: [devowl-wp-real-utills](#) / Version: 1.6.4 / Location: * applies site-wide / File Size: 3.45 KB

This JavaScript file is unloaded site-wide (everywhere).

☒ Keep site-wide rule ☐ Remove site-wide rule

All other unload rules (e.g. per page, RegEx) are overwritten by this site-wide rule.

Make an exception and always:

☐ Load it on this page ☐ Load it if the URL (its URI) is matched by a RegEx(es): ☐ Load it if the user is logged-in

Inline JavaScript code associated with the handle:

If kept loaded, set the following attributes: → ☐ on this page ☐ everywhere → ☐ on this page ☐ everywhere

More on Asset Cleanup's features later in the guide.

Optimize More! (Free)

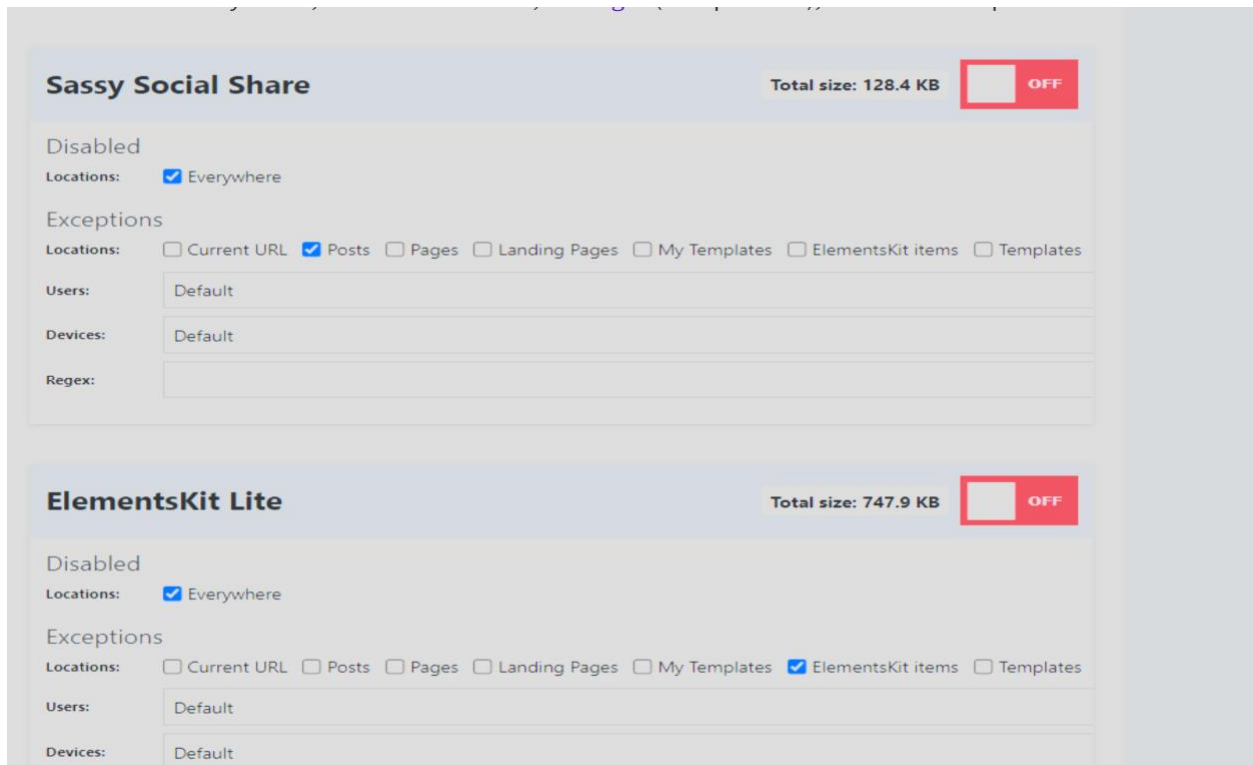
<https://wordpress.org/plugins/optimize-more/>

Optimize More! can selectively deactivate js and css files.

Perfmatters (Paid)

<https://perfmatters.io/>

Script Manager



Enable script manager in the perfmatters settings panel, then open script manager from the top bar in frontend to access the script manager UI.

WP Asset Manager (Free)

<https://wordpress.org/plugins/gonzales/>

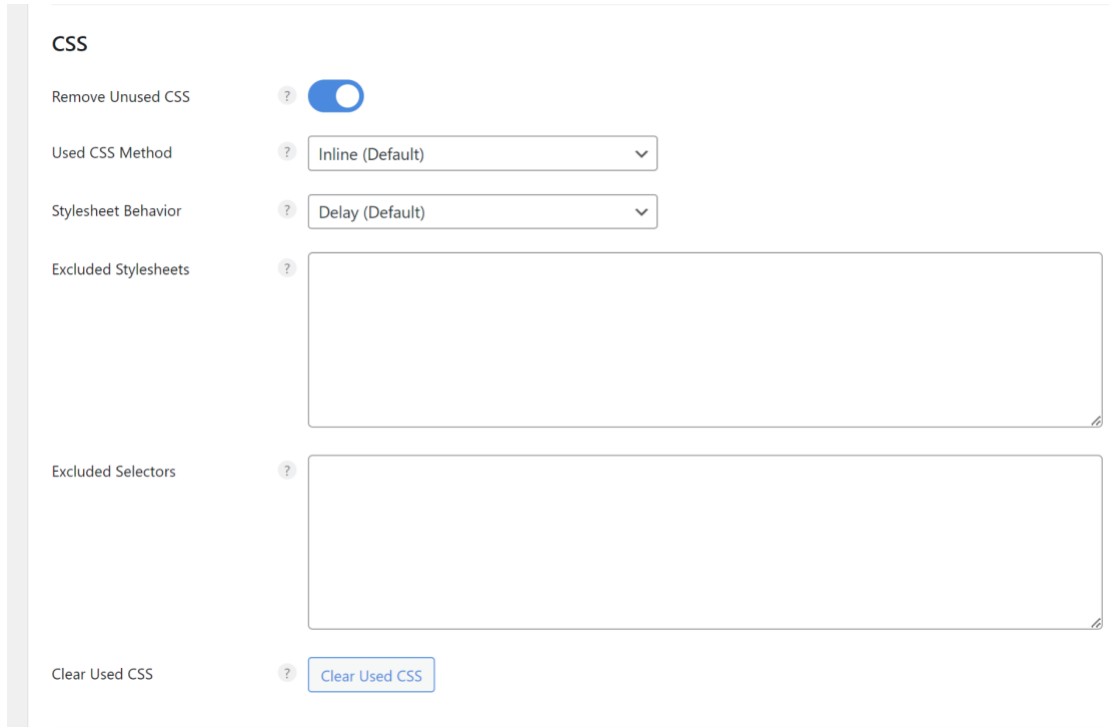
Neat free asset and plugin control manager. Full featured, can selectively deactivate css/js/plugins.

Remove Unused CSS

Instructions

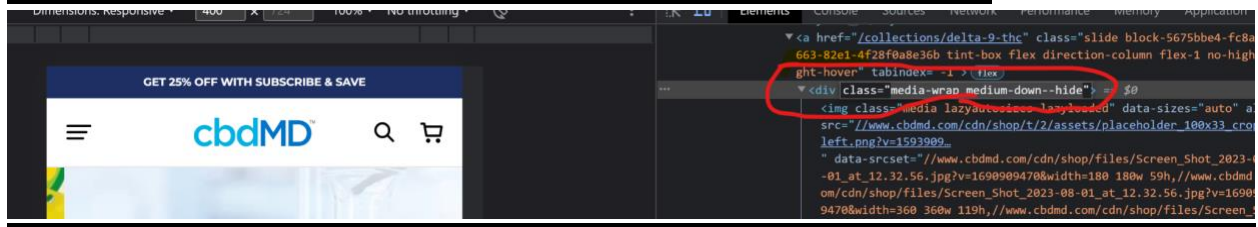
Note: Perfmatters interface used in the example below.

1. Enable the remove unused CSS function:

A screenshot of the Perfmatters CSS settings interface. The interface is titled "CSS" and contains several settings. "Remove Unused CSS" is a toggle switch that is turned on (blue). "Used CSS Method" is a dropdown menu set to "Inline (Default)". "Stylesheet Behavior" is a dropdown menu set to "Delay (Default)". "Excluded Stylesheets" is a large text area for listing exclusions. "Excluded Selectors" is another large text area for listing exclusions. At the bottom, there is a "Clear Used CSS" button. Each setting has a small question mark icon to its left.

2. Clear the caches of the caching plugin you are using, and the server cache if your host has one or you've enabled it on your VPS if you have one configured.
3. Check the frontend to ensure nothing has broken. **Check every page thoroughly throughout the entire site. Remove unused CSS will remove unused CSS on every page on your site.**
4. Add exclusions if necessary. Clear your unused css using the button at the bottom, or the clear unused css function of whatever plugin you are using.
5. Clear your caches again and verify that nothing is broken. If something is still broken, repeat adding exclusions until the design no longer breaks.
6. Once everything that you need to exclude is excluded(if you need to add exclusions at all that is), move on to delaying javascript

How to identify classes that need to be excluded

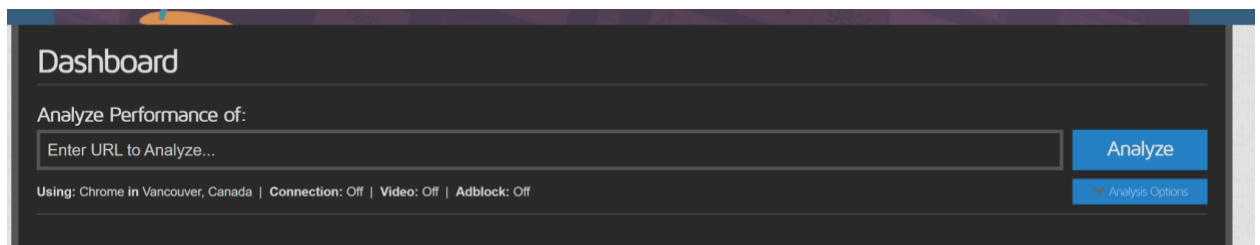


1. Inspect element over the css element you are trying to exclude.
2. Double click the class so the text is highlighted and then copy it.
3. Paste it in the exclude classes box with a period before the text. For example, “.media-wrap medium-down--hide”

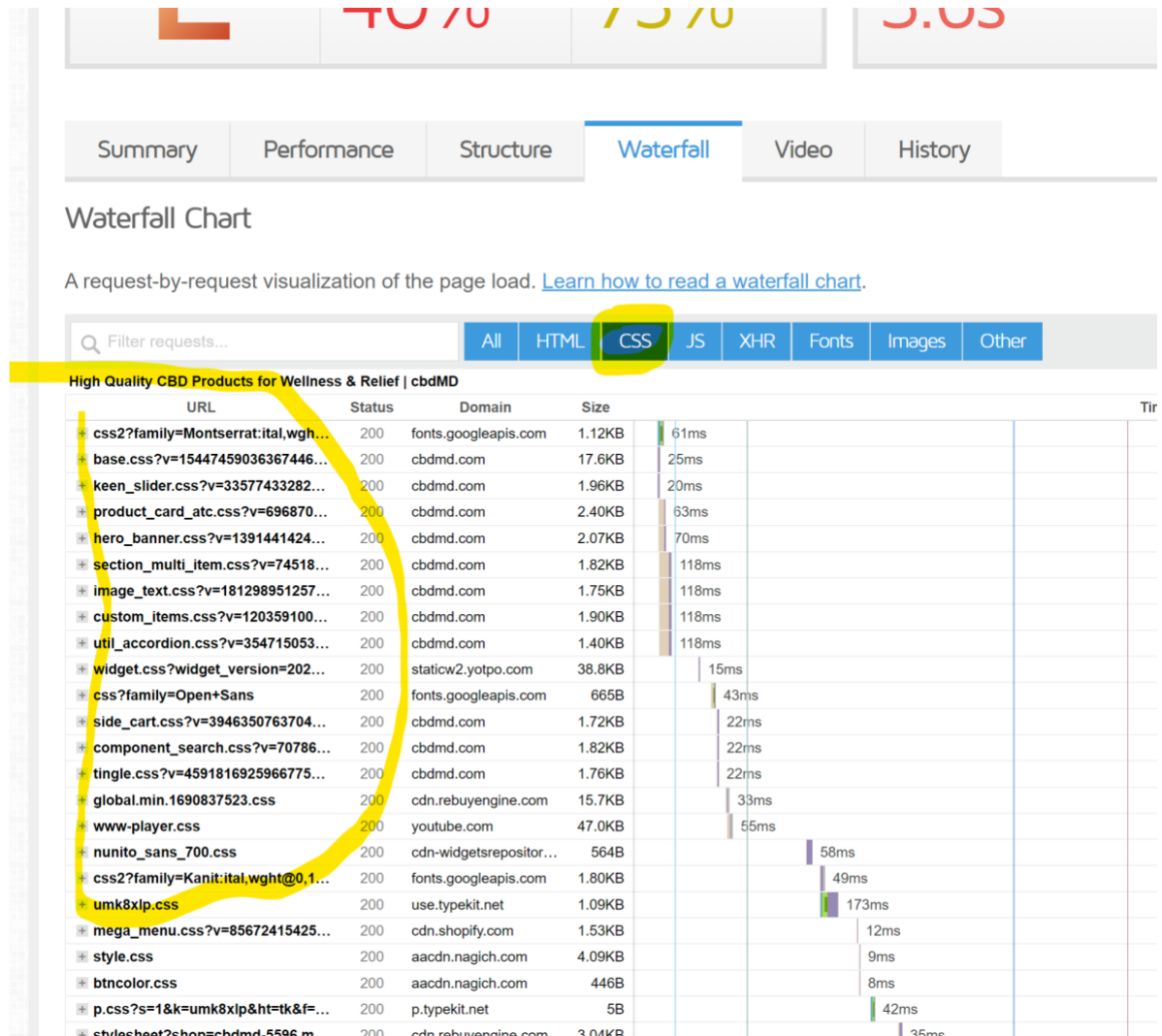
How to Identify which stylesheets need to be excluded

If excluding a class or a selector does not work, you may have to exclude an entire css stylesheet.

1. First, go <https://gtmetrix.com> or <https://www.debugbear.com/test/website-speed> and speed test the website you are trying to optimize. This requires you to copy paste your URL into the text box on the GTMetrix homepage:



2. Next go to the waterfall chart tab, and sort by CSS files:



3. Find the Stylesheets that need to be excluded and paste them into the exclusions box and hit save.

You will have to play around with it. I highly recommend you test the implementation of this feature on a staging site (**not your live site**), so the user experience is not affected on the live site due to design breakage.

Debloat (Free)

<https://wordpress.org/plugins/debloat/>

Great free option (and the only one that exists) for removing unused CSS.

Critical CSS for Wordpress (Free)

<https://wordpress.org/plugins/critical-css-for-wp/>

Removes Unused CSS and inlines critical CSS.

Flyingpress (Paid)

<https://flyingpress.com/>

Flyingpress is one of the best caching plugins and has the remove unused CSS feature.

Perfmatters (Paid)

<https://perfmatters.io/>

Perfmatters (covered below) is a great optimization plugin that has more configurability and can also remove unused css.

Manually Remove Unused CSS (Free)

<https://purifycss.online/>

Inline Small CSS/JS files

When to use inlining

This is useful for small files to reduce the amount of http requests, but inlining css/js files that are large will hurt performance. Use inlining sparingly, and always test to make sure nothing has broken after inlining a file. If a file depends on a specific loading order (say a specific js script), moving the load order to the root html may break functionality.

Optimize More! (Free)

<https://wordpress.org/plugins/optimize-more/>

Optimize More! has a function to inline both javascript and css files into the head of the html document.

Asset Cleanup (Free and Pro)

<https://wordpress.org/plugins/wp-asset-clean-up/>

Asset Cleanup can inline css files.

Automatic Inlining of CSS files based on file size

Asset Cleanup has functionality which will automatically inline CSS files below a certain file size (3 kb for example)

How To Optimize Your CSS Manually (Free)

Unnecessary if the other plugins listed above are used, but if a plugin is not desirable these resources explain how to remove it manually.

Micro-Optimize Your CSS

<https://perishablepress.com/how-to-micro-optimize-your-css/>

CSS Code Formatting

<https://perishablepress.com/series-summary-obsessive-css-code-formatting/>

Content Visibility Property (Free)

https://bloggingcommerce.com/en/improve-wordpress-loading-speed-with-css-content-visibility-property/#Que_hace_content-visibility

<https://web.dev/articles/content-visibility>

Conditional CSS loading via PHP functions

<https://webdesign.tutsplus.com/loading-css-into-wordpress-the-right-way--cms-20402t>

Critical CSS Generator

<https://www.corewebvitals.io/tools/critical-css-generator>

Manual CSS and JS Minification

<https://cssminifier.com/>

<https://www.toptal.com/developers/javascript-minifier>

<https://gist.github.com/Ruzgpegk/4fd666180e40bb8a0e0608ad35fb5a88#minify-your-css>

Lazy Load Custom Elements (Free)

<https://css-tricks.com/an-approach-to-lazy-loading-custom-elements/>

Minification

<https://wpengine.com/resources/how-to-improve-a-wordpress-sites-performance-with-minification/#:~:text=While%20there%20are%20several%20ways,therefore%2C%20more%20efficient%20to%20run.>

The top few paragraphs break down how minification works.

Minify HTML

WP Speed Of Light (Free)

<https://wordpress.org/plugins/wp-speed-of-light/>

Optimization kit and caching plugin, minifies html as well.

Minify HTML Markup (Free)

<https://wordpress.org/plugins/minify-html-markup/>

Purely minimizes HTML, no other functionality. Reduces the size of your HTML document, which speeds up TTFB. Use in conjunction with your caching plugin if it does not have an HTML minification feature (WP Rocket for instance, although WP Rocket should generally be avoided).

Minify JS and CSS files

Activate the css and js minification features using a cache plugin(most of them have a setting for css and js file minification), or Debloat. If using Debloat for unused css removal, activate minification within Debloat only. You should not be minifying your files with multiple plugins, only use one.

Debloat (Free)

<https://wordpress.org/plugins/debloat/>

Caching Plugins

The majority of page caching plugins have a feature to minify css/js files. Some come with html minification as well, although many do not. If a caching plugin lacks the ability to minify html, use the HTML minification plugin above.

Delay Javascript

Delaying javascript is **absolutely necessary** for big performance gains, as javascript is one of the worst offenders for reducing page speed performance. Delay JS is essentially lazyload for javascript, and it loads those javascript files only after user interaction to prevent them from downloading on initial page load.

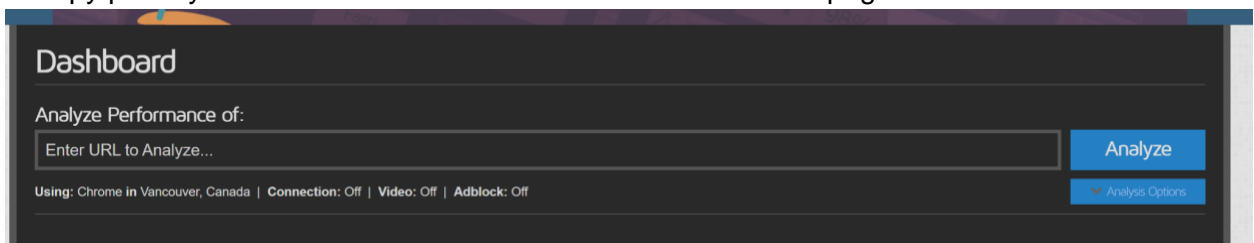
Exclusions

However, delaying javascript has a very high likelihood of breaking functionality and/or the design of your site, so you will most likely have to add exclusions for specific javascript files to prevent this. Double and triple check your pages and site features after activating this feature to ensure nothing has broken, because I have seen some nasty side-effects (such as forms not submitting, broken pop-ups, broken video players, broken maps and more) if exclusions are not added.

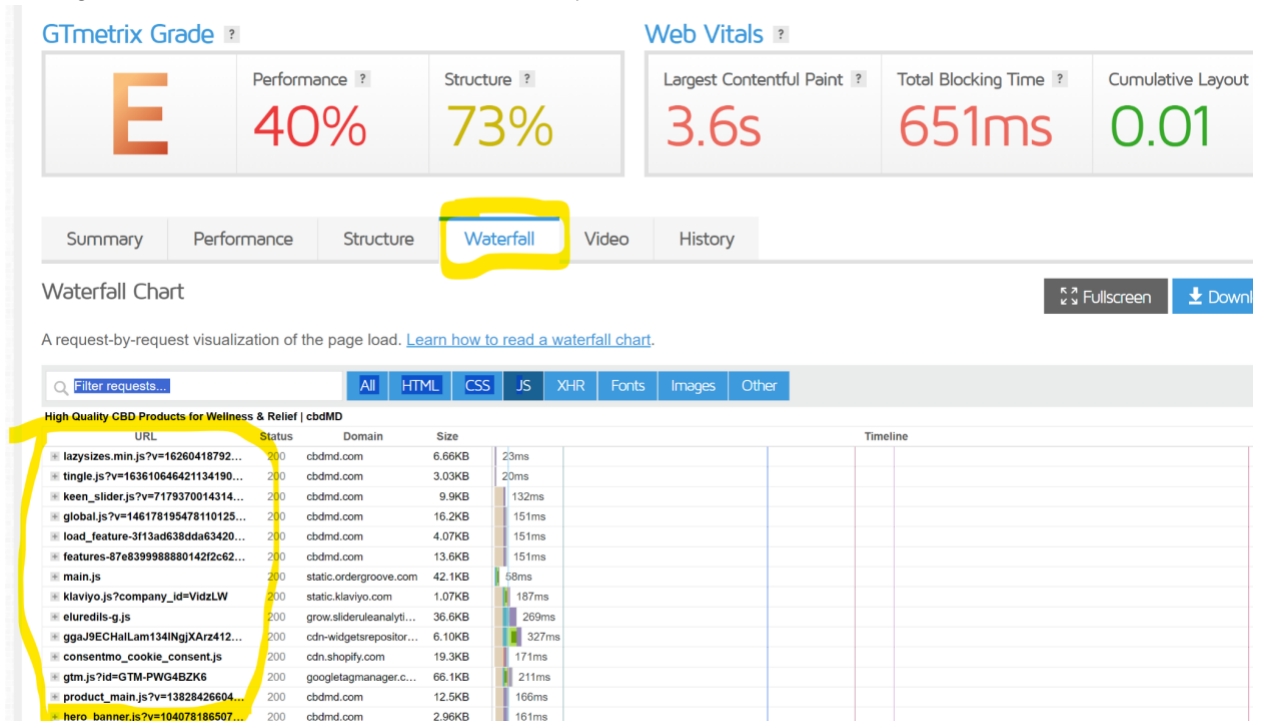
How to Identify which files need to be delayed or excluded from being delayed.

NOTE: You will have to delay scripts on every page. Certain plugins (such as some form plugins) will only load on the page in which they are used (the contact page in the case of the form plugin for example). **That means you will need to speed test every url and add the specific scripts to either the delay list or exclusions list. Speed test thoroughly!**

1. First, go <https://gtmetrix.com> (or whatever speed testing site you are using that has a waterfall chart) and speed test the website you are trying to optimize. This requires you to copy paste your URL into the text box on the GTMetrix homepage:



- Next go to the waterfall chart tab, and sort by JS files:



- For this example, I will use perfmatters to demonstrate how to choose which files to delay. You will pick the specific scripts you want to delay out of the waterfall chart. For example, **wp-polyfill.min.js**. Other files in the list are **bankful-script.js**, **w.js**, **hooks.min.js**, etc. You do not need to copy the full URL, just the name of the javascript file you wish to delay.
- Paste the name of the js files you want to delay(or exclude) into the box and hit save.
- Next you should go to the frontend of your site and check that nothing has broken either with the design, functionality or both. If a javascript file is essential and you delay it, stuff will break on the frontend, so make sure you test your site thoroughly after delaying a javascript file
- If instead of delaying specific scripts you wish to delay all of them wholesale **and then add exclusions for the scripts that need to run without being delayed**, perfmatters has that function.

JavaScript

Defer Javascript



Delay JavaScript



Delay Behavior



Quick Exclusions



Excluded from Delay



Delay Timeout



This may be preferable if you have a lot of scripts loading on your site. Instead of specifying scripts you want to delay here, you will want to list the scripts you wish to exclude here.

These instructions are applicable to *any* of the plugins that can delay javascript listed in this guide or any other plugin that contains functionality for delaying javascript.

Perfmatters (Paid)

<https://perfmatters.io/>

I highly recommend using Perfmatters delay specific JS feature in conjunction with WP-Meteors delay JS feature. Any javascript you want delayed to completely prevent it from loading on the initial page load has to be added manually. It is keyword based, so you can either specify a specific js file, or a string that is used in multiple js files.

If you want to delay all javascript files relating to a plugin, here is a random example of a hypothetical plugin which I'm just making up. if every file from "Sumo Wrestler forms" has the word wrestler in every javascript file, you can just use the word "wrestler" in the delay js box to delay all of those files.

Do not use Perfmatters Delay All JS feature(there is a dropdown where you can select either option) if you are using WP Meteor or WP-Rocket's delay javascript feature.

As mentioned above, Flying Pages Javascript file will show up in the GTMetrix waterfall of the page speed report and it will download on every page on the site when a user visits a page.

Delaying it with Perfmatters Specific Delay JS feature will prevent the file from loading entirely until user interaction with the page, and is **required** to remove the page weight impact of using Flying Pages. Otherwise, Flying Pages will hurt your page speed performance.

If a file is not needed at all until user interaction, delay it with Perfmatters Specific JS file delay feature. This also applies to your analytics files(Google Analytics, Hotjar, or any other analytics tracking software you are using). This can also apply to video players(**use this feature on the video player javascript files only if the video player is below the fold**), otherwise you will get pop-in when the video player JS file downloads, which is quite jarring and hurts the user experience.

Flyingpress (Paid)

Flyingpress has an integrated Delay Javascript function.

Debloat (Free)

<https://wordpress.org/plugins/debloat/>

Debloat also has a Delay JS feature.

Flying Scripts (Free)

<https://wordpress.org/plugins/flying-scripts/>

Simple plugin to delay javascript.

WP-Meteor (Free)

<https://wordpress.org/plugins/wp-meteor/>

WP-Meteor will delay **all** javascript, and this can and likely will break some elements on your site's pages. If this occurs, you need to manually add exclusions for the javascript files that are required for specific functionality of a plugin to work. Design elements may break as well by delaying javascript, so it is critical to go through your site after enabling this feature and check all of the pages(and/or post types) to ensure something did not break.

This feature will give you one of the largest page speed boosts you can get out of the methods listed in this guide, however, stuff **will likely break before you add file exclusions**. Double and triple check your site to ensure everything is functioning correctly. Test that your form submissions still works, add to cart process still works, popups, anything that is dependent on javascript. If it is a feature and not a design element, it is very likely to be implemented via javascript.

To identify the files you want to exclude, test your site in GTMetrix and filter it to only display javascript files with the javascript tab **before** you activate the delay javascript feature, so you know what you need to test excluding.

Optimize More! (Free)

<https://wordpress.org/plugins/optimize-more/>

Has tons of features, delaying javascript files among them.

Delay vs Defer Attribute

Delay vs Defer: Use cases

How delay works

- Initial Page Load: When a page loads, Perfmatters prevents certain specified JavaScript files from loading immediately. The rest of the page (HTML, CSS, and non-delayed JavaScript) loads as usual.
- User Interaction Trigger: The delayed scripts are loaded and executed after a specific user interaction, such as a mouse movement, scroll, or keyboard input.
- Since the javascript does not load at all until user interaction, the page speed impact of the javascript file is completely negated. Absolutely zero code of the file will be downloaded or executed before user interaction.

When delay is preferable

Delay will have the largest benefit for page speed, better than the defer or async attributes, and is the best option when it is possible to delay scripts when no functionality breakage occurs when delaying javascript files. You should always use delay when possible.

Delay may need to be applied to a single file, or multiple files depending on the functionality. Sometimes every file for the feature needs to be delayed for no interruptions in the specific functionality that is breaking when delay is applied, in which case you should delay all files related to the feature.

To be clear, the pagespeed benefit of delaying javascript is multiple times greater than deferring javascript. You should use them both in conjunction.

How Defer works

In some cases, delay will simply break functionality. While some scripts may not be able to be delayed, many of them can instead be deferred.

Scripts that depend on the DOM: Defer ensures that scripts execute only after the HTML document has been fully parsed. This is crucial for scripts that interact with or modify the DOM.

If such a script were loaded with delay, it won't execute until user interaction, leading to errors or broken functionality.

When Defer is preferable

Menu Dropdown Scenario

One scenario where defer may be preferable is for **menu dropdowns**. Menu dropdowns will generally not work until all necessary script dependencies have loaded in the DOM. If the scripts were to be delayed, if the menu is above the fold on mobile and a user taps the menu dropdown before any other interaction with the page, the dropdown may take two taps to open.

A user may believe the site to be broken/non-functional if the dropdown does not open immediately, potentially leading to an increased bounce rate. In this scenario, Defer is preferred, even though it will increase page weight compared to delay, while slightly increasing the load time.

User experience is king. So the slightly longer load time is preferable here. I recommend always defaulting to delay, then testing defer if the delay breaks functionality.

Defer may need to be applied to multiple files for certain functionality, or only a single file.

- Before Interaction: When the user initially visits the page, the dropdown menu might not be fully functional if its behavior depends on JavaScript that is being delayed.
- At First Interaction: Assuming the dropdown script is among those being delayed, the first user interaction on the page triggers the loading of the script. Depending on how quickly the script loads and executes, there might be a small delay before the dropdown becomes fully interactive.

Optimize More! (Free)

<https://wordpress.org/plugins/optimize-more/>

Optimize More! and many more plugins can apply the Defer attribute.

Async Attribute and Defer Attributes

One of the only scenarios where the async attribute is potentially preferable (and only if you need really accurate metrics) is for analytics scripts. This is going to sacrifice a lot of page speed due to how heavy and slow analytics scripts are.

However, the scripts will be loaded and active as soon as they download, and they will download quicker than either delay or defer. I would recommend async only for high traffic sites. Your metrics will still be very accurate when using delay (**never use defer for analytics scripts**), and for the fastest page speed, delay is highly recommended.

In my opinion, this is the least relevant loading attribute and should only be used in very specific scenarios such as analytics scripts where strictly accurate metrics are necessary. If you are willing to slightly sacrifice accuracy in your analytics metrics, always delay them.

If you are still interested in testing syncing your javascript files, you can use this plugin to do so:

Async Javascript (Free)

<https://wordpress.org/plugins/async-javascript/>

This plugin can apply the async or defer attribute to javascript files.

Remove Noscript Tags

Code Snippets

Fluent Snippets (Free)

<https://wordpress.org/plugins/easy-code-manager/>

Add this function to Fluent Snippets or functions.php in your theme's settings:

```
function remove_noscript_tags($html) {  
    return preg_replace('/<noscript>(.*?)</noscript>/is', '', $html);  
}  
add_filter('the_content', 'remove_noscript_tags');
```

This will remove inlined noscript tags globally

Warning

This will break functionality for users who have javascript disabled!

Lazyrender HTML Elements

Explanation:

<https://wpspeedmatters.com/reduce-dom-size-in-wordpress/>

<https://perfmatters.io/docs/lazy-load-elements/>

You can tell the browser to lazy render the contents (or elements) if it's not required for the above fold. It's just like lazy loading images, but for HTML elements. This is useful for delaying html elements such as the footer (and other objects below the fold) from loading on the initial page load.

Delay/Lazy Load anything (Free)

<https://wordpress.org/plugins/delay-load-any-content/>

Ability to Lazy Load any type of content by attaching a small CSS class to the element.

Perfmatters (Paid)

<https://perfmatters.io/docs/lazy-load-elements/>

Perfmatters has HTML element lazy rendering functionality.

Flyingpress (Paid)

<https://flyingpress.com/>

Best option, especially when using Elementor or Gutenberg. Has a simple toggle to lazy render the selected sections in the pagebuilder editors, as well as a box in the advanced section to manually add CSS selectors to lazy render.

Do Not Combine/Concentate Your CSS Or JS Files

Concentration (combining all javascript loaded on your site into a single file) will prevent certain optimization features listed in this guide from working. Many caching and optimization plugins

provide a combination/concentration feature for your js and css files which combine them into one file. **Don't turn it on!**

Css and js concentration is an outdated feature which many optimization plugins still include. Concentration is a legacy optimization feature, avoid it.

Wordpress Caching

All Caches Must Be Purged Together To Prevent Stale Caches

Note: To ensure that any changes to content or optimizations are reflected on the frontend, each of these three layers of cache must be purged: **Server Cache, Object Cache, and CDN Cache**. Purging all three layers ensures that the browser pulls the current version of the page.

The Cache Purger (Free)

<https://wordpress.org/plugins/the-cache-purger/>

Great plugin for purging multiple caches simultaneously, but it **does not** purge every caching layer. If you are using service worker caching, that must be purged separately.

Caching Layers

1. **Browser Cache:** This is client-side caching where the user's browser stores static files (like CSS, JavaScript, and images). It helps in reducing load times for repeat visits to the same site.
2. **Server Cache:** This usually refers to caching mechanisms on the server side. It can include various types of caching strategies implemented on the server.

Varnish/NGINX FastCGI cache sit on this layer.

3. **Page/Application Cache:** **This is the caching layer where Wordpress sits.** This is where Wordpress Caching plugins such as Flyingpress etc (HTML page caching plugins) come into play. When a user requests the page, the server can serve this cached version instead of dynamically generating the page every time.
4. **CDN Cache:** Content Delivery Networks (CDNs) cache your site's static assets in multiple geographic locations. This reduces the distance between the user and the

server holding your site's data, leading to faster load times. Cloudflare, BunnyCDN, Stackpath etc.

5. Object Cache (Database Caching): This involves caching the results of database queries. When a query is repeated (which happens often in dynamic applications), the server can serve the cached result instead of querying the database again. This is where Memcached, Redis, SQL Lite, Docket Cache etc. come into play.
6. PHP Cache (OPcache): This is specific to PHP and involves caching precompiled script bytecode. This means PHP doesn't have to parse and compile scripts on each request, which speeds up PHP execution.
7. Service Worker Caching: This is a form of client-side caching controlled by service workers, which are scripts that run in the background in the browser. They can manage caching of resources in a more sophisticated way, allowing for offline support and faster loading by intercepting network requests and serving cached responses.
8. DNS Caching

Why you need every layer of cache

Each layer of caching is complementary. If implemented correctly, all 8 layers of caching can be implemented simultaneously, with no conflicts, which leads to the greatest possible performance benefit.

Many articles claim that a CDN with full page HTML caching will magically resolve all page speed issues. This is a very common misconception. If you rely solely on a CDN for caching, you are leaving a lot of performance optimization opportunities on the table.

A CDN is incapable of having a 100% cache hit ratio at all times. It simply isn't going to happen. Even when a CDN is caching the majority of the content on the site(let's be generous and say an 85-90% cache hit ratio for the CDN cache), there are still hits(and likely many of them) to the origin server (the physical hardware that your site is hosted on).

When a connection does not hit the CDN cache, it is routed directly to the origin server. Without a correctly configured caching setup on the origin server, those users will encounter much slower load times.

Service worker caching is the lowest priority, but can have significant performance benefits if used correctly.

CDN Caching Layer

Using a CDN speeds up your website because it combines caching and multiple-server hosting to minimize network latency.

- Multiple Servers: The CDN hosts several web server applications on each computer across the globe.
- Caching: The CDN serves each page from a web server's cache.
- Latency: The website user is routed to the nearest web server when the CDN has cached the webpage.

Thorough section on CDNs covered further down in the guide

Application Caching Layer

Page Caching Plugins

Only use one page caching plugin

Using multiple page caching plugins will trigger errors and conflicts with unsynchronized caches.

Flyingpress (Paid)

<https://flyingpress.com/>

Best caching plugin, most fully featured, good developer support. Gijo (the developer) adds new features relatively frequently, is an extremely nice guy, and will help troubleshoot any issues you have and issue patches pretty quickly. Recommended.

WP Speed of Light (Free)

<https://wordpress.org/plugins/wp-speed-of-light/>

Great free caching plugin. Combining the free version of WP Speed of Light with other free plugins in the guide will nab you all the same features the premium version has.

Speed Booster Pack (Free)

<https://wordpress.org/plugins/speed-booster-pack/>

Pretty solid caching plugin.

Rapid Cache (Free)

<https://wordpress.org/plugins/rapid-cache/>

Rapid Cache is a fork of Comet Cache that focuses on new features, stability and simplicity.

Swift Performance (Free and Pro)

<https://swiftperformance.io/> (Pro)

<https://wordpress.org/plugins/swift-performance-lite/> (Free)

Swift Performance has a really nice featureset on paper, including features other caching and optimization plugins don't have. However, I personally have had poor results using it on my sites. However I'm leaving it in the guide due to how many unique features it has. It may perform better on other site setups, so hopefully someone finds the mention useful.

Cache Ajax Requests

Swift Performance can cache Ajax Requests

Caching Plugins To Avoid

WP-Rocket (Paid)

<https://wp-rocket.me/>

Do not use WP Rocket. WP Rocket **was** a solid page caching plugin. Now, it's buggy and the devs have been in **maintenance/removal of features mode** for ~3 years with barely any new features pushed out and more features entirely removed or had existing functionality flexibility eliminated rather than added.

Their code is filled with legacy bugs, and their update cadence is extremely slow. [There are over 700 open issues on their github.](#) It's an ultra buggy caching plugin.

WP Rocket constantly **removes** features, not adds them. They have continually taken away useful features, even ones that are necessary such as HTML minification. They used to be second best in my book when WP Rocket had more features, now it applies some optimizations automatically and they cannot be disabled without additional unnecessary ""helper"" plugins.

The continuous removal of functionality means the plugin is consistently getting worse. The best paid plugin is Flyingpress, hands down.

WP Rocket Helper Plugins (Free)

<https://github.com/wp-media/wp-rocket-helpers>

These should just be straight up unnecessary, the functionality should just be built into WP Rocket, ridiculous that these are even required. Perfect example of how they constantly remove functionality. Some of the removed functionality has been moved to the unnecessary helper plugins which are now required to regain it.

Pure HTML Page Caching Plugins

These page caching plugins have no other optimization functions besides html page caching (all other page caching plugins in the above items have more features in addition to HTML page caching).

Cache Master (Free)

<https://wordpress.org/plugins/cache-master/>

Supports up to 10 cache drivers such as File, Redis, Memcache, Memcached, APC, APCu, WinCache, MySQL, SQLite, and MongoDB

The Swiss army knife of caching. Try them all and find which type of caching works best for you!

Cache Enabler (Free)

<https://wordpress.org/plugins/cache-enabler/>

Simple page caching plugin, the only function is pure HTML page caching.

Surge (Free)

<https://wordpress.org/plugins/surge/>

Hyper Cache (Free)

<https://wordpress.org/plugins/hyper-cache/>

Hyper Cache is purely a caching plugin, and has no other additional functions. You can combine it with the other plugins listed in this guide to achieve all of the same optimization features you would get all in one with Flyingpress for free.

Which Caching plugin should I choose?

My personal preference is Flyingpress. Both are good options, and perform pretty similarly.

Remove Query Strings To Raise Cache Hit Ratio

<https://wpspeedmatters.com/ignore-query-strings/>

WP Stash (Free)

<https://github.com/inpsyde/WP-Stash>

<https://inpsyde.com/en/wordpress-caching-system-wpstash/>

Cache Warmer (Free)

<https://wordpress.org/plugins/cache-warmer/>

Visits website pages to warm (preload) the cache.

Proxy Cache Purge (Free)

<https://wordpress.org/plugins/varnish-http-purge/>

The Proxy Cache Purge plugin sends a request to bust NGINX or Varnish caches when a page or post is modified.

WPGraphQL Smart Cache (Free)

<https://wordpress.org/plugins/wpgraphql-smart-cache/>

WPGraphQL Smart Cache provides support for caching and cache invalidation of WPGraphQL Queries.

Rest Cache

WP Rest Cache (Free)

<https://wordpress.org/plugins/wp-rest-cache/>

This plugin will allow WordPress to cache the responses of the REST API, making it much faster.

- Caching of all default WordPress REST API GET-endpoints.
- Caching of (custom) post type endpoints.
- Caching of (custom) taxonomy endpoints.
- Automated flushing of caches if (some of) its contents are edited.
- Manual flushing of all caches.
- Manual flushing of specific caches.
- A counter how many times a cache has been retrieved.
- Specifying after what time the cache should be timed out.
- Registering custom endpoints for caching.
- Automatic cache regeneration.

Translations

Translation Cache (Free)

<https://wordpress.org/plugins/speed-up-translation/>

Cache Translation Language Files (Free)

<https://gist.github.com/Ruzgfpegk/4fd666180e40bb8a0e0608ad35fb5a88#cache-language-files>

Gravatar Cache

Optimum Gravatar Cache (Free)

<https://wordpress.org/plugins/optimum-gravatar-cache/>

If you need Gravatars, this is a solid Gravatar caching plugin. If they aren't needed, highly recommended to remove them.

FV Gravatar Cache (Free)

<https://wordpress.org/plugins/fv-gravatar-cache/>

Lightweight gravatar caching plugin

Image Caching

Cache Images (Free)

<https://wordpress.org/plugins/cache-images/>

Cache Images is a plugin that gives users the option to sideload images that are hosted on other domains to their own site. Sideloaded images are added to WordPress media library so you can use all tools related to images that you can use with images uploaded through WordPress. Image will be added as an attachment of the first post where it is found, and every post where the original URL is occurring will be updated with a new URL. Users can select from which domains to sideload images, including Blogger's domains.

It uses AJAX so it means you can sideload a large number of images even on slow servers.

Widget Output Cache (Free)

<https://wordpress.org/plugins/widget-output-cache/>

Use PHP output buffering to extract widget output and store it into WordPress transients for faster retrieval. It also adds a checkbox to widget controls to exclude it from being cached.

Translation Caching (Free)

<https://github.com/inpsyde/translation-cache>

Needs to be combined with Object caching. Allows you to cache Wordpress Translations.

Menu Cache

Docket Cache (Free)

<https://wordpress.org/plugins/docket-cache/>

Docket Cache has functionality to cache menus inside the configuration panel.

Menu Caching (Free)

<https://wordpress.org/plugins/menu-caching/>

This plugin will cache the menu HTML and show the cached version to your visitors, saving your database from far too many unnecessary calls. Compatible with Page Caching plugins (Flyingpress, WP Speed of Light, etc).

WP Nav Menu Cache (Free)

<https://wordpress.org/plugins/wp-nav-menu-cache/>

Cache Wordpress Nav Menus

WP-Admin Caching

WP-Admin Cache (Free)

<https://wordpress.org/plugins/wp-admin-cache/>

You will have to enable it after installing/activating, go to the settings for admin cache under settings>wp admin cache and press check all to enable it for your whole admin dashboard, then save the settings. You may have to clear the cache after installing a new plugin, if you extract it from a zip file via a file manager plugin, or it won't appear in the list due to a stale cache that did not have an update that triggered the cache.

Cache Ajax Requests

With Varnish

<https://guides.wp-bullet.com/how-to-cache-ajax-get-requests-with-varnish-4/>

With Swift Performance

<https://swiftperformance.io/blog/how-to-cache-ajax-get-requests-with-swift-performance-pro/>

Fragment Caching For Highly Dynamic Sites

Explanation

<https://seravo.com/en/speed-up-your-site-with-fragment-caching/>

When you can't cache the entire page, fragment caching comes into play. This will cache specific portions of a page as opposed to the entire thing.

W3 Total Cache (Paid)

<https://wordpress.org/plugins/w3-total-cache/>

W3 Total Cache has a fragment caching feature.

Borlabs Cache (Free)

<https://borlabs.io/borlabs-cache/>

The free version has a cache preload limitation. Use Cache Warmer so you don't have to buy a license(the page does have the free version, just need to scroll down).

Cache Warmer (Free)

<https://wordpress.org/plugins/cache-warmer/>

A nice workaround for cache preloading(cache warming) for the Borlabs plugin. Instead of buying a license, just pair it with Cache Warmer.

Fragment Cache Github (Free)

<https://github.com/Rarst/fragment-cache>

Fragment Caching Alternatives

No Cache Ajax Widgets (Free)

<https://wordpress.org/plugins/no-cache-ajax-widgets/>

With No Cache AJAX Widgets, you simply drag and drop the new AJAX powered widgets to any widget area/s, add your content and you are done. No more messing with your theme files, functions.php, fragmented caching, mfunc and others or excluding an entire page from caching.

Content No Cache (Free)

<https://wordpress.org/plugins/content-no-cache/>

This plugin is to exclude a part of the page from the cache. This allows you to keep caching enabled on highly dynamic sites.

Cache Blocks (Free)

<https://github.com/WordPressUtilities/wpucacheblocks>

More options coming soon

DIY Caching Methods

<https://www.smashingmagazine.com/2012/06/diy-caching-methods-wordpress/>

WP Stash (Free)

<https://github.com/inpsyde/WP-Stash>

<https://inpsyde.com/en/wordpress-caching-system-wpstash/>

WP Cache Remember (Free)

<https://github.com/stevegrunwell/wp-cache-remember>

WP Cache Remember is a simple WordPress caching plugin to introduce convenient new caching functions.

Browser Caching

Online Media Masters Article

<https://onlinemediamasters.com/serve-static-assets-with-an-efficient-cache-policy-wordpress/>

PHP APCu Caching

Atec Cache APCu (Free)

<https://wordpress.org/plugins/atec-cache-apcu/>

PHP APCu object and page caching

Object Caching

As this was expanded on at the top of the guide, I will simply be aggregating the links to the object caching resources here with no further explanation.

Redis Cache (Free)

<https://wordpress.org/plugins/redis-cache/>

Docket Cache (Free)

<https://wordpress.org/plugins/docket-cache/>

SQL Lite Object Cache (Free)

<https://wordpress.org/plugins/sqlite-object-cache/>

Memcache (Free)

<https://wordpress.org/plugins/object-cache-4-everyone/>

Varnish And Nginx **Server Caching Purging**

Proxy Cache Purge (Free)

<https://wordpress.org/plugins/varnish-http-purge/>

Automatically purges varnish/nginx caches when the content on the site is updated by WordPress. The plugin reaches out to the proxy cache service with the URL of the page, requesting the cache be deleted.

The Cache Purger (Free)

<https://wordpress.org/plugins/the-cache-purger/>

Great plugin for purging multiple caches simultaneously, but it **does not** purge every caching layer. If you are using service worker caching, that must be purged separately.

PHP OpCache (Free)

<https://tideways.com/profiler/blog/fine-tune-your-opcache-configuration-to-avoid-caching-surprises>

<https://gist.github.com/rohankhudedev/1a9c0a3c7fb375f295f9fc11aeb116fe>

Service Worker Caching

<https://philipwalton.com/articles/smaller-html-payloads-with-service-workers/>

<https://vimeo.com/362260166>

To be expanded on soon

DNS Caching

Cloudflare (Free and Pro)

If you set Cloudflare to be your DNS provider, they provide DNS caching.

Cache Statistics

Atec Cache Info (Free)

<https://wordpress.org/plugins/atec-cache-info/>

This plugin provides detailed status information and statistics for PHP cache features, namely OPcache, WP-object-cache, JIT, APCu, Memcached, Redis and SQLite-object-cache.

Cache Purging

Wordpress Layer Caching

The Cache Purger (Free)

<https://wordpress.org/plugins/the-cache-purger/>

This plugin attempts to purge all server-side caching methods.

This includes the most common caching plugins, some hosting based caches, most server based caches, built-in WordPress object caches, and even simple file based caches.

Built in automatic cache purging for the following caches:

- **Plugins**
 - **SiteGround Optimizer, Nginx Helper, LiteSpeed Cache, Cachify, Autooptimize, Fast Velocity Minify, WP Rocket, Swift Performance,**

Comet Cache, Hummingbird, WP Fastest Cache, WP Super Cache, W3 Total Cache, Hyper Cache, WP Optimize, Cache Enabler, NitroPack

- **Hosting / CDN**
 - WPEngine, Kinsta, GoDaddy Managed WordPress, Pantheon, Bluehost, Cloudways, Siteground, RunCloud
- **Server Based**
 - PHP FPM, Zend Opcache, APC and APCU, WinCache, Pagespeed Module, nginx, Static File Caches, Redis, Memcache, Memcached, Varnish
- **WordPress Built-In object caching, and persistent object caching**

Super Light Cache Buster (Free)

<https://wordpress.org/plugins/super-light-cache-buster/>

This simple plugin adds random version numbers to CSS & JS assets to prevent browser caching getting in the way of your happiness.

Server Caching

Proxy Cache Purge (Free)

<https://wordpress.org/plugins/varnish-http-purge/>

Clears the Varnish and NGINX Caches

Cache Warming

Warm Cache (Free)

<https://wordpress.org/plugins/warm-cache/>

Crawls your website-pages based on any XML sitemap. If you have a caching plugin this will keep your cache warm.

Cache Warmer (Free)

<https://wordpress.org/plugins/cache-warmer/>

Visits website pages to warm (create) the cache if you have any caching solutions configured.

Object caching

What is Object Caching?

Object caching stores frequently accessed data from the database in memory. When this data is needed again, it can be retrieved from the cache instead of querying the database. This reduces the number of database queries, lowering the load on the database server and improving response times.

Unlike the caching methods above which are Wordpress level caches, Object Caching is MySQL/MariaDB server level database caching. These plugins hook into a PHP service if using Redis or Memcached. Docket Cache on the other hand uses a slightly less performant method for object caching by creating PHP cache on the Wordpress level to achieve database caching functionality.

Redis or Memcached are the most performant object caching methods as they run in RAM, so they are preferable if possible. Most sites should be able to take advantage of Redis with just the free Redis Object Cache plugin as long as the Redis extension is enabled in PHP.

Docket Cache (Free)

<https://wordpress.org/plugins/docket-cache/>

Docket cache can be used when you cannot install Redis(some shared hosting plans don't have it for instance). Docket cache is an object cache, just like Redis is, except it functions at the Wordpress level as opposed to the server level. It can have better performance than Redis in many cases.

Redis (Free)

Dragonfly Redis Performance Optimization Guide

<https://www.dragonflydb.io/guides/redis-memory-and-performance-optimization>

Extensive article on optimizing Redis performance.

General Redis information for Wordpress

Redis is also a server level service, which you will need a Wordpress plugin to use as a connector to the server level redis service. REdis must be installed on your VPS/by your hosting provider for the plugin to work.

Redis Object Cache (Free)

<https://wordpress.org/plugins/redis-cache/>

Redis Object Cache developed by Till Krüss.

Note

If the Redis service is not active at the server level, this plugin will do absolutely nothing. The plugin will show that it cannot connect to the service in the settings page of the plugin, so make sure it can connect when you press the button in the settings page to activate the connection.

Make sure the Redis port(port #: 6379) that Redis is set to at the server level is set to the default that the plugin uses(in most cases it defaults to the port used by the plugin), otherwise the plugin will be unable to connect with the Redis service.

You also need to enable the Redis addon for PHP, which is a required dependency to activate Redis caching.

Once you enable the connection in the Redis plugin settings, Redis database caching is already configured and caching your database calls. No further configuration is required, and you are done. Easy peezy.

Once correctly configured, you will see a flush cache option when visiting the redis plugins settings. You will see your number of database queries in [Query Monitor](#) (if you have it active for diagnosis purposes) drop significantly after installing and configuring Redis, and should immediately notice a speed improvement on your site.

I noticed absolutely no difference in performance between the free version of the plugin vs the pro version. I tested the pro version with the zlib compression feature enabled(which is a pro only feature, and to enable zlib compression requires additional configuration at the server level), but if your site is heavily database dependent(an example would be a large e-commerce site), you may find the pro version improves performance further. YMMV. I recommend trying the pro version out anyways, it may suit your use case.

If you want to test the pro version, get a Pro license and request a refund if it doesn't add any additional performance.

Redis Object Cache

Overview

Metrics

Diagnostics

Overview

Client:	PhpRedis (v5.3.1)
Drop-in:	Valid

Connection

Status:	Connected
Host:	127.0.0.1
Port:	6379
Database:	0
Connection Timeout:	1s
Read Timeout:	1s
Redis Version:	5.0.5

Flush Cache

Disable Object Cache

Even with Redis, you want to keep database calls to a minimum. Reduce them to the absolute minimum by removing unnecessary cruft from old plugins and themes, as well as disabling database options from autoloading(which do not need to be auto-loaded of course) via [Advanced Database Cleaner](#), and have as few plugins active as possible to accomplish whatever features you want to implement. Make sure to check to that nothing has broken by preventing certain options from auto-loading

If you need some plugins that only have functionality on the backend, make sure to disable them on the frontend with Asset Cleanup's plugin manager. The lighter weight the plugins are, the better. Doing so will reduce the number of database queries on the frontend.

Atec Cache APCu (Free)

<https://wordpress.org/plugins/atec-cache-apcu/>

PHP APCu object and page caching

Memcached (Free)

Object Cache 4 Everyone

<https://wordpress.org/plugins/object-cache-4-everyone/>

Memcached based object cache connector. Only for sites with Memcached enabled at the server level.

SQLite Object Cache (Free)

<https://wordpress.org/plugins/sqlite-object-cache/>

Persistent object cache This one uses the widely available [SQLite3](#) extension to php (Make sure it is enabled on your server, or this plugin will not work). Many hosting services offer it.

If your hosting service does not provide memcached or redis, you may be able to use this plugin instead and get the benefit of object caching.

SQLite Object Caching: SQLite is a lightweight, disk-based database. Using it for object caching can be beneficial if you don't have access to more advanced caching solutions or if your site is relatively small and doesn't demand high scalability. However, it might not offer the same performance benefits as in-memory solutions like Redis or Memcached.

Which kind of object cache is best for my site?

PHP Object Caching (e.g., Docket Cache): PHP object caching in WordPress, like Docket Cache, can improve performance by caching the compiled opcode of PHP scripts. This can reduce the time it takes to load and execute PHP scripts. This type of caching is particularly useful if your server doesn't have opcode caching enabled at the PHP level (e.g., using OPcache).

Redis: Redis is an advanced key-value store known for its performance and flexibility. It can handle complex data types and offers features like replication and persistence. If your WordPress site has high traffic and needs to handle complex queries or large datasets, Redis might be a good choice.

Memcached: Memcached is a high-performance, distributed memory object caching system, often used to speed up dynamic web applications by alleviating database load. It's simpler than Redis and is designed for speed. It's a great choice for caching simple data and speeding up read-heavy workloads.

How to figure out which object cache is best for you

After testing, SQLite Object Cache has the worst caching performance, but YMMV. Docket Cache has better performance than Redis in many cases, however Redis may be desirable for high traffic sites since the cache is held in RAM instead of on Disk. However, since Docket Cache is implemented in a different way than Redis, even though it is caching the data on Disk, in many cases it can outperform Redis. Test both and choose whichever performs best.

Server Caching

NGINX PHP-FPM Cache (Free)

NGINX comes with a native server caching method called PHP-FPM/FastCGI cache. This is the primary server caching method used by sites configured to run on the NGINX webserver.

Apache Server Cache (Free)

<https://httpd.apache.org/docs/2.4/caching.html>

Official Apache Server Caching Documentation

LoadForge Article

<https://loadforge.com/guides/leveraging-caching-to-accelerate-apache-server-response-times>

Solid article on Apache server caching.

PHP OpCache

OpCache Configuration (Free)

<https://loadforge.com/guides/maximizing-your-php-applications-speed-with-opcache-optimization>

Solid article explaining how OpCache works on a surface level, and explainers on what the configurable OpCache settings work, and how to modify them in the PHP.ini configuration file.

How OpCache Works

Npopov Article

<https://www.npopov.com/2021/10/13/How-opcache-works.html>

Extremely in-depth technical article on the inner workings of PHP OpCache. This does not cover OpCache configuration whatsoever, and is not necessary information to correctly configure OpCache for improving PHP performance. Not critical information, but a really good thorough explainer for the curious.

MySQL Caching

SQL Query Caching

How SQL Query Caching Pairs With Object Caching

Different Caching Layers

ProxySQL (Query Caching)

Role

Caches the results of SQL queries at the database proxy level. This reduces the load on your MySQL/MariaDB server by serving cached query results directly from ProxySQL without querying the database.

Use Case

Particularly useful for read-heavy workloads where certain queries are executed frequently with the same parameters, such as fetching posts, comments, or user data in WordPress.

Object Caching

Role

Caches objects and data structures in memory, such as the results of expensive database queries, transients, and other non-SQL data. In WordPress, this is often used to cache the results of complex queries, session data, and other elements that don't change frequently.

Use Case

Ideal for caching the output of expensive operations like WP_Query, transients, and API calls. Redis is typically used for object caching, while Docket Cache is a specialized object caching plugin for WordPress that can work with Redis or alone.

Complementary Caching Benefits

Reduced Database Load

With ProxySQL caching SQL query results and Redis/Docket Cache storing the results of complex operations and objects in memory, the combined caching layers reduce the number of requests hitting your database. This helps to decrease database load and improve response times.

Improved Scalability

As traffic increases, having both SQL query results cached by ProxySQL and object data cached by Redis or Docket Cache allows the system to scale more efficiently. Each caching layer handles a different aspect of data retrieval, ensuring that the database and application layers are not overwhelmed

Faster Response Times

The combination of ProxySQL and Redis/Docket Cache ensures that frequently accessed data is served from memory or the proxy cache, leading to faster response times for end users.

ProxySQL Query Caching (Free)

<https://proxysql.com/>

MySQL Query caching has been deprecated, it is now recommended to use ProxySQL Cache for Query caching.

Shurutech Comprehensive Configuration Article

<https://shurutech.com/comprehensive-step-by-step-proxysql-guide/>

Comprehensive step by step guide for installing and configuring ProxySQL.

Materialized views

What Are Materialized Views

in MySQL are essentially precomputed and stored results of a query. Unlike standard views, which are virtual tables and do not store data physically, materialized views store the result of the query physically in the database. This allows the database to retrieve the data quickly without recalculating the result each time the view is queried.

Benefits of Using Materialized Views

Database Star Article

<https://www.databasestar.com/sql-views/>

Cooling Dude Article

<https://www.coding-dude.com/wp/databases/creating-mysql-materialized-views/>

From Dual Article

<https://fromdual.com/mysql-materialized-views>

Precomputed Results

Faster Query Execution

Since the results of complex queries are precomputed and stored, querying a materialized view can be much faster than running the original query, especially for complex joins, aggregations, or subqueries.

Offloading Computation

By precomputing and storing results, materialized views reduce the computational load on the database during runtime, which can be particularly beneficial for large datasets or complex queries.

Reduced Database Load

Combination with ProxySQL

Materialized views can reduce the frequency and complexity of the queries that ProxySQL needs to cache or route. ProxySQL can cache the results of queries against materialized views, leading to even faster response times and reduced database load.

Complementing Object Caching

While object caching (like Redis or Docket Cache) stores specific objects or query results, materialized views store more complex, precomputed data that can be quickly accessed. This can reduce the need for certain types of object caching or make cached objects smaller and more efficient.

Data Consistency and Freshness

Periodic Refresh

Materialized views can be refreshed periodically (e.g., daily, hourly) or manually, depending on the requirements of your application. This allows you to balance performance improvements with the need for up-to-date data.

Trigger-Based Refresh

You can set up triggers or procedures to refresh materialized views automatically when underlying data changes, ensuring the data is relatively fresh without manually refreshing the views.

How To Create An Auto-Refreshed Materialized View

Hitesh Gondalia Article

<https://hiteshgondalia.wordpress.com/manage-a-fast-refreshable-materialized-view/>

Fantastic long and thorough article on how to create a fast auto-refreshed Materialized View.

How to Combine Materialized Views with Other Caching Strategies

Materialized Views and ProxySQL

Query Caching

ProxySQL can cache queries that access materialized views. Since materialized views provide precomputed results, the combination of ProxySQL caching and materialized views can significantly reduce query execution times and the load on the database.

Query Routing

If you have different materialized views for different types of queries, ProxySQL can route queries to the appropriate view based on predefined rules, optimizing query performance.

Materialized Views and Redis/Docket Cache

Object Caching

For parts of your application where you need to cache specific pieces of data (like results of a query or an expensive calculation), Redis or Docket Cache can complement materialized views. For example, you could cache the result of querying a materialized view in Redis, ensuring even faster access times.

Hybrid Approach

Use materialized views for complex, frequently accessed datasets and Redis/Docket Cache for other transient, non-relational data. This approach can optimize performance by using the right tool for each type of data.

Materialized Views with Page Caching

Reducing Dynamic Content Generation

By using materialized views, you can reduce the computational effort required to generate dynamic content. This can, in turn, reduce the dependency on page caching plugins, as the pages can be generated more quickly from the materialized views.

Consideration for Maintenance and Refresh

Automatic Refresh

Set up materialized views to refresh automatically based on your application's needs. Ensure that the refresh frequency aligns with how frequently the underlying data changes.

Manual Refresh for Critical Views

For views that power critical parts of your application, consider manual refreshes during off-peak hours or when certain data updates are pushed to the database.

Potential Challenges

Disk Space

Materialized views consume additional disk space, as they store the result of the query physically in the database. Ensure that you have sufficient storage, especially if the views involve large datasets.

Trade-offs

Depending on how often you refresh the materialized views, there may be a trade-off between performance and data freshness. It's essential to determine the right balance for your application.

Complexity

Combining materialized views with multiple caching layers adds complexity to your architecture. Proper management, monitoring, and maintenance are crucial to ensure all layers work together harmoniously.

Varnish Caching (Free)

Cloudways Article

<https://www.cloudways.com/blog/configure-varnish-cache/>

Solid article on how to install and basic configuration for Varnish Caching.

Advanced Varnish Configuration

There is a thorough section on advanced Varnish configuration further down in the guide in the server optimization and configuration section

<https://docs.google.com/document/d/1ncQcxnD-CxDk4h01QYyrlOh1IEYDS-DV/#heading=h.c5fvp0ce1bdq>

HAProxy Caching

HAProxy Caching is covered further in the guide in the load balancing section with advanced configuration examples.

<https://docs.google.com/document/d/1ncQcxnD-CxDk4h01QYyrlOh1IEYDS-DV/#heading=h.5yfvgl0kkwc0>

Squid Proxy Cache

Why Use Two Separate Proxy Caches?

Different Roles in a Complex Architecture

Squid for Forward Proxy (External Content)

If your architecture requires caching of **external web content** (e.g., users frequently accessing the internet), Squid can be used as a forward proxy to reduce bandwidth and improve access speed for these requests.

Varnish for Reverse Proxy (Internal Content)

At the same time, Varnish can be used as a reverse proxy to cache and accelerate content served by your **internal web servers**, ensuring fast delivery of your web applications or sites to users.

Use Cases For Squid Cache

Squid is usually employed as a forward proxy cache for web content or as a reverse proxy for content from other servers. If used in conjunction with Varnish, Squid could handle content that Varnish does not cache effectively (e.g., very large files, specific web applications).

Squid Cache can be used alongside NGINX PHP-FPM caching, HAProxy caching, Apache caching, general load balancer caching, PHP OpCache caching, and Varnish caching, but there are some important considerations to ensure that this complex setup improves performance rather than introducing redundancy or conflicts.

Key Considerations

Cache Layer Redundancy

With multiple caching layers, such as Squid, Varnish, and HAProxy, there might be overlapping functionality. Each caching layer should be carefully configured to avoid redundant caching, which can lead to unnecessary complexity and potential performance degradation.

Server **Cache Hierarchy**

Define a clear hierarchy of caching layers. Typically, a web application would have caching at the server (e.g., PHP OpCache, PHP-FPM), reverse proxy (e.g., Varnish, Nginx), and content delivery (e.g., Squid, HAProxy) levels. Ensure each layer serves a distinct purpose:

PHP OpCache

Optimizes PHP code execution by storing precompiled script bytecode.

Nginx/Apache Caching

Performs basic web server-level caching.

NGINX PHP-FPM Caching

Handles caching at the FastCGI level.

Varnish Caching

High-performance HTTP accelerator.

Squid Cache

Typically used for caching external content (e.g., proxy cache) but can also cache dynamic content with careful configuration.

HAProxy Caching

Load balancer with optional caching functionality

General Load Balancer Caching

Balances traffic with caching capabilities.

Content Consistency

With multiple caches, managing cache invalidation and ensuring content consistency can become challenging. You must coordinate the cache purging across all layers to avoid serving stale content.

Network Overhead

Introducing Squid as an additional layer adds another point of network communication. The performance gains must outweigh the added latency and complexity.

Traffic Type

Consider the type of traffic and content served. For example, if most content is static and already well-cached by Varnish and HAProxy, adding Squid may not provide significant additional benefits.

Potential Configuration

- **Squid** can be placed at the edge, serving as a first-level cache for requests before they hit your load balancer or reverse proxy (e.g., Varnish). This could be useful for caching static assets or external content from third-party sources.
- **Varnish** and **HAProxy** can handle caching dynamic content and distributing traffic to your web servers.
- **Nginx/Apache** would handle server-level caching and pass requests to the application layer (PHP-FPM) after checking the cache.
- **PHP OpCache** optimizes the PHP code execution without affecting the rest of the caching layers.

Perfmatters (Paid)

<https://perfmatters.io/>

My Settings:

Disable Self Pingbacks	?	<input checked="" type="checkbox"/>
Disable REST API	?	Default (Enabled) v
Remove REST API Links	?	<input checked="" type="checkbox"/>
Disable Dashicons	?	<input checked="" type="checkbox"/>
Disable Google Maps	?	<input checked="" type="checkbox"/>
Disable Google Fonts	?	<input checked="" type="checkbox"/>
Disable Password Strength Meter	?	<input checked="" type="checkbox"/>
Disable Comments	?	<input checked="" type="checkbox"/>
Remove Comment URLs	?	<input checked="" type="checkbox"/>
Disable Heartbeat	?	Default v
Heartbeat Frequency	?	60 Seconds v
Limit Post Revisions	?	Default v

Disable Emojis	?	<input checked="" type="checkbox"/>
Disable Embeds	?	<input checked="" type="checkbox"/>
Remove Query Strings	?	<input checked="" type="checkbox"/>
Disable XML-RPC	?	<input checked="" type="checkbox"/>
Remove jQuery Migrate	?	<input checked="" type="checkbox"/>
Hide WP Version	?	<input checked="" type="checkbox"/>
Remove wlwmanifest Link	?	<input checked="" type="checkbox"/>
Remove RSD Link	?	<input checked="" type="checkbox"/>
Remove Shortlink	?	<input checked="" type="checkbox"/>
Disable RSS Feeds	?	<input checked="" type="checkbox"/>
Remove RSS Feed Links	?	<input checked="" type="checkbox"/>
Disable Self Pingbacks	?	<input checked="" type="checkbox"/>

WooCommerce

Disable specific elements of WooCommerce.

Disable Scripts



Disable Cart Fragmentation



Disable Status Meta Box



Disable Widgets



Save Changes

Restore Defaults

You should **always** preconnect and prefetch any external domains that your site links to. This includes sites like google tag manager, or google analytics, facebook, or any external sites your website is linking out to.

You can easily identify these external domains by looking at the waterfall chart in gtmetrix and looking at the domains listed, you can easily pick out which sites your site is connecting to.

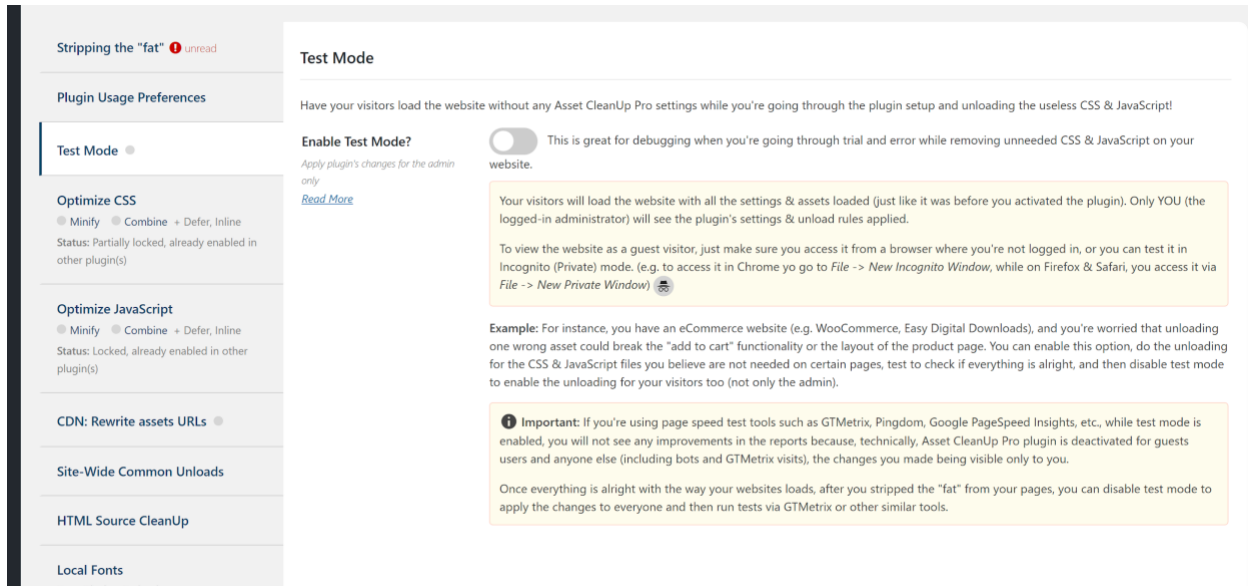
Asset Cleanup (Free and Pro)

Asset Cleanup is an essential optimization plugin.

<https://wordpress.org/plugins/wp-asset-clean-up/> (free)

<https://www.gabelivan.com/items/wp-asset-cleanup-pro/> (Pro)

Test Mode



The screenshot shows the 'Test Mode' settings in the Asset Cleanup plugin. On the left is a sidebar menu with options: 'Stripping the "fat"' (unread), 'Plugin Usage Preferences', 'Test Mode' (selected), 'Optimize CSS', 'Optimize JavaScript', 'CDN: Rewrite assets URLs', 'Site-Wide Common Unloads', 'HTML Source CleanUp', and 'Local Fonts'. The main content area is titled 'Test Mode' and contains the following text:

Have your visitors load the website without any Asset Cleanup Pro settings while you're going through the plugin setup and unloading the useless CSS & JavaScript!

Enable Test Mode? ☐ This is great for debugging when you're going through trial and error while removing unneeded CSS & JavaScript on your website.

Apply plugin's changes for the admin only
[Read More](#)

Your visitors will load the website with all the settings & assets loaded (just like it was before you activated the plugin). Only YOU (the logged-in administrator) will see the plugin's settings & unload rules applied.

To view the website as a guest visitor, just make sure you access it from a browser where you're not logged in, or you can test it in Incognito (Private) mode. (e.g. to access it in Chrome yo go to *File -> New Incognito Window*, while on Firefox & Safari, you access it via *File -> New Private Window*)

Example: For instance, you have an eCommerce website (e.g. WooCommerce, Easy Digital Downloads), and you're worried that unloading one wrong asset could break the "add to cart" functionality or the layout of the product page. You can enable this option, do the unloading for the CSS & JavaScript files you believe are not needed on certain pages, test to check if everything is alright, and then disable test mode to enable the unloading for your visitors too (not only the admin).

Important: If you're using page speed test tools such as GTMetrix, Pingdom, Google PageSpeed Insights, etc., while test mode is enabled, you will not see any improvements in the reports because, technically, Asset Cleanup Pro plugin is deactivated for guests users and anyone else (including bots and GTMetrix visits), the changes you made being visible only to you.

Once everything is alright with the way your websites loads, after you stripped the "fat" from your pages, you can disable test mode to apply the changes to everyone and then run tests via GTMetrix or other similar tools.

Enable test mode if you're optimizing a live site so that only the administrator user account can see the changes. Or install, migrate your website to a staging site or local wordpress development, then optimize away without having test mode enabled. **Otherwise you have to worry about breaking the frontend for a site which already has existing users and traffic.**

Asset Cleanup Disabling CSS and JS files in the Asset Cleanup Metabox on pages:

Asset CleanUp Pro: CSS & JavaScript Manager

page post type.

Page URL: <https://www.lonecbd.com/about/>

Please select the styles & scripts that are **NOT NEEDED** from the list below. Not sure which ones to unload? 🤔 Use "Test Mode" (to make the changes apply only to you), while you are going through the trial & error process. "Load in on this page (make an exception)" will take effect when a bulk unload rule is used. Otherwise, the asset will load anyway unless you select it for unload.

WordPress CSS & JavaScript core files are hidden as requested in the plugin's settings. They are meant to be managed by experienced developers in special situations.

Assets List Layout: * any new change will take effect after you use the "Update" button

Grouped by location (themes, plugins, core & external) ▾

→ Total enqueued files (including core files): 128

Contract All Groups

Expand All Groups

From Plugins (.css & .js) → Total files: 72

M Clever Mega Menu for Elementor v1.1.2 / 1 files

Unload on this page
[Check All](#) | [Uncheck All](#)

Handle: **cmm4e-mega-menu** * JavaScript (.js)

Source: </wp-content/plugins/clever-mega-menu-for-elementor/assets/frontend/js/cmm4e.min.js>

→ Preload (if kept loaded)? No (default) ▾ * applies site-wide ?

You should disable all unnecessary js and css files from plugins that are loading on a page if you do not want to completely disable them with the plugin manager that you can without breaking the page.

Handle: **devowl-wp-real-utils-helper** * JavaScript (.js)

There are JS "children" files depending on this file. By unloading it, the following will also be unloaded: [real-media-library-rm](#) → ☐ Ignore dependency rule and keep the "children" loaded

Source: </wp-content/plugins/real-media-library/vendor/devowl-wp-real-utils/dist/helper.js> → Preload (if kept loaded)? No (default) ▾ * applies site-wide ?

"Child" of one "parent" JS file: [devowl-wp-utils](#) / Version: 1.6.4 / Location: <BODY> * initial ▾ * applies site-wide / File Size: 3.4 KB

This JavaScript file is unloaded site-wide (everywhere).

☒ Keep site-wide rule ☐ Remove site-wide rule

All other unload rules (e.g. per page, RegEx) are overwritten by this site-wide rule.

Make an exception and always:

☐ Load it on this page ☐ Load it if the URL (its URI) is matched by a RegEx(es): ? ☐ Load it if the user is logged-in

Inline JavaScript code associated with the handle: Show / Hide +

If kept loaded, set the following attributes: **async** → ☐ on this page ☐ everywhere **defer** → ☐ on this page ☐ everywhere

Asset Cleanup File Preloading in the Asset Cleanup Metabox on pages:

→ Preload (if kept loaded)? * applies site-wide ?

You should only preload css files you want to load first that are critical to the page rendering.

Some files need basic preload, some can be asynced preloaded, other files cannot be preloaded at all due to breaking your page. You will have to clear caches and retest all related functionality to the file on the frontend to ensure you did not just introduce a new bug. For instance, I moved the simple lines icon file from a plugin called Woolentor. This damaged the ajax add to cart functionality of it's widgets gear icon for a split second after pressing the add to cart button. I would never have known that was an issue if I didn't test my changes.

You should preload css files above the fold(the content that loads in the viewport without scrolling) so that they load with priority and render your content faster. Ideally, your above the fold content should not need any javascript at all, which will improve perceived and actual load times. Set the preload to async if you can.

I generally would not preload a js file, as this will make it load in the head and give it priority, making your render-blocking score worse.

Move css/js files load location from head to the body to reduce render blocking

Handle: **font-awesome** * *Stylesheet (.css)*

Source: </wp-content/plugins/elementor/assets/lib/font->

Version: 4.7.0 / Location: <HEAD> * initial ▼ * app

If you cannot completely disable a css/js file on a page because it is necessary, you should move every js/css file that isn't immediately needed and doesn't break the page when doing so to the body of instead of the head to reduce render-blocking and improve load times.


Most js files do not need to be loaded in the head, and most plugins are poorly coded and load them in the head. You should move all or as many js files as you can from the head to the body.

You should also do this with any css file that isn't immediately needed, as this also reduces the render blocking score.

Keep in mind, you cannot change the location per page, moving the file from head to body will do so globally on every page on the site, so be aware that this could break something on one specific page vs another if the code needs to load in the head vs loading in the body.

Asset Cleanup Async, Defer for JS files:

If kept loaded, set the following attributes: async → ☐ on this page ☐ everywhere defer → ☐ on this page ☐ everywhere

No notes have been added about this JavaScript file (e.g. why you unloaded it or decided to keep it loaded) →  Add Note

I also set all css and js to async that I could.

You have to test the frontend functionality to make sure nothing broke if you set any file css or js to preload, async, or defer. You should only do it to CSS files that are above the fold, as these are critical and need to load first to avoid flashes of unstyled content. You will need Asset Cleanup Pro for async and defer settings, I don't think the free version has them.

You should be asyncing and deferring any javascript file you can. These properties can also break your page, so you will need to test it after you implement it for every file. I suggest doing them in small groups and then backtracking if any issues arise, for sake of time. 2-3 files per run, that way you don't make too many changes and can more easily narrow down the issue. 1 by one is horribly slow and inefficient to do.

CSS Inlining with Asset Cleanup

<https://blog.logrocket.com/improve-site-performance-inlining-css/>

Be careful not to inline too many files, this can hurt performance. Inlining CSS will integrate this CSS into your HTML head, which will likely download on every page. **Only inline critical CSS stylesheets that will download on every page, such as small CSS files from your theme, or a plugin that is used site-wide.**

Disable unnecessary hardcoded and external css/js on pages that they are not needed:

Hardcoded (non-enqueued) Styles & Scripts → Total: 5

i The following tags are NOT LOADED via the recommended `wp_enqueue_scripts()` action hook (despite the name, it is used for enqueueing both scripts and styles) which is the proper one to use when enqueueing scripts and styles that are meant to appear on the front end. The standard functions that are used inside the hook to do an enqueueing are: `wp_enqueue_style()`, `wp_add_inline_style()`, `wp_enqueue_script()` & `wp_add_inline_script()`. The tags could have been added via editing the PHP code (not using the right standard functions), directly inside posts content, widgets or via plugins such as "Insert Headers and Footers", "Head, Footer and Post Injections", etc. Be careful when unloading any of these tags as they might be related to Google Analytics/AdWords, StatCounter, Facebook Pixel, etc.

Hardcoded LINK (stylesheet) & STYLE tags

Belongs to: "WooCommerce" plugin

```
<style>.woocommerce-product-gallery{ opacity: 1 !important; }</style>
```

☒ Unload on this page

☐ Unload site-wide * everywhere

☐ Unload on All Pages of "page" post type

☐ Unload it for URLs with request URI matching this RegEx(es): ?

No notes have been added about this hardcoded STYLE tag (e.g. why you unloaded it or decided to keep it loaded) — [Add Note](#)

External 3rd Party (.css & .js) → Total files: 5

Note: External .css and .js assets are considered those who are hosted on a different domain (e.g. Google Font API, assets loaded from external CDNs) and the ones outside the WordPress "plugins" (usually /wp-content/plugins/), "themes" (usually /wp-content/themes/) and "uploads" (usually /wp-content/uploads/) directories.

Handle: **google-fonts-1** * Stylesheet (.css)

This resource is not loaded as "Remove Google Fonts" is enabled in "Settings" -> "Google Fonts".

Source: <https://fonts.googleapis.com/css?family=Roboto:100,100italic,200,200italic,300,300italic,400,400italic,500,500italic,600,600italic,700,700italic,800,800italic,900,900italic|Roboto+Slab:100,100italic,200,200italic,300,300italic,400,400italic,500,500italic,600,600italic,700,700italic,800,800italic,900,900italic|Muli:100,100italic,200,200italic,300,300italic,400,400italic,500,500italic,600,600italic,700,700italic,800,800italic,900,900italic|Poppins:100,100italic,200,200italic,300,300italic,400,400italic,500,500italic,600,600italic,700,700italic,800,800italic,900,900italic|Source+Sans+Pro:100,100italic,200,200italic,300,300italic,400,400italic,500,500italic,600,600italic,700,700italic,800,800italic,900,900italic|Work+Sans:100,100italic,200,200italic,300,300italic,400,400italic,500,500italic,600,600italic,700,700italic,800,800italic,900,900italic>

→ Preload (if kept loaded)? **No (default)** * applies site-wide

Version: 5.6.1 / Location: **<BODY> * initial** * applies site-wide / File Size: [Get File Size](#)

This stylesheet file is unloaded site-wide (everywhere).

☒ Keep site-wide rule ☐ Remove site-wide rule

i All other unload rules (e.g. per page, RegEx) are overwritten by this site-wide rule.

Make an exception and always:

☐ Load it on this page

☐ Load it if the URL (its URI) is matched by a RegEx(es): ?

☐ Load it if the user is logged-in


Some plugins do not load their code as a file that's identified under the plugin in the Asset Cleanup metabox that appears on pages. They appear at the bottom in the hardcoded section. One file that appeared there for me was from a form plugin that was loading a captcha. I couldn't figure out where the file was coming from until I disabled the hardcoded text that referenced the captcha.


It was loading the captcha file, even though I didn't even have the captcha feature activated. Some plugins will load useless files that you don't need, unnecessarily bogging your site's speed down.

Hardcoded code will not specify what plugin it's related to, but if you read the section labeled "source" that is hardcoded, it's relatively easy to identify what plugin it is related to. If you aren't sure what the hardcoded code does, either disable it and test the frontend to ensure nothing broke, or just leave it alone. These files generally do not have a huge performance impact, but this varies by plugin, so it may be worth disabling them.

Disable Theme JS and CSS using Asset Cleanup

✎ From Themes (.css & .js) → Total files: 8

 Astra v3.0.2

Handle: **astra-flexibility** * JavaScript (.js)  Loads only in Internet Explorer based on the following condition: if IE

Source: [/wp-content/themes/astra/assets/js/minified/flexibility.min.js](#) → Preload (if kept loaded)? No (default) * applies site-wide ?

Version: 3.0.2 / Location: <HEAD> * initial * applies site-wide / File Size: 15.21 KB

This JavaScript file is unloaded site-wide (everywhere).

☒ Keep site-wide rule ☐ Remove site-wide rule

! All other unload rules (e.g. per page, RegEx) are overwritten by this site-wide rule.

Handle: **astra-menu-animation** * Stylesheet (.css)

Source: [/wp-content/themes/astra/assets/css/minified/menu-animation.min.css](#) → Preload (if kept loaded)? No (default) * applies site-wide ?

Version: 3.0.2 / Location: <HEAD> * initial * applies site-wide / File Size: 3.33 KB

This stylesheet file is unloaded site-wide (everywhere).


☒ Keep site-wide rule ☐ Remove site-wide rule

! All other unload rules (e.g. per page, RegEx) are overwritten by this site-wide rule.

Make an exception and always:

☐ Load it on this page ☐ Load it if the URL (its URI) is matched by a RegEx(es): ? ☐ Load it if the user is logged-in

If kept loaded, make the browser download the file on any screen size (default) ?

No notes have been added about this stylesheet file (e.g. why you unloaded it or decided to keep it loaded) →  Add Note

Handle: **astra-theme-css** * Stylesheet (.css)

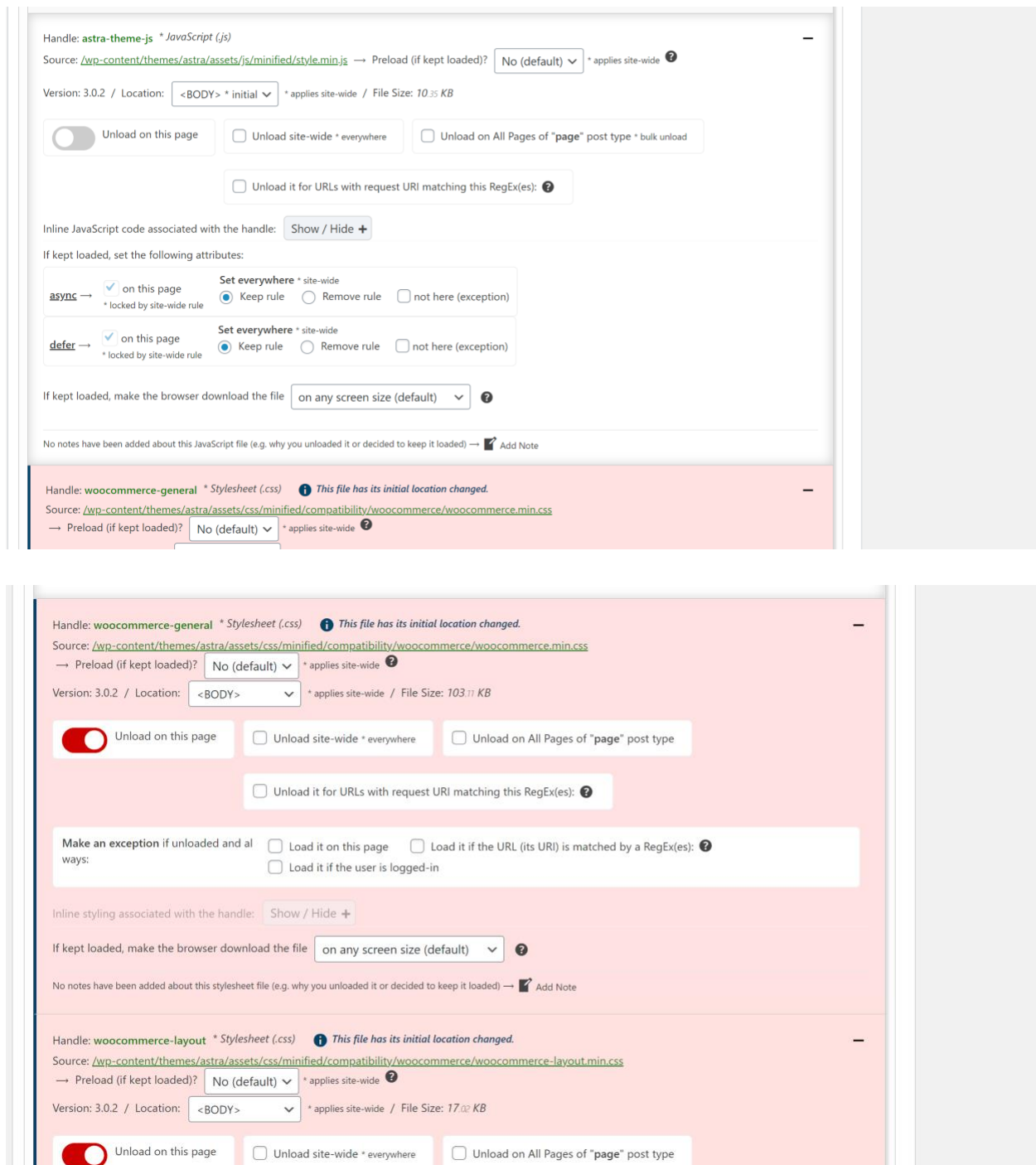
! There are CSS "children" files depending on this file. By unloading it, the following will also be unloaded: [astra-child-theme-css](#) → ☐ Ignore dependency rule and keep the "children" loaded

Source: [/wp-content/themes/astra/assets/css/minified/style.min.css](#) → Preload (if kept loaded)? Yes, async * applies site-wide ?

Version: 3.0.2 / Location: <HEAD> * initial * applies site-wide / File Size: 77.05 KB

☒ Unload on this page ☐ Unload site-wide * everywhere ☐ Unload on All Pages of "page" post type

☐ Unload it for URI's with request URI matching this RegEx(es): ?



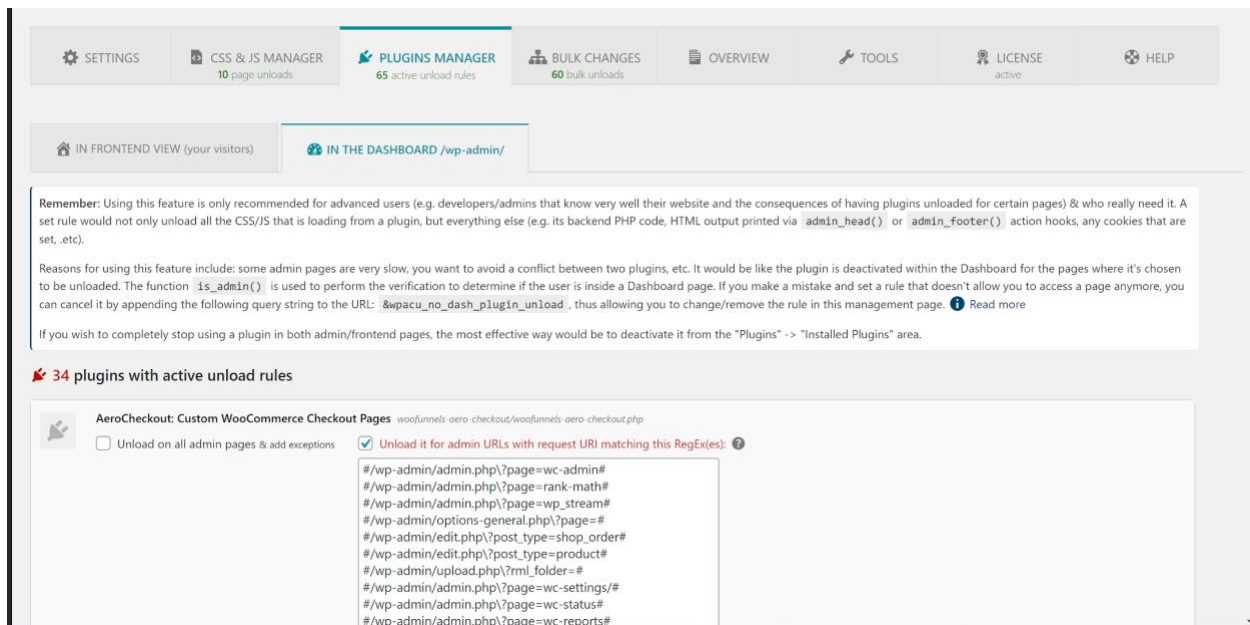
If you're using Elementor and using the theme builder, it replaces a lot of needed functionality from your theme. You should check through the css/js files your theme loads and see if any file of a theme can safely be disabled without breaking any functionality or the design.

Only disable Woocommerce general and Woocommerce layout on non-Woocommerce pages if you are using Astra. Otherwise it will break Woocommerce page layouts.

Asset Cleanup and Query Monitor

If you disable multiple js/css files with asset cleanup before saving the page and something breaks, Query Monitor is very useful for diagnosing which plugin is responsible, and you should know which scripts or css you need to re-enable. It will give you useful php error messages which are usually specific to the plugin having issues, so you'll more easily be able to tell which file you manipulated that's causing the issue.

Asset Cleanup's Plugin Manager



The screenshot shows the 'PLUGINS MANAGER' tab in the Asset Cleanup dashboard. The top navigation bar includes 'SETTINGS', 'CSS & JS MANAGER' (10 page unloads), 'PLUGINS MANAGER' (65 active unload rules), 'BULK CHANGES' (60 bulk unloads), 'OVERVIEW', 'TOOLS', 'LICENSE' (active), and 'HELP'. Below the navigation bar, there are two tabs: 'IN FRONTEND VIEW (your visitors)' and 'IN THE DASHBOARD /wp-admin/'. A warning box states: 'Remember: Using this feature is only recommended for advanced users (e.g. developers/admins that know very well their website and the consequences of having plugins unloaded for certain pages) & who really need it. A set rule would not only unload all the CSS/JS that is loading from a plugin, but everything else (e.g. its backend PHP code, HTML output printed via `admin_head()` or `admin_footer()` action hooks, any cookies that are set, etc). Reasons for using this feature include: some admin pages are very slow, you want to avoid a conflict between two plugins, etc. It would be like the plugin is deactivated within the Dashboard for the pages where it's chosen to be unloaded. The function `is_admin()` is used to perform the verification to determine if the user is inside a Dashboard page. If you make a mistake and set a rule that doesn't allow you to access a page anymore, you can cancel it by appending the following query string to the URL: `&wpacu_no_dash_plugin_unload`, thus allowing you to change/remove the rule in this management page. [Read more](#) If you wish to completely stop using a plugin in both admin/frontend pages, the most effective way would be to deactivate it from the "Plugins" -> "Installed Plugins" area.

Below the warning box, it says '34 plugins with active unload rules'. One plugin is highlighted: 'AeroCheckout: Custom WooCommerce Checkout Pages' with the file path 'woofunnels/aero-checkout/woofunnels/aero-checkout.php'. There are two options: 'Unload on all admin pages & add exceptions' (unchecked) and 'Unload it for admin URLs with request URI matching this RegEx(es):' (checked). A text area contains a list of regex rules for various admin pages.

```
#/wp-admin/admin.php?page=wc-admin#
#/wp-admin/admin.php?page=rank-math#
#/wp-admin/admin.php?page=wp_stream#
#/wp-admin/options-general.php?page=#
#/wp-admin/edit.php?post_type=shop_order#
#/wp-admin/edit.php?post_type=product#
#/wp-admin/upload.php?rml_folder=#
#/wp-admin/admin.php?page=wc-settings/#
#/wp-admin/admin.php?page=wc-status#
#/wp-admin/admin.php?page=wc-reports#
```

You should use the plugin manager(found under asset cleanup in the backend) and disable any plugins that don't need to run on specific pages on the frontend. Any plugin that runs entirely in the backend and does not have any frontend component can be disabled sitewide, you don't need to add an exception.

If you need to add exceptions, they need to be added as regex rules. Please check the asset cleanup documentation for further information on this:

<https://www.assetcleanup.com/docs/what-are-load-exceptions/>


It's more powerful than just disabling js/css. You should also disable plugins on the admin dashboard that are unnecessary on certain backend pages. For instance, you don't need elementor to load on the settings pages of other plugins. This will significantly speed up your wordpress admin dashboard.

Be cautious with this feature, as it will completely unload ALL the plugins code on that page, not just the CSS or JS. It's like the plugin was completely disabled on that page if you disable a plugin via the plugin manager function.


Disable Google Fonts via Asset Cleanup and preload all local fonts

Another thing you can use asset cleanup for to have a huge impact on your site is to disable all of your Google Fonts and host all your fonts locally. Asset Cleanup lets you preload local fonts and apply font display properties as well, i recommend setting the value for that to swap.



Sitewide and html source settings rules in Asset Cleanup's settings

Stripping the fat 

Plugin Usage Preferences



Test Mode 

Optimize CSS


 Minify  Combine + Defer, Inline

Status: Partially locked, already enabled in other plugin(s)

Optimize JavaScript

 Minify  Combine + Defer, Inline

Status: Locked, already enabled in other plugin(s)

CDN: Rewrite assets URLs 

Site-Wide Common Unloads

HTML Source CleanUp

Local Fonts


Font-Display, Preload

Google Fonts

Combine, Async Load, Font-Display, Preconnect, Preload, Removal


Site-Wide Unload For Common CSS & JS Files

This area allows you to quickly add the rule "Unload Site-wide" for the scripts below, which are often used in WordPress environments.

Disable Emojis Site-Wide  This will unload WordPress' Emojis (the smiley icons)

It will fallback to the default browser's Emojis and not fetch the ones from <https://s.w.org/>

As of WordPress 4.2, a new feature was introduced that allows you to use the new Emojis. While on many situations (especially when you are not using WordPress as a blog), you just don't need them. `release.min.js` (12 KB) is loaded (along with extra inline JavaScript code) which adds up to the number


Disable oEmbed (Embeds) Site-Wide  This will remove oEmbed feature

As of WordPress 4.4, a new oEmbed feature was introduced allowing you to embed videos (e.g. from similar things on your website by pasting a URL, which WordPress converts, providing a live preview editor. You can disable this feature if you do not need it.

The situations where you might want to keep it enabled include:

- You want other users to embed your WordPress blog articles on their site.
- You would like to embed other websites' articles, YouTube videos, Tweets etc. on your WordPress blog


If this option is kept enabled, the file `/wp-includes/js/wp-embed.min.js` is also loaded which adds up requests.

Disable Dashicons if Toolbar is hidden (Site-Wide)  This will unload Dashicons if the toolbar (top admin bar) is not showing up. This is because it loads properly, otherwise its layout will be broken. -> `/wp-includes/css/dashicons.min.css` (46 KB + the based on the browser)

The top admin bar (toolbar) requires Dashicons to function properly.

[Read about Dashicons](#) | [Read about Toolbar](#)

This is a CSS file that loads the official icon font of the WordPress admin as of 3.8. While needed for top admin bar and in the styling of other plugins such as Query Monitor, it is sometimes loaded site when it is not needed.

Disable Gutenberg CSS Block Library Site-Wide  This will unload Gutenberg Blocks CSS file -> `/wp-includes/css/dist/block-library/style.min.css` [it or not? Use Coverage from Chrome DevTools!](#)

If you're not using Gutenberg blocks in your posts/page (e.g. you prefer the Classic Editor), then you

In Asset Cleanup's settings, enable all the options you can in the site-wide common unloads tab, unless you need the functionality, such as emojis or oEmbeds. If your builder supports Emojis, they most likely have their own library for implementing them, and there is no reason to leave this feature enabled as it is the Wordpress default one for Gutenberg or the Classic editor.

Disable XML-RPC as well as almost no site relies on its functionality, and it's a security vulnerability. Also enable all options in the HTML CleanUp tab.

Speed Up WP-Admin

WP Admin Cache (Free)

<https://wordpress.org/plugins/wp-admin-cache/>

Admin Speedo (Free)

<https://wordpress.org/plugins/admin-speedo/>

Speed Up The Media Library (Free)

<https://www.itsupportguides.com/knowledge-base/wordpress/using-wordpress-media-library-months-with-files-php-filter/>

Snippets to improve the Media Library Performance.

Inspect HTTP Requests (Free)

<https://wordpress.org/plugins/inspect-http-requests/>

Monitor all the HTTP Request being made via WP HTTP Methods i.e. wp_remote_get, wp_remote_post. Block any request by just a click of a button.

Track how much time a request like updating core/plugin/theme taking (may be useful for bandwidth consumption analysis).

HTTP Requests Manager (Free)

<https://wordpress.org/plugins/http-requests-manager/>

Prevent WP HTTP requests from slowing down your WordPress website and admin interface. The HTTP Requests Manager plugin will log all WP HTTP requests with time taken to complete for each request. If there are multiple requests per page they will be color grouped.

Helps to prevent website slowdown by:

- Sets the request timeout period to 2 seconds. Where default is 5.
- Limits the number of requests per page by 3. Default is unlimited.
- Limit WP HTTP request if page load time is longer than 3 seconds. Default is unlimited.
- Option to block all external requests or allow only requests to wordpress.org for plugin, theme and core updates.

Docket Cache (Free)

<https://wordpress.org/plugins/docket-cache/>

In addition to being an object caching plugin, it has many options to speed up both the admin backend and the frontend as well in its configuration settings.

Translations

Performant Translations (Free)

<https://wordpress.org/plugins/performant-translations/>

This project uses a new approach to handle translation files in WordPress, making localization blazing fast.

An in-depth i18n performance analysis showed that localized WordPress sites load significantly slower than a site without translations.

Note: WordPress 6.5 and newer required

Important: this new functionality has been merged into WordPress 6.5! However, this plugin is still useful!

On WordPress 6.5+, this plugin automatically converts existing .mo files to .php and afterwards only loads the translations from the .php file.

This is useful for cases where language packs are not downloaded from WordPress.org but somewhere else.

Translation Cache (Free)

<https://wordpress.org/plugins/speed-up-translation/>

Has not been updated in over a year, test this on staging first!

Translatio (Free)

<https://wordpress.org/plugins/tmy-globalization/>

Translatio is an open source tool for internationalization and localization of WordPress based websites.

Sublanguage (Free)

<https://wordpress.org/plugins/sublanguage/>

Ultra lightweight translation plugin.

Performant Code Snippets Plugin

Fluent Snippets (Free)

<https://wordpress.org/plugins/easy-code-manager/>

FluentSnippets store your snippets in flat files, so it does not run SQL queries for your snippets. It is the Fastest Code Snippet Plugin for WordPress.

Performance Code Snippets (Free)

Luke Cavs Repo

<https://github.com/lukecav/code-snippets-wp-speed-up>

Web Squadron Snippets

<https://websquadron.co.uk/page-speed-performance-wordpress-code-snippets/>

Geekflare

<https://geekflare.com/wordpress-performance-optimization-without-plugin/>

WPMunk

<https://wpmunk.com/wordpress-performance-optimization-without-plugin/>

Code Validation (Free)

Code Validation errors can cause performance issues.

CSS Validators

Jigsaw W3 Validator (Free)

<https://jigsaw.w3.org/css-validator/>

CSS Portal (Free)

<https://www.cssportal.com/css-validator/>

Css Validator (Free)

<https://css-validator.org/>

Lambda Test (Free)

<https://www.lambdatest.com/free-online-tools/css-validator>

JS Validators

Code Beautify (Free)

<https://codebeautify.org/jsvalidate>

Validate Javascript (Free)

<https://validatejavascript.com/>

Site 24x7 Validator (Free)

<https://www.site24x7.com/tools/javascript-validator.html>

Esprima JS Validator (Free)

<https://esprima.org/demo/validate.html>

Fix HTML Validation Errors

W3 HTML Validator (Free)

<https://validator.w3.org/>

HTML Validation errors can cause performance issues. Use this tool to audit your site for html validation errors that need to be fixed.

HTML Validation (Free)

<https://wordpress.org/plugins/html-validation/>

HTML Validation Pro (Paid)

<https://www.alumnionlineservices.com/our-plugins/html-validation/#proext>

Automatically fixes a variety of HTML errors.

HTML Tidy (Free)

<https://www.html-tidy.org/>

Tidy is a console application for macOS, Linux, Windows, UNIX, and more. It corrects and cleans up HTML and XML documents by fixing markup errors and upgrading legacy code to modern standards.

HTMLtidy.net (Free)

<https://htmltidy.net/>

Static Sites

Site Generation

Pre-Generated HTML

A static site consists of pre-generated HTML files that are created during the development process. These files are directly uploaded to the web server and served to users as-is.

No Server-Side Processing

Since the content is static, there is no need for server-side processing such as PHP execution or database queries.

Server Resources and Performance

Low Server Load

Serving static HTML files is resource-efficient and requires minimal server processing power, resulting in faster load times and the ability to handle high traffic with ease.

No Database

Static sites do not require a database, which further reduces server complexity and potential points of failure.

Flexibility and Functionality

Limited Dynamic Content

Static sites are excellent for purely static content but require additional tools or frameworks (e.g., JavaScript, APIs) to handle dynamic functionality such as forms, user logins, or real-time updates.

Simpler Deployment

Deployment is straightforward since there are no server-side scripts to configure. Static files can be hosted on any web server or CDN.

Security

Low Attack Surface

With no server-side processing or database, static sites have a much lower attack surface, making them inherently more secure against common web vulnerabilities (e.g., SQL injection, PHP exploits).

Static Site Generators

Plugins

Simply Static (Freemium)

<https://wordpress.org/plugins/simply-static/>

Turn your website into a static site! Could significantly reduce hosting costs (potentially even to zero!), could increase site security and has many potential speed benefits. Your site will be converted to a purely html based site! This comes with many limitations, and if you have any dynamic resources this is most likely not going to be an option for you.

Staatic (Freemium)

<https://wordpress.org/plugins/staatic/>

Export Site To Static HTML (Freemium)

<https://wordpress.org/plugins/export-wp-page-to-static-html/>

Export WP page to static HTML/CSS plugin is a static html site or page generator that helps you to convert your wordpress site or pages to static html website and you can host it on your server, from a static html hosting provider or from a cdn.

WP2Static (Free)

<https://github.com/elementor/wp2static>

Hugo (Free)

<https://gohugo.io/>

Open source static site generator written in Go

Strattic (Paid)

<https://www.strattic.com/>

Strattic is an all-in-one static site generation and hosting platform that instantly optimizes WordPress by converting it to a static architecture. With Strattic, content managers and marketers can continue to manage content in WordPress as usual, while developers get to bask in the glory (and enjoy the peace of mind) of a fully headless website.

Pelican (Free)

<https://getpelican.com/>

Free Static Site Generator written in Python.

Limitations of Static Sites

1. No search (without a custom implementation such as Algolia, Elastic Search, Typesense etc)
2. No locally stored comments

These are just some of the potential pitfalls (some additional pitfalls not listed here), please read the links below.

Further reading

<https://blog.hubspot.com/website/static-vs-dynamic-website#:~:text=Static%20Website%20Disadvantages&text=The%20most%20apparent%20issue%20is,create%20a%20new%20HTML%20file>

<https://uk.indeed.com/career-advice/career-development/static-vs-dynamic-website>

<https://drewl.com/blog/advantages-and-disadvantages-static-website/>

As always, test thoroughly!

Quality of Life

Hide Admin Backend Notices

Admin Notification Center (Free)

<https://wordpress.org/plugins/wp-admin-notification-center/>

Lightweight Plugin To Hide Notices on the Admin Dashboard.

Show Page IDs

Echo Show IDs (Free)

<https://wordpress.org/plugins/echo-show-ids/>

Lightweight plugin which displays Page IDs on the admin dashboard.

Duplicate Pages

WP Duplicate Page (Free)

<https://wordpress.org/plugins/wp-duplicate-page/>

Lightweight Page Duplication plugin.

Tree Format Page Overview

CMS Tree Page Overview (free)

<https://wordpress.org/plugins/cms-tree-page-view/>

Adds a CMS-like tree overview of all your pages and custom posts to WordPress – much like the view often found in a page-focused CMS.

Within this tree you can edit pages, view pages, add pages, search pages, and drag and drop pages to rearrange the order.

Re-Enable The Classic Editor

Disable Gutenberg (Free)

<https://wordpress.org/plugins/disable-gutenberg/>

Restores the classic editor and disables the Gutenberg interface when editing a page.

HTAccess Editor

WP HTAccess Editor (Free)

<https://wordpress.org/plugins/wp-htaccess-editor/>

Htaccess editing from within the admin dashboard.

Media Library Folders

Real Media Library (Free)

<https://wordpress.org/plugins/real-media-library-lite/>

Adds folders and categories to the media library.

Nested Pages

WP Nested Pages (Free)

<https://wordpress.org/plugins/wp-nested-pages/>

- Provides a simple & intuitive drag and drop interface for managing your page structure and post ordering
- Enhanced quick edit functionality
- Adds an editable, sortable tree view of your site's page structure
- Automatically generates a native WordPress menu that matches your page structure
- A way to quickly add multiple pages & posts (ideal for development)
- Works with any post type
- Works on touch-enabled devices

Editor Tools

Advanced Editor Tools (Free)

<https://wordpress.org/plugins/tinymce-advanced/>

Advanced Editor Tools (previously TinyMCE Advanced) introduces a “Classic Paragraph” block for the block editor (Gutenberg).

If you are not quite ready to switch to the block editor, or have plugins that cannot be used there (yet), using the Classic Paragraph block is your best option. It lets you continue to use the familiar TinyMCE editor for most tasks, and at the same time gives you full access to all blocks and new features in the block editor.

Custom Statuses

WP Custom Status Manager (Free)

<https://wordpress.org/plugins/hw-wp-status-manager/>

Lightweight Custom Status Manager

Rollback Theme and Plugin Updates

WP Rollback (Free)

<https://wordpress.org/plugins/wp-rollback/>

Quickly and easily rollback any theme or plugin from WordPress.org to any previous (or newer) version without any of the manual fuss. Works just like the plugin updater, except you’re rolling back (or forward) to a specific version. No need for manually downloading and FTPing the files or learning Subversion. This plugin takes care of the trouble for you.

Dark Mode

WP Markdown Editor (Free)

<https://wordpress.org/plugins/dark-mode/>

Formerly titled “Dark Mode”, this is a ultra-lightweight plugin with editor enhancement features, as well as both a frontend and backend Dark Mode.

- Typography Settings – You can change the typography of your editor screen, using four different fonts, and controls for font size, line height, line width, and paragraph spacing.
- Ready-made Color Schema – Quickly choose your favorite color from the ready-made color schema available in WP Markdown. Highlight and edit your content the way you want.
- Syntax Shortcuts – A Markdown cheat sheet is available for you which shows other keyboard shortcuts for heading, formatting and the interface.
- Emoji Supported – You can make your content more interactive and attractive with emojis. WP Markdown supports all types of emojis in your write up.
- Custom Colors – You can customize the background text and color as you want with the custom color feature of our WP Markdown. Pick the color schema of your preference and make it yours!
- Makes it easy and fast – formatting in WordPress without thinking about the blocks.
- Dark Mode Switch – You can toggle between the dark and light mode using the admin bar dark mode switch menu anytime.
- Auto Match OS Theme – Automatically shows Darkmode if the OS preferred theme is dark.
- Remember Dark Mode – The dark mode remains saved once you activate the dark mode.

DarkMySite (Free and Pro)

<https://wordpress.org/plugins/darkmysite/>

Ultra-lightweight Dark Mode. Small codebase, unlikely to conflict with other plugins. As with everything else, test on staging first before installing it on live!

Note: Make sure you have external file manager access (SSH, Cpanel, etc) in case you get locked out of the dashboard somehow so you can remove the plugin from the plugins folder, just to be safe.

WP Admin Search

Admin Search (Free)

<https://wordpress.org/plugins/admin-search/>

Admin Search makes searching your WordPress website easy by bringing results from all your post types, media, taxonomies, comments, users and admin pages together in a single, simple-to-use interface, seamlessly integrated into the WordPress admin backend interface.

Choose which post types and taxonomies are searched and the number of results displayed for each. Admin Search also supports custom post types and taxonomies.

SSH Update Support

SSH SFTP Updater Support (Free)

<https://wordpress.org/plugins/ssh-sftp-updater-support/>

Update plugins and Wordpress over SFTP

Admin and Site Enhancements (Free and Pro)

<https://wordpress.org/plugins/admin-site-enhancements/>

Lightweight Site Enhancements Toolkit

File Manager

FileOrganizer (Free)

<https://wordpress.org/plugins/fileorganizer/>

Lightweight free file manager.

Filester (Free)

<https://wordpress.org/plugins/filester/>

Completely free performant file manager

File Manager, Code Editor, and Backup (Free)

<https://wordpress.org/plugins/softdiscover-db-file-manager/>

Completely free combo File Manager, Code Editor and Backup plugin. Medium-heavy weight, like every other backend only plugin, disable it on the frontend.

Maintenance Mode

Maintenance (Free)

<https://wordpress.org/plugins/maintenance/>

Free Lightweight Maintenance mode plugin.

Slim Maintenance Mode (Free)

<https://wordpress.org/plugins/slim-maintenance-mode/>

Another free lightweight maintenance mode plugin

Code Editor

File Manager, Code Editor, and Backup (Free)

<https://wordpress.org/plugins/softdiscover-db-file-manager/>

Completely free combo File Manager, Code Editor and Backup plugin. Medium-heavy weight, like every other backend only plugin, disable it on the frontend.

Backups

Make sure you disable your backup plugin from running in the frontend with Asset Cleanup or another plugin which can selectively disable plugins.

Backup Vault (Free)

<https://wordpress.org/plugins/backup-bolt/>

Lightweight backup plugin, completely free.

WP Vivid (Free)

<https://wordpress.org/plugins/wpvivid-backuprestore/>

Free backup and restore plugin, has staging functionality included in the free version. Make sure to disable the plugin on the frontend with one of the plugins in the “selectively deactivate plugins” section as it doesn’t need to run in the frontend!

File Manager, Code Editor, and Backup (Free)

<https://wordpress.org/plugins/softdiscover-db-file-manager/>

Completely free combo File Manager, Code Editor and Backup plugin. Medium-heavy weight, like every other backend only plugin, disable it on the frontend.

FastDup (Free)

<https://wordpress.org/plugins/fastdup/>

Lightweight backup plugin. Full featured other than missing cloud storage functionality.

Total Upkeep (Free)

<https://wordpress.org/plugins/boldgrid-backup/>

- Automated and manual backups
- Full file and database backup or customize settings based on your needs
- Remote backups via FTP / SFTP, Amazon S3 and Google Drive (Premium)
- Total Upkeep checks to ensure that your web server has the necessary features to properly create backup archives, which protects the integrity of your backups
- Clone, duplicate and/or migrate your site with just a few clicks
- Site Check monitors your site for issues that could lead to site crashes, provides a tool set to restore your site even if your WordPress installation is inaccessible
- Auto rollback feature creates a backup before updates, restores your site to the last backup if an update fails
- Create staging sites to test new plugins or themes with Total Upkeep + Cloud WordPress

Clone (Free)

<https://wordpress.org/plugins/wp-clone-by-wp-academy/>

Free backup plugin, medium weight.

Greenbackup (Freemium)

<https://wordpress.org/plugins/green-backup-by-greenwpix/>

(Free Plan): Unlimited sites, backups, and migrations. Backups locally.

Pro Version: Backups locally or on various cloud services like: Dropbox, Google Drive, FTP/SFTP, Amazon S3, Wasabi, OneDrive, Google Cloud, Storage, Backblaze B2, Microsoft Azure Blob, DigitalOcean Space.

Xcloner (Free)

This plugin is **heavy**. But it has a ton of pro features in a free plugin (like cloud backups). Make sure to selectively disable it on the frontend to mitigate its performance impact for users.

Duplicator (Paid)

<https://duplicator.com/>

Great backup plugin.

Search and Replace

Search & Replace Everything (Free)

<https://wordpress.org/plugins/search-replace-wpcode/>

An ultra lightweight search and replace plugin. Search & Replace Everything enables you to efficiently manage your website's content directly from the WordPress admin. This tool is essential for site migrations, content updates, or any situation where batch text and image replacements are needed.

You can perform detailed search and replace operations across your entire database. Select specific tables, toggle case sensitivity, and preview changes before committing. Designed to handle large websites.

Search Regex (Free)

<https://wordpress.org/plugins/search-regex/>

Search Regex adds a powerful set of search and replace functions to WordPress posts, pages, custom post types, and other data sources. These go beyond the standard searching capabilities, and allow you to search and replace almost any data stored on your site. In addition to simple searches you have the full power of PHP's regular expressions at your disposal.

Bulk Update URLs

Update URLs (Free)

<https://wordpress.org/plugins/update-urls/>

Bulk update URLs sitewide.

Media

Media Library Folders

Categorify (Free)

<https://wordpress.org/plugins/categorify/>

Lightweight Free Media Folders plugin.

Media Library Folders (Free)

<https://wordpress.org/plugins/media-library-plus/>

- Actual folders make it easier to organize your WordPress media library while reducing server burden.
- Add and build new Media library folders to label and organize as you wish instead of just month/date.
- Move, copy, rename and delete files and folders with a nice drag and drop interface
- Regenerate thumbnails.
- SEO Images to specify ALT and TITLE attributes when uploading.
- Sync folders/files when moving or uploading a folder via FTP.
- Create a MaxGalleria gallery.
- Block direct access for selected media library files

Replace Media

Enable Media Replace (Free)

<https://wordpress.org/plugins/enable-media-replace/>

A free, lightweight and easy to use plugin that allows you to seamlessly replace an image or file in your Media Library by uploading a new file in its place. No more deleting, renaming and re-uploading files! Only needs to run in the backend.

Media Cleaner (Free)

<https://wordpress.org/plugins/media-cleaner/>

Media Cleaner – removes all unnecessary media and broken entries.

Make sure to take backups before running this just in case

Protect Media Files

AAM Protected Media Files (Free)

<https://wordpress.org/plugins/aam-protected-media-files/>

Email

Ultimate WP Mail (Free)

<https://wordpress.org/plugins/ultimate-wp-mail/>

- Send automatic email messages or manual email messages to specific users or all users
- Send WooCommerce emails to purchasers and for specific actions (e.g. abandoned cart)
- Clean visual interface for composing emails
- Create smart lists of users to send emails to
- Trigger automatic email sending for specific actions (e.g. user registration, new post published, etc.)
- Great email previewing and scheduled sending features
- Included email templates for common emails
- Email logging

SMTP

Fluent SMTP (Free)

<https://wordpress.org/plugins/fluent-smtp/>

Lightweight full featured SMTP plugin

Newsletters

FluentCRM (Free and Pro)

<https://wordpress.org/plugins/fluent-crm/>

Performant, locally hosted Newsletter/CRM plugin. Unlimited emails on their free forever plan.

Email Address Encoding

Email Address Encoder (Free)

<https://wordpress.org/plugins/email-address-encoder/>

A lightweight plugin that protects/obscures plain email addresses and mailto links from email-harvesting robots, by encoding them into decimal and hexadecimal entities. Has an effect on the posts, pages, comments, excerpts, text widgets and other filtered content. Works without JavaScript — just simple spam protection.

Users

User Access Management

User Access Manager (Free)

<https://wordpress.org/plugins/user-access-manager/>

Lightweight User Access Manager

Allows you to manage the access of your content. This is useful if you need a member area, a private section at your blog or you want that other people can write at your blog but not everywhere.

Including all post types, taxonomies and files by creating user groups. Just assign the content you want to restrict and your registered users which should have access to a group. From now on the content is only accessible and writable for the specified group.

User Switcher

User Switching (Free)

<https://wordpress.org/plugins/user-switching/>

Ultra Lightweight User Switching Plugin

This plugin allows you to quickly swap between user accounts in WordPress at the click of a button. You'll be instantly logged out and logged in as your desired user. This is handy for helping customers on WooCommerce sites, membership sites, testing environments, or for any site where administrators need to switch between multiple accounts.

Login As Users (Free)

<https://wordpress.org/plugins/login-as-users/>

This plugin allows you to quickly change between user accounts at the click of a button.

This plugin adds the ability to access any user account without having the user password by just clicking one button and thus solve problems or provide better and faster customer support. With one click admin will be logged in as the specific user and then he can handle any situation without wasting much time.

Switch Post Authors

Post Author Switcher (Free)

<https://wordpress.org/plugins/post-author-switcher/>

Change the author of multiple posts simultaneously. Additionally you can switch post author by post type, post status and existing post author.

Multiple User Roles

Multiple Roles (Free)

<https://wordpress.org/plugins/multiple-roles/>

Allows you to assign more than one role to a user account.

User Deletion

Inactive User Deleter (Free)

<https://wordpress.org/plugins/inactive-user-deleter/>

Lightweight mass user deleter based on specified criteria.

Delete Me (Free)

<https://wordpress.org/plugins/delete-me/>

Allow users to delete their own user account.

Remote User Sync

WP Remote User Sync (Free)

<https://wordpress.org/plugins/wp-remote-users-sync/>

If you run multiple websites and want to keep users separated, but synchronize them automatically and securely for specific user operations, then WP Remote Users Sync is the plugin to use.

Taxonomies

Simple Taxonomies (Free)

<https://wordpress.org/plugins/simple-taxonomy-refreshed/>

Supports adding one or more taxonomies (either hierarchical or tag) to any objects registered on your installation. This plugin allows you to add a taxonomy just by giving them a name and some options in the backend. It then creates the taxonomy for you and takes care of the URL rewrites.

Media Library Attachment Taxonomies

Attachment Taxonomies (Free)

<https://wordpress.org/plugins/attachment-taxonomies/>

Lightweight Free Attachment Taxonomies Plugin.

The plugin adds two taxonomies to the WordPress media library which are then available to categorize and tag your attachments. By default, these taxonomies, although sharing the same names and behavior, are separate from the default post taxonomies, but this can easily be changed if desired.

Images

Taxonomy Images (Free)

<https://wordpress.org/plugins/taxonomy-images/>

Associate images from your media library to categories, tags and custom taxonomies.

Taxonomy Filtering

Beautiful Taxonomy Filters (Free)

<https://wordpress.org/plugins/beautiful-taxonomy-filters/>

The Beautiful Taxonomy Filters plugin is an easy and good-looking way to provide your visitors with filtering for your post types. With this you get a complete solution for adding filtering based on custom taxonomy terms/categories/tags. It will also automatically add rewrite rules for pretty looking filter URLs.

It's completely automatic, works without javascript and is based on the WordPress Plugin boilerplate for a standardized, organized and object-oriented codebase. It uses select2 for pretty looking and user friendly dropdowns but will fall back to ordinary ones if javascript is not supported.

Taxonomy Filter (Free)

<https://wordpress.org/plugins/taxonomy-filter/>

Taxonomy Filter is a simple and flexible plugin which allows users to filter hierarchical term taxonomies inside admin pages. If you need to simplify your tags and categories research on admin pages, this plugin will make it easier for you. It adds a custom input field (only for configured taxonomies) that you can use to filter every taxonomy list.

Custom Fields

ACF

ACF Extended Performance Mode (Paid)

<https://www.acf-extended.com/features/modules/performance-mode>

The Performance Mode is a unique ACF module that allows developers to optimize ACF metadata and enhance database load with two different methods: Ultra & Hybrid Engines.

ACF Better Search (Free)

<https://wordpress.org/plugins/acf-better-search/>

Ultra-lightweight ACF search plugin. This adds to the default WordPress search engine the ability to search by content from selected ACF fields.

ACF Caching

ACF Simple Cache (Free)

<https://wordpress.org/plugins/acf-simple-cache/>

Caches ACF Fields via json file for improved performance.

MetaBox

Optimizing MetaBox Custom Field Database Performance (Free)

<https://metabox.io/optimizing-database-custom-fields/>

Each custom field is stored as one row in the database. This approach allows developers to store unlimited data regardless of its structure. But, the downside is the rapid bloat of the database. Because the number of custom fields is usually very large.

This article will present solutions to optimize the storage of custom fields in the database to help boost your website performance.

MetaBox Groups (Paid)

<https://metabox.io/plugins/meta-box-group/>

Database optimization of sub-fields for reduced bloat

Custom Post Types & Custom Taxonomies (Free)

<https://wordpress.org/plugins/mb-custom-post-type/>

MB Custom Post Types & Custom Taxonomies helps you to register and edit custom post types and custom taxonomies easily in WordPress by providing an easy-to-use UI in the admin area.

SEO

All-In-One SEO plugins

SEO Pyramid (Free)

<https://wordpress.org/plugins/seo-pyramid/>

Neat, very lightweight SEO Toolkit. Schema functionality included (rare for a free plugin). Almost all the functionality you'd find in a paid SEO plugin. Frequently updated with enhancements and new features.

Note: Activate on staging first

The SEO Framework (Freemium)

<https://theseoframework.com/>

<https://wordpress.org/plugins/autodescription/>

Rankmath (Freemium)

Rankmath is a very well optimized plugin that is not detrimental to pagespeed.

Make sure you enable sitemap functionality in Rankmath. This will allow WP-Rocket to detect the Rankmath sitemap, which should increase preload efficiency.

Yoast (Freemium)

<https://wordpress.org/plugins/wordpress-seo/>

Hide SEO Bloat (Free)

<https://wordpress.org/plugins/so-clean-up-wp-seo/>

Removes Yoast backend bloat

WP Meta SEO (Free)

<https://wordpress.org/plugins/wp-meta-seo/>

Tons of features in the free version. There's a pro add-on, but not necessary as the functionality can be achieved via other free plugins. Free version has quite a lot of useful features. **This plugin is heavy, make sure to disable it on the frontend as all of the changes it makes are inserted via the backend.**

- Bulk edit all website meta on a single view
- Edit meta in content with live SEO analysis
- Fix HTML image resizing in content
- Bulk edit image file name and meta
- Bulk edit SEO link title
- 404 errors redirect and internal broken link checker
- Add Google Analytics tracking information and display statistics on WordPress
- Generate XML and HTML sitemaps
- Facebook and Twitter social sharing custom elements per content
- Breadcrumb generator
- Gutenberg SEO content checker
- Canonical URL management for post, pages and categories
- Elementor full integration with live onPage SEO in the Elementor page editor

SEO Engine (Free)

<https://wordpress.org/plugins/seo-engine/>

- Auto-generate meta titles and descriptions
- Automatically use the {TITLE} | {SITENAME} format for titles
- Convenient UI to override titles and descriptions, with AI
- Automatic sitemap.xml and robots.txt generation
- Social Cards for SNS and Open Graph
- Search Engine Ranking
- WooCommerce Assistant (AI)
- Prevent GPTBot (OpenAI) from using your content

Topical Map

Topical Map and Internal Link Flow (Free)

<https://wordpress.org/plugins/internal-link-flow-topical-authority-topical-map/>

Keyword Research

Keyword Research Tool (Free)

<https://wordpress.org/plugins/keyword-research-tool/>

Free keyword research plugin.

RankMyWP (Free)

<https://wordpress.org/plugins/rank-my-wp/>

Keyword Research Tool

- Over 140 Keyword Research Countries
- Check Keyword Country
- Check Keyword Competition
- Check Keyword Search Volume
- Check Keyword Social Popularity
- Save Keyword for SEO Optimization
- Save Keyword in Briefcase

Keywords Briefcase

- Unlimited Keyword Storage
- Categorize Keywords by SEO Strategy
- Keywords Research Information
- Save Keyword for SEO Optimization
- Keyword Research History
- Keywords Suggestion
- Save Keyword for SEO Optimization
- Add Labels to Keywords
- Filter Keywords by Labels

Google Ranking

- Unlimited Keywords in Google Ranking
- Connect to Google Search Console (GSC)
- GSC Keywords Synchronization
- Add Keywords from Briefcase
- Google Ranking with GSC average position
- Monitor Google average position over time

Broken Link Checking

Broken Link Checker (Free)

<https://wordpress.org/plugins/broken-link-checker/>

Lightweight Broken Link Checker

Instant Indexing

Index Now (Free)

<https://wordpress.org/plugins/mihdan-index-now/>

It can take days to weeks for search engines to discover that the content has changed, as search engines don't crawl every URL often. IndexNow will let search engines know immediately when content has been updated.

Internal Linking

SEO Booster (Free)

<https://wordpress.org/plugins/seo-booster/>

Free neat little SEO toolkit with automatic internal linking.

WordPress Internal Link Optimiser (Free)

<https://wordpress.org/plugins/internal-link-finder/>

Elevate your website's search engine optimisation (SEO) game by effortlessly identifying and capitalizing on valuable internal linking opportunities while efficiently managing your existing network of internal links.

A robust internal link structure is paramount for driving exceptional search engine performance and on-page SEO.

Link Whisper (Paid)

<https://linkwhisper.com/>

Solid internal linking plugin

Auto Internal Links (Free and Pro)

<https://wordpress.org/plugins/automatic-internal-links-for-seo/>

Automatic Internal Linking plugin.

Backlink Monitoring

SEO Booster (Free)

<https://wordpress.org/plugins/seo-booster/>

Backlink discovery. Free version requires manual scans, paid version has automated monitoring.

External Links

WP External Links (Free)

<https://wordpress.org/plugins/wp-external-links/>

Manage all external & internal links on your site. Control icons, nofollow, noopener, ugc (User Generated Content), sponsored and if links open in new window or new tab.

404 Monitoring

SEO Booster (Free)

<https://wordpress.org/plugins/seo-booster/>

Built in 404 page monitoring

Redirects

Redirection (Free)

<https://wordpress.org/plugins/redirection/>

Open Source Free Redirect Plugin. Most SEO plugins have this feature built in so this shouldn't be needed, but I'll leave it here just in case.

External Tools Toolkit

Opace Essential SEO Toolkit (Free)

<https://wordpress.org/plugins/opace-essential-seo-toolkit/>

Add links to multiple external tools directly within the admin dashboard to simplify and speed up your SEO workflow.

Schema

SEO Pyramid (Free)

<https://wordpress.org/plugins/seo-pyramid/>

Neat lightweight SEO plugin, schema functionality included for free (rare).

WPSSO Core (Free)

<https://wordpress.org/plugins/wpsso/>

Complete Structured Data Plugin for WordPress

Provides complete structured data for WordPress to present your content at its best for social sites and search results, no matter how URLs are shared, reshared, messaged, posted, embedded, or crawled.

Performant Sitemaps

Super Speedy Sitemaps (Free)

<https://www.superspeedyplugins.com/product/super-speedy-sitemaps/>

Super Speedy Sitemaps is a lightning fast sitemap index with Google News and Multisite support.

Seamlessly replace Yoast and Rank Math sitemaps and you can benefit from the ultra-fast sitemap generation in this plugin while still using the rest of the features from Yoast/Rank Math that you love.

Breadcrumbs

Flexy Breadcrumb (Free)

<https://wordpress.org/plugins/flexy-breadcrumb/>

Lightweight breadcrumbs plugin, works via shortcode.

Open Graph Tagging

Open Graph and Twitter Card Tags (Free)

<https://wordpress.org/plugins/wonderm00ns-simple-facebook-open-graph-tags/>

Improves sharing of your WordPress pages, posts, WooCommerce products, or any other post type on social media, by setting the correct Facebook Open Graph Tags.

DataCaptia (Free)

<https://wordpress.org/plugins/data-captia/>

Lightweight Open Graph Tags for Social Media Sharing. Helps with SEO and Social Sharing Reach.

Content

Related Posts

Super Related Posts (Free)

<https://wordpress.org/plugins/super-related-posts/>

Related posts plugin to improve traffic & bounce-rate with a superior algorithm. Zero server load & highly configurable related post plugin.

YARPP (Free)

<https://wordpress.org/plugins/yet-another-related-posts-plugin/>

Another performant related posts plugin.

Content Writing

AI Content Writing Assistant (Free)

<https://wordpress.org/plugins/ai-content-writing-assistant/>

AI Content Writing Assistant – Content Writer, ChatGPT, Image Generator All in One generates high-quality, unique content and stunning images with just one click.

RSS Feeds

FeedWordPress (Free)

<https://wordpress.org/plugins/feedwordpress/>

FeedWordPress is an Atom/RSS aggregator for WordPress. It syndicates content from feeds that you choose into your WordPress weblog, and then the content it syndicates appears as a series of special posts in your WordPress posts database. If you syndicate several feeds then you can use WordPress's posts database and templating engine as the back-end of an aggregation ("planet") website.

Workflow Management

Oasis Workflow (Free)

<https://wordpress.org/plugins/oasis-workflow/>

Oasis Workflow is a powerful feature rich plugin designed to automate any editorial workflow process using a simple, intuitive drag and drop interface.

Post Scheduling

Editorial Calendar (Free)

<https://wordpress.org/plugins/editorial-calendar/>

Neat little free plugin to schedule posts. Disable it on the frontend!

Schedule Posts Calendar (Free)

<https://wordpress.org/plugins/schedule-posts-calendar/>

Extremely lightweight scheduled post scheduling plugin. It only needs to run in the backend, make sure to disable it in the frontend to eliminate its performance impact!

Content Importing

Wordpress Importer (Free)

<https://wordpress.org/plugins/wordpress-importer/>

The WordPress Importer will import the following content from a WordPress export file:

- Posts, pages and other custom post types
- Comments and comment meta
- Custom fields and post meta
- Categories, tags and terms from custom taxonomies and term meta
- Authors

Accessibility

Accessibility Plus (Free)

<https://wordpress.org/plugins/accessibility-plus/>

Nice free plugin which implements some accessibility improvements.

Accessibe (Free)

<https://wordpress.org/plugins/accessibe/>

Free Accessibe Accessibility Plugin

WP Accessibility (Free)

<https://wordpress.org/plugins/wp-accessibility/>

Lightweight. This plug-in helps with a variety of common accessibility problems in WordPress themes. While most accessibility issues can't be addressed without directly changing your theme, WP Accessibility adds a number of helpful accessibility features with a minimal amount of setup or expert knowledge.

Automation

WP Pipes (Free)

<https://wordpress.org/plugins/wp-pipes/>

Cool and entirely free automation plugin. Doesn't have tons of features, but it's pretty neat.

AutomatorWP (Freemium)

<https://wordpress.org/plugins/automatorwp/>

Open Source free automation plugin. Lightweight than Uncanny Automator. Make sure you have object caching enabled and reduce any autoload options in your database to the minimum required to allow the plugin to continue functioning unimpaired. Also selectively unload it on the frontend, it only needs to run in the backend.

Note: Paid addons

Uncanny Automator (Freemium)

<https://wordpress.org/plugins/uncanny-automator/>

Uncanny Automator is heavy and can have a large negative performance impact, so I would strongly suggest selectively disabling it on the frontend. Automation in general is database query heavy, so make sure you have some form of object caching enabled! Automation actions are frequently called with cron jobs, so you'll have to play around with disabling some Uncanny Automator Database autoload options to reduce the performance impact.

Those options should still be loaded when they are called via cron or another API, so they *should* largely be safe to remove from autoload, but please test carefully after changing database option autoload settings to ensure nothing has broken! **Always take backups before changing autoload options.**

If-So (Freemium)

<https://wordpress.org/plugins/if-so/>

Allows you to set trigger conditions for automated actions.

AutomateWoo (Paid)

<https://automatewoo.com/>

Official Automattic created WooCommerce automation plugin. I've previously experienced jank with this plugin before, but the last time I used it was a few years ago. The interface is a bit wonky, and like any other automation plugin it will be database heavy. Same deal, selectively disable it on the frontend (it only needs to run in the backend), and make sure you have some form of object caching.

Site Search

ElasticSearch (Free)

ElasticPress (Free)

<https://wordpress.org/plugins/elasticpress/>

Why you should implement ElasticSearch

<https://guides.wp-bullet.com/install-configure-elasticsearch-to-speed-up-wordpress-search/>

How to implement ElasticSearch Locally (Free)

<https://mayvenstudios.com/blog/implement-elasticsearch-wordpress-site>

Optimizing ElasticSearch Relevance Article

<https://bigdataboutique.com/blog/optimizing-elasticsearch-relevance-a-detailed-guide-c9efd3>

Typesense (Free)

How to locally install TypeSense

<https://typesense.org/docs/guide/install-typesense.html>

Typesense Plugin (Free)

<https://wordpress.org/plugins/search-with-typesense/>

Super Speedy Search (Paid)

<https://www.superspeedyplugins.com/product/super-speedy-search/>

WP Extended Search (Free)

<https://wordpress.org/plugins/wp-extended-search/>

- Search in selected meta keys
- Search in category, tags or custom taxonomies
- Search in the post author name
- Include or exclude any public post type
- Control whether to search in title or content or excerpt or all

- Compatible with WooCommerce. Search in product SKU, Attributes, variations, and custom fields, etc.
- Create unlimited search settings to use with custom search forms.
- Add search forms using a widget, shortcode or PHP function. Also works with custom searchform.php
- Exclude old content from search results (Older than admin specified date)
- Customize the number of posts to display on the search results page
- Customize SQL relation (AND/OR) between search terms
- Customize order of search results
- Control whether to match search query exactly or partially
- Limit attachment results by mime type e.g. display only pdf files in search results.
- Translation ready
- Compatible with WPML

Filtering

Super Speedy Filters (Paid)

<https://www.superspeedyplugins.com/product/super-speedy-filters/>

Performant filters plugin, works for regular Wordpress post types and Woocommerce.

Filters and Grids (Free)

<https://wordpress.org/plugins/ymc-smart-filter/>

Free Filters plugin

Themify Woocommerce Product Filters (Free)

<https://wordpress.org/plugins/themify-wc-product-filter/>

Free Woocommerce Product Filters

Beautiful Taxonomy Filters (Free)

<https://wordpress.org/plugins/beautiful-taxonomy-filters/>

The Beautiful Taxonomy Filters plugin is an easy and good-looking way to provide your visitors with filtering for your post types. With this you get a complete solution for adding filtering based on custom taxonomy terms/categories/tags. It will also automatically add rewrite rules for pretty looking filter URLs.

It's completely automatic, works without javascript and is based on the WordPress Plugin boilerplate for a standardized, organized and object-oriented codebase. It uses select2 for pretty looking and user friendly dropdowns but will fall back to ordinary ones if javascript is not supported.

Design

Visual CSS Editor

Yellow Pencil (Freemium)

<https://wordpress.org/plugins/yellow-pencil-visual-theme-customizer/>

A performant visual css editor. The plugin allows you to customize any page and theme without coding. Click on an element and start visual editing. Adjust colors, fonts, sizes, positions and a lot more.

Import Figma Designs

UIChemmy (Free and Pro)

<https://wordpress.org/plugins/uichemmy/>

Allows you to import 10 designs monthly on the free plan. To use the free plan to import unlimited templates, just make a new temp site on instawp.com and register another account.

Imports and Exports

One Click Demo Import (Free)

<https://wordpress.org/plugins/one-click-demo-import/>

Lightweight One Click Demo Importer

Link Shortening

GreenWPX Shortlink (Free)

<https://wordpress.org/plugins/short-link-by-greenwp/>

Free linkshortener

Link Shortener (Free)

<https://wordpress.org/plugins/link-shortener/>

Extremely basic link shortener.

Linker (Free)

<https://wordpress.org/plugins/linker/>

Through Linker's short url tool you can know on which links your visitors are clicking. Linker is the easiest tool to use in order to create a short link in your own domain & track outbound link clicks from your website, using software like Google Analytics.

Create short links to your post, manage your 301 redirects, track affiliate links and do many other URL related tasks easily.

JotUrl Link Shortener (Free)

<https://wordpress.org/plugins/joturl-link-shortener/>

Ultra-lightweight link shortener. Hasn't been updated recently, make sure to test on staging to check for compatibility issues.

Ads

Advanced Ads Pro (Paid)

<https://wpadvancedads.com/lazy-load-ads/>

Lazy load ads: Only loads ads when the users scroll to them in their browser.

Optimization

Adsense

Lazyload Adsense (Free)

<https://wordpress.org/plugins/lazy-load-adsense/>

Other Ad Providers

For other Ad providers, you can delay their javascript from loading using any of the plugins listed in the Delay Javascript section. These can be identified from a waterfall chart. **Only delay ad JS if all Ads are below the fold! (are not visible on the initial page load as they are offscreen).**

Ad Management

FlatPM (Free)

<https://wordpress.org/plugins/flatpm-wp/>

Free ultra lightweight ad management

CRM

FluentCRM (Free)

<https://wordpress.org/plugins/fluent-crm/>

Lightweight locally hosted CRM plugin.

Live Chat

Help Dialog (Free)

<https://wordpress.org/plugins/help-dialog/>

Completely Free Lightweight Chat Plugin. It includes links for WhatsApp and phone support, FAQs, a search tool, and a contact form, ensuring quick and personalized assistance.

Indigital Live Chat (Free)

<https://wordpress.org/plugins/indigital-web-push-notifications/>

This plugin has 2 in 1 functionality, live chat and push notifications. Self-hosted version is entirely free.

Simple Chat Plugin (Free)

<https://wordpress.org/plugins/simple-chat-button/>

Free ultra lightweight (no javascript!) Whatsapp live chat plugin

Elementor Pro (Paid)

Elementor now has integrated native Live Chat functionality via the new “Floating Buttons” feature which can be enabled in the advanced settings in Elementor Pro.

Pro Elements (Free)

<https://github.com/proelements/proelements>

Pro Elements can be used to enable all Elementor Pro features, including the Floating Buttons for Live Chat.

Support

Support Plugins

Fluent Support (Free)

<https://wordpress.org/plugins/fluent-support/>

Lightweight support plugin

WooCommerce

Dequeue Woocommerce Cart Fragments (Free)

<https://wordpress.org/plugins/disable-cart-fragments/>.

Woocommerce Cart Fragments are disabled and re-enabled on the fly. There are no options to configure.

Disable WooCommerce Bloat (Free)

Debloat for WooCommerce

<https://wordpress.org/plugins/disable-analytics-for-woocommerce/>

Selectively disable all of the listed features below:

- All Admin Features
- Selective Features
- Connection to WooCommerce.com notification
- Marketplace Suggestions
- Extensions submenu
- Marketing Hub
- Remove Styles
- Remove Scripts
- Remove Gutenberg blocks styles on frontend
- Remove Widgets
- Remove Cart Fragmentation

Disable Bloat for WordPress & WooCommerce (Free)

<https://wordpress.org/plugins/disable-dashboard-for-woocommerce/>

Great plugin with a lot of Debloat features for both Woocommerce and Wordpress. Any paid “pro” options can be achieved with other free plugins. **There is no reason to purchase the pro version of the plugin. The pro version is completely unnecessary.**

Enable HPOS (Free)

<https://woo.com/document/high-performance-order-storage/>

High Performance Order Storage is a newer feature that the Woocommerce devs have implemented. It is enabled by default on new Woocommerce installations, however it needs to be manually activated on existing Woocommerce sites. Please ensure all of your plugins are compatible with HPOS before enabling it to prevent any errors. Activate it on staging first before doing so on a live site.

Checkout and Cart Pages

Debloat (Free)

<https://wordpress.org/plugins/debloat/>

Debloat works on the cart and checkout pages, unlike Perfmatters and Flyingpress. Perfmatters and Flyingpress disable their functionality on both of those pages, so use Debloat to remove unused css and delay javascript instead.

Defer Transactional Emails for Woocommerce (Free)

<https://wordpress.org/plugins/defer-transactional-emails-for-woocommerce/>

It will improve checkout process performance deferring transactional emails.

Speed Up Order Lookup

Fast Woocommerce Order Lookup (Free)

<https://wordpress.org/plugins/fast-woo-order-lookup/>

Speeds up order lookup by adding indexes. Speeds up your site by running a background process to create a special-purpose index table, a table of trigrams, to speed up that search. Then it uses those trigrams to search for orders.

Woo Prune Old Orders (Free)

<https://wordpress.org/plugins/woo-prune-orders/>

Greatly improve the performance of a WooCommerce site bogged down by tens of thousands of historic orders. Back orders up using your favorite Order Exports plugin, or rely upon integrated accounting software to keep history beyond the currently active orders in processing.

Test Checkout Process

Woo Order Test (Free)

<https://wordpress.org/plugins/woo-order-test/>

Sets up a testing payment gateway for WooCommerce to confirm your checkout works like it should. You can complete a full and real checkout on your site to see if everything is running smoothly. Runs for admin users only.

Store Toolkit

Woocommerce Store Toolkit (Free)

<https://wordpress.org/plugins/woocommerce-store-toolkit/>

Lightweight multi-function Woocommerce toolkit.

Booster For Woocommerce (Free and Pro)

<https://wordpress.org/plugins/woocommerce-jetpack/>

Multi-function (100+) all-in-one modular Woocommerce toolkit (unneeded functionality can be disabled to minimize performance impact).

Disable Woocommerce Usage tracking (Free)

Usage Tracking

Gathering usage data allows us to make WooCommerce better — your store will be considered as we evaluate new features, judge the quality of an update, or determine if an improvement makes sense.

Enable tracking ☐ Allow usage of WooCommerce to be tracked

To opt out, leave this box unticked. Your store remains untracked, and no data will be collected. Read about what usage data is tracked at: [WooCommerce.com Usage Tracking Documentation](#).

Woocommerce>Settings>Advanced>Woocommerce.com

Woocommerce Database Indexing

Scalability Pro (Paid)

<https://www.superspeedyplugins.com/product/scalability-pro/>

Adds high performance indexes to Wordpress and Woocommerce MySQL tables. Significantly improves database performance.

Product Search

Super Speedy Search (Paid)

<https://www.superspeedyplugins.com/product/super-speedy-search/>

Fast Product Search plugin.

Email Marketing

ShopMagic (Free)

<https://wordpress.org/plugins/shopmagic-for-woocommerce/>

Automated email marketing for Woocommerce, completely free.

FluentCRM (Free and Pro)

<https://wordpress.org/plugins/fluent-crm/>

FluentCRM is a full featured lightweight CRM with an amazing free tier plugin. Fully self-hosted, has Woocommerce complex, granular email marketing features for Woocommerce.

Bulk Edit

Smart Manager (Free and Pro)

<https://wordpress.org/plugins/smart-manager-for-wp-e-commerce/>

Bulk edit multiple products at the same time. Only needed on the backend.

Filters

Ultimate WooCommerce Filters (Free)

<https://wordpress.org/plugins/color-filters/>

Ultimate WooCommerce Filters is a plugin that allows you to filter WooCommerce products by color, size, attribute, category or tag. Easy to set up and implement via a block, shortcode or widget.

Themify Woocommerce Product Filters (Free)

<https://wordpress.org/plugins/themify-wc-product-filter/>

Free Woocommerce Product Filters

Super Speedy Filters (Paid)

<https://www.superspeedyplugins.com/product/super-speedy-filters/>

Ultra fast product filters for Woocommerce.

Note: Free option coming soon

Additional Order Filters For Woocommerce (Free)

<https://wordpress.org/plugins/additional-order-filters-for-woocommerce/>

Additional Order Filters for WooCommerce adds additional filters, with which you can easily and quickly find the orders you need among hundreds of others.

Avalon23 Products Filter (Free)

<https://wordpress.org/plugins/avalon23-products-filter-for-woocommerce/>

Lightweight product filter plugin

Beautiful Taxonomy Filters (Free)

<https://wordpress.org/plugins/beautiful-taxonomy-filters/>

The Beautiful Taxonomy Filters plugin is an easy and good-looking way to provide your visitors with filtering for your post types. With this you get a complete solution for adding filtering based on custom taxonomy terms/categories/tags. It will also automatically add rewrite rules for pretty looking filter URLs.

It's completely automatic, works without javascript and is based on the WordPress Plugin boilerplate for a standardized, organized and object-oriented codebase. It uses select2 for pretty looking and user friendly dropdowns but will fall back to ordinary ones if javascript is not supported.

Product Gallery

Image Gallery and Slider (Free)

<https://wordpress.org/plugins/gallery-slider-for-woocommerce/>

Ultra lightweight image gallery and slider for Woocommerce.

My Account Menu Editing

Gou Manage My Account Menu (Free)

<https://wordpress.org/plugins/gou-wc-account-tabs/>

Free My Account Menu Editor.

Checkout Field Editing

Qodax Checkout Manager (Free)

<https://wordpress.org/plugins/qodax-checkout-manager/>

Ultra lightweight checkout field editor

Cost Of Goods

Cost Of Goods For Woocommerce (Free)

<https://wordpress.org/plugins/cost-of-goods-for-woocommerce/>

Free feature rich cost of goods plugin.

Invoices

PDF Invoices and Packing Slips (Free)

<https://wordpress.org/plugins/woocommerce-pdf-invoices-packing-slips/>

Note: HEAVY plugin, make sure to selectively disable it on the frontend

Order Statuses

Order Status History for WooCommerce (Free)

<https://wordpress.org/plugins/order-status-history-for-woocommerce/>

Order Status History for WooCommerce (OSHWOO) speeds up your daily analysis and processing of orders. Unobtrusive visual cues in all the orders screens show when someone has ordered from your shop before, your (most) repeating customers, and spurts of unusual activity (like customers with due payments and cancellations).

Inventory Management

Shelf Planner (Free)

<https://wordpress.org/plugins/shelf-planner/>

Free HEAVY inventory management system.

Note: Make sure to selectively disable it on the frontend!

Multi-Currency

Woocommerce Currency Switcher (Free and Pro)

<https://wordpress.org/plugins/woocommerce-currency-switcher/>

Note: Free version has all features enabled, but supports only 2 currencies. To have more than 2 currencies pro is needed.

Shipping Distance

WooReer (Free)

<https://wordpress.org/plugins/wcsdm/>

Shipping distance calculator

Order Splitting

Woo Order Splitter (Free)

<https://wordpress.org/plugins/woo-order-splitter/>

It enables you to split, consolidate, clone, your crowd/combined/bulk orders using intelligent rules.

Product Questions

Product Questions and Answers (Free)

<https://wordpress.org/plugins/product-questions-answers-for-woocommerce/>

Product Feed

Woo Product Feed Pro (Free)

<https://wordpress.org/plugins/woo-product-feed-pro/>

Very lightweight product feed with fully enabled pro features with no limits.

Create XML feeds, CSV feeds, TXT feeds, and YML feeds for Google Shopping feed (merchant center), Meta feed, Facebook feed, Remarketing product feed, Bing product feed (for Bing

Shopping & Bing Ads), Billiger.de feed, Pricerunner feed, Skroutz, Catch.com.au feed, and many more.

Upsells

Offermative (Free)

<https://wordpress.org/plugins/offerivative-discount-pricing-related-products-upsell-funnels-for-woocommerce/>

The easiest and most powerful AI-driven WooCommerce upsell plugin to boost sales and maximize your average order value.

Offermative intelligently generates hundreds of conversion-focused upsells, related product recommendations, cross-sells, dynamic pricing deals, order bumps, BOGO deals and other compelling offer campaigns within your sales funnel, within minutes.

Product Stock Sync

Google Sheets Product Sync (Free)

<https://wordpress.org/plugins/product-sync-master-sheet/>

Easily manage and synchronize your WooCommerce product stock with the power of Google Sheets – Product Stock Sync with Google Sheets for WooCommerce.

Markup By Attribute

Markup By Attribute for Woocommerce (Free)

<https://wordpress.org/plugins/markup-by-attribute-for-woocommerce/>

Markup by Attribute Adds 'Markup' to Attribute Terms

Markup by Attribute allows you to add a markup (or markdown) to global attribute terms. If the attribute is 'color', then Markup by Attribute allows you to add "+5" to Blue while leaving Green and Yellow alone. When you set regular and sale prices, every blue product will be \$5.00 more.

Reviews

Google Business Reviews Ratings (Free)

<https://wordpress.org/plugins/g-business-reviews-rating/>

Lightweight Google Reviews plugin.

Absolute Reviews (Free)

<https://wordpress.org/plugins/absolute-reviews/>

Extremely lightweight reviews plugin. Slick designs.

Site Reviews (Free)

<https://wordpress.org/plugins/site-reviews/>

Amazing free, very lightweight open source review plugin. Works for regular site reviews as well as Woocommerce/EDD. It has many more integrations as well, all free.

Common Administration Tasks

Woocommerce Store Toolkit (Free)

<https://wordpress.org/plugins/woocommerce-store-toolkit/>

Store Toolkit for WooCommerce is a growing set of commonly-used WooCommerce administration tools.

Related Products

WebToffee Woocommerce Related Products (Free)

<https://wordpress.org/plugins/wt-woocommerce-related-products/>

Free Related Products plugin

Visual Email Designer

Decorator (Free)

<https://wordpress.org/plugins/decorator-woocommerce-email-customizer/>

Follow-Up Emails

FluentCRM (Free + Paid)

<https://wordpress.org/plugins/fluent-crm/>

CRM and Email Automation Plugin with Woocommerce support.

Shopmagic (Free + Paid Addons)

<https://wordpress.org/plugins/shopmagic-for-woocommerce/>

Giftcards

Gift4U (Free)

<https://wordpress.org/plugins/gift4u-gift-cards-all-in-one-for-woo/>

Free lightweight gift card manager

Product Compare

WCBoost (Free)

<https://wordpress.org/plugins/wcboost-products-compare/>

When dealing with numerous similar products, customers frequently find it difficult to make a purchase decision. As a result, making it simple for buyers to compare similar products is an important aspect of e-commerce websites. This feature is not included with the basic installation of WooCommerce.

Pre-Orders

Product Pre-Orders For Woo (Free)

<https://wordpress.org/plugins/product-pre-orders-for-woo/>

Lightweight pre-order plugin

Variation Swatches

Variation Swatches For Woocommerce (Free)

<https://wordpress.org/plugins/variation-swatches-for-woocommerce/>

Lightweight variation swatch plugin.

Fields

WC Fields Factory (Free)

<https://wordpress.org/plugins/wc-fields-factory/>

Custom Fields manager for Woocommerce.

Export Comments

Woocommerce Comments Export (Free)

<https://wordpress.org/plugins/comments-import-export-woocommerce/>

Wishlist

WCBoost (Free)

<https://wordpress.org/plugins/wcboost-wishlist/>

Free wishlist plugin. Extremely lightweight, less than 80 kb for the entire plugin.

WishSuite (Free)

<https://wordpress.org/plugins/wishsuite/>

Completely Free Feature Complete Wishlist Plugin.

WebToffee Wishlist For Woocommerce (Free)

<https://wordpress.org/plugins/wt-woocommerce-wishlist/>

Free Wishlist Plugin. Medium weight

Sales Notifications/FOMO

Woocommerce Sales Notification (Free)

<https://wordpress.org/plugins/wc-sales-notification/>

WC Sales Notification is a plugin for Showing your orders or fake notification on your website. You can show the notification all pages in your website. It gives you awesome features, is to generate for fake notifications. The plugin Has Wonderful customization option and incoming and outgoing animation options.

Accessories

F12 Woocommerce Accessories (Free)

<https://wordpress.org/plugins/f12-wc-accessories/>

- Add accessories to products.
- Add accessories to categories.
- Show accessories on the product page.
- Automatically show accessories for products added to the cart on the cart page.
- Add custom title & descriptions for the accessories on the product page.
- Add custom title & descriptions for the accessories on the cart page.
- Use placeholders [product_title] to customize the text and improve your seo.
- Set the tags for the titles h1-h6 or use a div container.
- Add mandatory accessories to products
- Add mandatory accessories to categories

Compare Products

Ever Compare (Free)

<https://wordpress.org/plugins/ever-compare/>

Free Lightweight Product Comparison Plugin

Compe Woo Compare Products (Free)

<https://wordpress.org/plugins/compe-woo-compare-products/>

Another lightweight compare plugin

Dynamic Pricing and Discounts

YayPricing (Free)

<https://wordpress.org/plugins/yaypricing/>

Lightweight discounts plugin

Woo Dynamic Pricing and Discounts (Free)

<https://wordpress.org/plugins/redis-woo-dynamic-pricing-and-discounts/>

Lightweight free dynamic pricing and discounts plugin

Dynamic Pricing and Discounts (Free)

<https://wordpress.org/plugins/discount-and-dynamic-pricing/>

Another lightweight dynamic pricing and discounts plugin

Coupon Reminders

Woo Coupon Reminder (Free)

<https://wordpress.org/plugins/woo-coupon-reminder/>

Lightweight automatic coupon email reminders

Coupon Generator

Coupon Zen (Free)

<https://wordpress.org/plugins/coupon-zen/>

Coupon Zen is a WordPress coupon code generator plugin that allows you to generate custom coupon codes for your website. Free and easy to use.

Rewards Points

Woocommerce Points and Rewards (Paid)

<https://woocommerce.com/products/woocommerce-points-and-rewards/>

The official points and rewards plugin, very lightweight.

EPOI – WP Points and Rewards (Free)

<https://wordpress.org/plugins/epoi-wp-points-and-rewards/>

Another lightweight rewards plugin

Product Size Chart

Product Size Chart For Woo (Free)

<https://wordpress.org/plugins/product-size-chart-for-woo/>

Lightweight size charts for Woocommerce products

Suggestions

Suggestion Engine (Free)

<https://wordpress.org/plugins/woo-suggestion-engine/>

Quick View

QuickSwish (Free)

<https://wordpress.org/plugins/quickswish/>

Subscriptions

SUBRE – Product Subscription (Free)

<https://wordpress.org/plugins/subre-product-subscription-for-woo/>

Great lightweight free full featured subscriptions plugin.

Extra Checkout Fields

FEWC Extra Checkout Fields (Free)

<https://wordpress.org/plugins/fewc-extra-checkout-fields-for-woocommerce/>

FEWC – Extra Checkout Fields For WooCommerce is an easy and effective method for customizing the checkout page. You can add custom fields, turn on/off any fields, and modify field position,... And preview the result on customizing.

Automation

AutomateWoo (Paid)

<https://automatewoo.com/>

Official Automattic created WooCommerce automation plugin. I've previously experienced jank with this plugin before, but the last time I used it was a few years ago. The interface is a bit wonky, and like any other automation plugin it will be database heavy. Same deal, selectively disable it on the frontend (it only needs to run in the backend), and make sure you have some form of object caching.

Custom Product Options

EPOW Custom Product Options (Free)

<https://wordpress.org/plugins/epow-custom-product-options-for-woocommerce/>

Lightweight product options customization plugin.

Affiliates

Affiliate Manager (Free)

<https://wordpress.org/plugins/affiliates-manager/>

Lightweight Affiliate Management plugin.

Memberships

Members (Free)

<https://wordpress.org/plugins/members/>

Lightweight Membership plugin.

Memberpress (Paid)

<https://memberpress.com/>

Paid Membership plugin from the Members Developers, more features, very lightweight.

Memberful (Free)

<https://wordpress.org/plugins/memberful-wp/>

Another membership plugin.

Restrict Content

QuentnWP (Free)

<https://wordpress.org/plugins/quentn-wp/>

Free content restriction plugin with Elementor integration.

User Access Manager (Free)

<https://wordpress.org/plugins/user-access-manager/>

Lightweight User Access Manager

Allows you to manage the access of your content. This is useful if you need a member area, a private section at your blog or you want that other people can write at your blog but not everywhere.

Including all post types, taxonomies and files by creating user groups. Just assign the content you want to restrict and your registered users which should have access to a group. From now on the content is only accessible and writable for the specified group.

Anti-Spam - Do not use a Captcha

Anti-Spam: Spam Protection (Free)

<https://wordpress.org/plugins/stop-spammer-registrations-plugin/>

Stop spam emails, spam comments, spam registration, and spam bots and spammers in general. Run diagnostic tests, view activity and more. Lightweight.

Wordpress Zero Spam (Free)

<https://wordpress.org/plugins/zero-spam/>

Captchas introduce additional lag, the above is a far better solution. If zero-spam doesn't work for you, find another captchaless option in the Wordpress plugin directory. **Always** seek to minimize calls to third party websites.

If zero-spam isn't working for you, try to find another non-captcha anti-spam plugin.

Fullworks Anti-Spam (Free)

<https://wordpress.org/plugins/fullworks-anti-spam/>

Fluent Comments (Free)

<https://wordpress.org/plugins/fluent-comments/>

Lightweight Fluent anti-spam plugin for comments.

Anti-Spam Reloaded (Free)

<https://wordpress.org/plugins/anti-spam-reloaded/>

Fluent Forms Anti-Spam (Paid)

<https://wpmanageninja.com/wp-fluent-form/wp-fluent-forms-block-spams/>

Form Plugins

Fluent Forms (Free)

<https://wordpress.org/plugins/fluentform/>

Full featured lightweight form plugin

Fluent Forms Anti-Spam (Paid)

<https://wpmanageninja.com/wp-fluent-form/wp-fluent-forms-block-spams/>

Cforms2 (Free)

<https://wordpress.org/plugins/cforms2/>

Ultra-lightweight forms plugin. It's just 220 KB for the entire plugin, which is very, very light. Flexible and powerful form builder plugin, covering a variety of use cases and features.

Contact Coldform (Free)

<https://wordpress.org/plugins/contact-coldform/>

Extremely lightweight (although barebones) form plugin, very well optimized. It even works without any javascript!

Disqus

Lazy Load Disqus Comments (Free)

<https://wordpress.org/plugins/disqus-conditional-load/>

Push Notifications

Zestpush (Free)

<https://wordpress.org/plugins/zestpush-web-push-notifications/>

Completely free push notifications plugin when self-hosted.

FCM Push Notification (Free)

<https://wordpress.org/plugins/fcm-push-notification-from-wp/>

Entirely free push notification plugin which hooks into Google Firebase (Generous Free plans with a paid tier for over 50k monthly users).

Ultimate Push Notifications (Free)

<https://wordpress.org/plugins/ultimate-push-notifications/>

Integrates with Google Firebase, the plugin is entirely free, and ultra-lightweight (~300 kb for the entire plugin).

Push Notifications For Web (Free)

<https://wordpress.org/plugins/push-notifications-for-web/>

Free Push Notification plugin that integrates with Google Firebase.

Push Notification for Post and BuddyPress(Free)

<https://wordpress.org/plugins/push-notification-for-post-and-buddypress/>

Entirely free push notification plugin that hooks into major notification providers (firebase, onesignal, etc)

WhatsApp Push Notifications

Indigital Web Push Notifications (Free)

<https://wordpress.org/plugins/indigital-web-push-notifications/>

Unlimited free push notifications over WhatsApp.

OneSignal (Free and Pro)

<https://wordpress.org/plugins/onesignal-free-web-push-notifications/>

Visitors opt-in to receive push notifications when you publish a new post. Visitors receive these notifications even after they've left your website, thus driving re-engagement.

You can configure notification delivery at preset intervals, create user segments, and customize the opt-in process for visitors.

OneSignal's free plan allows targeting up to 10,000 subscribers with push notifications.

Avatars

Simple Local Avatars (Free)

<https://wordpress.org/plugins/simple-local-avatars/>

Adds an avatar upload field to user profiles if the current user has media permissions. Generates requested sizes on demand just like Gravatar! Simple and lightweight.

Galleries

Responsive Lightbox (Free)

<https://wordpress.org/plugins/responsive-lightbox/>

Performant Popup and Gallery Plugin

Syntax Highlighter

Prismatic (Free)

<https://wordpress.org/plugins/prismatic/>

The only 3-in-1 syntax highlighter! Display beautiful code snippets with Prism.js, Highlight.js, or plain code escaping.

Bulk Page Generation

Bulk Page Generator (Free)

<https://wordpress.org/plugins/bulk-page-generator/>

Bulk Page Generator can generate multiple pages simultaneously. Big time saver.

Note: I recommend using this on staging or dev and then pushing to production (live). Bulk Page Generation can be resource intensive.

Sitewide Notices

Sitewide Notice WP (Free)

<https://wordpress.org/plugins/sitewide-notice-wp/>

Free lightweight sitewide notice plugin.

Sliders

Slider Pro (Free)

<https://wordpress.org/plugins/sliderpro/>

Lightweight free slider plugin with premium features built in.

Smart Slider 3 (Free and Pro)

<https://wordpress.org/plugins/smart-slider-3/>

Lightweight slider with a bunch of features.

Popups

Popup Maker (Free)

<https://wordpress.org/plugins/popup-maker/>

Lightweight popup plugin

Ultimate Lightbox (Free)

<https://wordpress.org/plugins/ultimate-lightbox/>

Lightweight Popup plugin

FancyBox (Free)

<https://wordpress.org/plugins/fancybox-for-wordpress/>

Seamlessly integrates FancyBox into your blog: Upload, activate, and you're done. Additional configuration optional.

You can easily customize almost anything you can think about fancybox lightbox: the border, margin width and color, zoom speed, animation type, close button position, overlay color and opacity and even more advanced option like several options to group images into galleries.

Food Ordering

RestroPress (Free)

<https://wordpress.org/plugins/restropress/>

Free food ordering system.

Surveys

NgSurvey (Free)

<https://wordpress.org/plugins/ngsurvey/>

Free survey plugin. Disable it on pages it is not being used on.

FAQs

Ultimate FAQ Solution (Free)

<https://wordpress.org/plugins/ultimate-faq-solution/>

Lightweight FAQ Plugin.

Private Messaging

Better Messages (Free)

<https://wordpress.org/plugins/bp-better-messages/>

Better Messages – is realtime private messaging system for WordPress, BuddyPress, BuddyBoss Platform, Ultimate Member, PeepSo and any other WordPress site.

Appointments and Meetings

Ultimate Appointment Scheduling (Free and Pro)

<https://wordpress.org/plugins/ultimate-appointment-scheduling/>

The free version has almost every feature unlocked.

- The premium version has these additional features, which I personally don't find necessary, but here's the list:
 1. Accept mandatory or optional payments for appointments either via PayPal or WooCommerce
- Add reCAPTCHA to your form, to help prevent spam
- Option to require login to WordPress before being able to create an appointment
- Additional booking form layout option
- Labelling options
- Customize the look of your form with an array of styling options to help you fit it in seamlessly with the rest of your site

Learning Management Systems (LMS)

LearnDash (Paid)

<https://www.learndash.com/>

LearnDash is a performant LMS plugin. Like almost every plugin, it comes with unused CSS and JS that must be removed, however it uses a low number of database calls, so once the CSS and JS are optimized, it performs very well.

To optimize LearnDash's CSS and JS, please refer to the Remove Unused CSS, Javascript Delay, Javascript Defer, and Asset Unloading sections.

Team Members

WP Team (Free)

<https://wordpress.org/plugins/ht-team-member/>

Lightweight Free Team Member Plugin

Progress Bars

Progress Bars (Free)

<https://wordpress.org/plugins/progress-bars/>

Ultra Lightweight Free Progress Bars Plugin

Polls

Yop Poll (Free)

<https://wordpress.org/plugins/yop-poll/>

Lightweight basic polls

Democracy Poll (Free)

<https://wordpress.org/plugins/democracy-poll/>

Another lightweight basic polling plugin.

CrowdSignal (Free)

<https://wordpress.org/plugins/polldaddy/>

Lightweight poll plugin made by Automattic, the developers of Wordpress.

CP Polls (Free)

<https://wordpress.org/plugins/cp-polls/>

Lightweight Polls Plugin with advanced features. Tons and tons of features for a free poll plugin. It's got more features than any of the free plugins listed so far.

The fact that their screenshots look like the plugin jumped straight out of 2002 is not encouraging, but give it a try to see if it fits your needs. Can likely customize the polls to look modern.

Quizzes

Involve Me (Free)

<https://wordpress.org/plugins/involve-me/>

CodeVery Quiz (Free)

<https://wordpress.org/plugins/codevery-quiz/>

Free lightweight plugin to make beautiful quizzes

Interactive Content

H5P (Free)

<https://wordpress.org/plugins/h5p/>

<https://h5p.org/content-types-and-applications>

Free Open Source plugin that can be used to create any interactive content. Quizzes, Games, Polls, Sliders, Image Galleries, Flash Cards, Collages, Charts, Accordions, Timelines, Summaries, Interactive Videos, Presentations and more.

Charts

M-Chart (Free)

<https://wordpress.org/plugins/m-chart/>

Free Lightweight Chart plugin

Easy Charts (Free)

<https://wordpress.org/plugins/easy-charts/>

Free Chart Plugin.

WP Charts and Graphs (Free)

<https://wordpress.org/plugins/wp-charts-and-graphs/>

Free Simple Chart Plugin

SQL Chart Builder (Free)

<https://wordpress.org/plugins/sql-chart-builder/>

Build Charts from SQL Queries

Accordion

Lightweight Accordion (Free)

<https://wordpress.org/plugins/lightweight-accordion/>

Ultra-lightweight accordion plugin. The entire plugin is 10 KB, and uses no javascript.

Disclaimers

Disclaimify (Free)

<https://wordpress.org/plugins/disclaimify/>

Directories

Simple Directory Listing (Free)

<https://wordpress.org/plugins/sirve/>

Simple Directory Listing plugin

Recipes

RecipePress Reloaded (Free)

<https://wordpress.org/plugins/recipepress-reloaded/>

RecipePress Reloaded is a lightweight, powerful and very flexible tool to manage your blog's recipe collection. It was designed with the editor in mind and created for easy and fast usage.

Mediavine Create (Free)

<https://wordpress.org/plugins/mediavine-create/>

Slightly heavier free Recipe plugin. Great templates.

Blossom Recipe Maker (Free)

<https://wordpress.org/plugins/blossom-recipe-maker/>

Free Recipe plugin

Text Messaging

Texteller (Free)

<https://wordpress.org/plugins/texteller/>

Texteller is an all-in-one text message integration solution for WordPress, supporting multiple third-party SMS and messaging gateways.

Note: Hasn't been updated in a while, check compatibility/functionality on a staging site.

Patreon

Patreon Connect (Free)

<https://wordpress.org/plugins/patreon-connect/>

The Lightweight Official Patreon Plugin.

Lightweight Youtube Player

Faster Youtube Embed (Free)

<https://wordpress.org/plugins/faster-youtube-embed/>

Faster YouTube embed is a powerful solution for embedding YouTube videos into your WordPress site that is exceptionally quick, straightforward, and easy to use.

WP Youtube Lyte (Free)

<https://wordpress.org/plugins/wp-youtube-lyte/>

WP YouTube Lyte allows you to lazy load your video's, by inserting responsive "Lite YouTube Embeds". These look and feel like normal embedded YouTube, but only call the "fat" YouTube-player when clicked on, thereby reducing download size & rendering time substantially

Real Estate

Essential Real Estate (Free)

<https://wordpress.org/plugins/essential-real-estate/>

There are full features needed for a real estate website, easy to customize and overwrite templates. Allow to manage property listings and submit property via back-end and front-end. Supported global payment, build market, agent management,...etc.
FREE plugins but already all PREMIUM FEATURES.

Addons (Free)

<https://plugins.g5plus.net/ere/add-ons/>

Essential Real Estate Free addons.

Real Estate Manager (Free)

<https://wordpress.org/plugins/real-estate-manager>

Instagram Feed

HT Instagram Feed (Free)

<https://wordpress.org/plugins/ht-instagram/>

Free lightweight Instagram feed.

Google Social Login

Login With Google (Free)

<https://wordpress.org/plugins/login-with-google/>

Lightweight Google Social Login plugin.

Podcasting

Seriously Simple Podcasting (Free)

<https://wordpress.org/plugins/seriously-simple-podcasting/>

Relatively lightweight podcasting plugin.

Powerpress (Free)

<https://wordpress.org/plugins/powerpress/>

Full featured podcasting plugin. (**Warning: Heavy, especially crucial to delay its javascript**).

PowerPress enables you to publish and manage one or multiple podcasts directly from your WordPress website. Strengthen your brand and reach audiences across all major platforms, such as Apple Podcasts, Spotify, YouTube Music, Amazon Music, and more.

Comment Subscribing

Subscribe To Comments Reloaded (Free)

<https://wordpress.org/plugins/subscribe-to-comments-reloaded/>

Subscribe to Comments Reloaded is a robust plugin that enables commenters to sign up for e-mail notification of subsequent entries. The plugin includes a full-featured subscription manager that your commenters can use to unsubscribe to certain posts or suspend all notifications.

Allows administrators to enable a double opt-in mechanism, requiring users to confirm their subscription clicking on a link they will receive via email or even One Click Unsubscribe.

Buy Me A Coffee Button

Buy Me A Coffee (Free)

<https://wordpress.org/plugins/buymeacoffee/>

Lightweight Buy Me A Coffee Plugin.

Google Maps

Google Maps Lite (Freemium)

<https://wordpress.org/plugins/advanced-google-maps-lite/>

Lightweight google maps plugin.

CBX Google Map (Freemium)

<https://wordpress.org/plugins/cbxgooglemap/>

Another lightweight Google maps plugin.

Manually optimize Google maps (Free)

The best way to optimize Google maps is by taking a screenshot of your map location. You can compress the image as much as possible, which will prevent any calls to google's third party domain and means your site will load 0 additional javascript to display the map.

Social

Social Sharing

HT Social Share (Free)

<https://wordpress.org/plugins/ht-social-share/>

Lightweight Completely Free Social Sharing Plugin

Hubbub Lite (Freemium)

<https://wordpress.org/plugins/social-pug/>

Very lightweight social sharing plugin.

Shared Counts (Free)

<https://wordpress.org/plugins/shared-counts/>

Shared Counts is a WordPress social media share buttons plugin. Sharing buttons increase traffic and engagement by helping readers share your posts and pages to their friends on social media.

AddToAny Share Buttons (Free)

The AddToAny Share Buttons plugin for WordPress increases traffic & engagement by helping people share your posts and pages to any service. Services include Facebook, Mastodon, Pinterest, WhatsApp, LinkedIn, Threads, Bluesky, Tumblr, Reddit, X, WeChat, and many more sharing and social media sites & apps.

Social Autoposting

WP Scheduled Posts (Free)

<https://wordpress.org/plugins/wp-scheduled-posts/>

Lightweight Social Auto Poster plugin.

Note: Make sure you selectively disable it in the frontend with any of the plugins for that functionality listed below(Asset Cleanup Pro, Perfmatters Script Manager, Freesoul plugin deactivator, etc.). It only needs to run in the backend.

Gamification

Gamipress (Freemium)

<https://gamipress.com/>

Lightweight Gamification plugin

Tables

Ninja Tables (Freemium)

<https://wordpress.org/plugins/ninja-tables/>

Lightweight Table Plugin.

Reviews

Site Reviews (Free)

<https://wordpress.org/plugins/site-reviews/>

Amazing free, lightweight open source review plugin. Works for regular site reviews as well as Woocommerce/EDD. It has many more integrations as well, all free.

Donations

WC Donation Platform (Free)

<https://wordpress.org/plugins/wc-donation-platform/>

Awesome, completely free open source donation plugin. Extremely well optimized(the entire plugin is 230 kb), great feature set.

Charts

Ninja Charts (Freemium)

<https://wordpress.org/plugins/ninja-charts/>

Performant Charts plugin

Cookie Notices

Cookie Dough (Free)

<https://wordpress.org/plugins/cookie-dough-compliance-and-consent-for-gdpr/>

Ultra lightweight Cookie Notice GDPR compliance banner. Completely free.

WP Full Picture (Free and Pro)

<https://wordpress.org/plugins/full-picture-analytics-cookie-notice/>

The free version has practically every feature unlocked. Local Analytics and Cookie Consent, fully featured. Very lightweight. The free version even has Woocommerce tracking.

Smart Cookie Kit (Free)

<https://wordpress.org/plugins/smart-cookie-kit/>

Neat little plugin with decent configurability, it can block all scripts until the user consents to cookies. Lightest weight cookie plugin out there, the entire plugin is less than 100 kb.

Cookie Fox (Free)

<https://wordpress.org/plugins/cookiefox/>

Completely free light weight cookie banner. Light on features, but useful.

Iubenda (Free and Pro)

<https://wordpress.org/plugins/iubenda-cookie-law-solution/>

Very lightweight cookie compliance plugin, tons and tons of features, free version is solid but has a monthly traffic limit. This is the best option.

<https://www.iubenda.com/en/help/379-basic-license-limits>

Lightweight Cookie Consent box (Free)

<https://wordpress.org/plugins/cookie-consent-box/>

Lightweight Cookie Notice by DAEXT

<https://codecanyon.net/item/lightweight-cookie-notice/30970799>

Free version

<https://wordpress.org/plugins/lightweight-cookie-notice-free/>

This is the lightest-weight cookie notice plugin I've found, it adds only 8 KB to your page weight.

Cookie Notice and Compliance (Free)

<https://wordpress.org/plugins/cookie-notice/>

Another lightweight cookie notice plugin.

Machete (Free)

<https://wordpress.org/plugins/machete/>

Machete has a good amount of features, one of them being a performance optimized cookie notice.

Server Level Optimizations

Apache (Free)

Htaccess (Free)

Perishable Press Article

<https://perishablepress.com/stupid-htaccess-tricks/>

Htaccess Optimizations

<https://gist.github.com/seoagentur-hamburg/c96bc796764baaa64d43b70731013f8a>

<https://awesomeopensource.com/project/phanan/htaccess>

More htaccess security headers

<https://github.com/drego85/httpw>

Apache MPM Settings (Free)

Prefork is the default. You can also test the Worker MPM or Event MPM with your site, theoretically they should offer performance benefits. I haven't tested either extensively. The event MPM should be the fastest of the 3 if your Wordpress install works with it.

You want to increase these values in your httpd.conf configuration file:

```
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# ServerLimit: maximum value for MaxClients for the lifetime of the server
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
```

```
<IfModule prefork.c>
StartServers    xxxx
MinSpareServers xxxx
MaxSpareServers xxxx
ServerLimit     xxxx
MaxClients      xxxx
MaxRequestsPerChild xxxxx
</IfModule>
```

```
# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule worker.c>
StartServers    xxxx
MaxClients      xxxx
MinSpareThreads xxxx
MaxSpareThreads xxxx
ThreadsPerChild xxxx
MaxRequestsPerChild 0
</IfModule>
```

The values you set here are heavily dependent on the amount of RAM allotted to your server. My server had 12 GB of RAM, so I was able to set very large values here. Tweak the values incrementally to increasingly large values up to a max of 1024(where the xxxx are), again, depending on the amount of available RAM. These values are probably too high, but in my testing 1024 had the best performance.

Do this for everything except MaxRequestsPerChild under prefork MPM(leave the worker MPM one at 0) until you either max your server resources, or hit 1024.

You should look at the graphs for your RAM and CPU utilization in your hosting provider's backend panel. Make sure you aren't going over whatever your host deems acceptable, and definitely make sure it isn't hitting 100%. The max you would want to go on either of them would likely be 80%, but aim for 60% max utilization. You want to leave room for spikes due to heavy load from a large influx of simultaneous users. If your site becomes popular all of a sudden because of a link from a big blog, or a social media influencer or some other traffic source, the usage *may* spike.

Apache tuning documentation

<https://httpd.apache.org/docs/2.4/misc/perf-tuning.html>

Awesome Open Source Apache Server Configs

<https://awesomeopensource.com/project/h5bp/server-configs-apache>

Hostens Article

<https://www.hostens.com/knowledgebase/optimize-apache-server/>

Host advice article

<https://hostadvice.com/how-to/web-hosting/apache/how-to-tune-and-optimize-performance-of-apache/>

Bob Cares Apache optimization article

<https://bobcares.com/blog/speed-up-apache/>

Tecmint Apache Performance Tuning Article

<https://www.tecmint.com/apache-performance-tuning/>

Quickhost Apache Tuning Article

<https://www.quickhost.uk/blog/2021/09/22/how-to-tune-apache-web-server-for-performance/>

8G Firewall (Free)

<https://perishablepress.com/8g-firewall/>

Lightweight highly configurable firewall

Apache2Buddy (Free)

<https://github.com/richardforth/apache2buddy>

What is Apache2Buddy?

Use the Apache2Buddy script to help make recommendations for changes to the Apache® web server configuration that assist with performance and stability. Apache2Buddy has no impact on the server itself and makes no changes. After the script runs, it suggests Apache fixes that you can examine and consider applying.

Apache2Buddy checks at least the following areas of concern:

- PHP® memory limits
- Memcache
- MaxClients
- Memory usage
- Max potential memory usage
- Percentage of total RAM allocated to Apache
- MaxRequestWorkers

How to tune Apache in seconds with apache2buddy.pl

This is a fork of Apache2Buddy.

<https://www.techrepublic.com/article/how-to-tune-apache-in-seconds-with-apache2buddy-pl/>

Other Apache Resources (Free)

Apache Load Testing

<https://awesomeopensource.com/project/alexfernandez/loadtest>

Apache Server Hardening

<https://www.tecmint.com/apache-security-tips/>

LibModSecurity

<https://kifarunix.com/configure-libmodsecurity-with-apache-on-centos-8/>

Htaccess Editor (Free)

<https://wordpress.org/plugins/wp-htaccess-editor/>

Edit the htaccess file directly within the WP Admin Dashboard.

Nginx (Free)

<https://github.com/fcambus/nginx-resources>

Nginx Configuration files

<https://github.com/pothi/wordpress-nginx/>

Awesome NGINX

<https://github.com/agile6v/awesome-nginx>

Awesome NGINX Security

<https://github.com/wallarm/awesome-nginx-security>

WP Rocket NGINX configuration

<https://github.com/SatelliteWP/rocket-nginx>

Premade configuration for NGINX by the WP Rocket team

Mod Pagespeed Module

https://www.tecmint.com/install-mod_pagespeed-to-boost-apache-nginx-performance/#

Nginx Tuning Repo

<https://github.com/denji/nginx-tuning>

Test on a staging environment!

Cache Control (Free)

<https://wordpress.org/plugins/cache-control-by-cacholong/>

Advanced control NGINX's FastCGI Cache. Also allows scheduled cache purging via Cronjobs.

TCP BBR

Medium

<https://atoonk.medium.com/tcp-bbr-exploring-tcp-congestion-control-84c9c11dc3a9>

BBR Quick Start

<https://github.com/google/bbr/blob/master/Documentation/bbr-quick-start.md>

Cloudflare + NGINX

<https://blog.cloudflare.com/http-2-prioritization-with-nginx/>

Nginx Configuration Generators

Nginxconfig.io

<https://github.com/digitalocean/nginxconfig.io>

Crossplane

<https://github.com/nginxinc/crossplane>

Reliable and fast NGINX configuration file parser and builder

Nginx Configuration Formatters

Nginx Config Formatter Go

<https://github.com/rwx-----/nginxConfigFormatterGo>

Nginx Config Formatter

<https://github.com/1connect/nginx-config-formatter>

Nginx Minify Conf

<https://github.com/lovette/nginx-tools/tree/master/nginx-minify-conf>

Static Analyzers

Gixy

<https://github.com/yandex/gixy>

Syntax Highlighting

SSLSecure.vim

<https://github.com/chr4/sslsecure.vim>

NGINX.vim

<https://github.com/chr4/nginx.vim>

Vim

<https://github.com/nginx/nginx/tree/master/contrib/vim>

NGINX vhost Traffic Status Module

<https://github.com/vozlt/nginx-module-vts>

Nginx Virtual host traffic status module. Extensive NGINX vhost information, including cache hit rate.

Nginx HTTP2 Server Push

<https://www.nginx.com/blog/nginx-1-13-9-http2-server-push/>

CPU Pinning and Numa Optimization

Adjusting NGINX's Worker Processes

The number of Nginx worker processes should ideally match the number of CPU cores available on your server. This ensures that each worker process can run on a separate CPU core, optimizing CPU usage.

You can set the `worker_processes` directive in Nginx's configuration file (`nginx.conf`) to the number of CPU cores. For example, if your server has 4 CPU cores, you can set `worker_processes 4`.

It's important to monitor the performance after making these changes and adjust if necessary. Sometimes setting worker processes to a higher number than the CPU cores may yield better performance, especially for serving static content.

Managing Worker Connections

The `worker_connections` setting defines the maximum number of connections each worker can handle. It's located in the events block of the `nginx.conf` file.

Increasing `worker_connections` allows each worker to handle more simultaneous connections, which can improve throughput. A common initial value to try is 1024.

As with worker processes, it's essential to monitor server performance after making changes to `worker_connections` and adjust as needed.

Utilizing Keepalive Connections

Keepalive connections can be enabled in `nginx.conf` to maintain persistent connections, reducing CPU and network overhead associated with opening and closing connections frequently.

Optimizing for NUMA Architecture

If your server uses a NUMA architecture, ensuring that Nginx processes are using local memory can reduce memory access latency.

Tools like “numactl” can be used to run Nginx with memory allocated from specific NUMA nodes, though this is a more advanced optimization that requires a good understanding of your server’s NUMA configuration.

Linode’s Configuration suggestions

<https://www.linode.com/docs/guides/getting-started-with-nginx-part-2-advanced-configuration/>

Nginx SSL Connection Speed Optimization

Nginx Official Blog Article

<https://www.nginx.com/blog/improve-seo-https-nginx/>

RT Camp Article

<https://rtcamp.com/tutorials/nginx/ssl-pci-compliance-performance/>

Changing Nginx Ciphers for better performance

Nginx’s default set of ciphers includes:

ssl_ciphers HIGH:!aNULL:!MD5;

This enables the strong “Diffie–Hellman_key_exchange algorithm” which is slow. Since PCI compliance doesn’t need this algorithm, we can safely disable it by explicitly defining ssl_ciphers.

ssl_ciphers HIGH:!aNULL:!MD5:!kEDH;

Turbocharging Nginx SSL TTFB/Latency

<https://linuxblog.io/nginx-tuning-tips-tls-ssl-https-ttfb-latency/>

7G Firewall for Nginx

<https://perishablepress.com/7g-firewall-nginx/>

Note: Jeff Starr is working on an update for the updated 8G firewall which will be compatible with NGINX(currently only compatible with Apache), however there is no estimated release time for it yet.

A/B Testing with Nginx

<https://www.nginx.com/blog/performing-a-b-testing-nginx-plus/>

HTTP/3

Enable HTTP/3 for NGINX

<https://codedamn.com/news/backend/leveraging-http3-with-nginx>

Short walkthrough on how to enable HTTP/3 for all NGINX installs.

WordOPs (Free)

HTTP/3 is now available in WordOPs. Built in and auto-configured when you set up Wordpress via WordOPs.

OpenLiteSpeed (Free)

Linux Config Article

<https://linuxconfig.org/openlitespeed-wordpress-installation>

Big Dinosaur Article

<https://blog.bigdinosaur.org/configuring-wordpress-openlitespeed/>

How to Configure OpenLiteSpeed for Maximum WordPress Performance

<https://www.uncleb.tech/how-to-configure-openlitespeed-for-maximum-wordpress-performance/>

Webserver Control Panel Optimization

Medium Article

<https://medium.com/@g24.sumit/litespeed-web-server-control-panel-optimizing-website-performance-with-speed-and-security-4daf69a9ec6>

Optimize the OpenLiteSpeed Webadmin GUI

<https://azdigi.com/blog/en/linux-server-en/web-server/how-to-optimize-openlitespeed-webadmin-gui/>

Advanced Configuration

<https://openlitespeed.org/kb/category/configuration/advanced-configuration/>

PHP suEXEC Max Conn

<https://docs.litespeedtech.com/lsws/cp/cpanel/tunings/>

PHP suEXEC Max Conn specifies the maximum number of concurrent PHP processes that can be created by LSWs for each user when running PHP scripts in suEXEC mode. The default setting is 10.

Generally this should be a good enough limit for the shared hosting environment, since you may not want too many processes running for each account on a server with many accounts. However, if you do have a need to allow more PHP processes for each account, you can adjust the value from 10 to a higher number, such as 40, or 100

OpenLiteSpeed one click auto-configuration script for Wordpress

<https://github.com/litespeedtech/ols1clk>

OpenLiteSpeed tuning for large servers/high traffic sites

<https://docs.litespeedtech.com/lsws/tuning-dedicated/>

Pre-optimized OpenLiteSpeed Docker Images with a focus on PHP performance

<https://github.com/sitepilot/docker-lsphp>

Docker LSPHP (LSAPI + PHP) is a collection of optimized and extensible container images for running PHP applications in production.

All images are based on their Docker Runtime image, an optimized and extensible Ubuntu container image.

Ansible Playbooks for a pre-optimized OpenLiteSpeed stack for Wordpress

<https://github.com/sitepilot/stack-openlitespeed>

Ansible playbooks for provisioning optimized web servers for WordPress and Laravel. These playbooks are used by Autopilot (our cloud server control panel) and are perfect for:

Local development environments.

High-performance production servers (with caching).

HTTP/2 and HTTP/3

HTTP/2 and HTTP/3 work at the **server level** (as well as using a helper plugin) and the functionality must be enabled in your **web server** in order to take advantage of it.

HTTP/2 and HTTP/3 Explained

HTTP/2 Server Push allows for additional assets to be directly pushed to a visitor's browser without the need for subsequent HTTP requests to fetch them. Helping to even further reduce load times and help sites load more efficiently.

Github Article

<https://alexandrehtb.github.io/posts/2024/03/http2-and-http3-explained/>

Thorough technical explanation of HTTP/2 and HTTP/3

SensePost Article

<https://sensepost.com/blog/2021/adventures-into-http2-and-http3/>

Another thorough technical breakdown with additional information

How To Enable HTTP/2

Apache and NGINX

<https://www.linkedin.com/pulse/http2-apache-nginx-detailed-guide-samuel-akopyan/>

Short Article on how to enable HTTP/2 for Apache and NGINX.

OpenLiteSpeed

https://www.litespeedtech.com/support/wiki/doku.php/litespeed_wiki:config:enable-http2

HTTP/2 Server Push Plugin

<https://wordpress.org/plugins/http2-push-content/>

- Push / Pre-load all JS files in site with one simple option
- Push / Pre-load all the CSS files in your website
- Push / Pre-load other resources throughout the site or based on the page types

HTTP/3

Enable HTTP/3 for Apache

Apache currently has no native HTTP/3 module and therefore no native support. Apache can still take advantage of HTTP/3 if NGINX is configured as a reverse proxy once HTTP/3 support is enabled in NGINX.

Note: Apache can be paired with the HAProxy load balancer to enable HTTP/3 Support on Apache servers (HAProxy has native HTTP/3 support).

Enable HTTP/3 for NGINX

<https://codedamn.com/news/backend/leveraging-http3-with-nginx>

Short walkthrough on how to enable HTTP/3 for NGINX.

Enable HTTP/3 for OpenLiteSpeed

<https://azdigi.com/blog/en/linux-server-en/web-server/enable-quick-http-3-on-openlitespeed/>

Article on how to enable HTTP/3 for OpenLiteSpeed.

HTTP/2 and HTTP/3 Server Push (Free)

<https://wordpress.org/plugins/better-resource-hints/>

Free Server Push Plugin

SSL Performance Optimization (Free)

There are several areas to improve SSL connection (and general SSL traffic latency). Some of these may impact security in some small or great way.

(These are for ssl.conf using Apache)

Disable the Domain lookup, and allow Non-SNI to reach primary domain

- SSLStrictSNIVHostCheck off

Instead of named hosts, use IP address for the :443 VHOST entry

Cache SSL Sessions

- SSLSessionCache shmcb:/run/httpd/sslcache(512000)
- SSLSessionCacheTimeout 300

Allow for faster SSL Protocol order, general compatibility, medium security:

- SSLProtocol all -SSLv3
- SSLCipherSuite ALL:+HIGH:+TLSv1:!ADH:!EXP:!SSLv2:!MEDIUM:!LOW:!NULL:!aNULL
- SSLHonorCipherOrder on

OCSP Stapling (speeds up SSL)

- SSLUseStapling on
- SSLStaplingResponderTimeout 5
- SSLStaplingReturnResponderErrors off
- SSLStaplingCache shmcb:/run/ocsp(128000)

HSTS (can do preload so http:// requests become https:// requests in the browser)

- Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains; preload"

Consider the following more advanced implementations:

- HTTP/2 with HPACK
- Brotli Compression

Other non-SSL issues that may help speed up connection

Another directive for Apache in httpd.conf

- HostnameLookups Off

HAProxy SSL performance tweaks

<https://www.suse.com/c/speeding-ssl-need-know-haproxy/>

Additional SSL Performance Optimizations

<https://www.linkedin.com/advice/0/what-best-ways-optimize-ssl-network-performance-1xfae#:~:text=To%20optimize%20SSL%20network%20performance%2C%20use%20hardware%20acceleration%2C%20employ%20session,SSL%2FTLS%20termination%20points%20strategically.>

1. Choose the right cipher suite
2. Enable session resumption
3. Tune TCP parameters

SuperUser Thread

<https://superuser.com/questions/1066477/how-to-speed-up-ssl-connection-establishment>

Nginx SSL Connection Speed Optimization

Nginx Official Blog Article

<https://www.nginx.com/blog/improve-seo-https-nginx/>

Turbocharging Nginx SSL TTFB/Latency

<https://linuxblog.io/nginx-tuning-tips-tls-ssl-https-ttfb-latency/>

Stack Overflow Thread

<https://stackoverflow.com/questions/36672261/how-to-reduce-ssl-time-of-website>

1. SSL caching should be on
2. HTTP Keep-Alives should be turned on so the connection is not dropped after each HTTP Request
3. Use the most recent TLS version
4. OCSP Stapling saves network trip to check cert is valid, using OCSP or CRL
5. Certificate Chain. You'll want a short certificate chain (ideal 3 cert long: your server, intermediary and the CAs root certificate).

6. Ciphers used. There are older and slower ciphers and faster and new ones. Can get complicated here really quickly but generally you should be looking for ECDHE ciphers for most clients (and preferable ECDHE...GCM ones) and want to specify that server order should be used so you get to pick the cipher used rather than the client.

Reduce TLS Handshake Time

Raaz Kumar Article

<https://raazkumar.com/tutorials/nginx/improve-ssl-performance/>

Baeldung's SSL Overhead Optimization Tutorial

<https://www.baeldung.com/cs/ssl-overhead-causes-countermeasures>

HSTS SSL Preloading

<https://hstspreload.org/>

CAUTION: If your site does not have a valid SSL certificate after you add your site to the preload list, it MUST have a valid SSL certificate to remain accessible. A site will be unreachable/inaccessible if the site does not have a valid SSL certificate.

Submission Requirements

If a site sends the preload directive in an HSTS header, it is considered to be requesting inclusion in the preload list and may be submitted via the form on this site.

In order to be accepted to the HSTS preload list through this form, your site must satisfy the following set of requirements:

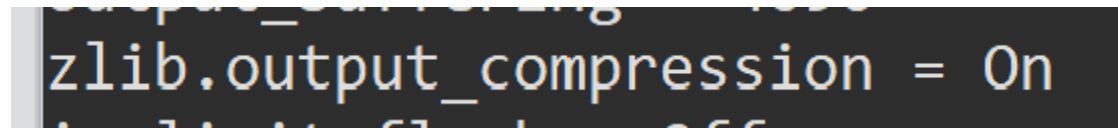
- Serve a valid certificate.
- Redirect from HTTP to HTTPS on the same host, if you are listening on port 80.
- Serve all subdomains over HTTPS.
- In particular, you must support HTTPS for the www subdomain if a DNS record for that subdomain exists.
- Note: HSTS preloading applies to all subdomains, including internal subdomains that are not publicly accessible.
- Serve an HSTS header on the base domain for HTTPS requests:
 - The max-age must be at least 31536000 seconds (1 year).
 - The includeSubDomains directive must be specified.
 - The preload directive must be specified.
- If you are serving an additional redirect from your HTTPS site, that redirect must still have the HSTS header (rather than the page it redirects to).

PHP Settings (Free)

If PHP 8 doesn't have incompatibilities with the plugins you use, you should use 8-8.3. Always use the most updated version of PHP that you can. Use the newest version of MariaDB or MySQL as well.

PHP execution time should be increased to 300 seconds, input time increased to 300 seconds, input variables increased to 20000, memory limit set to 256 MB-768 MB of RAM. It is not advisable to allocate the entire portion of RAM allotted to your VPS, as doing so can cause performance issues/crash the server.

Enable Zlib output compression in PHP.ini



```
zlib.output_compression = On
```

PHP Opcache

Increase OPcache memory to at least 512 MB, preferably 1024 or 2048 if your server has more ram. The max accelerated files limit should be increased as much as your server resources will allow, the cap is 100000.

Moving PHP session data to /dev/shm for improved performance

Zip up your WordPress site with a move of the PHP session data to /dev/shm! This will give you faster page loading speeds and great user experiences. Here's how:

1. Find the php.ini file in the /etc directory.
2. Edit the file with a text editor.
3. Look for the line starting with "session.save_path".
4. Replace the current value with "/dev/shm".

“Performance Improvement: By moving PHP session data to /dev/shm, you are essentially storing this data in RAM. This can significantly speed up the read and write operations for session data, which can be beneficial for WordPress sites with high traffic and heavy session usage.

Considerations and Trade-offs:

RAM Usage

Since `/dev/shm` uses RAM, it's important to ensure that your server has enough memory. Storing session data in RAM will consume more memory, which could be an issue if your server is already memory-constrained.

Persistence

Data stored in `/dev/shm` is temporary and will be lost on reboot. This is generally fine for session data, which is temporary by nature, but it's something to keep in mind.

Scalability

If your WordPress site is served from multiple servers in a load-balanced environment, this approach may not be suitable unless you have a way to share session data across servers.

Security

Ensure that your `/dev/shm` partition is secured appropriately, as storing session data in a shared memory space can have security implications.

Extending PHP's lifespan by increasing the "Max Idle Time" setting

Locate the PHP config file(`php.ini`). Edit the file and modify the `"maxIdleTime"` parameter. Increase its value to give PHP longer idle time. After updating the `"maxIdleTime"` value, save changes and apply them to the server.

Test Performance

Monitor the website's performance after extending PHP's lifespan. Check for better response times and site performance.

Improve front end performance with just one line of PHP

<https://www.spacedmonkey.com/2023/06/29/improve-front-end-performance-with-just-one-line-of-php/>

StashPHP Cache

<https://www.stashphp.com/>

WP Stash (Free)

<https://github.com/inpsyde/WP-Stash>

<https://inpsyde.com/en/wordpress-caching-system-wpstash/>

OPCache Manager (Free)

<https://wordpress.org/plugins/opcache-manager/>

OPcache statistics and management right in the WordPress admin dashboard.

Create Multiple Asynchronous PHP Processes To Speed Up Processing Time

Torque Mag Article

<https://torquemag.io/2016/01/use-asynchronous-php-wordpress/>

MySQL performance optimizations (Free)

Reduce Auto-loaded Data

<https://kinsta.com/knowledgebase/wp-options-autoloaded-data/>

Most plugins will set many of their database options to autoload at all times, which hurts your database performance, which also uses more server resources. Reducing the auto-loaded data as much as possible without harming any functionality will improve your database performance.

Please test disabling any database options on a staging server, this could potentially break your site when optimizing and testing. Always test on a staging server when modifying autoload database options.

Advanced Database Cleaner (Free and Pro)

<https://wordpress.org/plugins/advanced-database-cleaner/>

Many database optimization plugins can modify auto-load options. I use the Advanced Database Cleaner plugin, but there are many options available.

Database Cleaner (Free)

<https://wordpress.org/plugins/database-cleaner/>

Another free option for managing database options autoload status.

Automatic Database Tuning

MySQL Tuner (Free)

<https://github.com/major/MySQLTuner-perl>

MySQLTuner is a script written in Perl that allows you to review and benchmark a MySQL installation quickly and make adjustments to increase performance and stability. The current configuration variables and status data are retrieved and presented in a brief format along with some basic performance suggestions.

Releem (Paid)

<https://releem.com/>

Releem is a paid MySQL optimization service that dynamically adjusts your MySQL configuration to match your server resources and sets the values appropriately. It actively measures Database load and adjusts settings accordingly. MySQL Tuner does not actively monitor and tune the configuration based on Database load dynamically as Releem does.

Enable MySQL Query Cache

<https://mariadb.com/kb/en/query-cache/>

<https://www.freecodecamp.org/news/the-ultimate-guide-to-high-performance-wordpress/>

(Ctrl+f for query cache on the free code camp page)

Set in memory table size limits

Set In-memory Table Size Limits: complex queries will cause the SQL server to create temporary tables. The SQL server will attempt to create tables in memory up to a certain size for performance. It is important to set the max size to an appropriate value to avoid creating on-disk temporary tables.

The variables that control this behavior are `tmp_table_size` and `max_heap_table_size`. You can compare the number of internal on-disk temporary tables created to the total number of internal

temporary tables created by comparing `Created_tmp_disk_tables` and `Created_tmp_tables` values.

Set the buffer pool size

<https://www.freecodecamp.org/news/the-ultimate-guide-to-high-performance-wordpress/>

“The buffer pool is the area of memory that MariaDB uses to cache data and indexes. It is important to configure the buffer pool size appropriately based on the amount of available memory and the size of the database. A good rule of thumb is to set the buffer pool size to about 40% of available memory, but keep in mind this percentage is typically applicable only on stand-alone database servers. The buffer pool size can be controlled using the `innodb_buffer_pool_size` variable. See <https://mariadb.com/kb/en/innodb-buffer-pool/> for more information.

One way to analyze buffer pool performance is by observing the `innodb_buffer_pool_wait_free` status variable. If it is increasing, then you don't have enough buffer pool (or your flushing isn't occurring frequently enough). In this case, you should set the `innodb_buffer_pool_size` variable higher if your system resources allow for it.”

Check these links for more information:

https://mariadb.com/kb/en/server-system-variables/#tmp_table_size

https://mariadb.com/kb/en/server-system-variables/#max_heap_table_size

Manual optimization and learning resources

<https://www.percona.com/blog/mysql-101-parameters-to-tune-for-mysql-performance/>

<https://dev.mysql.com/doc/refman/8.0/en/optimization.html>

<https://www.turing.com/kb/best-practices-for-mysql-performance-tuning>

<https://www.infoworld.com/article/3675552/10-more-essential-mysql-performance-tuning-tips.html>

<https://tecadmin.net/mysql-performance-tuning-tips/>

<https://dev.mysql.com/doc/mysql-perfschema-excerpt/8.0/en/performance-schema.html>

<https://phoenixnap.com/kb/improve-mysql-performance-tuning-optimization>

<https://blog.logrocket.com/5-ways-rapidly-improve-mysql-database-performance/#advanced-operations-tricks>

<https://medium.com/releem/optimizing-mysql-for-peak-performance-a-comprehensive-guide-881ba1ad3ee1>

Building the best INDEX for a given SELECT

https://mysql.rjweb.org/doc.php/index_cookbook_mysql

You have a SELECT and you want to build the best INDEX for it. This blog is a "cookbook" on how to do that task.

How To Fix Slow MySQL queries

<https://www.kevinleary.net/blog/fixing-slow-queries-wordpress-database/>

Partitioning Tables

Table partitioning divides large tables into smaller, more manageable pieces, making it easier to query and manage data. This can improve query performance by reducing the amount of data scanned during queries.

Partition by Range/List/Hash/Key

You can partition tables by range, list, hash, or key, depending on your data and query patterns. For example, partitioning by date range can be useful for tables that store historical data.

Indexes on Partitions

Ensure that each partition is properly indexed to maximize performance gains.

Partitioning can significantly reduce query times for large tables by limiting the amount of data that needs to be scanned.

Temporary Tables and Memory Tables

Temporary tables and memory tables can be used for intermediate calculations or to store transient data that doesn't need to be persisted to disk. Memory tables, in particular, are stored in RAM and are very fast.

Memory Engine

Use the MEMORY storage engine for tables that store small amounts of frequently accessed data. These tables reside in memory, allowing for very fast access.

Temporary Tables

Use temporary tables for complex queries that require multiple steps. Temporary tables are created on the fly and dropped after the session ends, reducing the need to repeatedly query the same data.

These techniques can offload complex operations from your main tables and speed up query processing.

Read Replicas

Set up MySQL read replicas to handle read-only queries, which can reduce the load on your primary database server.

Official MySQL Documentation

<https://dev.mysql.com/doc/refman/8.4/en/replication-solutions-performance.html>

Replication Explainer

<https://dev.mysql.com/doc/refman/8.4/en/replication.html>

PlanetScale Article

<https://planetscale.com/blog/what-is-mysql-replication-and-when-should-you-use-it>

Medium Article

<https://medium.com/@truongbui95/scaling-sql-read-operations-effective-strategies-and-detailed-examples-9b590da68819>

Solid article with configuration examples.

Further Research

AI Chatbots can relay a wealth of information on the topic of Read Replicas, and also assist with implementation.

Asynchronous Processing For Read Operations

Medium Article

<https://medium.com/@denisanikin/asynchronous-processing-with-in-memory-databases-or-how-to-handle-one-million-transactions-per-36a4c01fc4e4>

Tarantool (Free)

https://www.tarantool.io/en/doc/latest/platform/ddl_dml/sql/improving_mysql/

Tarantool is an interesting way to improve MySQL performance and also assists with Database scaling performance.

MySQL Connector for Tarantool

<https://github.com/tarantool/mysql>

Database Sharding

Split your data across multiple databases (shards) based on a shard key. This can reduce the size of each database, making queries faster and reducing the load on any single server.

These approaches allow for horizontal scaling, which is essential for handling massive amounts of traffic and data.

PlanetScale Article

<https://planetscale.com/blog/how-to-scale-your-database-and-when-to-shard-mysql>

Vitess (Free)

<https://github.com/vitessio/vitess>

<https://vitess.io/>

Vitess is a database clustering system for horizontal scaling of MySQL through generalized sharding.

By encapsulating shard-routing logic, Vitess allows application code and database queries to remain agnostic to the distribution of data onto multiple shards. With Vitess, you can even split and merge shards as your needs grow, with an atomic cutover step that takes only a few seconds.

MySQL Performance Cheat Sheet

<https://severalnines.com/blog/mysql-performance-cheat-sheet/>

InnoDB Buffer Pool Instance Division

Percona Article

https://www.percona.com/blog/how-many-innodb_buffer_pool_instances-do-you-need-in-mysql-8/

If you have a high workload and a multi-core system, consider dividing the InnoDB buffer pool into multiple instances. This can help reduce contention for memory structures within InnoDB. To do this, adjust the `innodb_buffer_pool_instances` parameter according to your system's cores and workload.

Tuning InnoDB Log File Size

Releem Article

https://releem.com/docs/mysql-performance-tuning/innodb_log_file_size

The size of the InnoDB log files (`innodb_log_file_size`) has a significant impact on performance. Larger log files can reduce disk I/O for write-intensive workloads but might increase recovery time after a crash. Balancing the log file size based on the nature of your workload can provide performance benefits.

Thread Pool Utilization

For servers handling many connections and queries, using a thread pool can improve performance. Instead of one thread per connection, MySQL can manage a pool of threads and efficiently handle numerous connections.

Optimize Join Buffers

Releem Article

https://releem.com/docs/mysql-performance-tuning/join_buffer_size

When MySQL performs joins, it uses a join buffer. The size of this buffer (`join_buffer_size`) determines how much data MySQL can handle at once during a join operation. Increasing the size of the join buffer can improve the performance of non-indexed joins.

MySQL Reference Manual

<https://dev.mysql.com/doc/refman/8.0/en/buffering-caching.html>

Query Cache Sizing and Usage

MySQL query cache reference manual

<https://dev.mysql.com/doc/refman/5.7/en/query-cache.html>

In versions of MySQL where query cache is still supported, properly sizing (`query_cache_size`) and configuring the query cache can lead to performance improvements for read-heavy workloads. However, note that the query cache is deprecated in later versions of MySQL and removed in MySQL 8.0.

Fine-tuning Sort and Read Buffers

Releem Sort Buffer Size Article

https://releem.com/docs/mysql-performance-tuning/sort_buffer_size

MySQL uses sort buffers (`sort_buffer_size`) and read buffers (`read_buffer_size`) during various operations. Adjusting these buffers can improve performance, especially for complex sorting and reading operations.

MySQL Compression

MySQL reference manual article

<https://dev.mysql.com/doc/refman/8.0/en/innodb-page-compression.html>

For tables with large amounts of text or blob data, consider using MySQL's compression capabilities to reduce disk space and improve I/O efficiency.

Use of Covering Indexes

Toadworld article

<https://blog.toadworld.com/2017/04/06/speed-up-your-queries-using-the-covering-index-in-mysql>

Covering indexes include all the fields required by a query. They can dramatically improve query performance as the database can retrieve all the necessary data from the index without accessing the table data.

Optimizing Subqueries

MySQL Reference Manual article

<https://dev.mysql.com/doc/refman/8.0/en/subquery-optimization.html>

MySQL Tutorial Article

<https://www.mysqltutorial.org/mysql-basics/mysql-subquery/>

In MySQL, subqueries can sometimes be inefficient. Rewriting queries to avoid subqueries or transform them into joins can lead to performance improvements.

Partitioning Large Tables

Sling Academy Article

<https://www.slingacademy.com/article/table-partitioning-in-mysql-8-a-practical-guide/>

For very large tables, consider using table partitioning. This divides a table into parts and can improve query performance, especially for range-based queries.

Data Geek Lab Article

<https://datageeklab.com/mastering-mysql-partitioning/>

Optimizing Temporary Tables and Disk Swaps

MySQL Reference Manual Article

<https://dev.mysql.com/doc/refman/8.0/en/optimize-table.html>

Keep an eye on the use of disk-based temporary tables. If possible, tune your server so that most temporary tables are created in memory (`tmp_table_size` and `max_heap_table_size`).

Disabling Performance Schema

MySQL Reference manual Article

<https://dev.mysql.com/doc/refman/8.0/en/performance-schema-startup-configuration.html>

If you're not using MySQL's performance schema, consider disabling it to save resources. This is especially useful for systems with limited resources.

Binary Log Group Commit

MySQL Reference Manual Article

<https://dev.mysql.com/doc/refman/8.0/en/replication-options-binary-log.html>

For replication setups, optimizing the binary log group commit settings can significantly reduce replication lag and improve the overall performance of the master server.

Resource Groups for Priority Tuning

Resource Group Tuning MySQL Reference Manual Article

<https://dev.mysql.com/doc/refman/8.0/en/resource-groups.html>

Use MySQL 8.0's Resource Groups feature to assign high-priority queries to specific processors or processor cores, improving performance for critical queries.

Linux Blog MySQL Performance Tuning Article

<https://docs.informatica.com/data-engineering/common-content-for-data-engineering/10-2-2/performance-tuning-guide/preface.html>

Very extensive article with some unique optimization strategies.

Database Query Optimization Analysis

Improve WP Query Performance When Using External Object Cache

https://www.spacedmonkey.com/2023/01/17/improve-wp_query-performance-if-using-external-object-cache/

Redash (Free)

<https://github.com/getredash/redash>

SQL users leverage Redash to explore, query, visualize, and share data from any data sources. Their work in turn enables anybody in their organization to use the data. Redash supports more than 35 SQL and NoSQL data sources. It can also be extended to support more.

- **Browser-based:** Everything in your browser, with a shareable URL.
- **Ease-of-use:** Become immediately productive with data without the need to master complex software.
- **Query editor:** Quickly compose SQL and NoSQL queries with a schema browser and auto-complete.
- **Visualization and dashboards:** Create beautiful visualizations with drag and drop, and combine them into a single dashboard.
- **Sharing:** Collaborate easily by sharing visualizations and their associated queries, enabling peer review of reports and queries.
- **Schedule refreshes:** Automatically update your charts and dashboards at regular intervals you define.
- **Alerts:** Define conditions and be alerted instantly when your data changes.
- **REST API:** Everything that can be done in the UI is also available through REST API.
- **Broad support for data sources:** Extensible data source API with native support for a long list of common databases and platforms.

Solarwinds

Database Performance Analyzer (Paid)

<https://www.solarwinds.com/database-performance-analyzer>

Analyze SQL Server and query performance, identify problems, and give optimization and tuning advice

SQL Sentry (Paid)

<https://www.solarwinds.com/sql-sentry>

Database performance monitoring for SQL Server and Azure SQL databases, with fast root cause analysis and visibility across the data estate

Database Observability (Paid)

<https://www.solarwinds.com/solutions/solarwinds-observability/database-observability>

Visualize database health

Plan Explorer (Free)

<https://www.solarwinds.com/free-tools/plan-explorer>

Quickly get to the root of query problems with a free query analysis tool

Varnish (Free)

My Varnish VCL custom config

https://docs.google.com/document/d/18IrewYSQBt0AChDn_wENbDnKHlr0BWt_zkhfttfVbks/edit?usp=sharing

You should be able to just copy paste the contents into your Varnish Default.vcl config file.

Configuration examples

<https://gist.github.com/fevangelou/84d2ce05896cab5f730a>

Awesome Varnish

<https://github.com/huayra/awesome-varnish>

Increase your Varnish Cache Hit Rate

<https://varnish-cache.org/docs/trunk/users-guide/increasing-your-hitrates.html>

Medium Article

<https://nbeguier.medium.com/3-tips-to-boost-the-performance-of-your-varnish-cache-3f4ce44be3c1>

Sharded Caching

Dynamic backend resolution

Cache Ajax Requests

<https://guides.wp-bullet.com/how-to-cache-ajax-get-requests-with-varnish-4/>

Varnish Gather

<https://github.com/varnish/varnishgather>

Varnish Gather is a simple script designed to gather as much relevant information as possible on a Varnish Cache setup.

Additional Varnish Modules

<https://github.com/varnish/varnish-modules>

This is a collection of modules ("vmods") extending Varnish VCL used for describing HTTP request/response policies with additional capabilities.

Gzip, Brotli and Zstandard Text Compression (Free)

Gzip, Brotli and Zstandard compression are text compression methods that work at the **webserver** level. That means you need support for Gzip/Brotli/Zstandard compression enabled via NGINX, Apache, or LiteSpeed and to properly configure them to serve compressed content.

Red Switches Article

<https://www.redswitches.com/blog/brotli-vs-gzip/>

Fantastic in-depth article that explains both Brotli and Gzip compression methods.

WP Rocket Articles

<https://wp-rocket.me/google-core-web-vitals-wordpress/enable-text-compression>

Gzip, Brotli and Zstandard Compression Levels

The articles above have a good explanation of compression levels, their benefits, potential pitfalls from raising the compression level, etc. However, the quick explanation is that higher compression levels produce smaller file sizes but increase server processing time. Increasing compression levels has diminishing returns, and you need a high powered server to use the maximum compression level for Brotli and Gzip.

The default compression level for Brotli and Gzip should be 6. Higher levels are better up to a point. Very high levels of gzip and brotli compression may actually be a performance detriment and hurt your pagespeed scores, so it is highly recommended to run speed tests after adjusting these configuration settings to measure their impact on performance and determine whether further adjustments are needed.

I recommend raising the level one increment at a time until you reach the point where you see a performance hit instead of a benefit on tests.

How to enable Brotli or Gzip for Apache

<https://wp-rocket.me/blog/brotli-vs-gzip-compression/>

WP Rocket has a section in this article on how to implement Gzip/Brotli compression via .htaccess for **Apache only**

How to enable Brotli with Gzip fallback on NGINX

<https://bytepursuits.com/nginx-enabling-brotli-compression-with-gzip-fallback>

Zstandard Compression

Gzip vs Brotli vs Zstandard Compression Article

<https://paulcalvano.com/2024-03-19-choosing-between-gzip-brotli-and-zstandard-compression/#:~:text=Bro%20li%20level%205%20usually%20result,faster%20than%20gzip%20and%20Bro%20li.>

While Zstandard compression is the best and fastest compression algorithm of the three, **Zstandard Compression is currently only supported by Chrome as of May 2024**. Firefox and other browsers will add support for Zstandard compression soon, Zstandard compression support is currently in beta testing for Firefox.

However due to how new Zstandard is, when implemented, you should always have a fallback configured to serve Gzipped/Brotli Compressed content to browsers that do not support Zstandard Compression (any device with an older browser, so the majority of all browsers across user devices).

Experimental NGINX Zstandard Module

<https://github.com/tokers/zstd-nginx-module>

Load Balancing

Haproxy (Free)

HAProxy Caching

LoadForge Article

<https://loadforge.com/guides/advanced-haproxy-caching-strategies>

Great article on how to set up HAProxy caching, as well as advanced caching configuration strategies.

Official HAProxy Article

<https://www.haproxy.com/blog/accelerate-your-apis-by-using-the-haproxy-cache>

Article on HAProxy caching directly from the HAProxy blog.

Resources

<https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts>

<https://www.cloudbees.com/blog/performance-tuning-haproxy>

<https://medium.com/@pawilon/tuning-your-linux-kernel-and-haproxy-instance-for-high-loads-1a2105ea553e>

<https://delta.blue/blog/haproxy-timeouts/>

<https://www.papertrail.com/solution/tips/haproxy-logging-how-to-tune-timeouts-for-performance/>

<https://www.freecodecamp.org/news/how-we-fine-tuned-haproxy-to-achieve-2-000-000-concurrent-ssl-connections-d017e61a4d27/>

<https://www.suse.com/c/speeding-ssl-need-know-haproxy/>

Configuration Recommendations/What to avoid

https://www.reddit.com/r/Proxmox/comments/10r814x/optimizing_web_traffic_to_vms_and_services_haproxy/

Roxy WI

<https://roxy-wi.org/>

GUI for HAProxy and other server stack infrastructure automatic configuration. Free tier, as well as paid.

Pre-Made HAProxy Configs

<https://github.com/HariSekhon/HAProxy-configs>

MPTCP

<https://www.multipath-tcp.org/>

(More info soon)

Redhat Article

<https://www.redhat.com/en/blog/understanding-multipath-tcp-networking-highway-future>

AWS Hosting Stack

Ars Technica Article

<https://arstechnica.com/information-technology/2023/07/ars-on-aws-01/>

Elastic Load Balancing (Paid)

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

Elastic Load Balancing automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones. It monitors the health of its registered targets, and routes traffic only to the healthy targets. Elastic Load Balancing scales your load balancer as your incoming traffic changes over time. It can automatically scale to the vast majority of workloads.

Elastic Load Balancing supports the following load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. You can select the type of load balancer that best suits your needs.

AWS Fargate (Paid)

<https://aws.amazon.com/fargate/>

AWS Fargate is a serverless, pay-as-you-go compute engine that lets you focus on building applications without managing servers.

AWS Lambda (Paid)

<https://aws.amazon.com/lambda/>

AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources (Automatic Scaling), making it the fastest way to turn an idea into a modern, production, serverless applications.

Amazon Aurora (Paid)

<https://aws.amazon.com/rds/aurora/>

Amazon Aurora provides unparalleled high-performance and availability at global scale with full MySQL and PostgreSQL compatibility, at 1/10th the cost of commercial databases. Aurora has 5x the throughput of MySQL and 3x of PostgreSQL.

Amazon Elastic File System (Paid)

<https://aws.amazon.com/efs/>

Serverless, fully elastic file storage. Amazon Elastic File System (Amazon EFS) automatically grows and shrinks as you add and remove files with no need for management or provisioning.

Amazon Elastic Container Service (Paid)

<https://aws.amazon.com/ecs/>

Amazon Elastic Container Service (ECS) is a fully managed container orchestration service that helps you to more efficiently deploy, manage, and scale containerized applications.

Amazon OpenSearch (Paid)

<https://aws.amazon.com/opensearch-service/>

Amazon OpenSearch Service makes it easy for you to perform interactive log analytics, real-time application monitoring, website search, and more. OpenSearch is an open source, distributed search and analytics suite derived from Elasticsearch. Amazon OpenSearch Service offers the latest versions of OpenSearch, support for 19 versions of Elasticsearch (1.5 to 7.10 versions), as well as visualization capabilities powered by OpenSearch Dashboards and Kibana (1.5 to 7.10 versions).

Amazon ElastiCache (Paid)

<https://aws.amazon.com/elasticache/>

Amazon ElastiCache is a fully managed, Redis- and Memcached-compatible service that delivers real-time, cost-optimized performance, and up to 99.99% high availability for modern applications. ElastiCache is ideal for high-performance use cases such as data caching, web, mobile apps, healthcare apps, financial apps, gaming, ad-tech, IoT, media streaming, session stores, leaderboards, machine learning (ML), and microservices-based applications.

ElastiCache speeds up database and application performance, scaling to hundreds of millions of operations per second with microsecond response time. Benefits include enhanced security, reliability, scalability, and performance when compared to open source alternatives. It also unlocks cost savings for read-heavy workloads and provides cost-optimization features like data tiering for memory-intensive workloads.

With Amazon ElastiCache Serverless, you can create a highly available cache in under a minute without infrastructure provisioning or configuration. You can create an ElastiCache Serverless cache in a few steps by specifying a cache name in the AWS Management Console, AWS Software Development Kit (SDK), or AWS Command Line Interface (CLI).

VPS Optimizations (Free)

Optimizing Linux

Disable THP

<https://www.pingcap.com/blog/transparent-huge-pages-why-we-disable-it-for-databases/>

Tuned-ADM

<https://manpages.ubuntu.com/manpages/jammy/man7/tuned-profiles.7.html>

Change Init Level to 3

Set your init level to 3 if the default is set to 5. Use the command “init 3” to set the init level to 3, then reboot for the changes to take effect. You can click the link below for instructions on how to do so.

<https://www.cyberciti.biz/howto/question/linux/changing-run-levels-3-5.php>

[More info on that here](#)

This link is how to do the reverse, changing 5 to 3 if the “init 3” command does not work. **If your run level is set to 5, change it to 3.**

Optimizing Linux Github

<https://github.com/sn99/Optimizing-linux>

Great repo for how to optimize linux performance.

Stacer

<https://www.site24x7.com/learn/linux/linux-performance-optimization.html>

Stacer is a tool used for optimizing Linux that deletes unwanted cache and sorting processes by looking at memory usage and CPU. Stacer reveals how services and processes are performing and which user is executing the processes.

Performance Optimization Tools and Techniques

<https://linuxconfig.org/linux-performance-optimization-tools-and-techniques>

Enable Z-Swap

<https://blog.jacobstoner.com/zswap-improve-memory-performance-on-linux/>

```
sudo perl -pi -e 's/^(GRUB_CMDLINE_LINUX_DEFAULT.*)"$/$1 zswap.enabled=1  
zswap.compressor=lz4 zswap.zpool=z3fold zswap.max_pool_percent=25"/' /etc/default/grub  
sudo update-grub  
sudo echo -e 'lz4\nlz4_compress\nz3fold' >> /etc/initramfs-tools/modules  
sudo update-initramfs -u  
sudo reboot
```

Tune the Linux Filesystem for increased performance

Foss Linux Article

<https://www.fosslinux.com/111937/tips-and-tricks-for-optimizing-linux-device-performance.htm>

Semi-surface level, but slightly different info than the linux.com article, best to use both of the many listed techniques together.

Use a Custom Kernel

<https://www.fosslinux.com/111937/tips-and-tricks-for-optimizing-linux-device-performance.htm>

Disable Unnecessary Linux Services

To list all services, you can use this command:

systemctl list-unit-files --type=service

Then, to disable a service, use:

sudo systemctl disable SERVICE_NAME

Jumbo Frames

Enabling Jumbo Frames:

- Check if your network adapter supports Jumbo Frames: `ethtool eth0`

- Enable Jumbo Frames by setting a larger MTU (Maximum Transmission Unit), typically 9000 bytes for Ethernet: `ifconfig eth0 mtu 9000` or `ip link set dev eth0 mtu 9000`
- Verify the changes: `ifconfig eth0` or `ip addr show eth0`
- Remember to configure Jumbo Frames on all devices in your network to avoid packet fragmentation.

Jumbo Frames can help reduce CPU load by decreasing the number of packets processed, but it's crucial to ensure that all network equipment supports them to avoid potential network issues. **Contact your hosting provider to ensure enabling this feature is supported.**

TCP BBR

Increase Linux Internet speed with TCP BBR congestion control

<https://www.cyberciti.biz/cloud-computing/increase-your-linux-server-internet-speed-with-tcp-bbr-congestion-control/>

<https://www.techrepublic.com/article/how-to-enable-tcp-bbr-to-improve-network-speed-on-linux/>

<https://www.linuxbabe.com/ubuntu/enable-google-tcp-bbr-ubuntu>

<https://github.com/systemd/systemd/issues/9725#issuecomment-413369212>

<https://djangocas.dev/blog/huge-improve-network-performance-by-change-tcp-congestion-control-to-bbr/>

Linux Network Performance Parameters

<https://github.com/leandromoreira/linux-network-performance-parameters>

Intel's VPS Wordpress Tuning guide for Xeon Systems

<https://www.intel.com/content/www/us/en/developer/articles/guide/wordpress-tuning-guide-on-xeon-systems.html>

Linux Optimization suggestions for the Steam Deck

These are Linux optimizations for the Steam Deck, however they are also applicable to Linux VPS optimization. Below is the Medium article I pulled these commands from, as well as reversion commands for each of these optimization features in case you want to revert for any reason.

Paste all of these commands as one single code block, not single lines one at a time. **The reversion commands should not be run together with the commands to enable these features, only if you want to disable them and revert back to stock.**

Medium Article

<https://medium.com/@a.b.t./here-are-some-possibly-useful-tweaks-for-steamos-on-the-steam-deck-fcb6b571b577#6c57>

Switching over to the “performance” CPU governor at start-up

(paste this into your terminal as one giant code block, do not run the commands individually)

```
cat << EOF | sudo tee /etc/systemd/system/cpu_performance.service
[Unit]
Description=CPU performance governor
[Service]
Type=oneshot
ExecStart=/usr/bin/cpupower frequency-set -g performance
[Install]
WantedBy=multi-user.target
EOF
sudo systemctl daemon-reload
sudo systemctl enable cpu_performance.service
```

To undo or revert this change, run:

```
sudo systemctl disable cpu_performance.service
```

MGLRU (Multi-Gen. Least Recently Used)

This is one of the most recent additions to the upstream Linux kernel, literally just merged for the 6.1 LTS (Long-Term Support). However, the upstream kernel doesn't enable this functionality by default, yet.

(paste this into your terminal as one giant code block, do not run the commands individually)

To enable this feature, run the terminal command:

```
cat << EOF | sudo tee /etc/tmpfiles.d/mglru.conf
w /sys/kernel/mm/lru_gen/enabled - - - - 7
w /sys/kernel/mm/lru_gen/min_ttl_ms - - - - 0
```

EOF

Reversion command:

```
sudo rm /etc/tmpfiles.d/mglru.conf
```

Unlocking the memory lock

By default, the Linux kernel sets this particular value to just 64KiB, which is the maximum amount of memory the kernel will lock within a single operation. This is an old recommendation.

This command will set this value to 2 GB, adjust as needed depending on your server's RAM capacity:

(paste this into your terminal as one giant code block, do not run the commands individually)

```
cat << EOF | sudo tee /etc/security/limits.d/memlock.conf
* hard memlock 2147484
* soft memlock 2147484
EOF
```

This site will help you convert the values so you can replace the above value “2147484” with an appropriate value for your VPS's memory.

<http://www.csgnetwork.com/memconv.html>

To revert the change, run:

```
sudo rm /etc/security/limits.d/memlock.conf
```

Changing the I/O (Input/Output) Scheduler

We are going to permanently swap the I/O-scheduler to Kyber with the following block (note that this will work with all cases, meaning NVMe SSD, eMMC & microSD):

(paste this into your terminal as one giant code block, do not run the commands individually)

```
cat << EOF | sudo tee /etc/udev/rules.d/64-ioschedulers.rules
ACTION=="add|change", KERNEL=="nvme[0-9]*", ATTR{queue/scheduler}="kyber"
ACTION=="add|change", KERNEL=="sd[a-z]mmcblk[0-9]*", ATTR{queue/rotational}=="0",
ATTR{queue/scheduler}="kyber"
EOF
```

To undo/revert this change:

```
sudo rm /etc/udev/rules.d/64-ioschedulers.rules
```

TCP Network Stack Optimization

TCP Keepalive

Adjusts how long TCP should wait before sending keepalive messages to confirm that an idle connection is still active. This can help in managing connection overloads.

(Adjust command values as needed)

```
sysctl -w net.ipv4.tcp_keepalive_time=600  
sysctl -w net.ipv4.tcp_keepalive_probes=5  
sysctl -w net.ipv4.tcp_keepalive_intvl=60
```

TCP Window Scaling

Enables TCP window size adjustments, allowing for larger buffers, which can improve data throughput for connections.

(Adjust command value as needed)

```
sysctl -w net.ipv4.tcp_window_scaling=1
```

TCP SACK

The Selective Acknowledgment (SACK) feature allows devices to inform senders about lost packets, facilitating more efficient retransmissions.

(Adjust command value as needed)

```
sysctl -w net.ipv4.tcp_sack=1
```

TCP FIN Timeout

Adjusts the timeout to close a connection, freeing up resources sooner.

(Adjust command value as needed)

```
sysctl -w net.ipv4.tcp_fin_timeout=30
```

Increase TCP Max Syn Backlog

Determines the number of incoming connections that can be queued for handling.

(Adjust command value as needed)

```
sysctl -w net.ipv4.tcp_max_syn_backlog=2048
```

Increase TCP Max Tw Buckets

Controls the maximum number of TCP sockets in TIME_WAIT state, managing memory use.

(Adjust command value as needed)

```
sysctl -w net.ipv4.tcp_max_tw_buckets=400000
```

TCP Fast Open

Allows data to be sent during the initial SYN handshake, reducing latency for connections.

(Adjust command value as needed)

```
sysctl -w net.ipv4.tcp_fastopen=3
```

Reversion Commands

To revert TCP settings, you'll need to know the default values, which might vary based on your Linux distribution and version. Here are commands to set typical default values:

```
sysctl -w net.ipv4.tcp_keepalive_time=7200  
sysctl -w net.ipv4.tcp_keepalive_probes=9  
sysctl -w net.ipv4.tcp_keepalive_intvl=75  
sysctl -w net.ipv4.tcp_window_scaling=1  
sysctl -w net.ipv4.tcp_sack=1  
sysctl -w net.ipv4.tcp_fin_timeout=60  
sysctl -w net.ipv4.tcp_max_syn_backlog=128  
sysctl -w net.ipv4.tcp_max_tw_buckets=5000  
sysctl -w net.ipv4.tcp_fastopen=0
```

Enable Asynchronous I/O (AIO)

Enable AIO to allow multiple I/O operations to execute concurrently, improving performance for database servers like MySQL

```
innodb_use_native_aio = 1
```

Offload Processing

Enable offloading features on your network interface to reduce CPU load

```
ethtool -K eth0 gro on  
ethtool -K eth0 gso on  
ethtool -K eth0 tso on
```

Increase Open File Limits

Ensure your system can handle a large number of open files

```
echo "fs.file-max = 500000" >> /etc/sysctl.conf  
ulimit -n 500000
```

Enable Writeback Caching for Disk I/O

Improve write performance by enabling writeback caching

```
hdparm -W1 /dev/sda
```

Store temporary files in memory to speed up read/write operations

```
mount -t tmpfs -o size=512M tmpfs /tmp
```

Optimize Memory Usage

Use zram to compress RAM data to increase the amount of available memory

```
apt-get install zram-config
```

Optimize Virtual Memory Settings

Adjust virtual memory settings for better performance

```
sysctl -w vm.dirty_background_ratio=5  
sysctl -w vm.dirty_ratio=10
```

Use CPU Isolation

Isolate CPU cores for critical applications to ensure they have dedicated resources

```
isolcpus=1,2
```

Configure HugeTLB Pages

For applications requiring large memory pages, configure HugeTLB to reduce TLB misses

```
echo 'vm.nr_hugepages=128' >> /etc/sysctl.conf
```

Optimize DNS Resolution

Use a Local DNS Cache

Install and configure a local DNS caching server like dnsmasq

```
apt-get install dnsmasq
```

Adjust Out-of-Memory (OOM) Killer Settings

Prevent the OOM killer from terminating critical processes by adjusting oom_score_adj

```
echo '-1000' > /proc/$(pidof critical_process)/oom_score_adj
```

Use RAM Disks for High-Performance Temporary Storage

Mount a portion of your RAM as a disk for extremely fast I/O operations

```
mount -t tmpfs -o size=1G tmpfs /mnt/ramdisk
```

Optimize Swap Settings

Use zswap

Enable and configure zswap for better swap performance

```
echo 1 > /sys/module/zswap/parameters/enabled  
echo lz4 > /sys/module/zswap/parameters/compressor  
echo 20 > /sys/module/zswap/parameters/max_pool_percent
```

Optimize Boot Parameters

Modify boot parameters to improve performance

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash noapic noacpi nosplash irqpoll"
```

Reduce Kernel Latency

Enable kernel preemption and configure low-latency settings

```
echo 1 > /sys/kernel/debug/sched_features
```

Enable and Configure HugeTLBfs

Use HugeTLBfs to allocate large pages for applications that can benefit from reduced TLB misses

```
mount -t hugetlbfs none /dev/hugepages
```

Configure Network Bonding

Combine multiple network interfaces for increased bandwidth and redundancy

```
apt-get install ifenslave  
echo "bonding" >> /etc/modules
```

Fine-Tune TCP SYN Cookies

Enable SYN cookies to protect against SYN flood attacks while allowing legitimate traffic.

```
sysctl -w net.ipv4.tcp_syncookies=1
```


Optimize ARP Cache Settings

Tune ARP cache settings to handle large numbers of IP addresses efficiently

```
sysctl -w net.ipv4.neigh.default.gc_thresh1=1024  
sysctl -w net.ipv4.neigh.default.gc_thresh2=2048  
sysctl -w net.ipv4.neigh.default.gc_thresh3=4096
```

Enable IP Spoofing Protection

Configure settings to prevent IP spoofing

```
sysctl -w net.ipv4.conf.all.rp_filter=1  
sysctl -w net.ipv4.conf.default.rp_filter=1
```

Optimize Kernel Shared Memory Parameters

Adjust shared memory settings to improve performance for applications that use IPC (Inter-Process Communication)

```
sysctl -w kernel.shmmax=68719476736  
sysctl -w kernel.shmall=4294967296
```

Adjust VM Overcommit Settings

Fine-tune memory overcommit behavior for better memory management

```
sysctl -w vm.overcommit_memory=1  
sysctl -w vm.overcommit_ratio=50
```

Enable and configure query caching in MySQL/MariaDB for faster query responses

```
query_cache_size = 64M  
query_cache_type = ON
```

Configure TCP Timestamps

Enable or disable TCP timestamps based on your workload to improve performance

```
sysctl -w net.ipv4.tcp_timestamps=1
```

Adjust UDP Buffer Sizes

Increase UDP buffer sizes for applications that use UDP heavily

```
sysctl -w net.core.rmem_max=26214400  
sysctl -w net.core.wmem_max=26214400
```

Optimize nf_conntrack Settings

Adjust connection tracking settings for high-volume traffic

```
sysctl -w net.netfilter.nf_conntrack_max=131072  
sysctl -w net.netfilter.nf_conntrack_tcp_timeout_established=1200
```

Enable Disk Write Caching

Enable write caching on disks to improve write performance

```
hdparm -W1 /dev/sda
```

Tune Systemd for Performance

Reduce systemd's default timeout values for faster boot and shutdown

```
DefaultTimeoutStartSec=10s  
DefaultTimeoutStopSec=10s
```

Optimize NTP Synchronization

Ensure accurate time synchronization with minimal overhead using chrony instead of ntpd

```
apt-get install chrony
```

Enable and Configure Network Interface Offloading

Enable offloading features on network interfaces to reduce CPU usage

```
ethtool -K eth0 gro on  
ethtool -K eth0 gso on  
ethtool -K eth0 tso on
```

Implement Kernel Same-Page Merging (KSM)

Enable KSM to merge identical memory pages between virtual machines

```
echo 1 > /sys/kernel/mm/ksm/run
```

Use High-Resolution Timers

Enable high-resolution timers for applications that require precise timing

```
echo "highres=on" >> /etc/default/grub  
update-grub
```

Optimize sysctl Parameters for Network Performance

Adjust various sysctl parameters to enhance network throughput and reduce latency

```
sysctl -w net.ipv4.tcp_rfc1337=1  
sysctl -w net.ipv4.tcp_mtu_probing=1  
sysctl -w net.ipv4.tcp_base_mss=536
```

Implement Disk Quotas for User Management

Enable and configure disk quotas to manage user disk usage efficiently

```
apt-get install quota
```

Use Page-Cache Management Tools

Use tools like vmtouch to manage and lock frequently accessed files in memory

```
apt-get install vmtouch  
vmtouch -dl /path/to/important/files
```

Optimize Swapping Behavior

Configure vm.dirty_expire_centisecs and vm.dirty_writeback_centisecs for better swap performance

```
sysctl -w vm.dirty_expire_centisecs=2000
```

```
sysctl -w vm.dirty_writeback_centisecs=1000
```

Implement Software RAID for Disk Performance

Use software RAID to improve disk performance and redundancy

```
mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
```

Optimize Kernel Timer Frequency

Adjust the kernel timer frequency to reduce overhead

```
echo 1000 > /proc/sys/kernel/timer_frequency
```

Enable IPv6

Enable and optimize IPv6 for better network performance and future-proofing

```
sysctl -w net.ipv6.conf.all.disable_ipv6=0
```

Use xfs_fsr for XFS File System Defragmentation

Defragment XFS file systems to improve performance

```
xfs_fsr /dev/sda1
```

Optimize TCP Backlog Settings

Increase TCP backlog settings to handle more simultaneous connections

```
sysctl -w net.core.somaxconn=1024
```

```
sysctl -w net.core.netdev_max_backlog=5000
```

Use mdadm for Software RAID Management

Manage software RAID arrays with mdadm

```
mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
```

Implement TCP MSS Clamping

Adjust TCP MSS (Maximum Segment Size) to avoid fragmentation

```
iptables -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Optimize TCP SYN Flood Protection

Configure TCP SYN flood protection to prevent DoS attacks

```
sysctl -w net.ipv4.tcp_syncookies=1
```

Enable and Configure dnsmasq for DNS Caching

Use dnsmasq to cache DNS queries and reduce lookup times

```
apt-get install dnsmasq  
systemctl enable dnsmasq  
systemctl start dnsmasq
```

Enable tcp_tw_recycle for Faster Connection Reuse

Enable tcp_tw_recycle to speed up connection reuse

```
sysctl -w net.ipv4.tcp_tw_recycle=1
```

Use blktrace for Block Layer Tracing

Install and use blktrace to trace block I/O operations

```
apt-get install blktrace  
blktrace -d /dev/sda
```

Adjust Kernel Panic Settings

Configure the system to automatically reboot after a kernel panic

```
sysctl -w kernel.panic=10  
sysctl -w kernel.panic_on_oops=1
```

Optimize Inotify Watches

Increase the number of inotify watches to handle more file events

```
sysctl -w fs.inotify.max_user_watches=524288
```

Enable TCP Window Scaling

Improve TCP performance by enabling window scaling

```
sysctl -w net.ipv4.tcp_window_scaling=1
```

Enable TCP Keepalive Probes

Configure TCP keepalive probes to manage idle connections

```
sysctl -w net.ipv4.tcp_keepalive_time=300
```

```
sysctl -w net.ipv4.tcp_keepalive_intvl=60
```

```
sysctl -w net.ipv4.tcp_keepalive_probes=5
```

Optimize TCP Congestion Control Algorithms

Experiment with different TCP congestion control algorithms to find the best fit for your workload

```
sysctl -w net.ipv4.tcp_congestion_control=bbr
```

Use schedtool for Advanced CPU Scheduling

Install and use schedtool to control CPU scheduling policies

```
apt-get install schedtool
```

```
schedtool -R -p 20 -e myapp
```

Enable Kernel Samepage Merging (KSM)

Use KSM to reduce memory usage by merging identical memory pages

```
echo 1 > /sys/kernel/mm/ksm/run
```

Optimize Network Buffer Sizes

Increase network buffer sizes to handle high traffic volumes

```
sysctl -w net.core.rmem_max=16777216  
sysctl -w net.core.wmem_max=16777216
```

Optimize Kernel Scheduler

Adjust kernel scheduler settings for better performance

```
echo 1 > /proc/sys/kernel/sched_autogroup_enabled
```

Adjust rcu Settings for Low-Latency Systems

Tune Read-Copy Update (RCU) settings for low-latency performance

```
echo 0 > /sys/kernel/rcu_expedited
```

Optimize Filesystem Journaling

Choose the appropriate journaling mode for your filesystem (e.g., data=writeback for ext4)

```
mount -o remount,data=writeback /dev/sda1
```

Enable and Configure BBRv2 TCP Congestion Control

Use BBRv2 for even better TCP congestion control performance

```
echo "net.ipv4.tcp_congestion_control=bbr" >> /etc/sysctl.conf  
sysctl -p
```

Optimize File System Performance with tune2fs

Use tune2fs to optimize ext4 filesystem parameters

```
tune2fs -o journal_data_writeback /dev/sda1
```

Use ethtool to Optimize NIC Settings

Fine-tune NIC settings for better network performance

```
ethtool -K eth0 tso off
```

Enable and Configure Network Interface Teaming

Use network interface teaming for load balancing and failover

```
apt-get install ifenslave
```

Use irqbalance for Balancing Interrupts

Install and configure irqbalance to distribute interrupts across CPUs

```
apt-get install irqbalance
```

Tune Kernel Semaphore Parameters

Adjust semaphore settings for better performance with IPC-heavy applications

```
sysctl -w kernel.sem="250 32000 100 128"
```

Configure and Use ethtool for Network Optimization

Use ethtool to optimize network interface settings

```
ethtool -K eth0 rx off tx off
```

Enable TCP Small Queues (TSQ)

Optimize the TCP stack by enabling TSQ

```
sysctl -w net.ipv4.tcp_limit_output_bytes=131072
```


Enable and Configure EarlyOOM

Use EarlyOOM to handle out-of-memory situations more gracefully

```
apt-get install earlyoom
```

Optimize Inode Cache

Adjust inode cache settings to handle large numbers of files

```
sysctl -w fs.inode-nr
```

```
sysctl -w fs.inode-state
```

Optimize Disk Caching

Adjust `vm.dirty_ratio` and `vm.dirty_background_ratio` to control disk caching behavior

```
sysctl -w vm.dirty_ratio=20
```

```
sysctl -w vm.dirty_background_ratio=10
```

Implement IP Address Aliasing

Use IP address aliasing to manage multiple IP addresses on a single network interface, improving network performance and management

```
ip addr add 192.168.1.2/24 dev eth0
```

Enable Asynchronous I/O for Database Servers

Enable asynchronous I/O (AIO) for database servers like MySQL or PostgreSQL to improve disk I/O performance

```
innodb_use_native_aio=1
```

Optimize TCP Reassembly Settings

Adjust TCP reassembly settings to handle fragmented packets more efficiently

```
sysctl -w net.ipv4.tcp_reordering=3
```

Enable jemalloc for Memory Allocation

Use jemalloc as a memory allocator for applications like Nginx, MySQL, or Redis to improve memory management performance

```
LD_PRELOAD=/usr/lib/x86_64-linux-gnu/libjemalloc.so.2
```

Adjust IO Elevator Algorithms

Switch to a more suitable I/O elevator algorithm like mq-deadline or none for SSDs

```
echo mq-deadline > /sys/block/sda/queue/scheduler
```

Use fstrim for SSDs

Periodically run fstrim to inform the SSD of unused blocks, improving performance

```
fstrim -v /
```

Optimize TCP Keepalive Settings

Fine-tune TCP keepalive settings to manage idle connections better

```
sysctl -w net.ipv4.tcp_keepalive_time=120
```

```
sysctl -w net.ipv4.tcp_keepalive_intvl=30
```

```
sysctl -w net.ipv4.tcp_keepalive_probes=5
```

Optimize Virtual Memory Overcommit

Adjust virtual memory overcommit settings to better handle memory allocation

```
sysctl -w vm.overcommit_memory=2  
sysctl -w vm.overcommit_ratio=50
```

Use ethtool for Network Interface Tuning

Adjust network interface settings for better performance

```
ethtool -K eth0 rx off tx off
```

Implement BBRv2 TCP Congestion Control

Use BBRv2 for enhanced TCP congestion control performance

```
sysctl -w net.ipv4.tcp_congestion_control=bbr
```

Optimize I/O Performance with blk-mq (Multi-Queue Block I/O Queueing Mechanism)

Enable and configure blk-mq for better I/O performance on SSDs

```
echo "mq-deadline" > /sys/block/sda/queue/scheduler
```

Use schedtool to Control CPU Scheduling Policies

Install and use schedtool to manage CPU scheduling

```
apt-get install schedtool  
schedtool -R -p 20 -e /usr/bin/myapp
```

Optimize TCP Retransmission Timeout (RTO)

Adjust TCP RTO settings for better performance

```
sysctl -w net.ipv4.tcp_retries2=5
```

Optimize TCP SYN Cookies

Enable TCP SYN cookies to protect against SYN flood attacks

```
sysctl -w net.ipv4.tcp_syncookies=1
```

Optimize TCP Window Scaling

Enable TCP window scaling for better performance

```
sysctl -w net.ipv4.tcp_window_scaling=1
```

Use cgroups for Resource Management

Implement cgroups to limit, prioritize, and isolate resource usage of processes

```
cgcreate -g cpu,memory:mygroup  
cgset -r cpu.shares=512 mygroup  
cgset -r memory.limit_in_bytes=512M mygroup
```

Optimize TCP Memory Buffers

Adjust TCP memory buffer settings for better performance

```
sysctl -w net.ipv4.tcp_rmem='4096 87380 16777216'  
sysctl -w net.ipv4.tcp_wmem='4096 65536 16777216'
```

Optimize Kernel Schedulers

Choose the appropriate I/O scheduler for your workload (noop for SSD)

```
echo "noop" > /sys/block/sda/queue/scheduler
```

Optimize Kernel Shared Memory Parameters

Adjust shared memory settings for better performance with IPC-heavy applications

```
sysctl -w kernel.shmmax=68719476736  
sysctl -w kernel.shmall=4294967296
```

Implement ZFS for Advanced File System Features

Use ZFS for better data integrity, snapshots, and performance

```
apt-get install zfsutils-linux  
zpool create mypool /dev/sda
```

Enable TCP Selective Acknowledgements (SACK)

Optimize TCP performance with SACK

```
sysctl -w net.ipv4.tcp_sack=1
```

Enable and Configure zram for Compressed Swap

Use zram to create a compressed block device in RAM

```
apt-get install zram-config
```

Implement TCP Timestamping

Enable TCP timestamps for better performance and reliability

```
sysctl -w net.ipv4.tcp_timestamps=1
```

Use zpool for ZFS Pool Management

Manage ZFS storage pools with zpool

```
zpool create mypool /dev/sda
```

Optimize File System Access Time

Use noatime and nodiratime mount options to reduce disk I/O

```
mount -o remount,noatime,nodiratime /
```

Optimize TCP Backlog Size

Increase TCP backlog size to handle more incoming connections

```
sysctl -w net.core.somaxconn=1024
```

Optimize Kernel Virtual Memory Settings

Adjust Virtual Memory Settings For Better Performance and Optimize Kernel File Handle Limits

Increase the number of file handles the kernel can manage
`sysctl -w fs.file-max=2097152`

Enable TCP Low Latency

Configure TCP low latency mode for better responsiveness

`sysctl -w net.ipv4.tcp_low_latency=1`

Use pm2 for Node.js Process Management

Install and use pm2 to manage Node.js processes

```
npm install pm2@latest -g
pm2 start app.js
```

Note: Only applicable if using Node.js in Wordpress

Implement Squid for Caching and Proxying

Install and configure Squid for web caching and proxying

```
apt-get install squid
```

Optimize TCP Congestion Control for High Latency Networks

Use a suitable TCP congestion control algorithm for high-latency networks

```
sysctl -w net.ipv4.tcp_congestion_control=htcp
```

Use cputlimit to Limit CPU Usage of Processes

Install and use cputlimit to control the CPU usage of processes

```
apt-get install cputlimit
cputlimit -e myapp -l 50
```

Optimize Disk Write Caching

Enable disk write caching for better write performance

```
hdparm -W1 /dev/sda
```

Use noatime and nodiratime Mount Options

Reduce disk I/O by disabling access time updates on filesystems

```
mount -o remount,noatime,nodiratime /
```

Adjust Maximum Transmission Unit (MTU) for Network Interfaces

Increase MTU size for better performance on high-speed networks

```
ip link set dev eth0 mtu 9000
```

Tune Kernel Preemption Model

Configure the kernel preemption model for low-latency desktop or high-throughput server workloads

```
echo 0 > /proc/sys/kernel/preempt
```

Optimize Disk I/O

Increase Read-Ahead Buffer

Adjust the read-ahead buffer size for disk I/O

```
blockdev --setra 2048 /dev/sda
```

Optimize IRQ (Interrupt Request) Balancing

Distribute IRQ load across multiple CPU cores to prevent bottlenecks

```
apt-get install irqbalance
```

```
systemctl enable irqbalance
```

Use HugePages for Memory-Intensive Applications

```
echo 'vm.nr_hugepages=128' >> /etc/sysctl.conf
```

Use Lightweight Logging

Reduce the verbosity of system logs to decrease disk I/O

```
sed -i 's/^#LogLevel=info/LogLevel=error/' /etc/systemd/journald.conf
```

Network Queue Management

Queue Length

Increasing the network interface queue length can help handle more packets during traffic bursts, reducing packet drops.

(Adjust value as needed)

```
ifconfig eth0 txqueuelen 10000
```

Reversion Command

Reset the transmission queue length to its default value (usually 1000):

```
ifconfig eth0 txqueuelen 1000
```

I/O Scheduler

Choosing the right I/O scheduler can improve performance based on your workload characteristics (e.g., SSD vs HDD).

Set I/O Scheduler

For SSDs, noop or deadline might be better.

```
echo "deadline" > /sys/block/sda/queue/scheduler
```

Swappiness

The vm.swappiness parameter controls the relative weight given to swapping out runtime memory, versus dropping pages from the system cache.

Adjust Swappiness

```
sysctl -w vm.swappiness=10
```

Preventing the superfluous book-keeping of File Access Times

The Linux kernel updates the access time of every file every time it would access it, meaning that even a read-only operation would immediately result in a write operation, because the access time of the file in question had to be updated.

Therefore, let's permanently switch over to the noatime setting with the following command:

```
sudo sed -i -e '/home/s/\bdefaults\b/&,&,noatime/' /etc/fstab
```

To undo/revert this change:

```
sudo sed -i -e 's/,noatime//' /etc/fstab
```

Reversion Commands

File Descriptor Limits

Reverting file descriptor limits involves setting them back to typical defaults:

```
# System-wide
echo "fs.file-max = 65535" >> /etc/sysctl.conf
# Per-user
echo "* soft nofile 1024" >> /etc/security/limits.conf
echo "* hard nofile 4096" >> /etc/security/limits.conf
```

I/O Scheduler

To revert the I/O scheduler, you can switch back to the default. For many Linux systems, cfq (Completely Fair Queuing) is a common default for HDDs, and noop or mq-deadline for SSDs:

```
echo "cfq" > /sys/block/sda/queue/scheduler
```

Swappiness

The default swappiness value is usually set at 60, which is suitable for most general-use cases:

```
sysctl -w vm.swappiness=60
```

Making Reversions Permanent

To ensure these settings are permanent, update the `/etc/sysctl.conf` file with the reverted values and run:

```
sysctl -p
```

This command reloads the configurations from the `sysctl.conf` file.

SSH tools

WinSCP (Windows only) (Free)

<https://winscp.net/eng/index.php>

A great GUI SFTP file manager that works over SSH. I highly recommend it for managing files over SSH on Windows. Has an integrated terminal that can be launched from the GUI.

Filezilla (Multi-platform) (Free)

<https://filezilla-project.org/index.php>

Multi-platform SFTP GUI file manager, works on Mac, Windows and Linux.

SolarPutty (Windows only) (Free)

<https://www.solarwinds.com/free-tools/solar-putty>

Fantastic terminal program with some nice QOL features.

Putty lacks the ability to do basic things such as copy paste, or saving login credentials.

Note

If you go the VPS route like I did, you are going to be using SSH to modify config files frequently(at least during the initial setup), and it will be **absolutely necessary** to use it if you somehow manage to crash your server and get locked out of the frontend/backend of Wordpress. You can always revert your changes to the config file if they cause issues, because Wordpress is just a program running on your server. You will always be able to access the config files via SSH as long as your server is on.

Always configure your SSH credentials and log in with an SSH client before you make any modifications to Apache, PHP, MySQL, Varnish, HAProxy, Nginx, or any

other server level software so you don't get locked out and have to start from scratch. You will need to do this with any VPS you manage.

This is absolutely critical for the servers your live sites run on. Do not forget to set your SSH credentials up.

CI/CD Pipeline (Continuous integration and continuous delivery/deployment)

Github (Free)

WPPusher

<https://wppusher.com/>

Free tier supports public repositories.

Github Embed

<https://wordpress.org/plugins/github-embed/>

Plugin that allows you to embed details from github just by pasting in the URL as you would any other embed source. Currently supports:

- Repositories
- User profiles
- Project milestone summaries
- Project contributors

Github Updater

<https://github.com/afragen/git-updater>

A simple plugin to enable automatic updates to your GitHub hosted WordPress plugins, themes, and language packs. Additional API plugins available for Bitbucket, GitLab, Gitea, and Gist.

Git and Composer Pipeline

Delicious Brains Article

<https://deliciousbrains.com/storing-wordpress-in-git/#custom-plugins-themes>

Jenkins (Free)

<https://www.jenkins.io/>

<https://github.com/jenkinsci/jenkins>

Medium Article

<https://medium.com/@mudasirhaji/complete-step-by-step-jenkins-cicd-with-github-integration-aae3961b6e33>

Extremely comprehensive Medium guide on Jenkins CICD github development

CI/CD Jenkins Github Pipeline

Blazemeter Article

<https://www.blazemeter.com/blog/cicd-pipeline-jenkins-github>

Hosting

Managed vs VPS Hosting

Make sure you choose a fast host. In my testing VPS hosting is fastest, but managed hosting providers such as WP Engine/Cloudways/Kinsta etc, are close. Managed hosting is suggested for major clients so they have hosting support if necessary when you (the developer) is unavailable.

VPS hosting is generally going to be faster than those hosts, and give you much more flexibility to implement performance optimizations.

Avoid shared hosting

Avoid shared hosting like the plague. Extremely limited server resources, almost no configuration flexibility, expensive, and **SLOW**.

Hosting Server Location

When I choose a location for my servers, I like to place it in as close to a central location to my users geographic region as possible. As an example, in the US I place my servers in the center

of the country so that visitors on the East and West Coast will have somewhat similar latency, so the user experience is relatively homogenous regardless of their state.

VPS Configuration Panels and Software

ClusterCS (Paid)

<https://clustercs.com/>

I use ClusterCS to configure my VPS automatically. Personally, they are my preferred panel. All you need are SSH credentials, they handle everything else. You can also tweak many, many more settings than using a hosting providers panel directly, they don't place any limitations on what you can configure. They're the priciest panel option however.

aaPanel (Free)

<https://www.aapanel.com/new/index.html>

aaPanel is free, and the closest to my LAMP setup(meaning you can follow the apache optimization section. Unfortunately I have no experience with NGINX).

WordOPs (Free)

<https://wordops.net/>

WordOPs is another free option to automatically configure your VPS to run Wordpress. WordOPs is a command line tool to install Wordpress on a VPS over SSH.

HTTP/3

HTTP/3 support is now included in WordOPs!

Other VPS panel options

The other paid panel options that I'm aware of that you have for VPS configuration are services such as **Runcloud**, **SpinUpWP**, and some other cloud control panels like **Cyberpanel**. There are many other options available as well that are not listed here.

Cheap and Fast VPS Hosting Providers

Digital Ocean

<https://www.digitalocean.com/>

Digital Ocean Droplets are pretty cheap, and they have great performance. Of all the mainstream providers that are commonly recommended in hosting articles, they are by far the best. They certainly outclass **Vultr and Linode**, which are extremely slow, and should not be used.

Genesis Hosting

<https://genesishosting.com/>

A little known hosting provider that has relatively performant VPS's (even though their specs may appear to be low compared to other providers). They give you bare metal VPS's (just like VPSDime listed below), which you will have to configure yourself over SSH with one of the aforementioned VPS configuration panels.

My VPS Host

VPSDime

I host with [VPSDime](#). VPSDime has incredibly cheap (and blazing fast!) Linux VPS's. I host my sites on their cheapest Linux VPS plan, which is a 4 vCPU, 6 GB ram machine for only \$7/month. However their RAM upgrade options only cost an additional \$7/month for 6 GB more RAM up to a cap of 36 GB.

The system resources they give you for what they charge is insane, and out of the 50+ hosting providers I have tested, no one has better performance and certainly no one has better pricing than what they charge.

Disclosure: The above link to VPSDime contains an affiliate code which earns me a commission for purchases made at no additional cost to you. Thank you for helping me provide you with useful information!

If you decide to go with VPSDime, I recommend getting their cheapest VPS plan for \$7/month (Don't add any of their add-ons that cost additional money (such as Varnish Configuration), they are unneeded), then configuring it using one of the aforementioned VPS management panels over SSH. If you wanted to, you could easily run multiple websites off of one single VPSDime VPS due to their generous system resource allocation. If you need additional ram over 6 GB, each additional 6 GB of ram is an additional \$7/month.

Note: I will be adding a section on how to manually configure Varnish Caching soon!

Hosting Performance Benchmarking

WP Performance Tester (Free)

<https://wordpress.org/plugins/wppercormancetester/>

Wordpress hosting benchmarking plugin

CDN/DNS

Do I need a CDN?

Not every site needs a CDN! If you are only serving traffic local to the country the server is hosted in, users are near the hosting origin server's geographic location, and do not have excessive volumes of traffic do not need CDNs. In such cases, a CDN will actually have a net negative performance impact due to the increased network latency and unnecessary network hops.

If your site has a decent amount of traffic which would heavily load the server, a CDN is absolutely something to consider to reduce hosting bandwidth usage and traffic limits (if your host imposes any). If the site is serving multiple geographic regions and not just the country of origin, a CDN is a must.

A CDN will provide performance consistency across geographic regions since they have proxy servers which host a local copy of your site on the CDNs server, which allows a visitor to access the site from a server which is much closer to their physical location. If your site is based in the US but receives traffic from outside the country of origin for example, the visitor outside would experience worse load times than the visitor who is closer to the hosting server. In this scenario a CDN is preferable.

Beleaf article

<https://beleaf.au/blog/anatomy-of-a-performant-and-sustainable-webpage/>

How Does A CDN Work?

SwitchUpCB Article

<https://switchupcb.com/blog/1-second-wordpress-website-page-speed/>

Note: Even though SwitchUpCB has a really solid breakdown of how CDN's function, absolutely do not trust his information on anything else in the article. His analysis on everything else related to Pagespeed Performance in this article is wrong.

CDN Rewrite

CDN Enabler (Free)

<https://wordpress.org/plugins/cdn-enabler/>

Enables CDN rewrite (true CDNs, not cloudflare as a reverse proxy as this functionality is unneeded when using Cloudflare).

Cloudflare (Freemium)

Cloudflare has a free tier, and is one of the most reliable CDNs out there.

Be aware that if you choose not to proxy your traffic through Cloudflare, you will also lose most(almost all really) of the benefits of Cloudflare's CDN, so you should speed test your site with the CDN active(orange cloud), and with the CDN inactive(gray cloud) both.

Cloudflare functions solely as your DNS provider if the cloud is gray. This will still result in a performance benefit if your original DNS provider was slow, however the impact will be minimal.

Cloudflare Super Page Cache (Free)

<https://wordpress.org/plugins/wp-cloudflare-page-cache/>

Enables many Cloudflare paid features for free!

Cache Ajax Requests

Cloudflare Super Page Cache (Free)

<https://wordpress.org/plugins/wp-cloudflare-page-cache/>

Cloudflare Super Page Cache can cache Ajax get requests (not post requests sadly)

<https://imgur.com/xWmKHmL>

Uncheck the box in "do not cache dynamic content" titled "Ajax Requests"

Limitations

If the Ajax requests are coming from a plugin that uses Ajax Nonces, the page/ajax will not be able to be cached.

See the developer's explanation here:

<https://wordpress.org/support/topic/how-to-correctly-cache-or-bypass-ajax-requests/>

Signed Exchanges

<https://blog.cloudflare.com/automatic-signed-exchanges-desktop-android>

Note: This functionality is available for the domains proxied through Cloudflare. Please ensure that the proxy status is set to Cloudflare (orange cloud).

App For Cloudflare

<https://appforcf.com/>

Unlock advanced Cloudflare features without being a network administrator or developer. Works with any Cloudflare plan (including Free), no APO subscription needed.

Cloudflare + NGINX

<https://blog.cloudflare.com/http-2-prioritization-with-nginx/>

Cloudflare Argo Traffic Routing (Paid)

<https://www.cloudflare.com/application-services/products/argo-smart-routing/>

“Argo delivers web traffic over the fastest links available resulting in noticeably faster web assets and improved end-user experience, resulting in faster loading times.

Cloudflare APO (Automatic Platform Optimization) For Wordpress (Paid)

<https://www.cloudflare.com/application-services/products/automatic-platform-optimization/>

Powered by the Cloudflare's Edge

“Automatic Platform Optimization serves both static and dynamic content from Cloudflare edge reducing costly round trips to the origin and server processing time. All the communication occurs between the user's device and Cloudflare edge and

Automatic Platform Optimization bypasses the cache on standard WordPress and WooCommerce cookies for authenticated users. This ensures customized content for a specific user is only visible to that user. Fonts are rehosted and served from our cache.”

Enhanced HTTP/2 Prioritization

Optimizes the order of resource delivery, independent of the browser. Greatest improvements will be experienced by visitors using Safari and Edge browsers.

Cache Reserve (Paid)

Increase your cache hit ratio and avoid unnecessary trips to the origin

Automatically store all cacheable files in Cloudflare's object storage buckets for longer periods and minimize egress costs.

Response Buffering (Paid, and Expensive)

Enable or disable buffering of responses from the origin server

(Requires the Expensive Enterprise Plan)

Rocket Loader

Do not enable Rocket Loader

Rocket Loader is almost always problematic most of the time and can often break functionality, leave it disabled. I recommend never using this feature.

Batch Install Every Free Optimization Plugin

Note: This list has not been updated in a few months and there are a significant number of new additions in the guide that will soon be included.

WPFavs (Free)

<https://wordpress.org/plugins/wpfav/>

<https://wpfav.com/>

Plugin which allows you to import batch plugin installation templates from the WP Fav's website to install a group of plugins automatically from wordpress.org/plugins. You can choose to install them all, or just choose the ones you want out of the premade list, then install them in bulk in one click.

WP Fav's List Code:

JAuOGP5BZICR5LmBsPANN9kpKHfiie

Wp Favs v1.2.1.1

If you are new to Wp favs, first go to <https://wpfavs.com> and create your first list! Or you can [browse](#) the existing Wp favs

API Key

Get Wp Favs

Quick Key

Quick Load

Wordpress.org username

Load Favorites

Paid Plugins are not included in the batch installation

Most free plugins in the guide has been included in the batch installation list above. Just enter the code in the WP Favs plugin settings, choose which plugins you want to batch install and you're good to go to start configuring them.

MISCELLANEOUS

Grafana - Observability monitoring (Free)

<https://grafana.com/>

Compose and scale observability with a free and open-source stack

Rest API Performance

<https://www.spacedmonkey.com/2023/02/03/rest-api-performance-with-jonny-harris/>

Instant Temporary Wordpress Sites

TasteWP (Freemium)

<https://tastewp.com/>

InstaWP (Freemium)

<https://instawp.com/>

Wordpress Playground (Free)

<https://wordpress.org/playground/>

Temporary Wordpress site which exists only in your browser and all data is deleted once the tab is closed.

Wordpress Playground Sandbox/Instant Dev Site (Free)

<https://wordpress.org/plugins/playground/>

With this plugin, you can:

Create a copy of your site in a private WordPress Playground instance.

Test plugins from the WordPress plugin directory without actually installing them on your site.

Your site's data is shipped directly to your web browser where it stays only as long as you keep your browser tab open. That's right! WordPress Playground runs a copy of your site directly on your device.

Admin And Site Enhancements (Free)

<https://wordpress.org/plugins/admin-site-enhancements/>

Awesome plugin which combines the features of various plugins to modify the admin dashboard, all into one! I'd personally put this one on every site as it has so many nice QOL features.

Website Sitemap and Wireframe Generator

Relume (Freemium)

<https://www.relume.io/>

Speed Logger

<https://speedlogger.io>

Speed Logger runs regular speed tests on your pages to monitor pagespeed on a day to day basis.

Official Matrix Chat for Wordpress (Free)

<https://wordpress.org/plugins/chatrx/>

An official free Matrix Chat client for Wordpress

Accessibility Scanners

Wave (Free)

<https://wave.webaim.org/>

WAVE is a suite of evaluation tools that helps authors make their web content more accessible to individuals with disabilities. WAVE can identify many accessibility and Web Content Accessibility Guideline (WCAG) errors, but also facilitates human evaluation of web content.

Editoria11y Accessibility Checker (Free)

<https://wordpress.org/plugins/editoria11y-accessibility-checker/>

Editoria11y (“editorial accessibility ally”) is a quality assurance tool built for an author’s workflow:

- It provides instant feedback. Authors do not need to remember to press a button or visit a dashboard to check their work.
- It checks in context on pages, not just within the post editor, allowing it to test content edited in widgets or theme features.
- It focuses exclusively on content issues: assisting authors at improving the things that are their responsibility.

It is meant to supplement, not replace, testing your code and visual design with developer-focused tools and testing practices.

Accessibility Checker (Free)

<https://wordpress.org/plugins/accessibility-checker/>

Accessibility Checker automatically scans your content each time you save a draft or hit publish, giving you real-time feedback on your post’s or page’s accessibility. Receive a visual overview of how your content stacks up against more than 40 different accessibility checks created to meet Web Content Accessibility Guidelines (WCAG) 2.1 success criteria.

For each error or warning flagged, Accessibility Checker lists the code that triggered the error or warning. Each item is identified as an error or warning depending upon its severity, so that you can prioritize your fixes. Thorough documentation is available to guide you on how to fix accessibility problems on your website and personalized accessibility remediation assistance and accessibility consulting is available if desired.

Kevin Leary's Tools

<https://www.kevinleary.net/blog/test-website-ada-compliance-wcag-accessibility/>

Remove Backgrounds from PNGS (Free)

<https://gemoo.com/tools/remove-background/>

Remove backgrounds from PNGs to make them transparent

Website Architecture Analysis

Built With (Free)

<https://builtwith.com/>

Great site for a quick analysis of a Website's tech stack.

Free Open Source Project Management Tool

Taiga (Free)

<https://taiga.io/>

Amazing free open source project management tool.

Load testing

Locust (Free)

<https://locust.io/>

Self-Hosted Open Source Load Testing software written in Python.

Loader.io (Freemium)

<https://loader.io>

Load testing tool with up to 100k concurrent user simulation on the generous free plan.

Yandex Tank (Free)

<https://github.com/yandex/yandex-tank>

Free load testing tool from Yandex

AI

CodeWP (Freemium)

<https://codewp.ai/>

Nifty site to create things for functions and plugins for Wordpress with AI.

ChatGPT (Freemium)

<https://chat.openai.com>

ChatGPT can produce code for you for Wordpress. The paid ChatGPT 4 is obviously going to produce better code.

Gemini (Freemium)

<https://gemini.google.com/app>

Google's Gemini AI can also write code for you. Different model than ChatGPT, can produce better results. Try them both, see which one you prefer.

Analytics

Perplexity (Freemium)

<https://www.perplexity.ai/>

Claude (Freemium)

<https://claude.ai/chats>

Meta (Freemium)

<https://www.meta.ai/>

Open Source Models

Hugging Face (Free)

<https://huggingface.co/>

LMStudio (Free)

<https://lmstudio.ai/>

GPT4ALL (Free)

<https://gpt4all.io/index.html>

LoLLMs (Free)

<https://github.com/ParisNeo/lollms>

Ollama (Free)

<https://ollama.com/>

Awesome AI (Free)

<https://github.com/re50urces/Awesome-AI>

AI mega-resource. The repository contains AI tools to do a variety of things. Great resource.

Fabric (Free)

<https://github.com/danielmiessler/fabric>

Community curated prompts

- Fabric has Patterns for all sorts of life and work activities, including:
- Extracting the most interesting parts of YouTube videos and podcasts.
- Writing an essay in your own voice with just an idea as an input.
- Summarizing opaque academic papers.
- Creating perfectly matched AI art prompts for a piece of writing.
- Rating the quality of content to see if you want to read/watch the whole thing.
- Getting summaries of long, boring content.
- Explaining code to you.
- Turning bad documentation into usable documentation.
- Creating social media posts from any content input.

And bunch more

Analytics

Fully Locally Host Your Analytics

Locally hosting your analytics js files will result in a performance improvement due to reduced network latency and roundtrip time. Always locally host your analytics JS files when possible!

Host Everything Locally (Free)

<https://github.com/Dan0sz/host-everything-locally>

Fullstory (Paid)

<https://www.fullstory.com/>

Comprehensive analytics platform. Could be useful.

Microsoft Clarity (Free)

<https://clarity.microsoft.com/>

Highly recommended! Drop-in Hotjar replacement, and 100% free.

Locally hosting analytics files will reduce network latency and roundtrip time, therefore increasing your website's performance. Always locally host analytics scripts!

Server Side Event Tracking

<https://www.kevinleary.net/blog/building-a-server-side-event-tracking-application/>

More users are using ad blockers than ever, preventing accuracy in client-side event metric tracking. Here's a great blog on how to implement server side tracking so that you get accurate metrics.

Open Source Analytics Tools (Free)

<https://posthog.com/blog/best-open-source-analytics-tools>

PostHog (Freemium)

<https://posthog.com>

Plausible (Free)

<https://plausible.io/>

Grafana (Free)

<https://grafana.com/oss/>

GrowthBook (Free)

<https://github.com/growthbook/growthbook>

OpenReplay (Free)

<https://openreplay.com/>

Apache Superset (Free)

<https://github.com/apache/superset>

Mojito A/B Testing (Free)

<https://github.com/mint-metrics/mojito>

<https://github.com/mint-metrics/mojito-js-delivery>

A modular, source-controlled split testing framework that lets you build, launch and analyze experiments via Git/CI. Mojito's JS file is just 5 kb, very well optimized.

Countly (Free)

<https://github.com/Countly/countly-server>

OpenWebAnalytics (Free)

<https://github.com/Open-Web-Analytics/Open-Web-Analytics>

- Track visitors, pageviews, e-commerce transactions, and configurable actions
- Track unlimited number of websites using a single instance of OWA Server
- First party Javascript tracker client
- Reporting Dashboard/Portal
- View and customize all reports
- Generate Heatmaps
- Generate "Domstream" session recordings
- Geolocation of visitors
- REST API for administration and data access
- Multi user reporting interface
- Extensible framework via custom modules

Open Web Analytics Wordpress Plugin (Free)

<https://wordpress.org/plugins/open-web-analytics/>

Matomo Analytics (Free)

<https://wordpress.org/plugins/matomo/>

Google Analytics replacement, full local hosting on your own server is possible!

Matomo On-Premise (Free)

<https://matomo.org/how-do-i-install-on-premise/>

Unleash A/B Testing (Free)

<https://github.com/Unleash/unleash>

Build No Code Mobile Apps (Free)

<https://thunkable.com/>

That's right! You can now build your own mobile app with zero code, entirely for free!

Headless Wordpress on JAMStack

Smashing Magazine

<https://www.smashingmagazine.com/2020/02/headless-wordpress-site-jamstack/>

Great article on how to build a headless Wordpress site with JAMStack.

Secure Datastores

OpenBao (Free)

<https://github.com/openbao/openbao>

OpenBao exists to provide a software solution to manage, store, and distribute sensitive data including secrets, certificates, and keys. The OpenBao community intends to provide this software under an OSI-approved open-source license, led by a community run under open governance principles.

Arbitrary key/value secrets can be stored in OpenBao. OpenBao encrypts these secrets prior to writing them to persistent storage, so gaining access to the raw storage isn't enough to access your secrets. OpenBao can write to disk, Consul, and more.

OpenBao can generate secrets on-demand for some systems, such as AWS or SQL databases. For example, when an application needs to access an S3 bucket, it asks OpenBao for credentials, and OpenBao will generate an AWS keypair with valid permissions on demand. After creating these dynamic secrets, OpenBao will also automatically revoke them after the lease is up.

Infrastructure As Code

OpenTofu (Free)

<https://github.com/opentofu/opentofu>

OpenTofu is an OSS fork of Terraform for building, changing, and versioning infrastructure safely and efficiently (Similar to Ansible). OpenTofu can manage existing and popular service

providers as well as custom in-house solutions. This allows a blueprint of your datacenter to be versioned and treated as you would any other code. Additionally, infrastructure can be shared and re-used.

Ansible (Free)

<https://github.com/ansible/ansible>

Ansible is a radically simple IT automation system. It handles configuration management, application deployment, cloud provisioning, ad-hoc task execution, network automation, and multi-node orchestration. Ansible makes complex changes like zero-downtime rolling updates with load balancers easy. More information on the Ansible website.

How to Pair them

OpenTofu - Provision infrastructure with metadata decoration in a programmatic repeatable way.

Ansible - Configure your newly provisioned infrastructure based on the previously created metadata decoration

Headline Analyzer Tools

Kinsta

<https://kinsta.com/blog/headline-analyzer/>

Fix Hanging Words in WordPress

<https://www.kevinleary.net/blog/fix-hanging-words-wordpress/>

Awesome Open Source (Free)

<https://awesomeopensource.com/>

Awesome WP Speed Up (Free)

Amazing page, has lists of most of the performance plugins out there, tons and tons more that are not listed in the guide:

<https://awesomeopensource.com/project/lukecav/awesome-wp-speed-up>

Most of these plugins won't be of any use to you as they are just poorer implementations of the features WP-Rocket, Flyingpress and Perfmatters contain here, but there are a few diamonds in the rough.

Luke Cavs Github Repos (Free)

<https://github.com/lukecav?tab=repositories>

Some of this guide was sourced from his repos, he also has a lots of stuff not in this guide that may be applicable to your site. He has a bunch of repos, each one is its own beast. Highly recommend browsing through it, tons of goodies. This one can be a real deep dive, prepare to get lost in how much stuff he has.

The repos he maintains also contain resources for many popular plugins. A few examples would be Awesome Elementor or Awesome Gravity Forms, which are mega wiki pages that contain almost every resource and addon available for both plugins. He has over 100 repos.

Purifycss (Free)

<https://purifycss.online/>

Critical CSS Generator

IcoMoon (Free)

<https://icomoon.io/app>

SVG Icons

Admin and Site Enhancements (Free and Pro)

<https://wordpress.org/plugins/admin-site-enhancements/>

Collection of quality of life tweaks for the Wordpress admin.

Onlinemediamasters

<https://onlinemediamasters.com/>

Amazing blog with a fantastic amount of optimization articles and guides.

Tools

<https://onlinemediamasters.com/tools/>

Collection of useful Wordpress tools.

WPSpeedMatters

<https://wpspeedmatters.com/>

Very useful blog, has some things not covered by this guide

Kinsta's blog

<https://kinsta.com/blog>

Useful info, techniques and other things not covered here. I implemented some of the suggestions from their articles that I have not listed in this guide. Also a fantastic resource for many other things related to web development, highly recommended reading.

Performance Checklist Github

<https://github.com/flowforfrank/performance-checklist>

Perishable Press

<https://perishablepress.com/>

Great blog. Interesting plugins, books which look interesting to explore in the future, lots of great articles. Solid resource for optimization.

Dynamic Link Insertion Via Javascript To Improve SEO

<https://perishablepress.com/insert-links-external-javascript/>

How to Modify GET and POST Requests with WordPress

<https://perishablepress.com/modify-get-post-requests-wordpress/>

expose_php, Easter Eggs, and .htaccess

<https://perishablepress.com/expose-php/>

How to Fix _blank Target Vulnerability

<https://perishablepress.com/fix-blank-target-vulnerability/>

Huge Collection of Code Snippets: HTAccess, PHP, WordPress, jQuery, HTML, CSS

<https://perishablepress.com/code-snippets/>

How to Modify HTTP Headers in the WordPress Admin Area

<https://perishablepress.com/modify-http-headers-wordpress-admin-area/>

Free Code Camp Guide

<https://www.freecodecamp.org/news/the-ultimate-guide-to-high-performance-wordpress/>

Inline SVGs (Free)

<https://yoksel.github.io/url-encoder/>

Fix plugin conflicts with the Elementor editor

Editor Cleanup For Elementor (Free)

<https://wordpress.org/plugins/editor-cleanup-for-elementor/>

Editor Cleanup For Elementor is an add-on of Freesoul Deactivate Plugins to clean up the editor of Elementor.

It will not only clean up the assets of other plugins, their PHP code will not run either.

Favicon Generator

Favicomatic (Free)

<https://favicomatic.com/>

Generate favicons

Nelio A/B Testing (Freemium)

<https://wordpress.org/plugins/nelio-ab-testing/>

A/B Testing plugin

OoohBoi Steroids for Elementor (Free)

<https://wordpress.org/plugins/ooohboi-steroids-for-elementor/>

An awesome set of tools, options and settings that expand Elementor defaults. Instead of creating new Elementor Widgets, these act like an upgrade of existing options or the self-standing panels thing.

Local WP Development

Dev Kinsta (Free)

If you're interested in developing your Wordpress site on a local server instead of a staging server, I recommend using Dev Kinsta: <https://kinsta.com/devkinsta/>

LocalWP (Free)

The other option for local Wordpress development is LocalWP: <https://www.localwp.com>

Temporary Wordpress Hosting

InstaWP (Freemium)

<https://instawp.com>

InstaWP is great for temporary wordpress hosting. Free tier sites last ~1 week. If you want to continue working on that temp site, create a backup, erase the existing site, create a new temporary site, restore the backup. You can do this ad infinitum (forever). Somewhat tedious, but free! They also have paid plans with more features.

Great if you need space to work on a demo for a client or can't afford another hosting plan but need somewhere to work which is accessible over the internet (as opposed to local development).

TasteWP (Freemium)

<https://tastewp.com/>

Free instant temporary Wordpress sites

Uptime Monitoring (Free)

Hetrix Tools (Free)

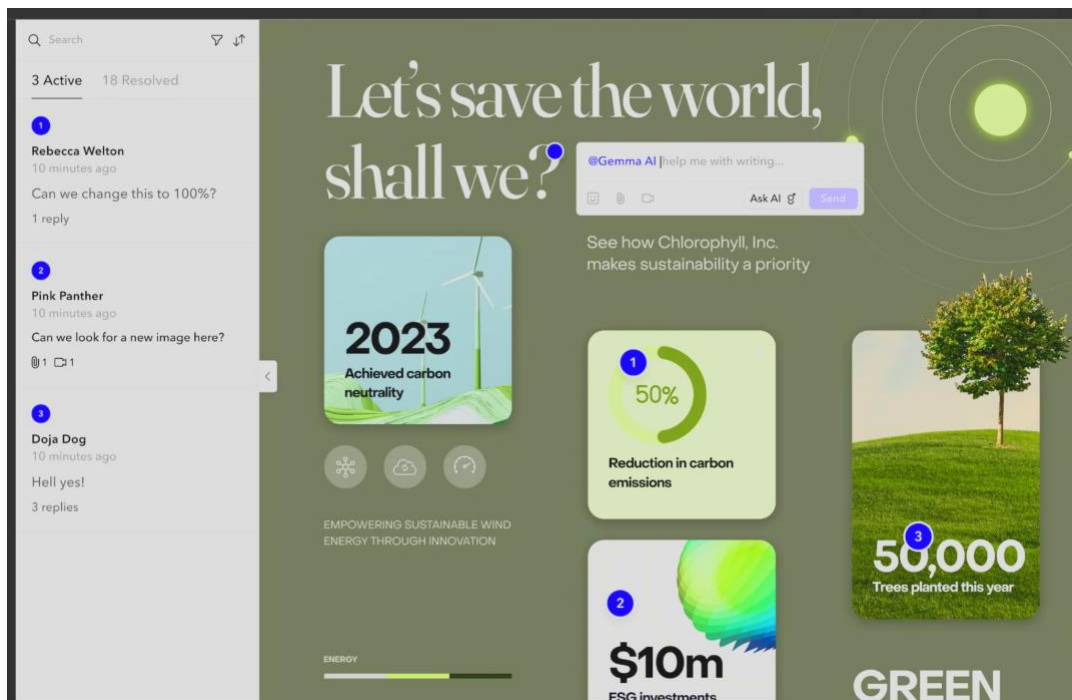
<https://hetrixtools.com/>

Free uptime and blacklist monitoring service.

Markup.io (Freemium)

<https://www.markup.io/>

Markup is a great tool for collaborating with clients and your designers, it lets people leave comments and feedback about your page with a live updated UI.



Automate tedious tasks

Zapier (Freemium)

<https://zapier.com/>

Great automation SAAS

Make (Freemium)

<https://www.make.com/en>

Comparable to Zapier.

Hexomatic (Paid)

<https://hexomatic.com/>

Webpage Rip (Free)

<https://webpagerip.com/>

Rip portions of web pages or the entire page.

SEO

Learningseo.io (Free)

<https://learningseo.io/>

The de facto best resource to learn about SEO on the internet. Has a full roadmap covering every SEO subject under the sun.

Annie Cushing's Tools (Free)

https://docs.google.com/spreadsheets/d/1Y_ozPDx9PJY8bM7sVSOrc4bT_8UGKG7qhtbUkW4HbU/edit#gid=0

Great collection of tools and lots of useful resources.

Buzzstream (Freemium)

<https://www.buzzstream.com/>

Say goodbye to tedious list building and ineffective emails with the world's best outreach CRM.

Search Engine Journal (Free)

<https://www.searchenginejournal.com/>

Great SEO Blog

Prowly (Paid)

<https://prowly.com/>

PR Outreach

Buy me a Coffee

<https://ko-fi.com/jazir5>

Please consider buying me a coffee! Your support helps me to continue providing you with useful information and to keep updating the guide with an ever growing amount of content!

Contact Me

Email

Send me an email at techtutormm@gmail.com.

Voicemail or Text

323-739-8552

This is a VOIP line, not my main phone number. If you leave me a voicemail or text, I will certainly call back or send a text back when I get a chance!

Discord

@jazir5

I look forward to hearing from you!