

Speed Up Your Website

8 Practical Tips that Work



Table of Contents

Click or tap to jump to any section;

Chapter 1
3 Choosing a Fast
Hosting Provider

Chapter 3
24 Selecting the Right Plugins and
Theme for Optimal Performance

Chapter 5
44 Minifying Your Website's
CSS and Javascript

Chapter 7
59 Integrating a Content
Delivery Network

Chapter 2
14 Setting Up Our
Test Website

Chapter 4
34 Optimizing Your
Images

Chapter 6
51 Enabling Browser
Caching

Chapter 8
70 Making Sure Your
Website Remains Fast

Launch Your Own Website With Unlimited Hosting

Use **HOSTINGERTUTORIALS**
discount code and get an additional
10% OFF for any web hosting plan.

[Use Code](#)



Introduction

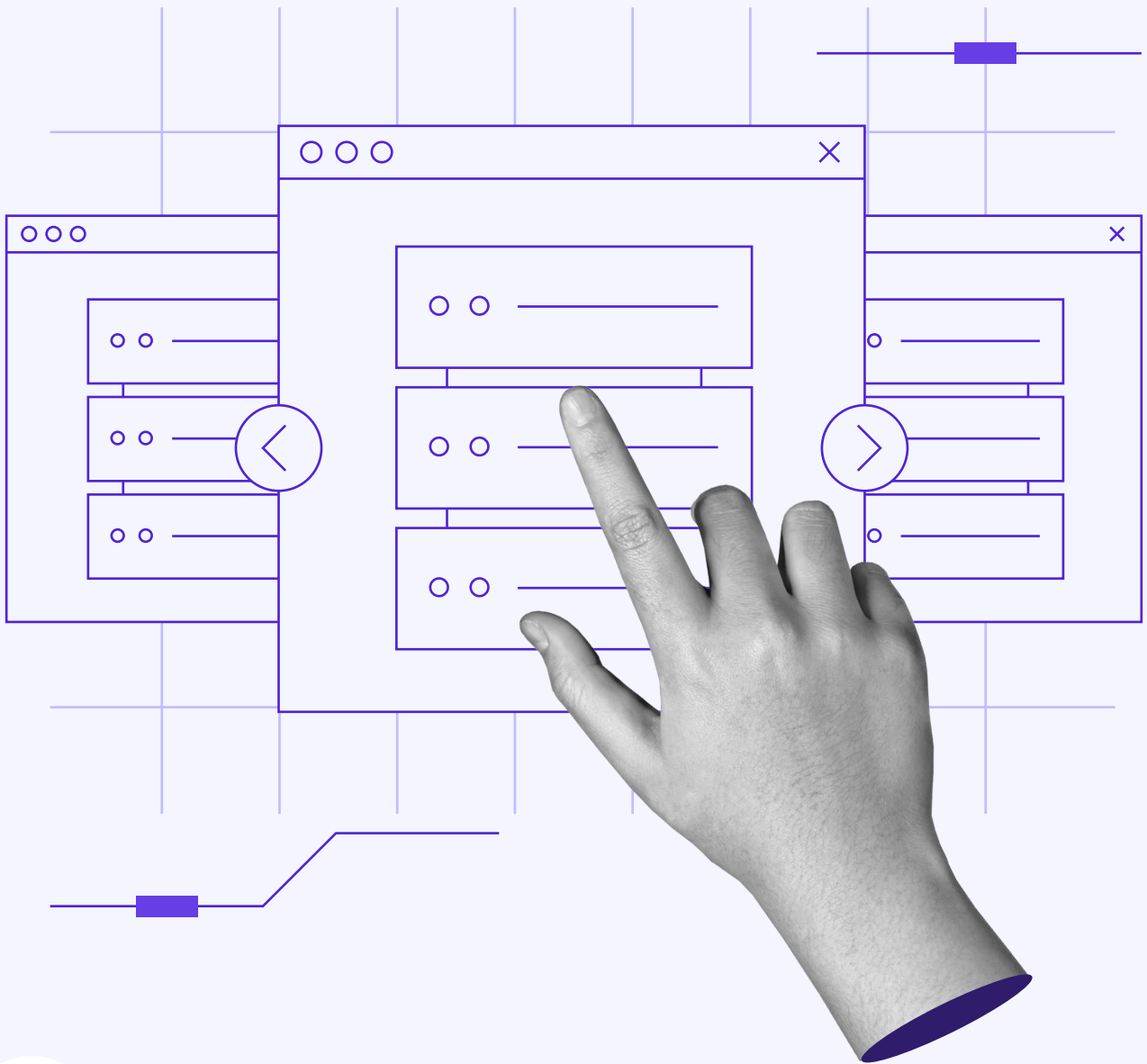
It's easy to overlook website performance, especially when you're just starting out. However, page loading times are vital to [search engine optimization](#) (SEO) as well as for providing a high-quality user experience (UX) to your visitors.

Fortunately, there are many ways to speed up your site. Even if you have little technical experience, there are plenty of beginner-friendly strategies you can apply to decrease page loading times.

Throughout this eBook, we'll share tried-and-true best practices for speeding up your WordPress site. To that end, we're going to create a new website and walk you through the performance optimization process step by step, including:

1. Selecting a fast hosting provider.
2. Choosing the right plugins and theme for optimal performance.
3. Compressing and optimizing images.
4. Minifying CSS and JavaScript files.
5. Enabling browser caching.
6. Integrating a Content Delivery Network (CDN).
7. Ensuring long-term website speed.

We have a lot of ground to cover, so let's jump right in.



1

Choosing a Fast Hosting Provider



[Click here to return to the table of contents](#)

Your choice of hosting provider plays a critical role in your website's performance. Some are more expensive, others offer different features, and a few are significantly faster.

Website performance boils down to two factors – server and website optimization. If you choose the right web host and plan, you'll easily take care of that first factor. It sets the right groundwork for your website.

In this chapter, we'll teach you how to pick a web host to provide a strong foundation for your site's performance. We'll go over what to look for in a provider, how to choose the right plan, and how to sign up at Hostinger. Let's get to work.

➤ What to Look for in a Hosting Provider

While there are a lot of factors that set web hosts apart, the most apparent one is [pricing](#). Some companies offer plans that cost hundreds of dollars per month, while others charge less than \$5.

However, price alone doesn't determine the quality of the service you'll receive. When it comes to selecting the right web hosting provider, there are several factors to pay attention to beyond your budget, including:

Scalability. As the goal is to see your website grow, you'd want to opt for a web host that offers a broad range of hosting plans.

Excellent support. 24/7 availability and multiple communication channels, such as phone, email, chat, and a ticket system, are strong indicators of quality support.

Tech stack. This includes server locations, content delivery network (CDN), and web servers.

Additional features. Automatic backups, WordPress auto-installer, and advanced security measures will make your setup and administration process easier.

The tech stack factor is particularly important in gauging a hosting provider's performance without purchasing a plan. Looking at reviews can only inform you so much, so we recommend paying attention to what technology your web host uses to support your website.

Hostinger, for example, has [data centers around the world](#). You can pick the one closest to your audience to reduce loading times for most users. Additionally, we use LiteSpeed web servers and offer an in-house CDN solution to accelerate your website's performance.

In short, we take care of server optimization to give you a leg up from the get-go. All you have to focus on is following WordPress performance best practices, which is what this eBook is all about.

➤ How to Choose the Right Hosting Plan for Your Website

As mentioned before, a quality web hosting provider will give you plenty of plan options. However, you still need to know which one fits your needs best.

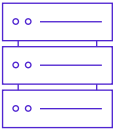
For the vast majority of websites, these are the [four hosting types](#) you should focus on:



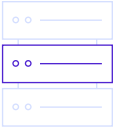
Managed WordPress hosting. This hosting type is optimized to meet WordPress' performance needs. It has a lot of technical similarities with shared hosting but with additional features to optimize WordPress websites.



Cloud hosting. This type of hosting combines the best features of both VPS and shared hosting plans. You benefit from dedicated resources, which are the main advantage of a VPS service while leaving server uptime, maintenance, and monitoring for your web host. It's a great choice for freelancers or agencies needing extra resources without worrying about technical details.



Shared hosting. This hosting type is the cheapest since you share server resources with other users. For this reason, it's mostly popular for hosting small to medium-sized websites with low traffic volume.



Virtual Private Servers (VPS). With a VPS, you get a set amount of dedicated resources. Depending on how much power you need, VPS can be exceedingly cheap or incredibly expensive.

If you are looking for a versatile hosting option without breaking the bank, we recommend going with cloud hosting. Overall, cloud plans provide plenty of resources and performance gains while leaving the technical aspects of running a server to your web host, freeing your time for other tasks.

Now that you've chosen the right hosting plan, let's sign up for a Hostinger account.

➤ Signing Up for a Hostinger Account (In 3 Steps)

For this tutorial, we'll use one of our own Hostinger plans. However, the steps remain the same regardless of the web host you choose.

Launch Your Own Website With Unlimited Hosting

Use **HOSTINGERTUTORIALS**
discount code and get an additional
10% OFF for any web hosting plan.

Use Code



1. Select and Purchase a Hosting Plan

Signing up for web hosting is pretty straightforward. With [Hostinger](#), you only need to identify the plan you want and click on the **Add to cart** button. For our test website, we're going to use the **Cloud Startup Hosting plan**.

You'll need to decide the term you want to sign up for. Generally, you will get the best deal by opting for a longer contract. Below, we've opted for a four-year commitment to maximize our savings.

All-in-one solution

- ✓ **Dedicated IP & Resources** ?
- ✓ **Unmetered Traffic** (Unlimited GB) ?
- ✓ **24/7 Customer Support** ?

Security

- ✓ **Unlimited Free SSL** ?
- ✓ **99.9% Uptime Guarantee** ?
- ✓ **Daily Backups** ?

WordPress

- ✓ **Managed WordPress** ?
- ✓ **WordPress Acceleration** ?
- ✓ **Object Cache for WordPress** ?

1. Choose a period

1 MONTH	12 MONTHS	24 MONTHS	48 MONTHS
<div>○</div>	<div>○</div>	<div>○</div>	<div>●</div>
	SAVE \$180.00	SAVE \$372.00	SAVE \$768.00
			+3 months FREE
\$24.99	\$24.99 \$9.99	\$24.99 \$9.49	\$24.99 \$8.99
USD / month	USD / month	USD / month	USD / month
Plan renews at \$24.99/month on 05/02/2024	Plan renews at \$23.99/month on 05/01/2025	Plan renews at \$22.99/month on 05/01/2026	Plan renews at \$19.99/month on 05/04/2028
	Free domain for 1st year ?	Free domain for 1st year ?	Free domain for 1st year ?

After reviewing your order, scroll down to create an account and select your payment method.

2. Create your account

Already have an account? [Log in](#)


OR


Facebook


Google


We will not share your information or send spam.


3. Select payment

☒ Credit Card 

☐ PayPal 

☐ Google Pay 


☐ AliPay 

☐ Coingate 

Cloud Startup – 48 Months Plan

✓ Daily Backup

✓ 3 Extra Months

✓ Domain Name 

✓ Domain WHOIS Privacy Protection

~~\$1,199.52~~ **\$431.52**


~~\$100.32~~ **\$0.00**

~~\$74.97~~ **\$0.00**


~~\$9.99~~ **\$0.00**

\$0.00

Country

United States 

State

Washington 

City


Street Address

ZIP code

Use this street format: 1234 Main Avenue

Plan Discount – 64%

–\$768.00

Taxes & Fees 

\$38.84

Total

~~\$1,307.48~~ **\$470.36**

[Have a coupon code?](#)


Name on card


0000 0000 0000 0000

MM / YY

CVC code

Submit Secure Payment

 30-Day Money-Back Guarantee


 Encrypted and Secure Payments

Once you've entered all the required information, click the **Submit Secure Payment** button.

2. Log in to Your New Hosting Account

Accessing your new hosting account is simple. Return to Hostinger's homepage and click on the **Log in** button on the top right corner of the screen. Clicking on it will lead you to a login page where you can enter the credentials you chose during Step 1.

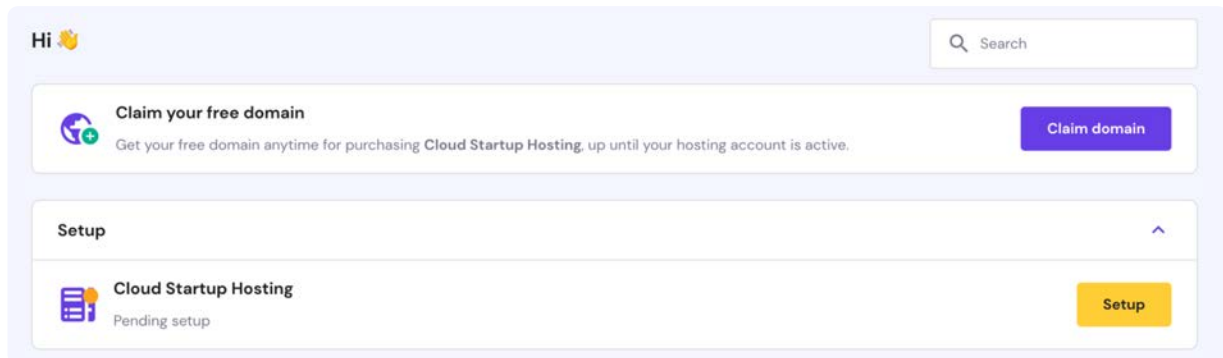
If the login process is successful, the page will redirect you to the hosting control panel, where you can view all the details about your current services. For now, let's finish configuring your hosting account.

 HOSTINGER

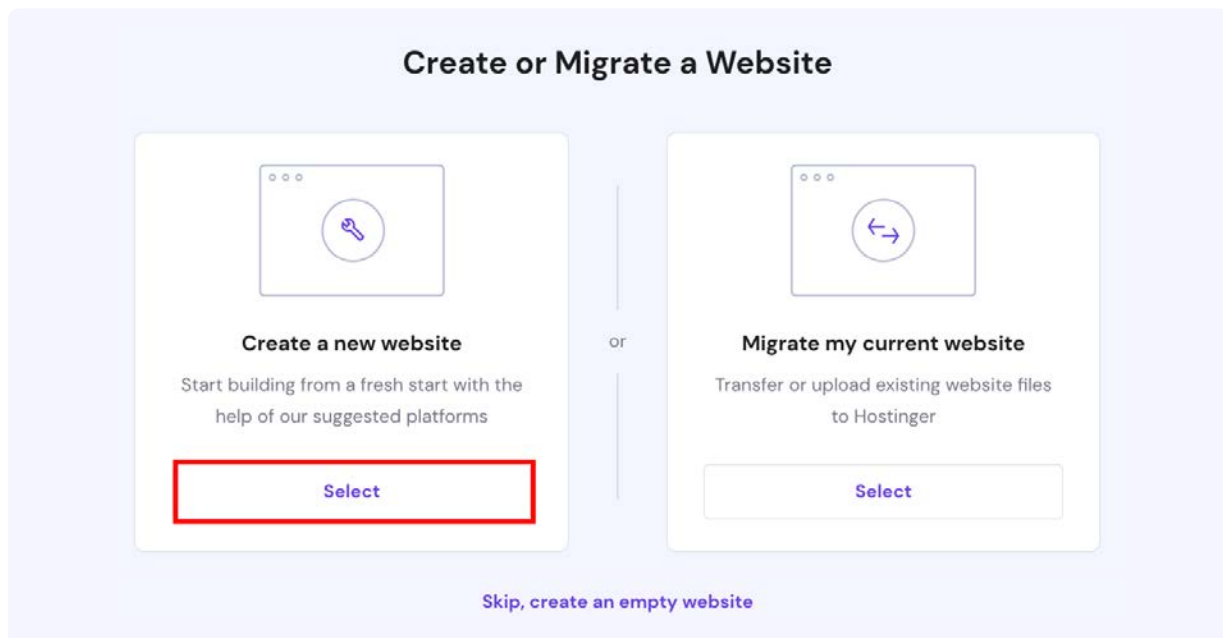
Three. Two. Online

3. Install WordPress on Your Hosting Plan

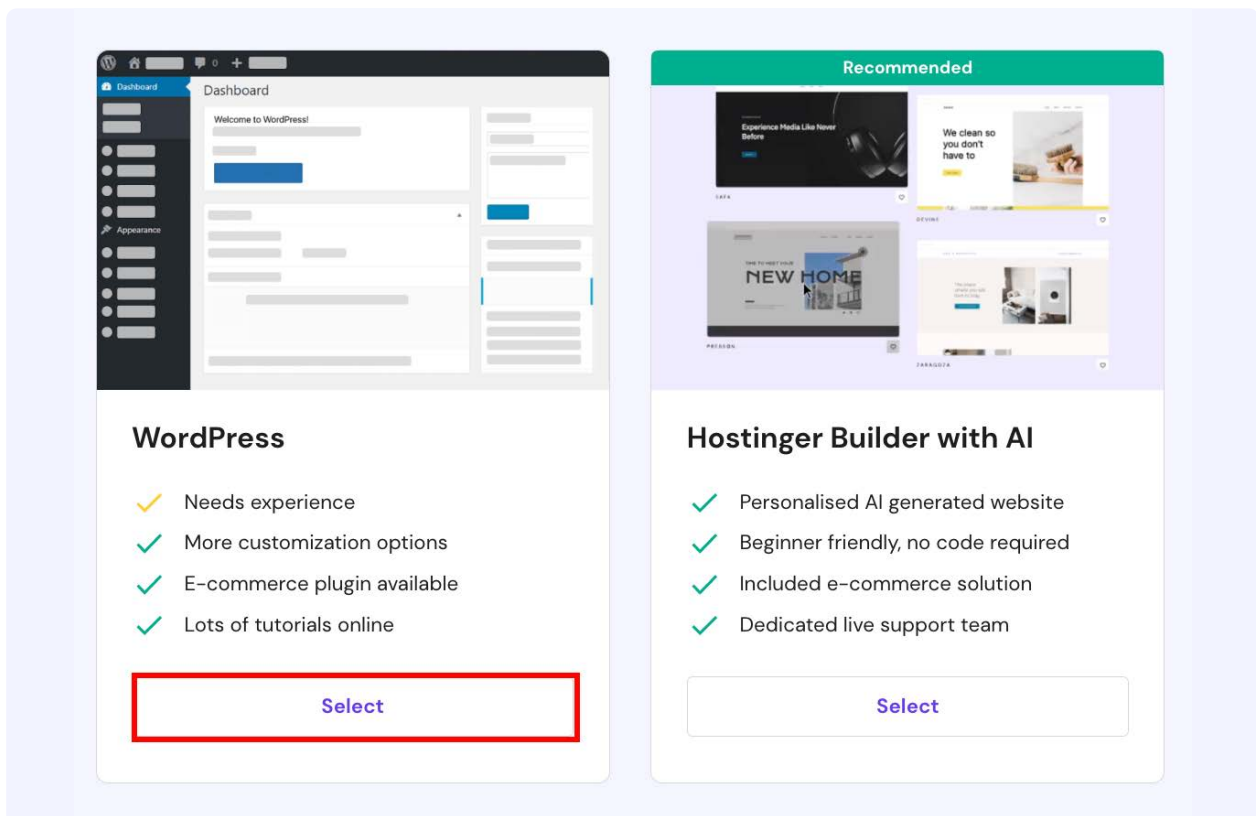
The front page of your hosting control panel will provide an overview of your plan and other features. Any elements that require further attention are indicated by a **Setup** button like with our **Cloud Startup Hosting plan** below:



Once you click the **Setup** button, answer a few questions, and you'll be guided through the configuration steps. First, decide whether to build or migrate your site. We'll be taking the first route.



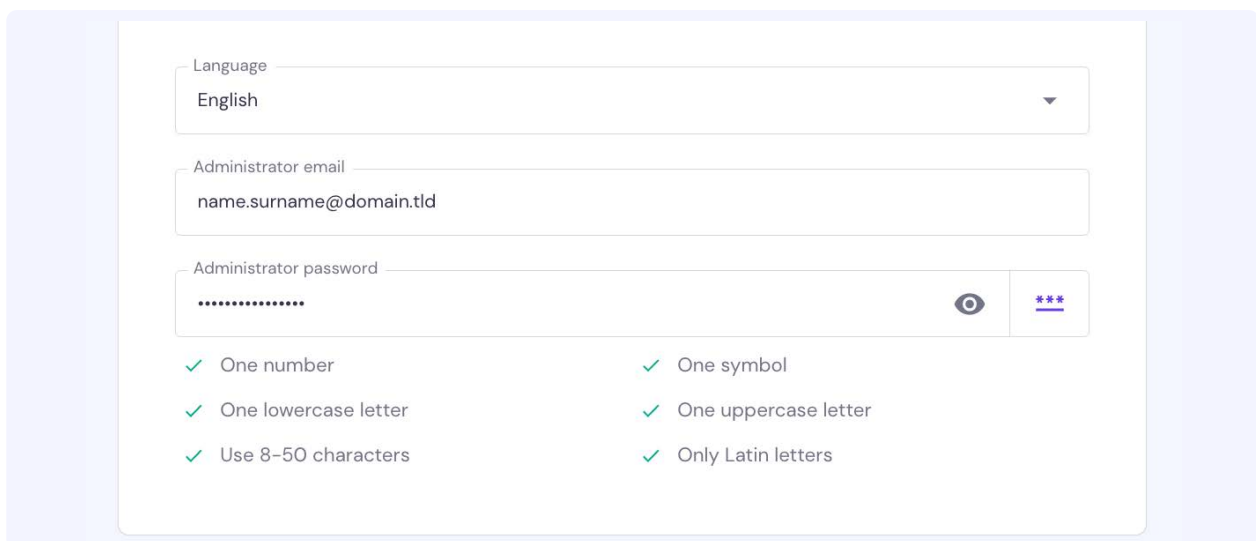
Now, you need to select a platform. In our case, we'll go with WordPress.



The image shows a comparison between two website building options. On the left is the WordPress dashboard, and on the right is the Hostinger Builder with AI interface. Below each interface is a list of features and a 'Select' button.

WordPress	Hostinger Builder with AI
<ul style="list-style-type: none">Needs experienceMore customization optionsE-commerce plugin availableLots of tutorials online	<ul style="list-style-type: none">Personalised AI generated websiteBeginner friendly, no code requiredIncluded e-commerce solutionDedicated live support team
Select	Select

Next, you'll need to provide the login details for your WordPress account. Keep them somewhere safe, as you will need the same login details to access your WordPress admin area later.



The image shows the WordPress login details form. It includes fields for Language, Administrator email, and Administrator password. Below the password field are checkboxes for password requirements.

Language: English


Administrator email: name.surname@domain.tld


Administrator password: [masked]


Requirements:


- One number
- One lowercase letter
- Use 8-50 characters
- One symbol
- One uppercase letter
- Only Latin letters


You can have certain plugins pre-installed during this setup process. Check the ones you need and click **Continue** or do it later by opting **Skip, I will manage plugins later**.

☐  **WooCommerce**
For eCommerce websites. Includes secure payments, configurable shipping options, and more.

☐  **MonsterInsights**
Track your website's statistics and view how people interact with your site

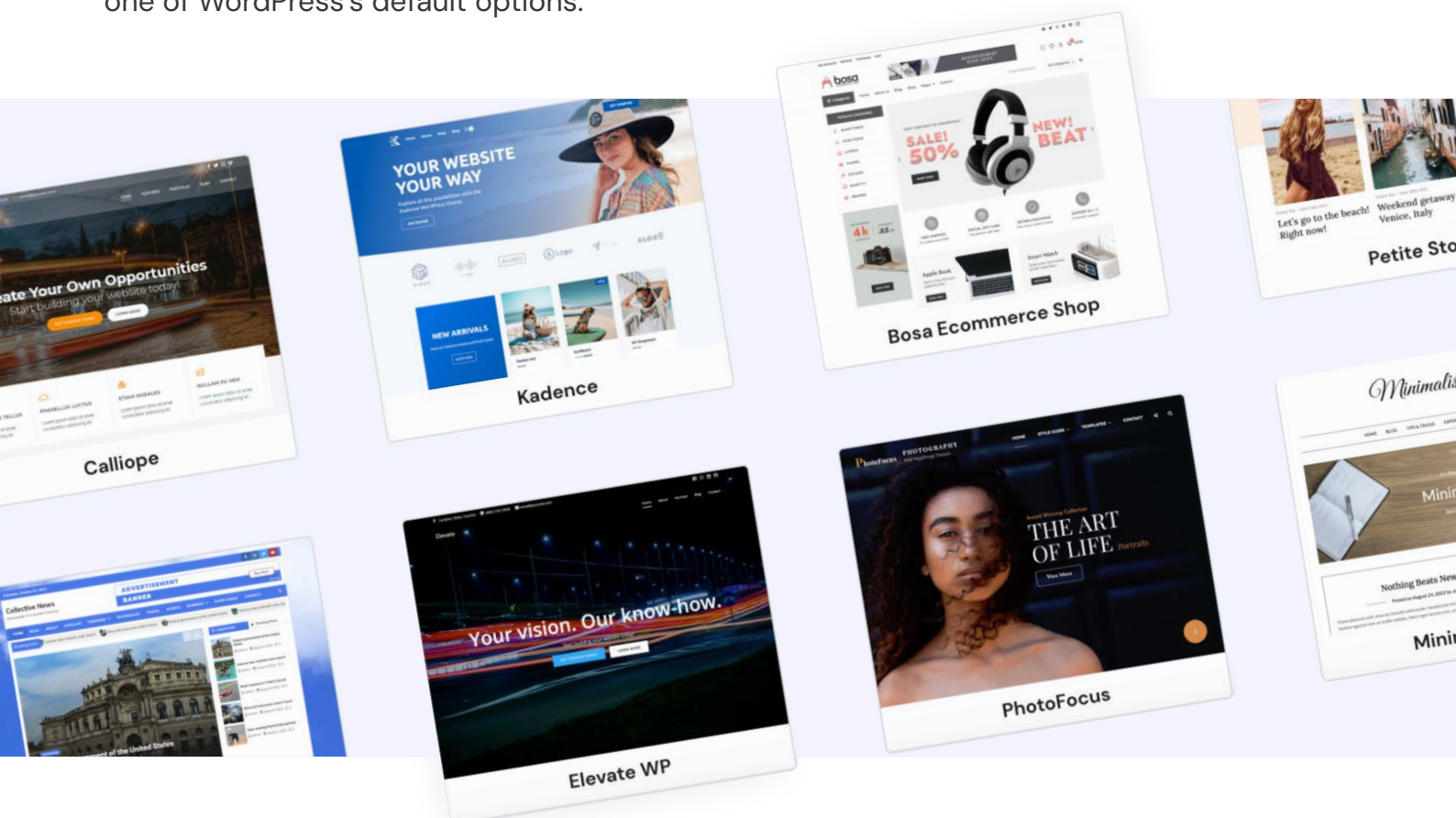
☐  **OptinMonster**
Marketing plugin that assists in getting more email subscribers and grow your business

☐  **All in one SEO**
For optimizing SEO. Helps your website to be visible, gain traffic and clients.

☐  **Google Site kit**
Discover insights about how people find and use your website

[Continue](#) [Skip, I will manage plugins later](#)

Next, you can set up a new theme for your WordPress site. We'll skip this step and start with one of WordPress's default options.



Leverage our AI content generator to populate your new website with unique content. Enter your brand name and description to their respective fields. The tool will generate content pieces like the banner copy, the About page text, and blog posts.

Automate Your Text Creation with AI

Provide your brand name and business details, and let ChatGPT handle the rest. We will generate a few content pieces for your website.

Brand name

Description

Enter at least 10 characters

e.g. We are a sustainable farming company based in the beautiful city of Newark, NJ. We have a passion for quality food and love for sustainable practices...


Continue

Skip, I will write content later

Hostinger will ask whether you want to use the free domain name you got with the **Cloud Startup Hosting plan**, buy one, or use an existing domain. We will claim our free domain name.

Name Your Website


FREE



Claim a Free Domain

Get a new domain name for your project for free


Select



Buy a Domain

Purchase a new domain name for your project

Select

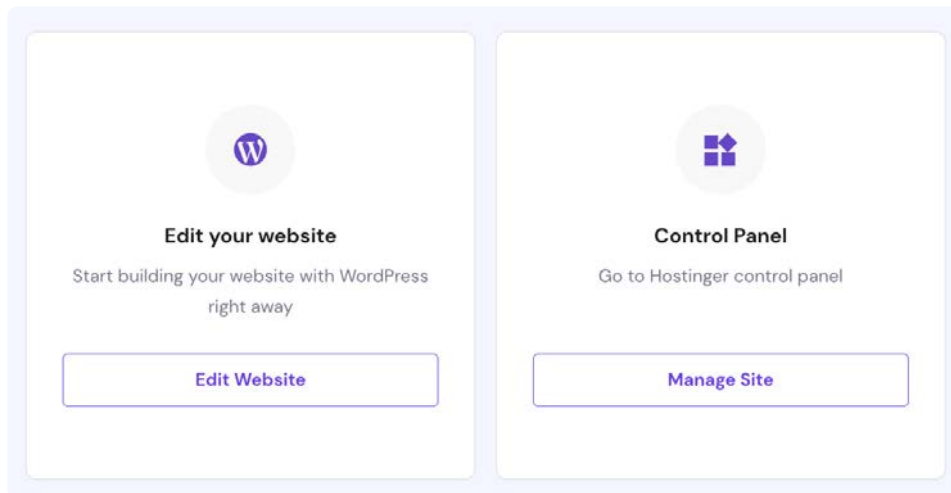


Use an Existing Domain

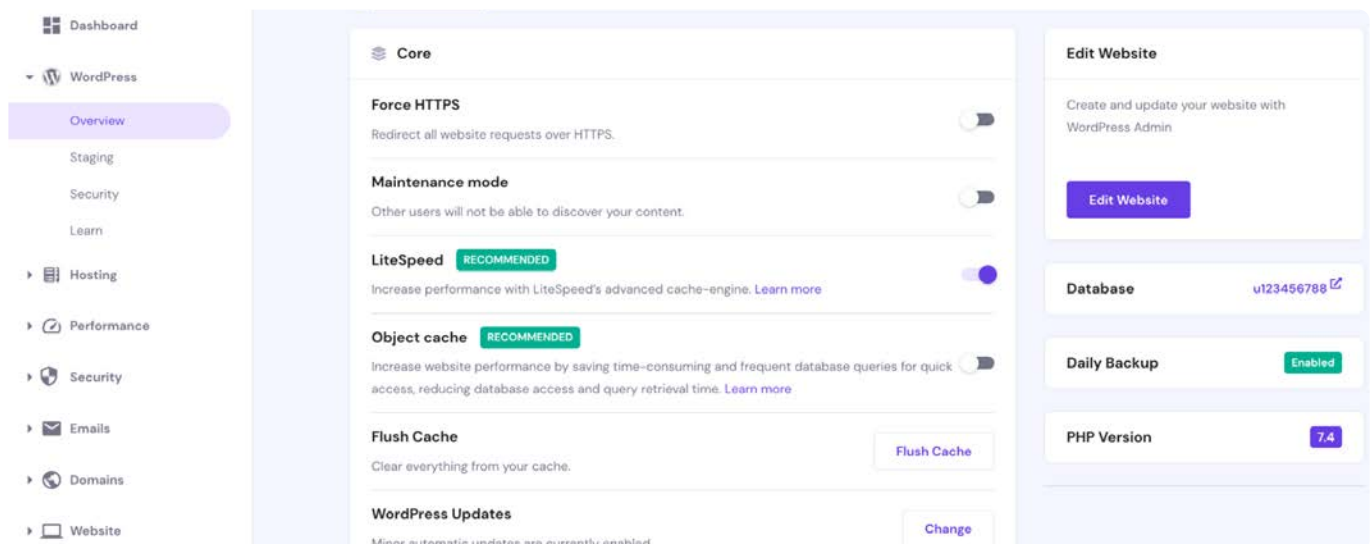
Use a domain previously purchased at another provider

Select

It will take a minute or two to install WordPress in the background. Once the installation is complete, you can start building your website or navigate to the control panel. We'll go with the latter.



Next, select the **Manage** button next to your new website. You'll see a quick overview of your hosting account on the left side. Select **WordPress** → **Overview** to access the dashboard, where you can control various features such as your PHP version, enable and disable LiteSpeed Cache, and force HTTPS.



To access the WordPress admin area, click the **Edit Website** button to the right of your screen.



2

Setting Up Our Test Website



[Click here to return to the table of contents](#)

If you choose the right web host, a website running on a fresh WordPress installation should take less than two seconds for pages to load, both on the front-end and the admin dashboard. However, loading usually takes longer as you start adding content to the site.

The more complex and populated your pages are, the longer they'll take to load. Understanding the impact your design and content choices can have on your website's performance is essential if you want to keep it running smoothly.

In this chapter, we will show how a new WordPress website feels in terms of performance. Then, we'll set up test content to see the impact various optimization techniques we explore in this eBook have on page speed.

► Why We're Using a Test Website to Demonstrate WordPress Performance Optimization

You can follow many online guides to [speed up](#) your WordPress website. However, they rarely provide a baseline to help you understand the impact of the practices outlined.

This eBook will show you what affects your website's loading times and how much of a difference these best practices can make.

To do that, we'll set up a fresh WordPress installation on the hosting plan purchased and configured at the end of Chapter 1. We'll use this site to provide a baseline of how the platform runs on a performance-oriented server before it has been customized.

Once we have those statistics, we'll incorporate content and add-ons throughout several chapters to mimic a real-life site. Then we'll run the same tests to show how those elements affect site speed.

Ideally, a well-optimized website should load in under one second. Anything between one and three seconds is decent. If it takes longer than that, your site's performance may start harming the user experience.

► How We Set Up Our Test Website (In 4 Steps)

Throughout the rest of this book, we'll be using a single website to run all our tests. The goal is to show you how performance can shift depending on the type of changes you make to WordPress.

1. Add a Fresh WordPress Installation to Our Hosting Account

In the previous chapter, we showed you how to sign up for and configure a Hostinger Cloud Startup plan. We also set up WordPress on our account.

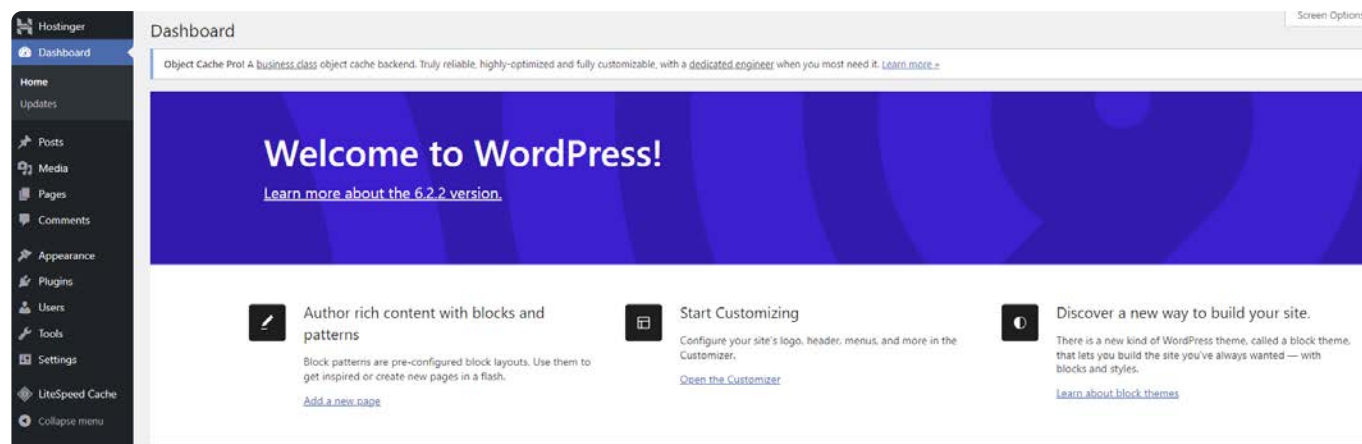
The WordPress installation process varies depending on your chosen hosting provider and plan. Overall, there are three approaches you can take:

Use an auto-installer. Many hosting companies let you set up WordPress using an automatic installer. This approach is ideal for beginners, as you only need to configure a few settings and your host will take care of the rest.

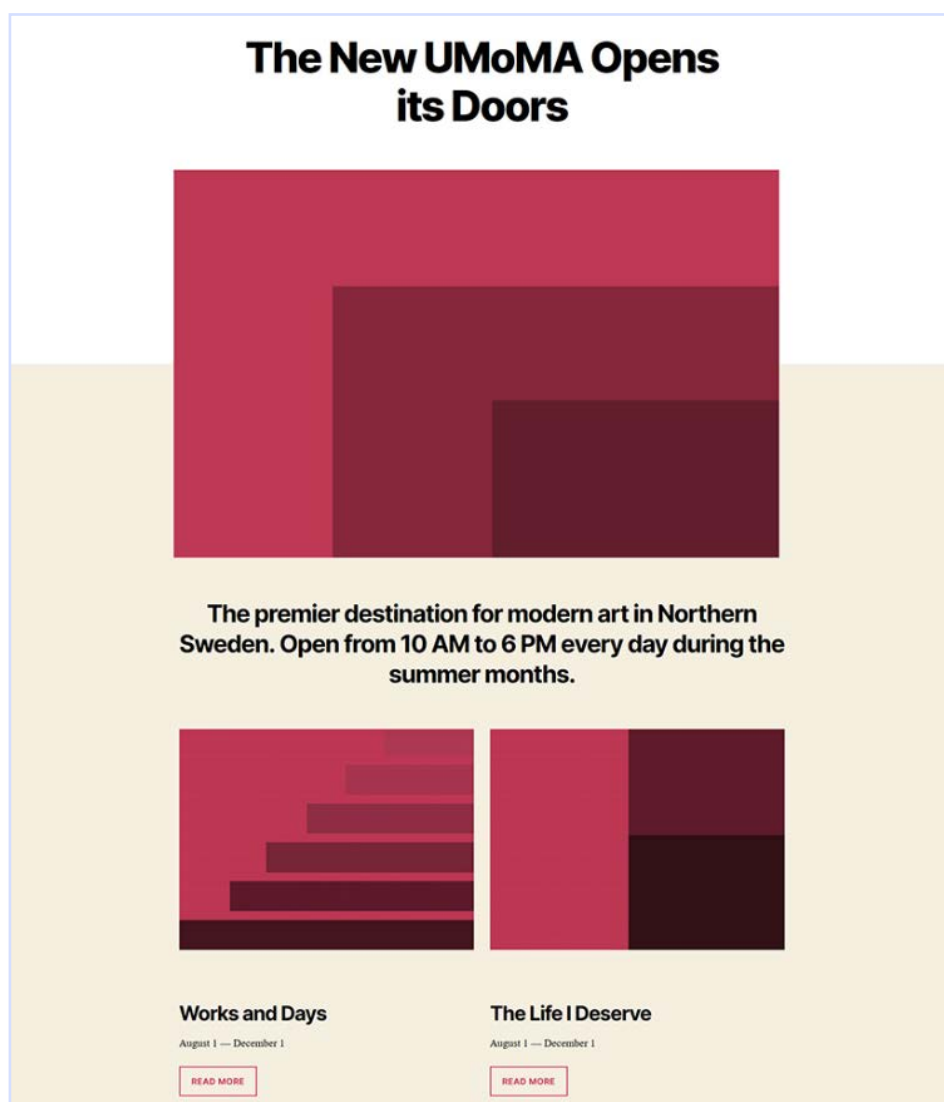
Set up WordPress manually. This method involves downloading WordPress files from the official website, uploading them to your server, and setting up the site using the CMS' built-in installer. The process requires some technical know-how.

Install WordPress through the command line. If you're using a VPS hosting plan or have SSH access, you can use the [WordPress command-line interface](#) (WP-CLI) to install the CMS. This method is a bit more advanced and utilizes multiple commands.

Regardless of the method you use, we've got you covered. We have a tutorial explaining [all three approaches](#) outlined above. Once WordPress is set up on your server, you can access your admin dashboard at <https://yourdomain.com/wp-admin>.



If you go to your website's front-end, you should see a barebones page or some dummy content. Here's one of the sample pages that come with the default **Twenty Twenty WordPress theme**:



At this point, our WordPress site is ready for us to customize it. But before we get to work, let's run a few tests.

2. Test Our Website's Baseline Performance

To run the performance tests for our barebones website, we're going to use two different platforms:

Pingdom

With this service, we can test a page's loading times from different server locations across the world.

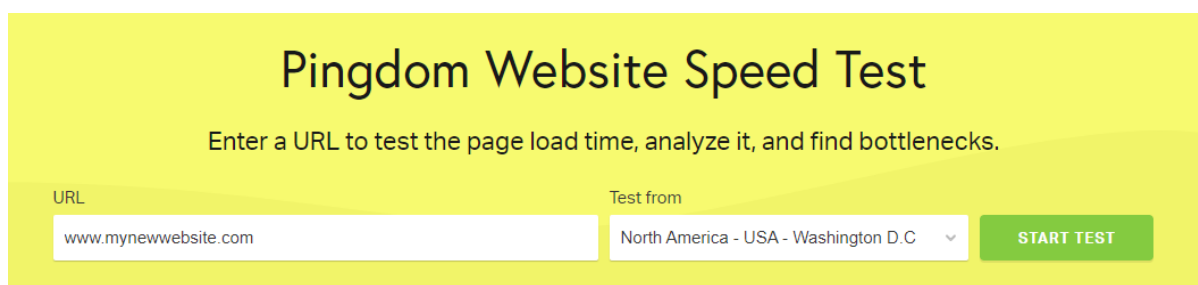
PageSpeed Insights

This Google tool gives you a performance score and actionable suggestions on how to improve the website.

Keep in mind that we're going to return to both of these platforms throughout the entirety of this eBook. Together, they paint a fairly comprehensive picture of a website's performance, so we'll rely on them to measure the impact of our optimizations.

➤ Performance Tests With Pingdom

Let's start with Pingdom. As soon as you visit its homepage, it'll ask you for the URL to analyze and the server location you want to use. For this test, we'll focus on our site's default homepage:



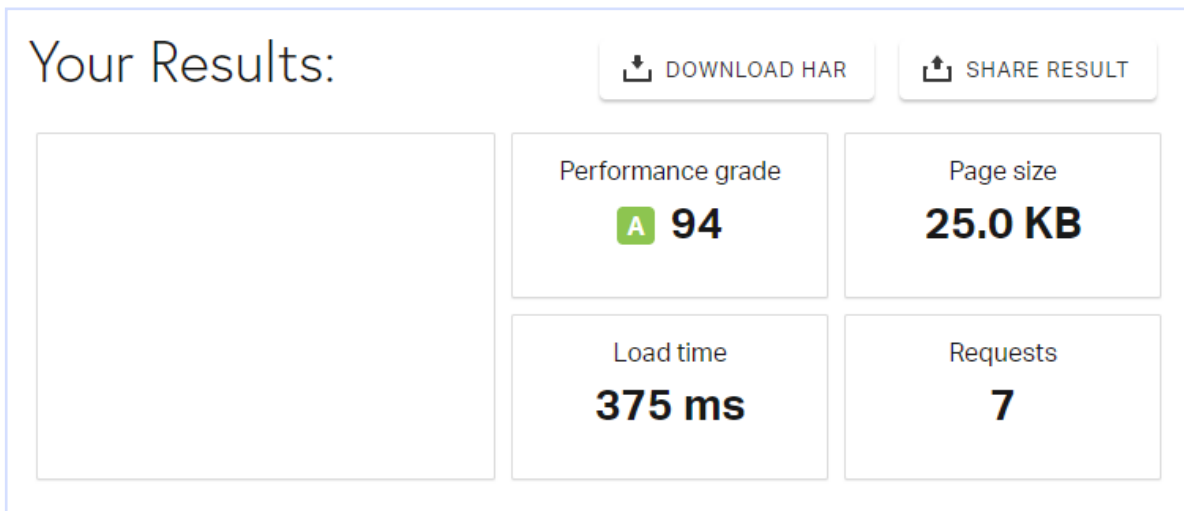
Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL:

Test from: ▼

The test itself should take a few seconds. Once it's done, a performance report will appear:



Our initial result is great. The page loads in less than half a second, getting a near-perfect performance grade. The issue, however, is the page size. In real life, you'll almost never come across a modern web page that's only 25 KB unless it's 100% text.

Moreover, it's important to understand we're talking about a brand new website with no frills. As soon as you start adding elements, the page size, number of requests, and overall loading times will increase.

Here are the results of our Pingdom tests from five different server locations, including the results above:

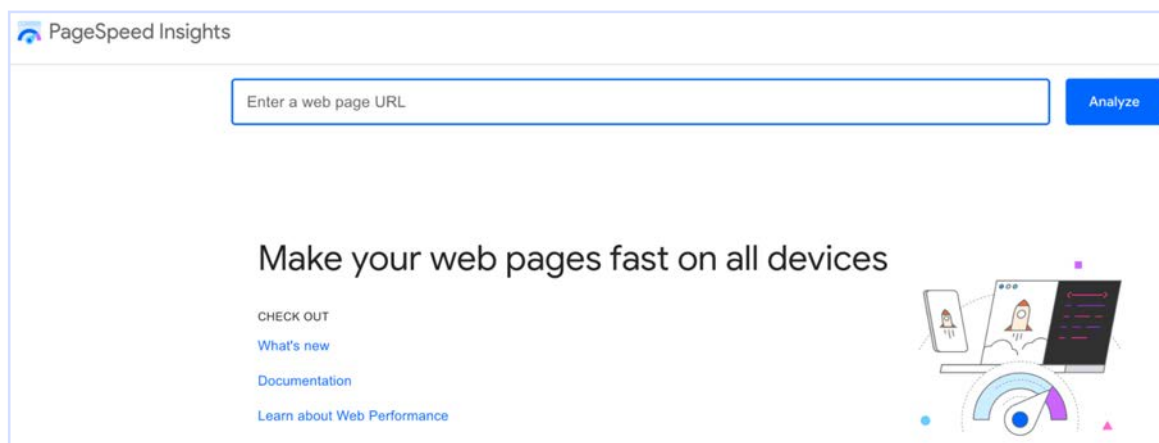
1. **San Francisco, USA:** 0.475 seconds
2. **London, United Kingdom:** 0.434 seconds
3. **Washington D.C., USA:** 0.375 seconds
4. **Tokyo, Japan:** 1.2 seconds
5. **Sydney, Australia:** 0.93 seconds
6. **Frankfurt, Germany:** 0.14 seconds

The point of using different server locations is to highlight the impact that physical distance has on loading times. Our website's server is located in the US, so the Washington D.C. test naturally got the best result.

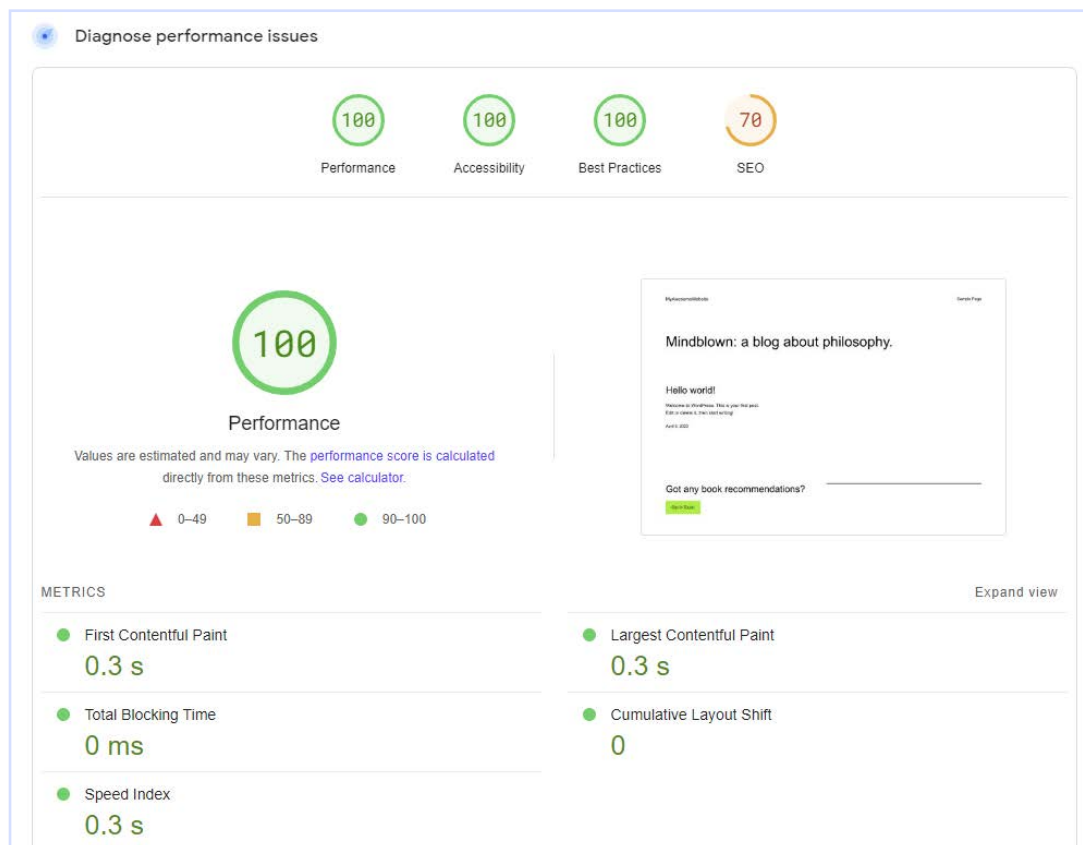
As the distance grows, so will loading times. For a small page like the one we're testing, the difference isn't too pronounced. However, it becomes much more apparent the more complex your website is.

► Analyzing Page Speed With Google

Now let's move on to Google PageSpeed Insights. As with Pingdom, this platform asks for a URL to test. We'll use our default homepage once more.



PageSpeed Insights breaks down a website's performance using different metrics. It tells you how long your website takes to render both partially and fully.



When most people talk about loading times, they're actually referring to the **Largest Contentful Paint (LCP)** metric. It measures how long it takes for the page's main content to display, even if there are still elements working in the background.

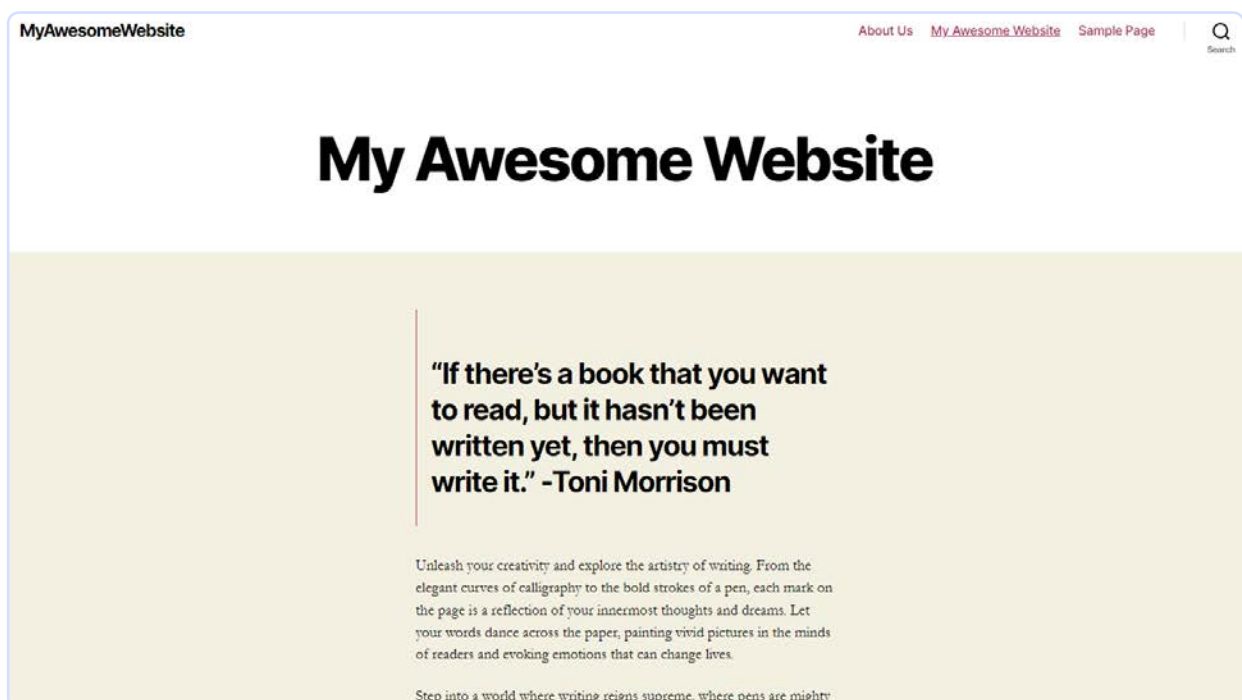
The numbers you see above are decent, as we're still below the two-second mark. Plus, we got a near-perfect optimization score, which means our servers are configured for speed from the start.

Although each tool provides slightly different numbers, the results tell us the same – a brand new WordPress website on an optimized server is fast. However, that doesn't mean there's no work to be done. Let's set up some dummy content for our site so that the tests become more realistic.

3. Create a Test Page With Sample Content

We're not going to dig into how to use the WordPress Block Editor to build pages here. However, if you need a refresher on this skill, you can check out our guide on [WordPress basics](#).

For the test site, we created a unique page filled with dummy content and several images. Here's a quick look at it:



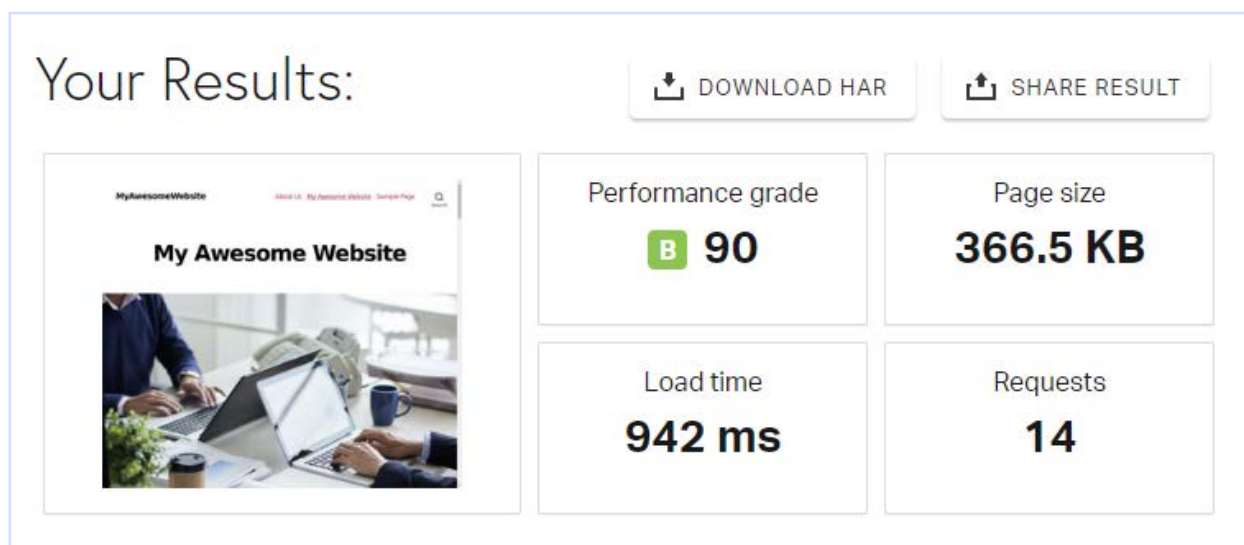
You can only see a fraction of the page, but it weighs 366.5 KB. We populated the test page with text and images, making it look close to an actual blog page.

Media files have an especially significant impact on loading times, as do any scripts that run in the background. We'll talk more about optimizing these elements later.

4. Re-Test Our New Page

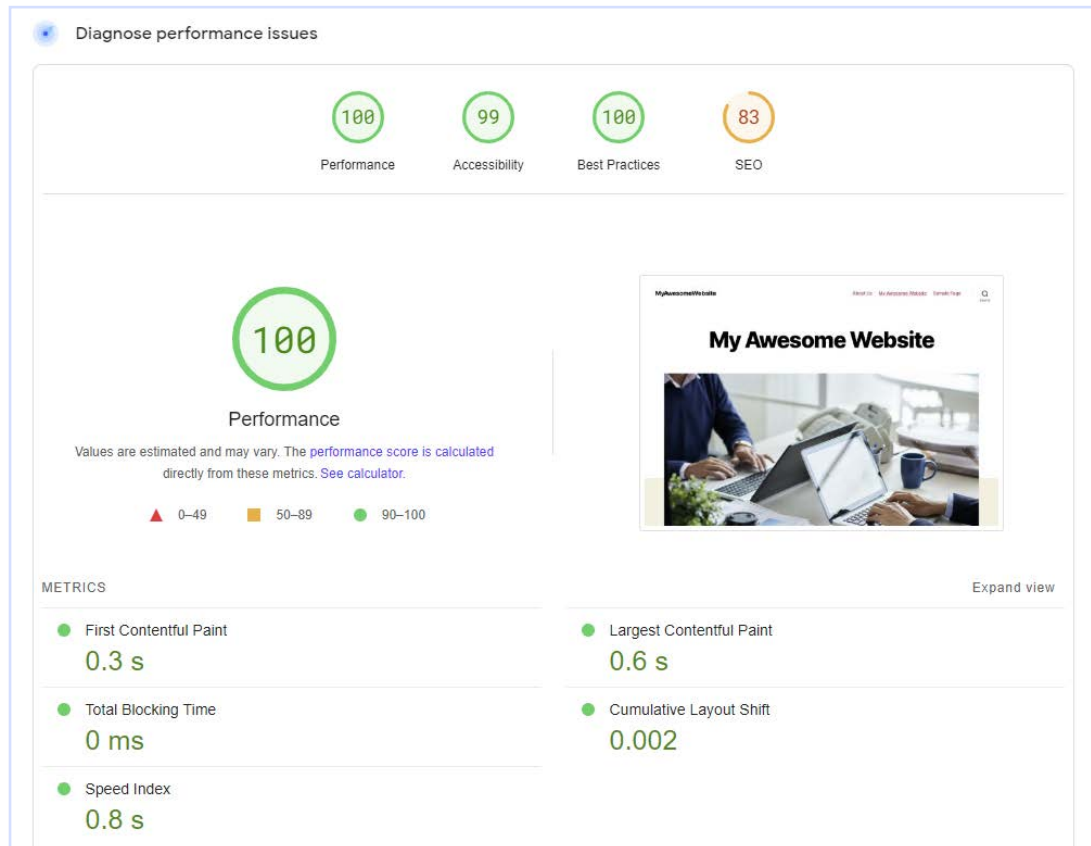
Now that we have our content set up, we will run the initial tests again to illustrate how heavier pages have longer loading times.

Here are the results of the Pingdom test from San Francisco for our new test page:



The number of requests is significantly higher than the default setup we tested, and the loading time also increased to almost three times its original value. Even so, we're still below one second, which is excellent news for anyone visiting our website.

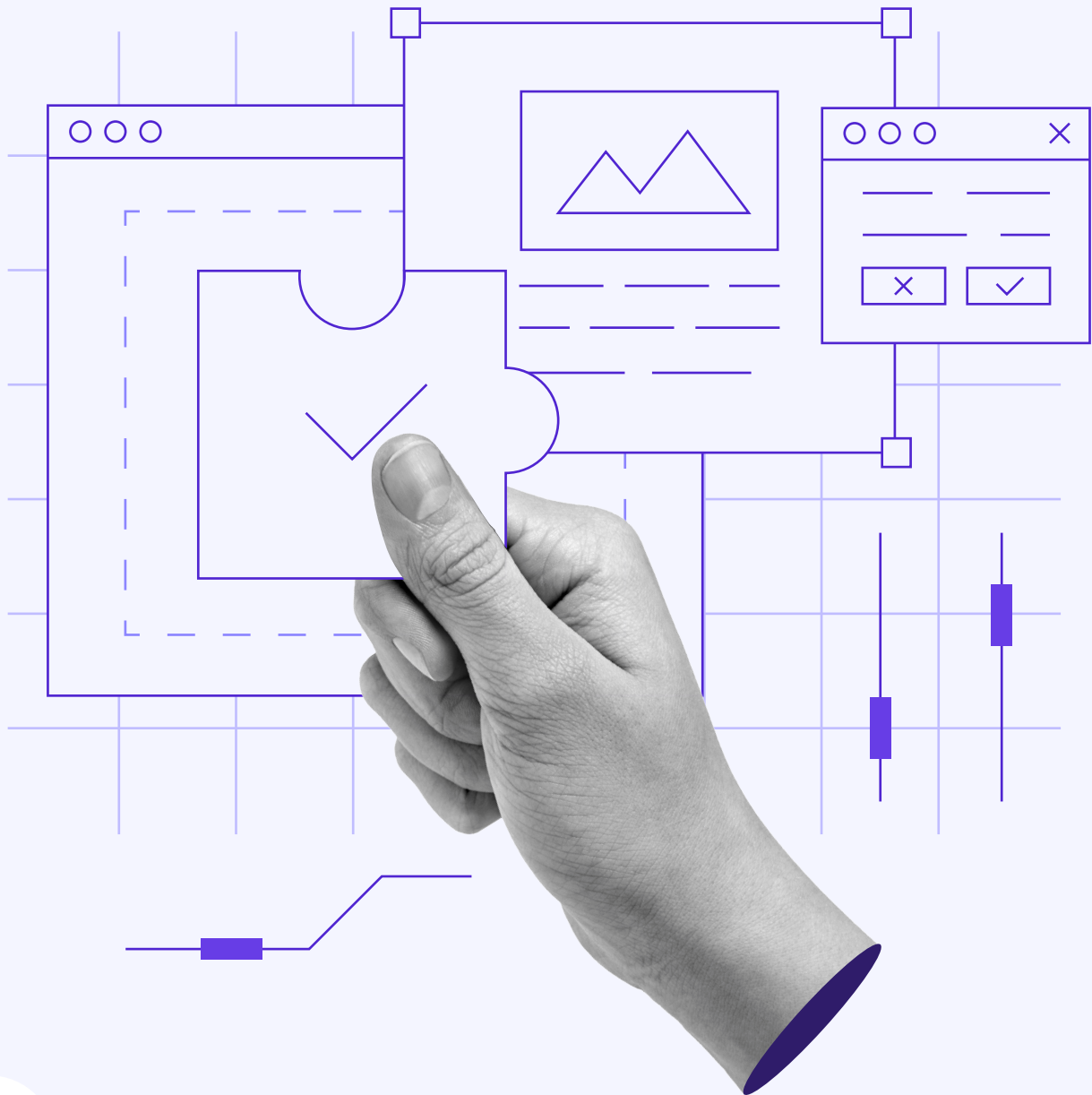
PageSpeed Insights also shows that the LCP value doubles. That is to say, the resources we've added so far have a measurable impact on performance:



In practice, you should never rely on a single test to gauge your website's performance. Only by aggregating data from multiple sources can you replicate what your users experience.

After all, you'll receive visitors from across the world with different devices and internet speeds.

Right now, our test website is fast by any metric. However, it's still very simple from a functional standpoint. In the real world, popular websites have a lot of advanced features and tools working in the background, such as plugins and themes, which we will address in further chapters.



3

Selecting the Right Plugins and Theme for Optimal Performance



[Click here to return to the table of contents](#)

Selecting the perfect WordPress theme and plugins from thousands of options can be tricky. The right choices can significantly improve your website's functionality and aesthetics. However, factoring in these extensions' influence over page speed is also important.

Fortunately, due to the wide selection of themes and plugins available, you can easily pick out those that will have a minimal impact on performance. All it takes is a bit of research.

In this chapter, we're going to break down factors to consider when choosing a WordPress theme and plugins. Let's get to work.

➤ What to Look for in a WordPress Theme

The main thing most people look out for in a [WordPress theme](#) is style. You want to pick one that embodies the right aesthetics for your brand and website. However, it's also important to factor other elements into your decision, such as:



Regular updates from the developer to ensure optimal performance and security.



Extra features tailored for your website's niche to provide key functionality.



Proper optimization to avoid adding unnecessary code that can slow your website down.

It's fairly easy to spot a theme's popularity, requirements, and update frequency. On WordPress.org, you can find this information in the short description under the **Download** button:

Astra

By *Brainstorm Force*

Commercial Theme [Support](#)

This theme is free but offers additional paid commercial upgrades or support.

[Preview](#) [Download](#)

Version: 4.1.5
 Last updated: May 18, 2023
 Active Installations: 1+ million
 WordPress Version: 5.3 or higher
 PHP Version: 5.3 or higher
[Theme Homepage](#) →

Ratings [See all](#)

You may also want to check out your prospective theme's **Development Log**. A quick look can give you an idea of how committed its developers are to maintaining it:

source: astra

☐ Added
 ☐ Modified
 ☐ Copied or renamed

[View changes](#)

Revision Log Mode:

☒ Stop on copy
☐ Follow copies
☐ Show only adds and deletes

View log starting at and back to
 Show at most revisions per page.
☐ Show full log messages [Update](#)

Diff	Rev	Age	Author	Log Message
<input checked="" type="radio"/>	@191905	4 weeks	themedropbox	New version of Astra - 4.1.5
<input type="radio"/>	@191229	6 weeks	themedropbox	New version of Astra - 4.1.4
<input type="radio"/>	@188679	2 months	themedropbox	New version of Astra - 4.1.3
<input type="radio"/>	@188095	3 months	themedropbox	New version of Astra - 4.1.2
<input type="radio"/>	@188040	3 months	themedropbox	New version of Astra - 4.1.1
<input type="radio"/>	@187989	3 months	themedropbox	New version of Astra - 4.1.0
<input type="radio"/>	@183643	5 months	themedropbox	New version of Astra - 4.0.2
<input type="radio"/>	@183255	5 months	themedropbox	New version of Astra - 4.0.1
<input type="radio"/>	@183162	5 months	themedropbox	New version of Astra - 4.0.0
<input type="radio"/>	@179283	7 months	themedropbox	New version of Astra - 3.9.4
<input type="radio"/>	@178838	8 months	themedropbox	New version of Astra - 3.9.3
<input type="radio"/>	@175427	9 months	themedropbox	New version of Astra - 3.9.2
<input type="radio"/>	@172347	11 months	themedropbox	New version of Astra - 3.9.1

However, it's hard to know if a theme is properly optimized without trying it firsthand. Testing out a free one from the WordPress Theme Directory is easy enough, but that's not the case with premium options.

Premium themes can cost up to **\$60** per one-year license. You don't want to pay that much money for a tool that's going to slow your site down.

There's no obvious solution to this problem. Most theme developers will tell you their products are fast and try to persuade you with high-performance scores from website performance testers such as PageSpeed Insights.

As we demonstrated in the previous chapter, scoring high on those tools doesn't automatically translate to fast loading times. One way to get a more accurate picture is to read user reviews. Keep your eyes peeled for mentions of slow performance or bloat and steer clear of those themes.

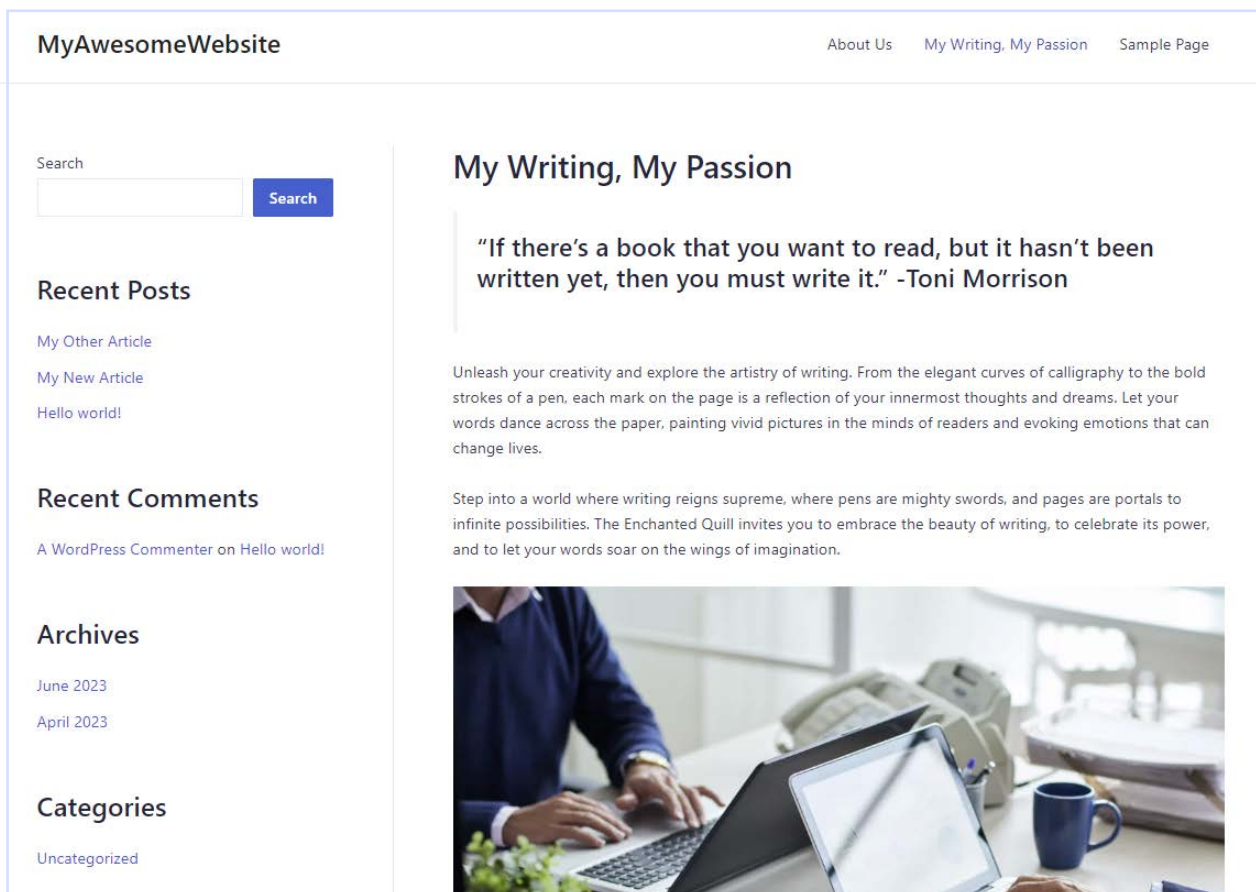
► Installing a Theme on Our Test Site

For our test site, we've decided to use [Astra](#), a free multipurpose theme:

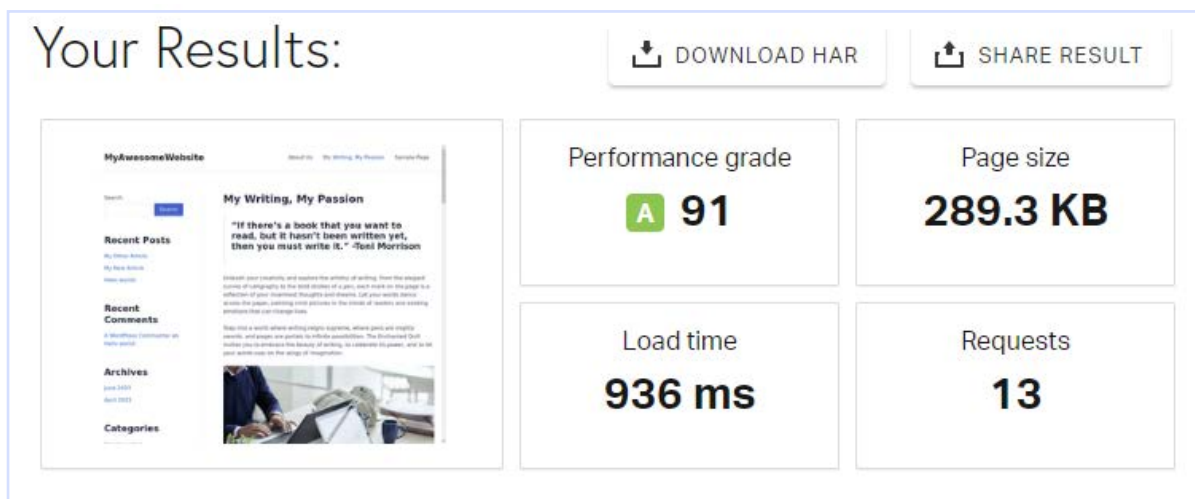


It offers solid performance, regular updates, and many desirable features. For more suggestions, you can check out our list of the [60+ Best Free WordPress Themes](#) as well as the [Fastest Premium WordPress Themes](#).

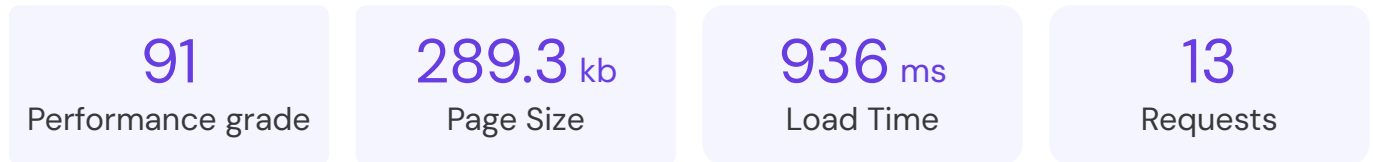
If you're new to WordPress and need help installing a theme, we also have a helpful [tutorial on that subject](#). For now, we'll skip right to our test page after activating Astra. It contains the same text and images, just with a new coat of paint:



In theory, switching themes should immediately impact our loading times, for better or worse. A quick Pingdom test shows that's true:

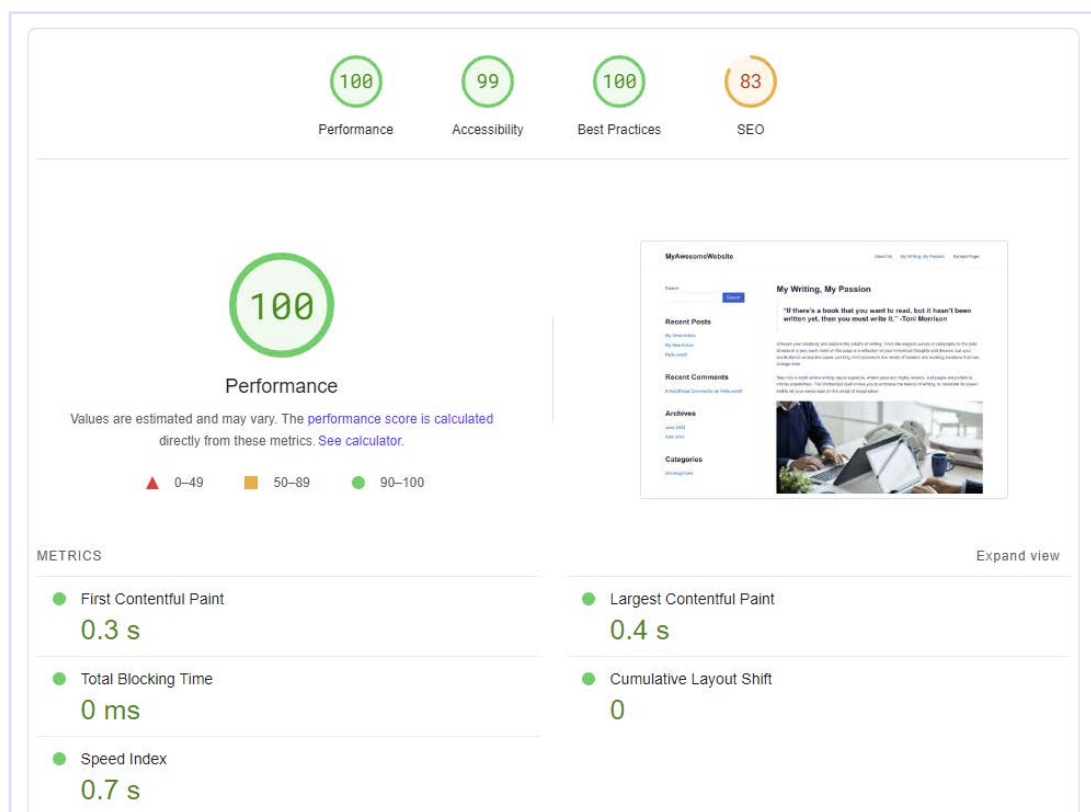


To remind you, our Pingdom results for this page using the default **Twenty Twenty** theme were as follows:

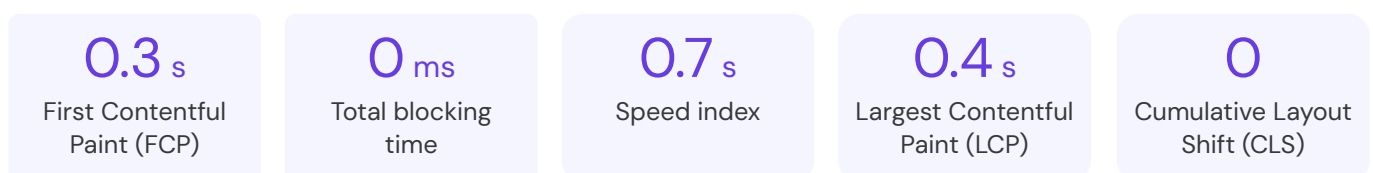


As you can see, page size, load time, and request drop by a small margin while the performance grade goes up by one point – all this just by switching to Astra.

PageSpeed Insights tells a similar story, with the Astra version of our test page loading slightly faster:



Here are the initial PageSpeed Insights results when using the default **Twenty Twenty** theme:

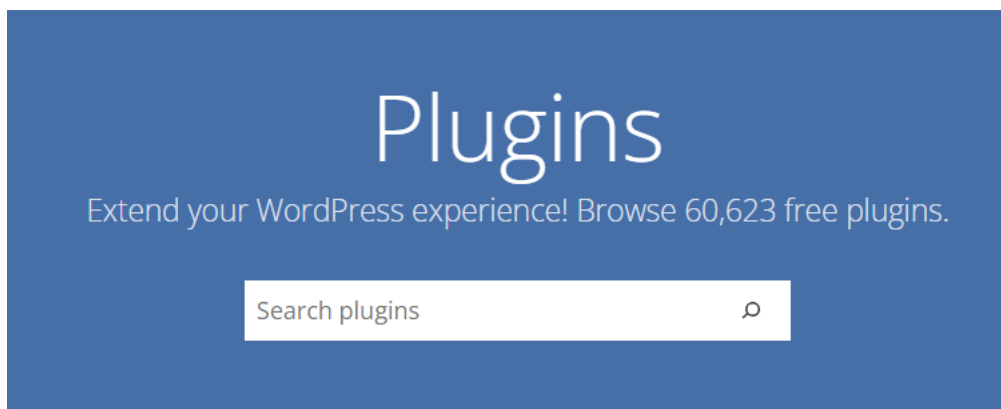


All default WordPress themes are already optimized for speed, but these tests prove that there are third-party themes that can further improve your website's performance. By the time we finish this eBook, those numbers will be even lower.

➤ How to Select and Install the Right Plugins for Your WordPress Website

Like themes, plugins can make or break your website's performance. [Installing too many plugins](#) can slow your site down, so it's best to only go with the ones you need most.

Narrowing your options down can be challenging, especially since 60,623 plugins are available for free in the WordPress Plugin Directory as of writing this eBook. If you factor in premium options, that number skyrockets.



If you need help choosing which plugins to install, follow our best practices for smart plugin selection:

Be selective. Don't install plugins just for the fun of it. Be realistic about your needs and the usefulness of any tool added to your site.

Avoid overlapping functionality. Try not to install multiple plugins that do the same thing. Consider any features your theme includes and avoid overlaps there as well.

Consider multi-functional vs. single-use plugins. Opt for a multi-functional plugin if it fulfills many of your needs. Otherwise, use lightweight plugins that offer a specific functionality.

Deactivate or delete unused plugins. Doing so not only eliminates unnecessary code from your site but also security risks due to unpatched vulnerabilities.

If you want some guidance in this area, see our list of [Best WordPress Plugins](#).

➤ Adding Plugins to Our Test Site

To illustrate how plugins impact performance, we installed five popular ones on our test website:

- **All-in-One WP Migration**

A plugin that handles website backup and migration.

- **Akismet Anti-Spam**

A comment spam prevention tool.

- **WPForms Lite**

A simple plugin for setting up contact forms.

- **Wordfence Security**

A comprehensive WordPress security plugin.

- **Yoast SEO**

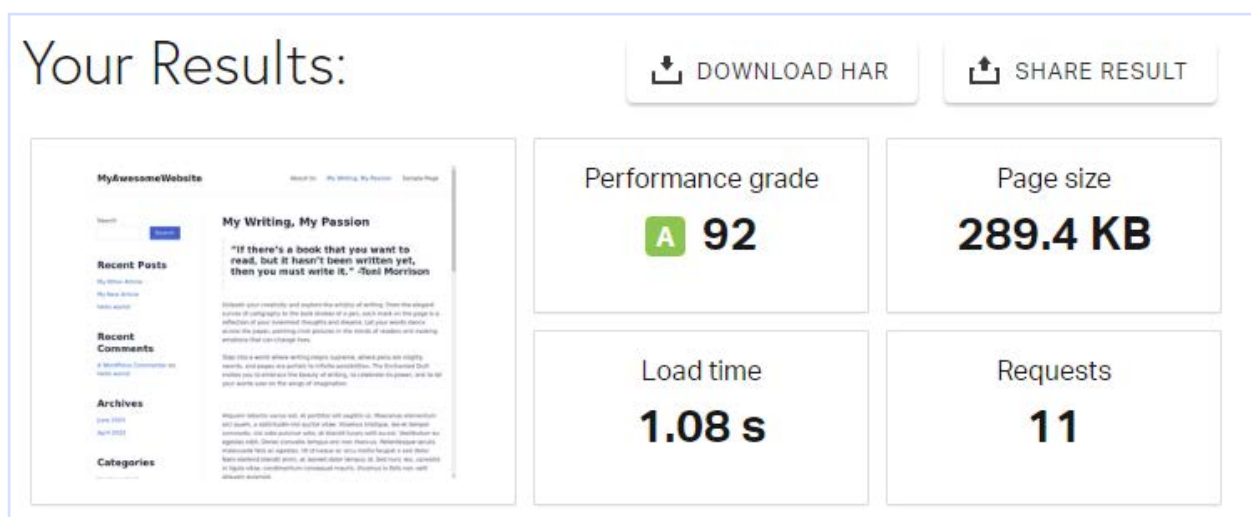
The most popular Search Engine Optimization (SEO) plugin.

For guidance installing WordPress plugins, you can read our [beginner's tutorial on this skill](#). Once we added the five listed above, we activated them:

<input type="checkbox"/> Plugin	Description	Automatic Updates
<input type="checkbox"/> Akismet Anti-Spam Settings Deactivate	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. It keeps your site protected even while you sleep. To get started, just go to your Akismet Settings page to set up your API key. Version 5.1 By Automattic View details	Enable auto-updates
<input type="checkbox"/> All-in-One WP Migration Deactivate	Migration tool for all your blog data. Import or Export your blog content with a single click. Version 7.75 By ServMask View details Contact Support Translate	Enable auto-updates
<input type="checkbox"/> Hello Dolly Activate Delete	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page. Version 1.7.2 By Matt Mullenweg View details	Enable auto-updates
<input type="checkbox"/> Hostinger Activate Delete	Hostinger WordPress plugin. Version 1.2.0 By Hostinger Visit plugin site	
<input type="checkbox"/> LiteSpeed Cache Activate Delete	High-performance page caching and site optimization from LiteSpeed Version 5.4 By LiteSpeed Technologies View details	Enable auto-updates
<input type="checkbox"/> Redis Object Cache Settings Deactivate	A persistent object cache backend powered by Redis. Supports Predis, PhpRedis, Relay, replication, sentinel, clustering and WP-CLI. Version 2.4.1 By Till Krüss View details ★ Upgrade to Pro	Enable auto-updates
<input type="checkbox"/> Wordfence Security Upgrade To Premium Deactivate	Wordfence Security - Anti-virus, Firewall and Malware Scan Version 7.9.3 By Wordfence View details	Enable auto-updates
<input type="checkbox"/> WPForms Lite Get WPForms Pro Settings Docs Deactivate	Beginner friendly WordPress contact form plugin. Use our Drag & Drop form builder to create your WordPress forms. Version 1.8.2.1 By WPForms View details	Enable auto-updates
<input type="checkbox"/> Yost SEO Get Premium Finish your first-time configuration FAQ Settings Deactivate	The first true all-in-one SEO solution for WordPress, including on-page content analysis, XML sitemaps and much more. Version 20.9 By Team Yoast View details	Enable auto-updates
<input type="checkbox"/> Plugin	Description	Automatic Updates

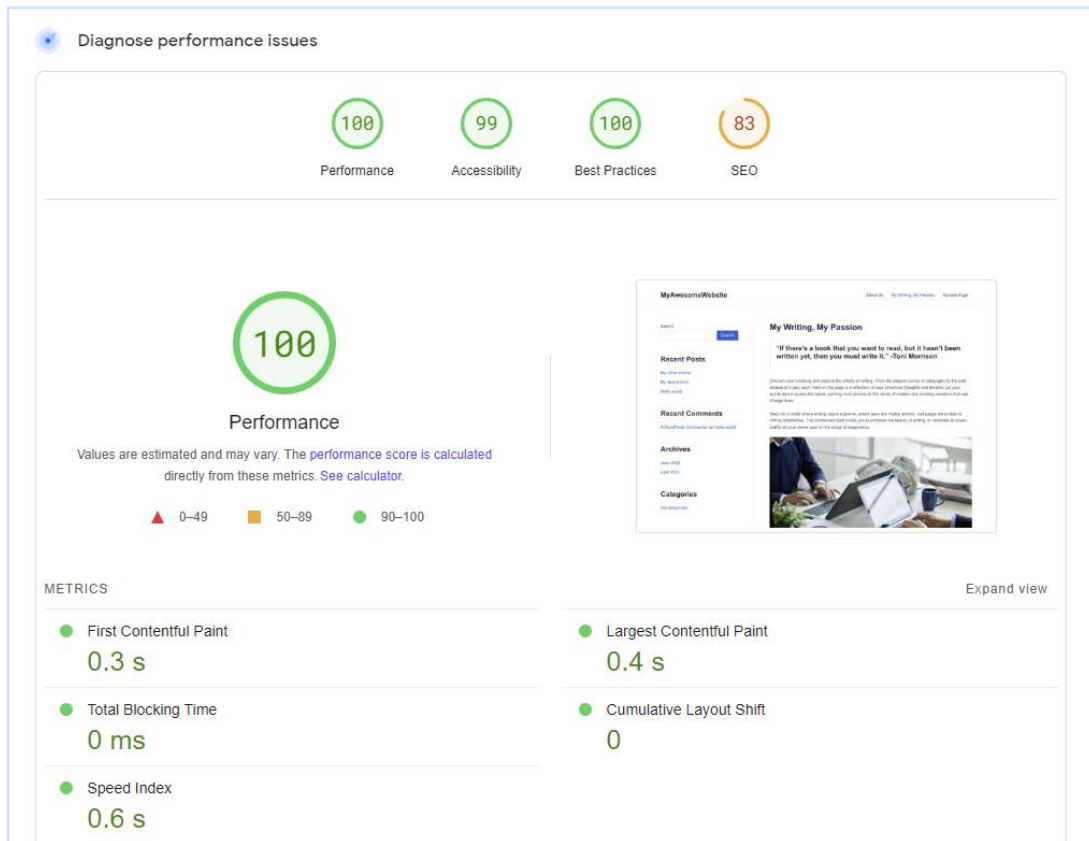
As you can see, there are three other plugins listed in our dashboard. **Hello Dolly** comes pre-installed on every WordPress site, while **LiteSpeed Cache** and **Hostinger** are included with our hosting plan. We'll discuss them in a later chapter.

For testing purposes, we're going to leave them deactivated. With that in mind, let's go through our standard tests once more, starting with Pingdom Tools:



Right away we see an increase in loading times, now requiring over a second to load. Meanwhile, the performance grade goes up by one point.

Interestingly, PageSpeed Insights showcases a 0.1 second decrease in speed index:

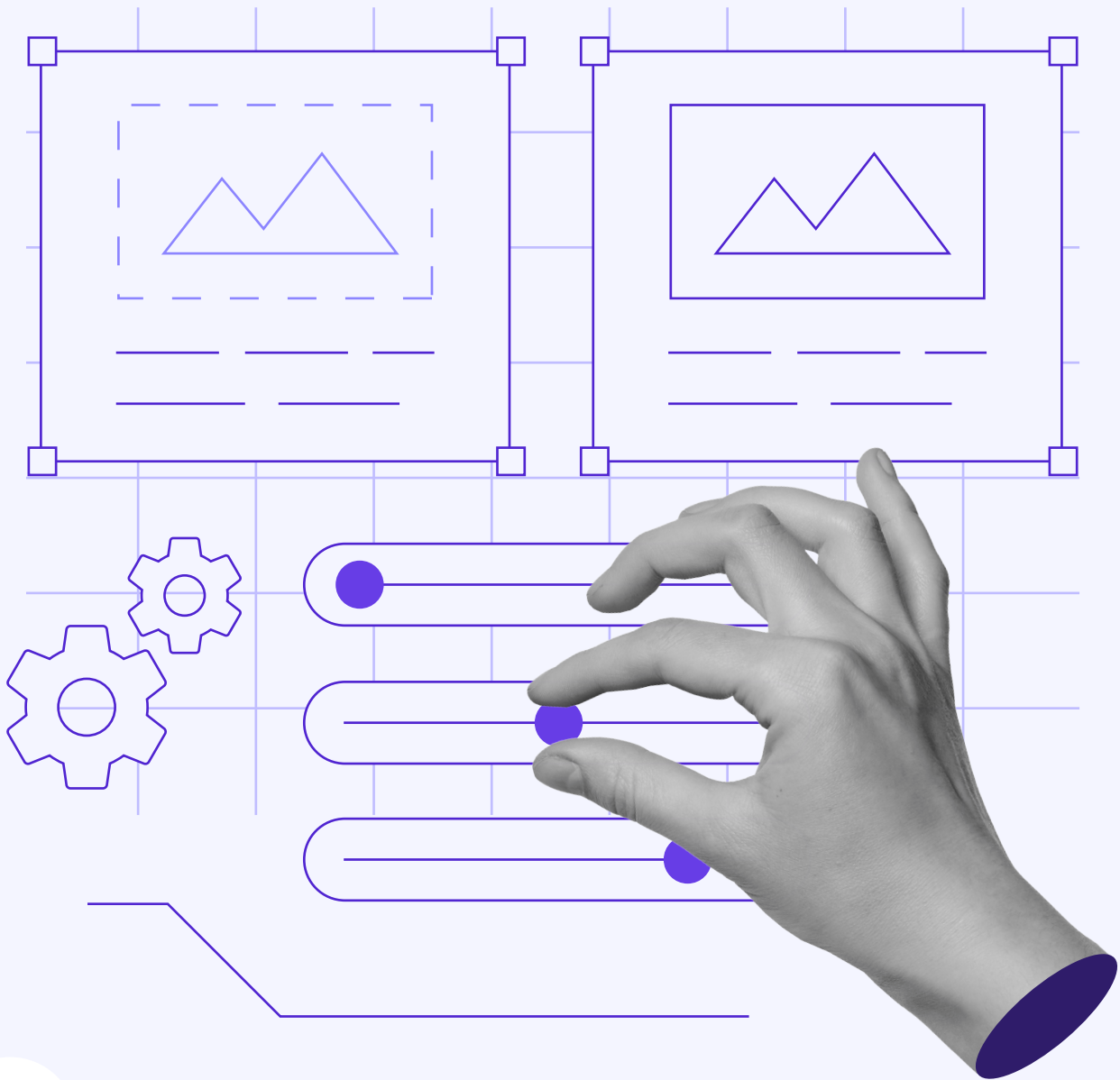


The more scripts you have running in the background and loading with your pages, the longer it takes to load them. However, that doesn't mean you can't use a broad range of plugins on your website.

A WordPress website with ten active plugins or less should perform just fine if other optimization strategies are applied. Keep in mind that performance drops from plugins may happen after installation and activation but will stabilize over time.

Plugins are one of the most useful aspects of WordPress. Not using them because they may impact your performance severely limits what you can do with the platform.

As long as you choose which plugins you use carefully and you don't have a hundred of them active at the same time, you should be fine.



4

Optimizing Your Images



[Click here to return to the table of contents](#)

Images are a key component of any website. Unfortunately, they're also one of the leading culprits when it comes to poor performance.

High-resolution images look better. However, they also take longer to load. By [optimizing](#) yours, you can cut down on your file sizes, therefore, cutting down on your loading times. Visitors shouldn't be able to see the difference in the quality of your media.

As for how to optimize your images, there are several methods you can use in WordPress. However, by far the most important is compression. Let's talk about how it works before we continue to work on our test website.

► An Introduction to Image Optimization and Compression

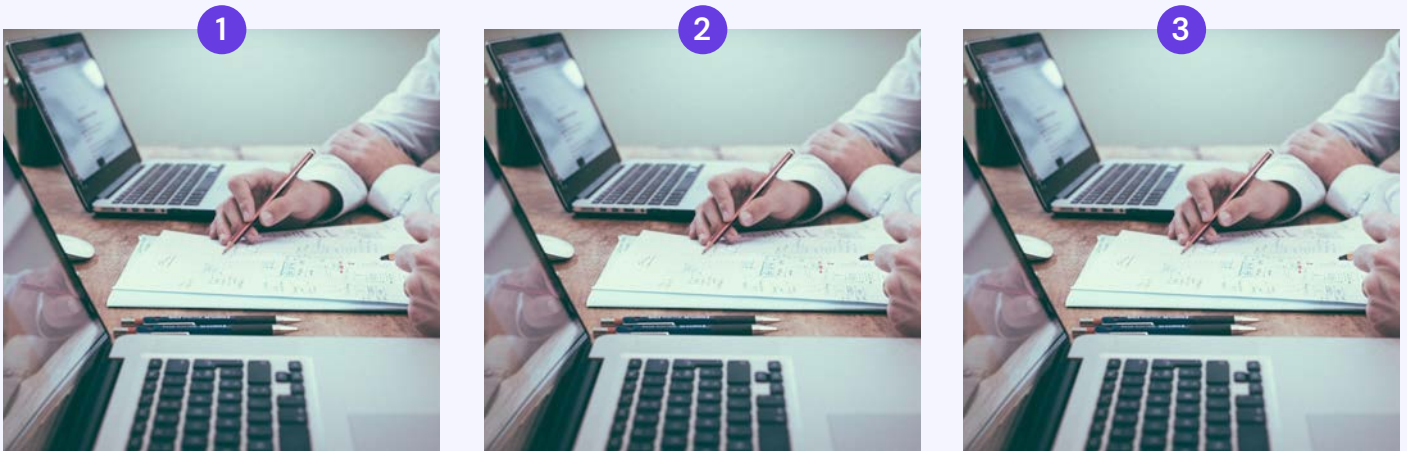
You've probably dealt with compressed files at some point. Essentially, compression software reduces the size of a file without affecting the data itself. This enables you to transfer it much faster.

Compression works similarly when used for image optimization. The file size is reduced without removing any data. This process is known as [lossless](#) compression. It's ideal for images with tons of small details that you want to preserve.

Unfortunately, it might not reduce your file sizes as much as you'd like since it focuses more on preserving quality. On the other end of the spectrum, you have **lossy** compression.

With this approach, there's a trade-off between smaller file sizes and a decrease in quality. Pixels are removed to reduce the amount of data. The resulting image might not look as crisp, but in real-life scenarios, most users still won't notice the difference.

Take this image for example: fig 1 is **uncompressed**, fig 2 has been through **lossless compression**, while fig 3 is the **lossy** version.



If you blow these images up to a higher resolution, you **might** be able to spot the differences. However, when it comes to desktop or mobile devices, you're usually working with much lower resolutions than it takes to fully detect those changes.

Therefore, all three images look pretty much identical, even though one of them is significantly larger than the others.

If you're uploading pictures from a mobile device or camera, you're probably working with massive initial file sizes. Loading those images can dramatically decrease website performance, even with fast servers and user connections.

By optimizing them and using more appropriate resolutions, you can speed up your website, and few people, if anyone, will be able to spot the difference.

➤ How to Optimize Your WordPress Images (In 4 Steps)

There are many online services you can use to optimize images. However, it's a time-consuming process that doesn't always scale well for websites. Considering you want to compress every media file you upload, your best bet is a plugin that automates the process. Let's start there.

1. Use an Image Optimization Plugin

As with most WordPress functionality extensions, there's more than one plugin to choose from to get the job done. Image optimization is a key step to improving your website's performance, so it's a highly saturated category.

Putting aside premium options, consider utilizing [TinyPNG](#) for [compressing JPEG and PNG images](#). As the name implies, it supports popular image file formats and optimizes them automatically when you upload them to your site. It can also handle the existing ones in your Media Library:



Beyond that, TinyPNG can also help you automatically resize dimensions by setting a maximum height and width. This can help ensure your visuals are displayed at a reasonable resolution.

For the plugin to work, you'll need to register for a free account with [Tinify](#), the company that powers the service. This enables you to optimize up to 500 images per month, which should be enough for most medium-sized websites.

You can register or connect an existing account from the **Settings > TinyPNG > JPEG, PNG & WebP image compression** tab in your WordPress dashboard.

Compress JPEG & PNG images

Speed up your website. Optimize your JPEG and PNG images automatically with TinyPNG.

Tinify account

Register new account

Provide your name and email address to start optimizing images.

Your full name

Register Account

Already have an account?

Enter your API key. Go to your [API dashboard](#) to retrieve it.

Save

Here, you'll find a link to retrieve your **Tinify API key**, which is a string of characters and numbers that it uses to authenticate your account.

Once you save your API key to WordPress, you can turn to the **New image uploads** settings within the same tab. Here you can configure whether the plugin should optimize images in the background or upon upload:

New image uploads

When should new images be compressed?

☐ **Compress new images in the background (Recommended)**
This is the fastest method, but can cause issues with some image related plugins.

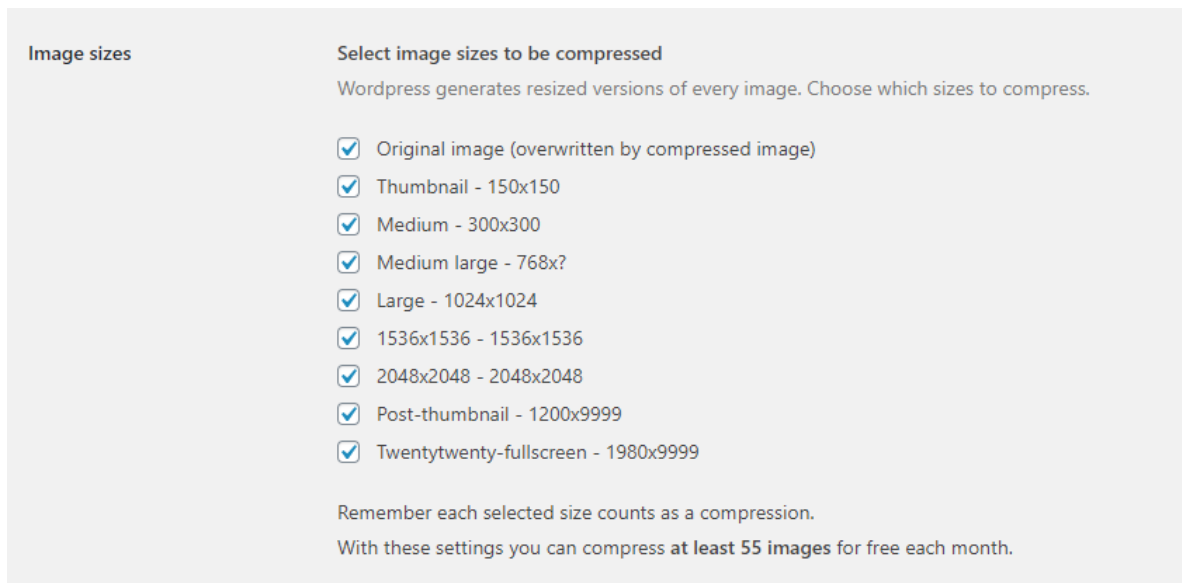
☒ **Compress new images during upload**
Uploads will take longer, but provides higher compatibility with other plugins.

☐ **Do not compress new images automatically**
Manually select the images you want to compress in the media library.

We suggest you go with the option to **Compress new images during upload**. It might slow down the image upload process a bit, but it's well worth it. Plus, the difference isn't significant if you're using a decent hosting provider.

2. Resize Images Upon Uploading

If you scroll down even further within the **Settings > TinyPNG > JPEG, PNG & WebP image compression** tab, you'll see a section that reads **Image sizes**. Here, you can configure which image resolutions the plugin should automatically optimize.



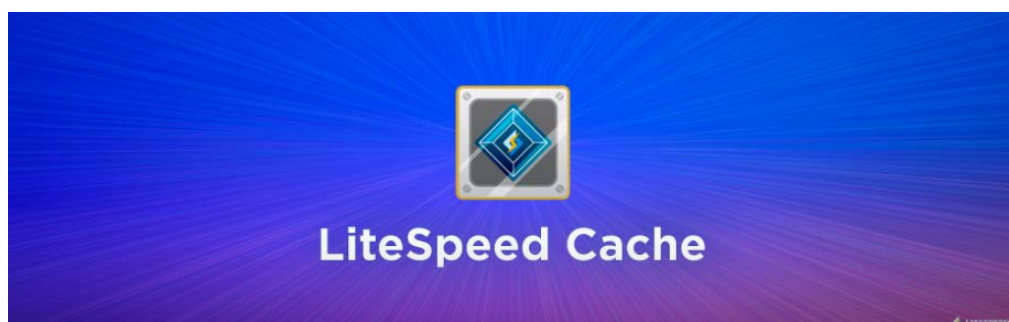
WordPress automatically generates multiple copies of every image you upload, each at a different resolution. The best example is thumbnails, which you can use to display a blog post's featured image within another page.

Each size you compress counts toward your 500-image limit. In our experience, you get the best results by optimizing only the original, full-size images and thumbnails, which are the most commonly used versions. Just make sure to select a resolution that's been compressed when you add media files to your library.

Once you're done, remember to save your changes to the plugin's settings.

3. Enable Lazy Loading

If you're using Hostinger to host your website, you'll notice we include a few plugins with all our customers' WordPress installations. One of these is called **LiteSpeed Cache**, which includes a whole host of features to help you optimize your site's performance:

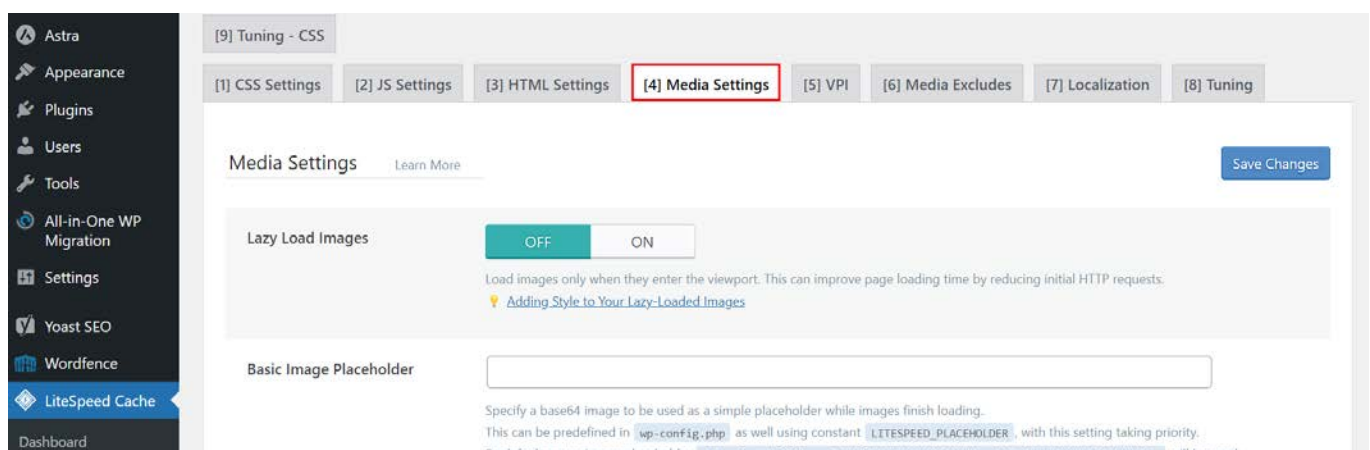


We will touch on many of those features throughout this eBook. For this chapter, we want to focus on a specific option called **lazy loading**.

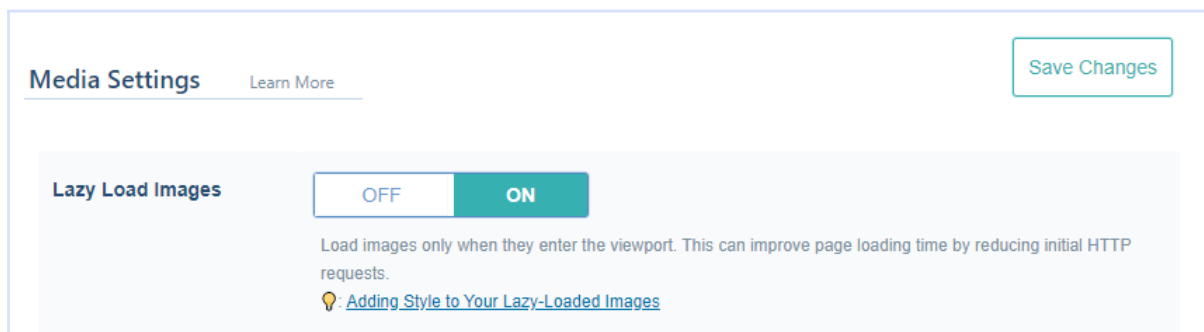
Usually, when you visit a website, your browser will attempt to load all its elements at once. If you visit a page that requires a fair bit of scrolling, your browser prioritizes elements at the top of the screen and at the bottom equally.

Lazy loading tells browsers to hold off on loading images until users scroll down and reach them. The difference may be subtle, but it can have an impact on your page's initial loading times. Combine it with compressed images and you should see even better results.

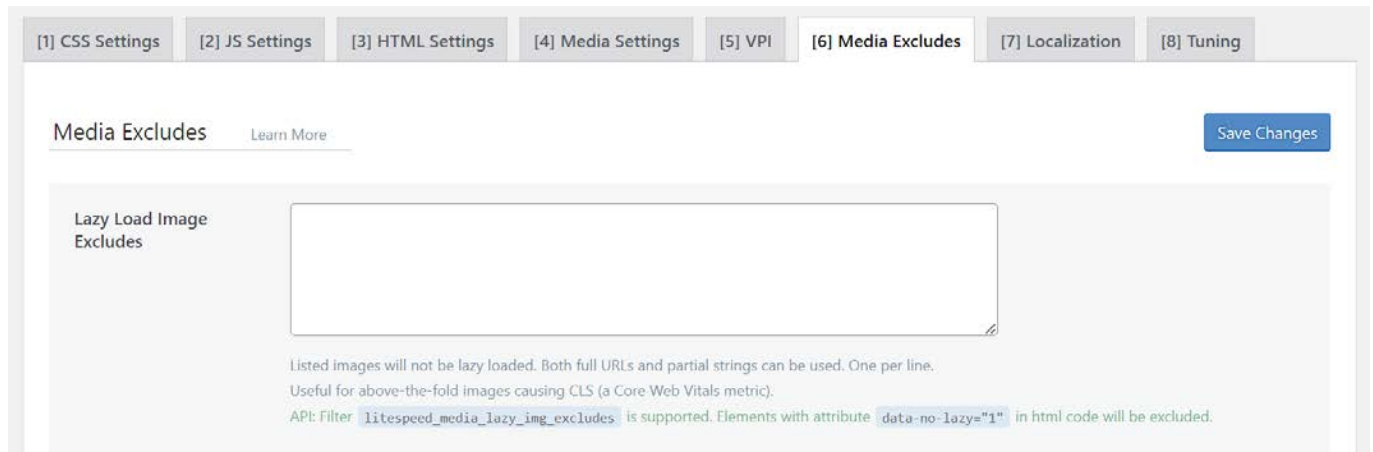
Once you activate LiteSpeed Cache, you can enable lazy loading by navigating to **LiteSpeed Cache > Page Optimization** and opening the **Media Settings** tab:



Turn on the **Lazy Load Images** option:



If you want to disable lazy loading for specific images and CSS classes, you can add them to the **Lazy Load Image Excludes** field under the **Media Excludes** tab:

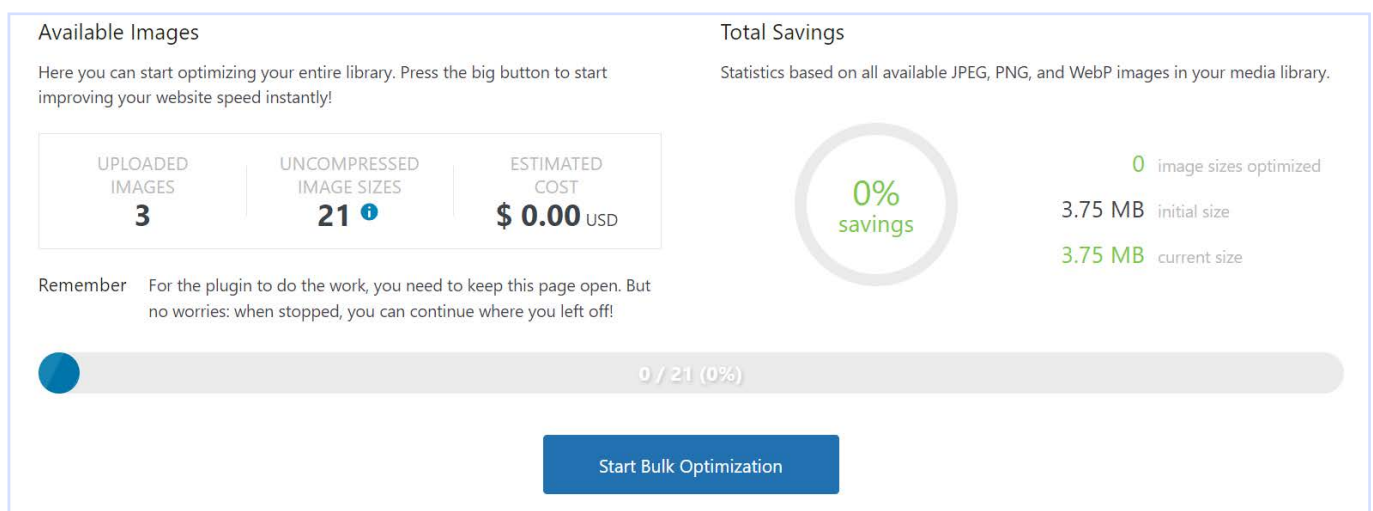


At this stage, you have all the pieces in place for top-to-bottom image optimization throughout your website. Now it's time to compress your existing media and see what kind of difference that will make.

4. Optimize Existing Images

During the last few chapters, we've been optimizing and testing the same website, which includes multiple images. The plugin we set up during Step 1 above takes care of optimizing new images you upload, but you have to run existing graphics through it manually.

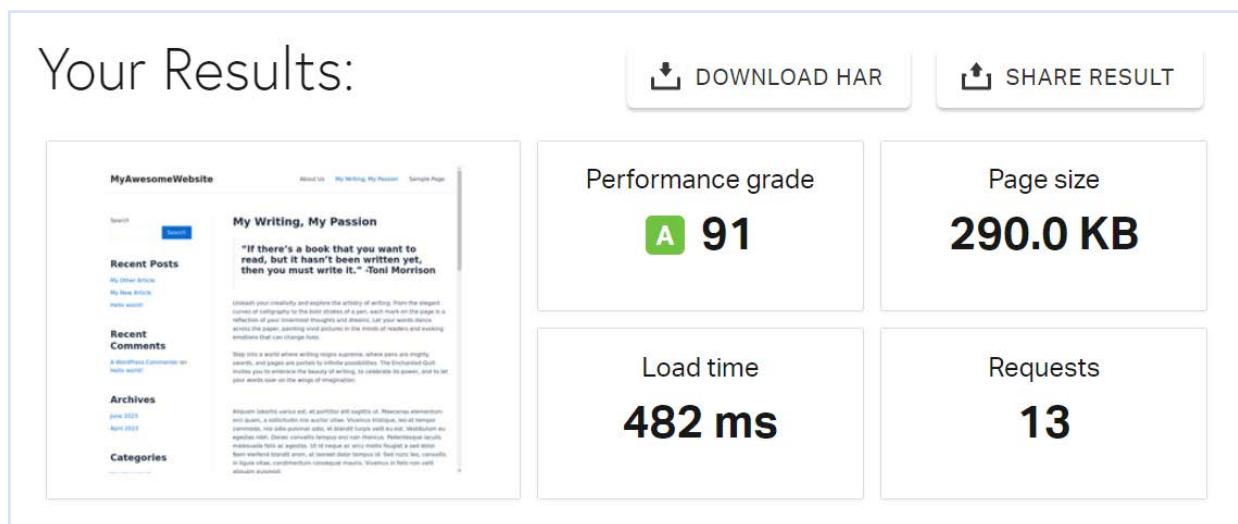
To do so, go to the **Media > Bulk Optimization** tab. Here, you'll see an overview of your Media Library and a **Start Bulk Optimization** button. Simply click on it to start compressing existing images:



With your free account, you can optimize up to 500 images for free per month. You don't need to worry about the estimated cost unless the number of uploaded images exceeds that limit.

► Optimizing Images on Our Test Site

We've now compressed all the images on our example site, so it's time to get back to testing. First, let's check out what Pingdom has to say about our sample page now.



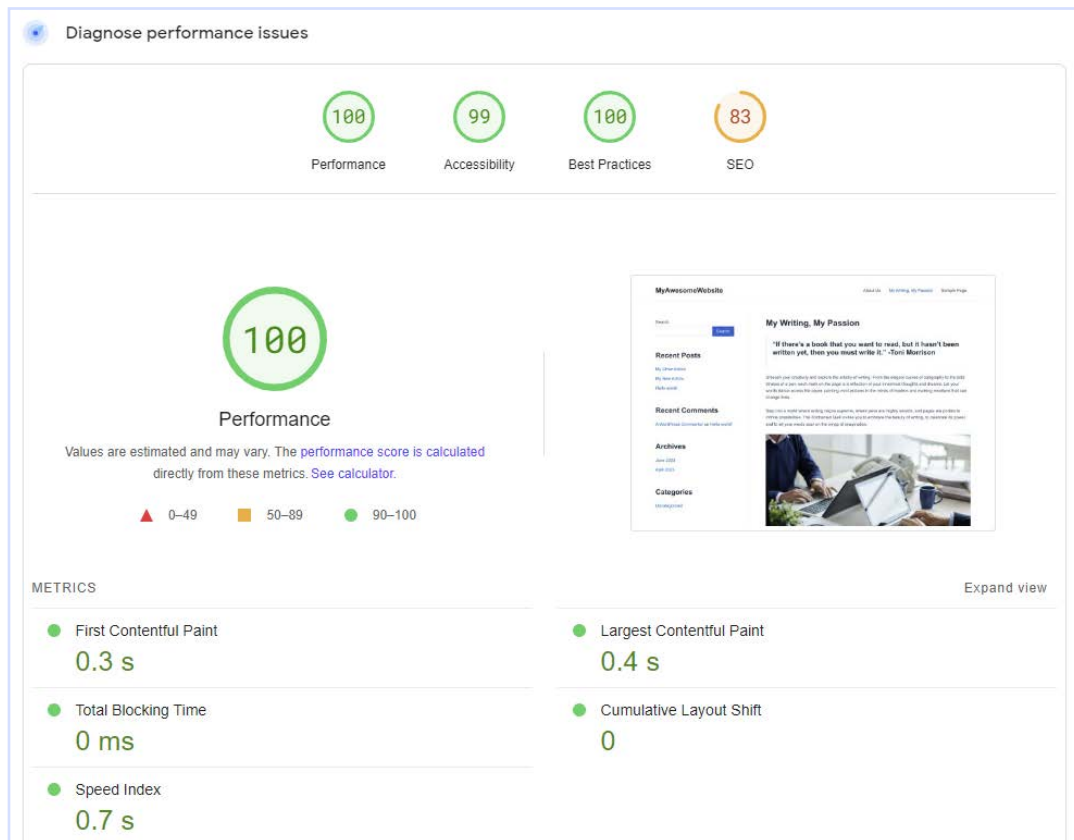
Let's start with the **Page Size** section. During the previous chapter it clocked in at 289.4 KB and now it's up by 600 bytes. In theory, compressing our images should've reduced this metric, but there are two things to consider:

We installed another plugin, which adds more scripts to each page.

Our images weren't all that heavy to begin with, so our compression savings weren't all that significant.

Although the test page's size is slightly larger, it actually loads a lot faster. During Chapter 3, Pingdom returned a result of 1.08 seconds. It's now down to 0.482 seconds, which is fantastic.

Now, let's compare those results with PageSpeed Insights.



The results here aren't as positive. With the additional scripts from TinyPNG, our speed index goes up by 0.1 second. Overall, we still achieve fairly fast loading times.

In theory, there should've been a broader margin between the **First rendering with content** value and the other metrics, thanks to lazy loading. However, that difference is often minimal in practice.

We can attribute our results to the fact that we're using well-optimized images and that our page is still remarkably light. For pages that are massive in size due to a wealth of elements, lazy loading is much more likely to have a noticeable impact.

We've seen significant improvements in WordPress performance so far. However, there are still a couple of optimization methods we can apply before we call it a day. They're some of the most important ones, too, so let's keep going.



5

Minifying Your Website's CSS and JavaScript



[Click here to return to the table of contents](#)

Chapter 3 made it clear that plugins have an impact on your WordPress website's performance. While the severity of their influence varies, every extension will add new scripts to your site. That additional code means your server needs to deal with more requests for every visitor.

Since we don't want to remove useful features from your website, the next best approach is to decrease the impact they have on loading times. We can achieve that through code minification, which is similar to compression in that it reduces file size.

For this chapter, we're going to break down how minification works and what types of scripts you can use it for in WordPress. Then we'll show you how to implement it on your website and the impact it can have on performance.

► An Introduction to Minification

Usually, when you compress a file, you need to decompress it before accessing it again. However, minification works differently. When you minify code, you take out redundant information. It's still usable, but if you look at the source, it will be very hard to read.

Take this example of [CSS](#), where we configure the border style, padding, and background color for a container:

```
div {  
  border: 1px solid black;  
  padding: 25px 50px;  
  background-color: lightblue;  
}
```

The minified version of that snippet would look something like this:

```
div{border:1px solid #000;padding:25px 50px;background-color:#add8e6}
```

To speed up a website, you can simplify its code by removing unnecessary spaces, comments, and sometimes even replacing variable names. These changes might seem significant, but they don't actually affect how the website works. Although the minified code becomes harder for humans to read, computers can still understand it just fine.

By minifying the code, you reduce its overall size, making it faster to download and load on web browsers. This optimization technique improves website performance without altering its functionality.

For the example above, minification is overkill. However, when you have a full WordPress website, there are thousands of lines of scripts and stylesheets to minify. Put together, you can achieve noticeable results when it comes to page size, which in turn impacts loading times.

However, minifying code manually isn't a viable solution, unless you're talking about a handful of simple files. In most cases, you'll want to rely on a tool that can minify the types of files you need to optimize.

Usually, you want to focus on JavaScript and CSS, which are among the most common types of scripts for most websites. You can also try to minify your HTML files if your web host doesn't [enable GZIP compression](#).

► How to Minify Your Website's CSS and JavaScript (In 2 Steps)

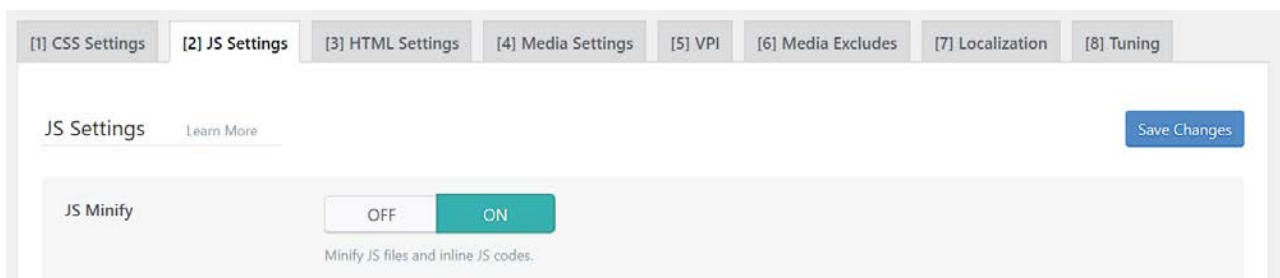
In the last chapter, we introduced you to the **LiteSpeed Cache** plugin as our preferred tool for implementing lazy loading. As we've mentioned, it can do a lot more to help speed up your website. In the following sections, we'll show you how to use it to minify key files on your website.

1. Configure LiteSpeed Cache for Minifying Your Website's Scripts

Enabling minification using LiteSpeed Cache is very straightforward. Navigate to **LiteSpeed Cache > Page Optimization**, then open the **CSS Settings** tab. Turn on the **CSS Minify** setting, and the plugin will do the rest:



Do the same to the **JS Minify** setting within the **JS Settings** tab:

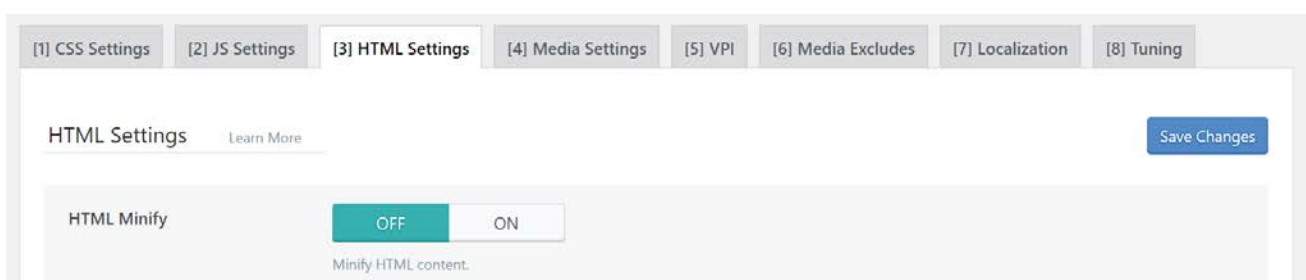


You'll notice some options for combining your CSS and JS files. If you enable those features, your visitors will have to load fewer files since the plugin will merge all your site's code to reduce the number of requests.

That sounds fantastic in practice, but in our experience, the improvements are often marginal. Merging scripts can even prompt errors with certain themes and plugins, so we don't recommend enabling this feature.

2. Enable Additional Code Optimization Settings

Moving on, you'll see the setting to minify [HTML files](#) as well. As we mentioned before, we want to skip this option, so make sure it's disabled.



The last two settings to focus on are useful for **deferring** CSS and JS files. Enabling them means that browsers won't rush to load scripts that aren't critical for the first stage of page rendering, cutting down on initial loading times.

The process is somewhat similar to lazy loading, which we described in Chapter 3. Go back to the **JS Settings** tab and pick **Deferred** for the **Load JS Deferred** setting:

Load JS Deferred

OFF

Deferred

Delayed

Deferring until page is parsed or delaying till interaction can help reduce resource contention and improve performance causing a lower FID (Core Web Vitals metric). [Learn More](#)

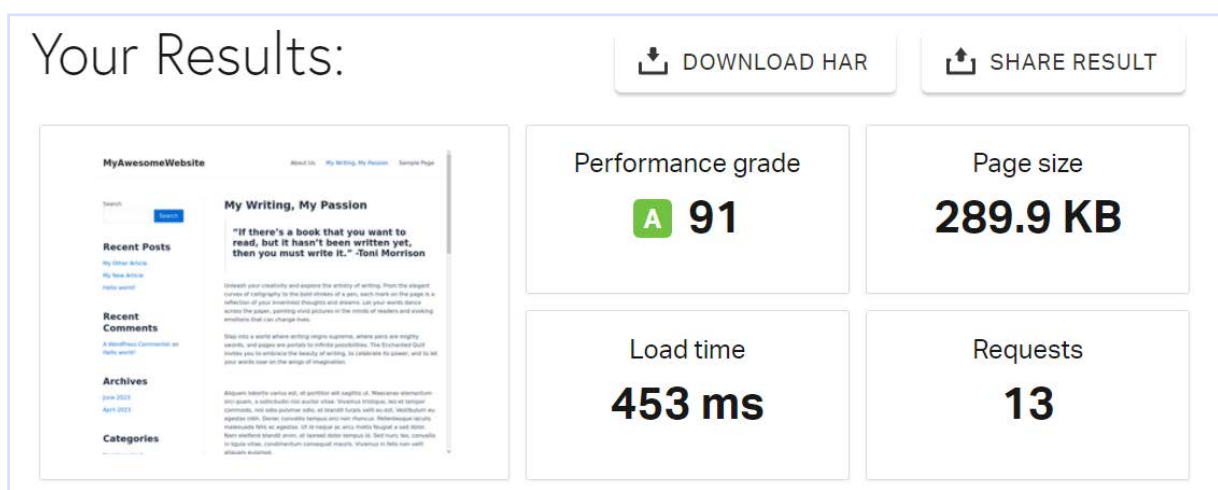
This can improve your speed score in services like Pingdom, GTmetrix and PageSpeed. [Learn More](#)

⚠️ This option may result in a JS error or layout issue on frontend pages with certain themes/plugins.

Remember to save your changes to the plugin's configuration before you move on to testing.

➤ Minifying and Deferring Code on Our Test Site

Minifying our page's scripts should, theoretically, reduce its file sizes and loading times. The question is how significant an impact it will have. Let's start by testing it using Pingdom:



In Chapter 4, this same page weighed in at 290 KB. Minifying its scripts and styles led to a reduction in file size of 100 bytes.


That metric isn't all that significant, but once you look at loading times, you see a more sizable difference. This round of testing clocked in at about 29 ms faster than our last test.

A hundred bytes of file size reduction isn't enough to account for that drop in loading times, however. That's largely due to deferring the loading of both **JavaScript and CSS** in the background. In fact, PageSpeed Insights uses this as one of its performance improvement suggestions:

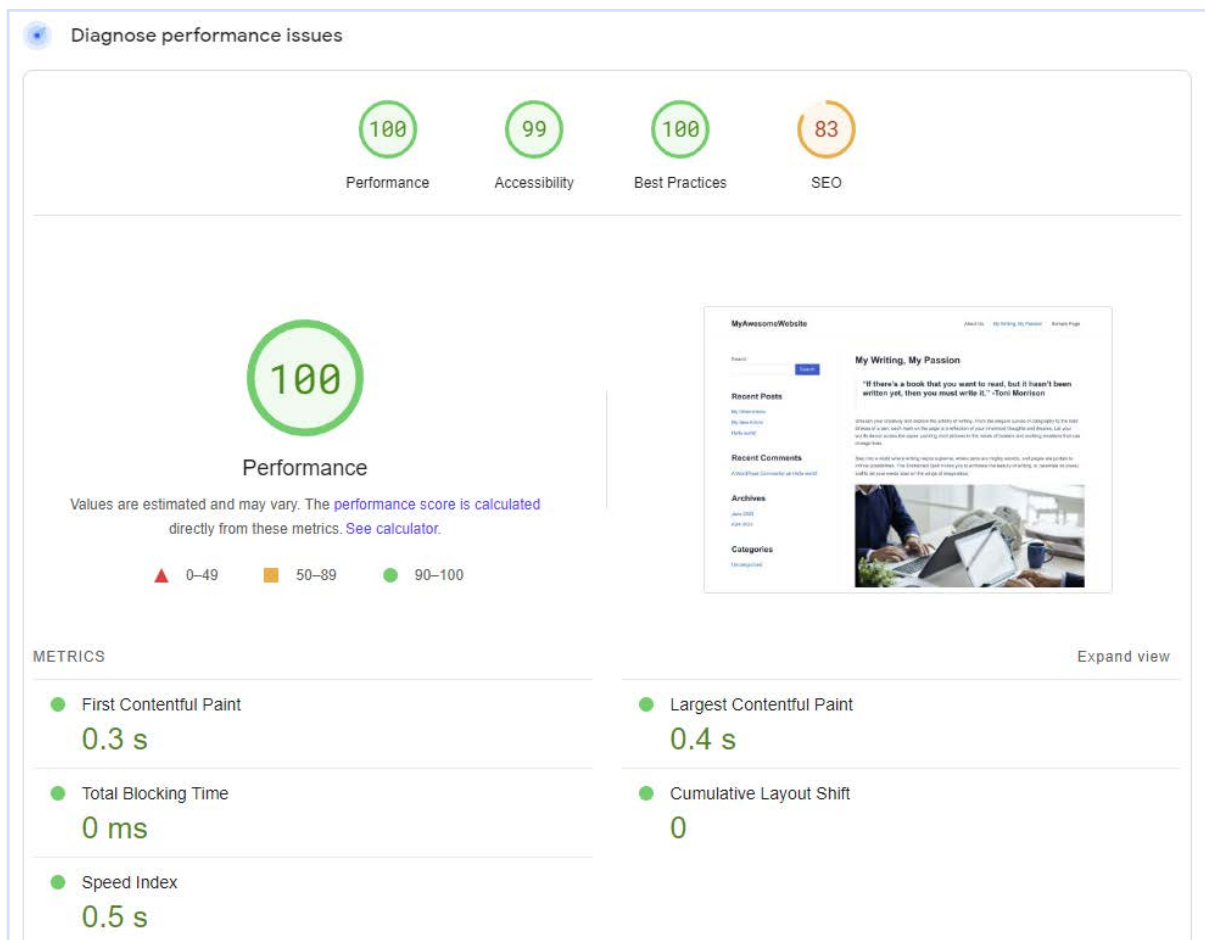
PASSED AUDITS (35) Hide

● Eliminate render-blocking resources

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn how to eliminate render-blocking resources.](#) FCP LCP

 There are a number of WordPress plugins that can help you [inline critical assets](#) or [defer less important resources](#). Beware that optimizations provided by these plugins may break features of your theme or plugins, so you will likely need to make code changes.

Speaking of PageSpeed Insights, here are the latest results after implementing minification and postponing the loading of non-critical resources:



Comparing these results with those from the previous chapter, there's a speed index reduction of about 0.2 seconds. This round of performance optimization offers the fastest overall loading times we've seen so far, including our initial tests in Chapter 2.

Remember, back when we started with a brand new website, we had loading times below half a second according to Pingdom. They increased to a little over one second after adding images and plugins.

Likewise, our initial PageSpeed Insights results were very similar to those we achieved in this chapter. However, our first significant rendering time increased after adding images and other content.

All this goes to show that you don't have to limit yourself to achieve high performance. With a little work, your content-rich site can load as well as a fresh [WordPress installation](#).



6

Enabling Browser Caching



[Click here to return to the table of contents](#)

Every time someone visits your website, they load the same elements over and over again. Unless you update your site every five minutes, it doesn't make much sense for browsers to perform the same task repeatedly, particularly since it can also negatively affect your server's performance.

The solution to this problem is called **caching**. This process prompts the browser to store parts of your website locally. That way, when visitors return to your website, they don't need to load entire pages from scratch. The fewer elements there are to render, the faster the process is.

For this chapter, we're going to dig into how browser caching works. Then we'll teach you how to implement it in [WordPress](#) and analyze its impact on our test website.

► Understanding Browser Caching

As you now know, a lot happens in the background every time someone visits your website. In most cases, the browser loads dozens of elements behind the scenes before anything appears on visitors' screens. Ideally, this process shouldn't take more than a second or two.

So far, we've focused on a variety of methods to cut down those loading times by:

Compressing or optimizing the elements your website requires to function, such as images and scripts.

Deferring elements to load later on, so visitors don't have to wait as long before they can see the above-the-fold content of your pages.

Reducing the number of elements that visitors have to load.

Usually, the third option is the last resort unless we're talking about website components that don't fulfill any purpose. Removing key elements just to reduce loading times is impractical and unnecessary.

There is, however, an interesting workaround that can reduce the number of elements users need to load without having to do away with any of them. It's called caching.

In simple terms, this process allows visitors to temporarily store parts of your website on their devices. Instead of repeatedly requesting content from your site's server, their browsers can simply retrieve the content from their caches.

Most of today's popular browsers support caching. However, it's up to you to configure it for your website.

For instance, you have to set timeout rules. This ensures visitors will load new content periodically and see changes to your website. Otherwise, visitors would keep seeing outdated versions of your content that are stored in their browsers.

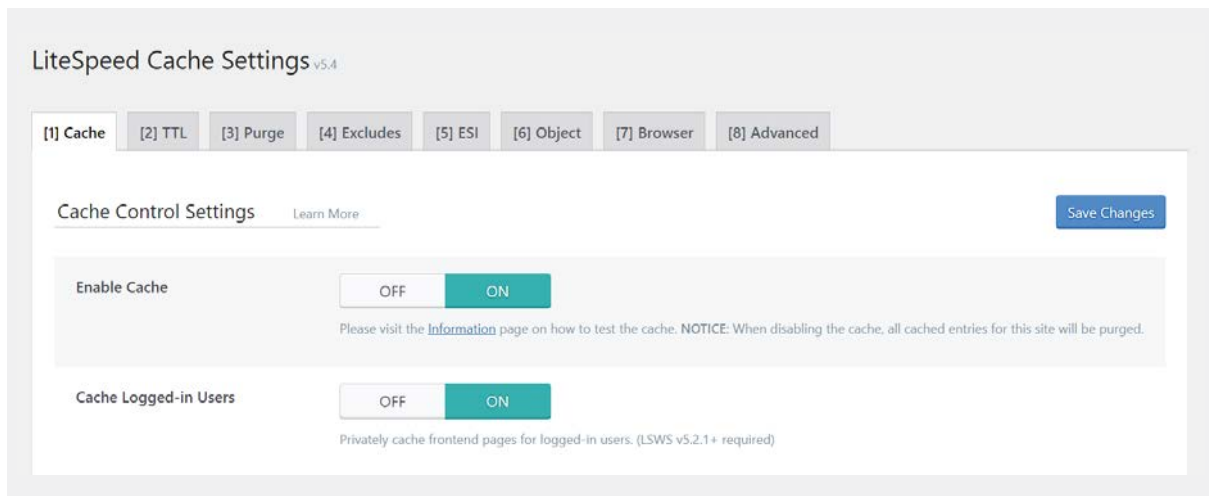
► How to Enable and Configure Browser Caching in WordPress (In 3 Steps)

Once more, we're going to rely on the [LiteSpeed Cache](#) plugin to help us implement this feature. While it's not the only option as far as caching plugins go, it does include a lot of settings that enable us to finetune our website's local storage configuration. Let's take a look at them.

1. Enable Caching and Configure Your Timeout Settings

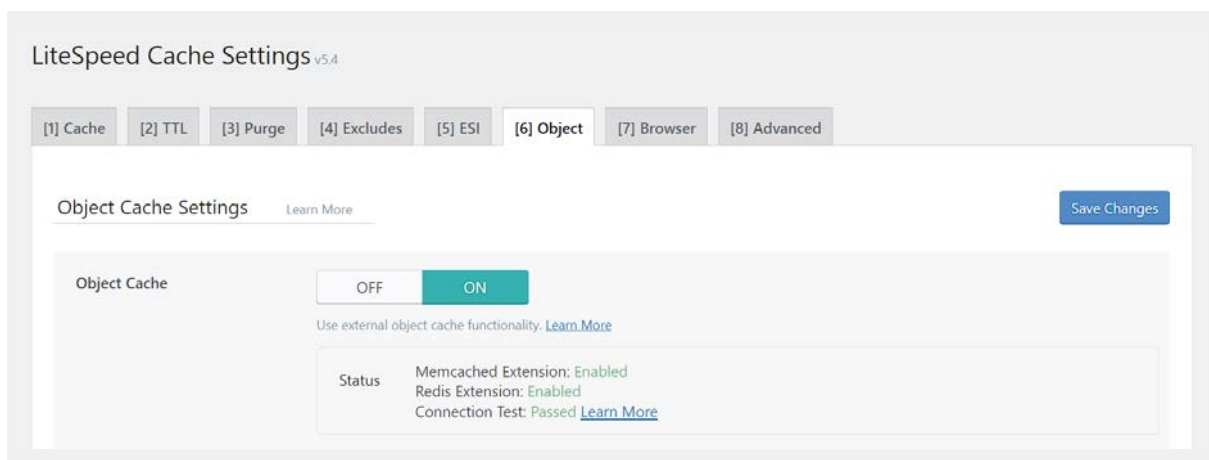
This time, we will redirect our attention to the **Cache** menu. Once you're in it, open the **Cache** tab.

By default, most caching settings here will be turned on. The plugin will automatically cache your website for logged-in users. Take a moment to ensure this option is enabled here before proceeding.

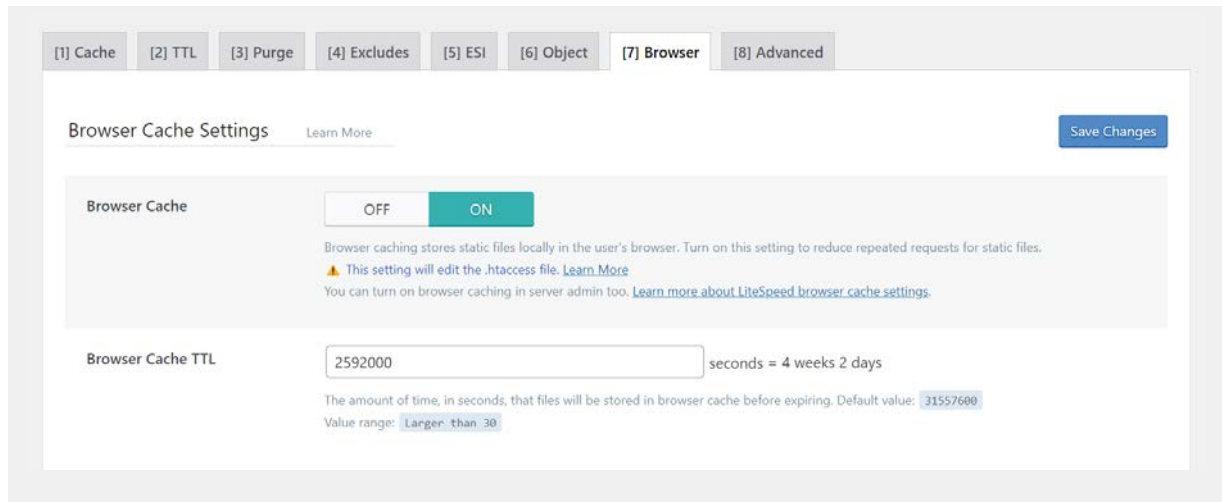


Next, navigate to the **Object** tab to enable object caching, which is different from browser caching. With object caching, you store database queries so users don't have to spend time processing them individually.

Let's leverage this feature by turning on the **Object Cache** setting.



Switching to the **Browser** tab, let's turn on the **Browser Cache** setting to speed up loading times for all your visitors. For our test website, we set the **Browser Cache TTL** value to 2592000 seconds, meaning the cache will expire after 30 days since users' first visit.



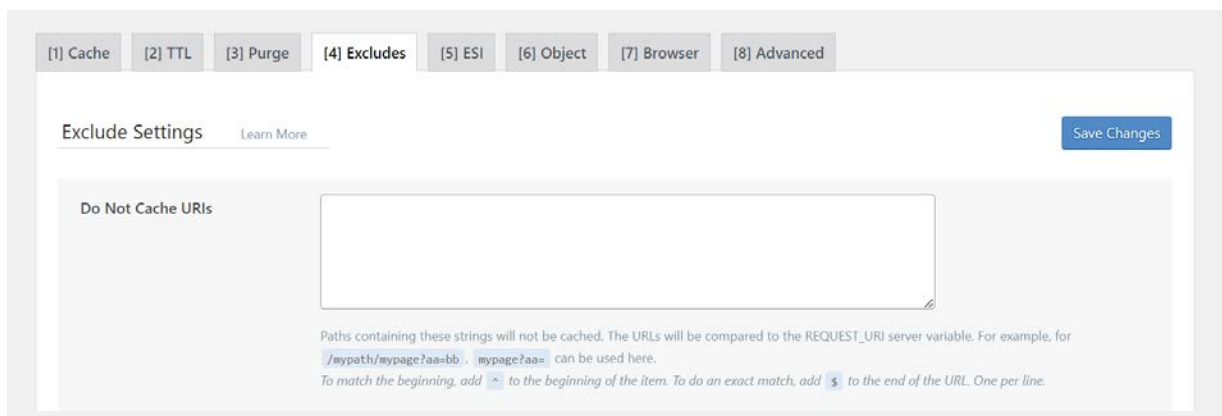
Don't forget to hit **Save Changes** once you're done.

Now that we've enabled caching for visitors and logged-in users, it's time to review which elements you want to store and which ones you don't.

2. Decide Which Elements to Cache (And Which Not To)

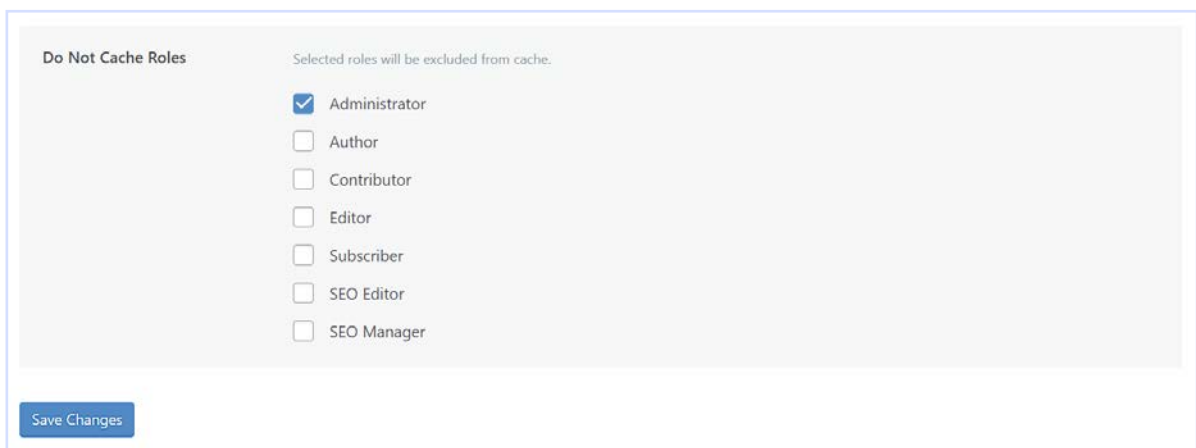
In practice, there are parts of your website you won't want to cache. For example, [WordPress](#) includes a search feature visitors can use to find content within your website. Caching search results doesn't work because that page generates dynamically depending on users' queries.

With LiteSpeed Cache, you can exclude specific pages, tags, categories, and user roles from caching. To exclude those pages, go to the **Excludes** tab and find the **Do Not Cache URLs** option. In the corresponding field, list the addresses for the content you deem too sensitive for caching:



Generally speaking, you want to enable caching for most pages on your website, particularly those with mostly static content. These might include your homepage, any other primary pages, and your blog content.

We also recommend excluding users with administrator roles from caching. This way, they'll always have access to the latest versions of your pages. Still in the **Excludes** tab, scroll down to Do Not Cache Roles and check the box next to Administrator.



Do Not Cache Roles Selected roles will be excluded from cache.

- ☒ Administrator
- ☐ Author
- ☐ Contributor
- ☐ Editor
- ☐ Subscriber
- ☐ SEO Editor
- ☐ SEO Manager

[Save Changes](#)

3. Tweak Your Cache Purge Settings

In Step 1, we set up a global expiry timer for your visitors' caches. However, you can further configure this element for specific parts of your site.

If you move to the **Purge** tab, you'll find options that lets you configure cache-busting or purge settings. This refers to specific situations where you want your website to tell visitors' browsers to reload a certain page, even though the expiration period hasn't passed yet.

First off, you'll notice a setting labeled **Purge All On Upgrade**. This option will clear users' caches whenever you update plugins, themes, or the WordPress core:



[1] Cache [2] TTL [3] **Purge** [4] Excludes [5] ESI [6] Object [7] Browser [8] Advanced

Purge Settings [Learn More](#) [Save Changes](#)

Purge All On Upgrade ☐ OFF ☒ ON

When enabled, the cache will automatically purge when any plugin, theme or the WordPress core is upgraded.

Further down, there's an option that reads **Auto Purge Rules for Publish/Update** alongside a broad range of page and content types. For each option you enable here, your website will purge the relevant files when you publish or update content within that category:

Auto Purge Rules For Publish/Update

Note
Select "All" if there are dynamic widgets linked to posts on pages other than the front or home pages. Other checkboxes will be ignored.
Select only the archive types that are currently used, the others can be left unchecked.

☐ All pages ☒ Front page ☒ Home page ☒ Pages

☒ All pages with Recent Posts Widget

☒ Author archive ☒ Post type archive

☐ Yearly archive ☒ Monthly archive ☐ Daily archive

☒ Term archive (include category, tag, and tax)

Select which pages will be automatically purged when posts are published/updated.

Let's say, for example, you tick the **Home page** option, as seen above. This means every time you make a change to your homepage, WordPress will automatically purge its cached files. By forcing users to reload the page, they won't miss out on your updated content.

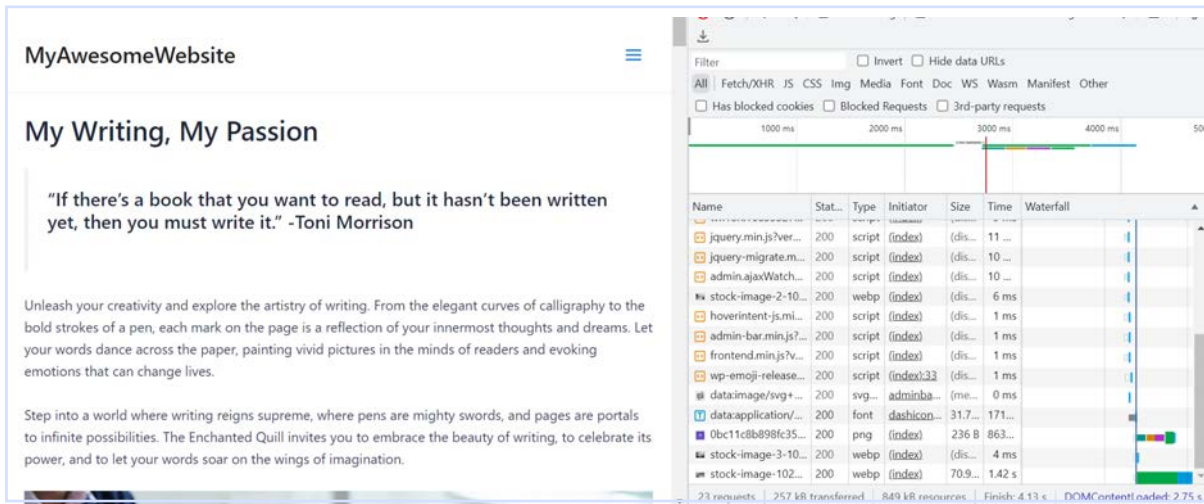
Combining the global timeout rules with granular cache purge settings, you can feel confident that when you make changes to your website, visitors will see them. At the same time, they should be able to enjoy drastically reduced loading times, which we're about to test now.

► The Impact of Browser Caching on Our Test Site

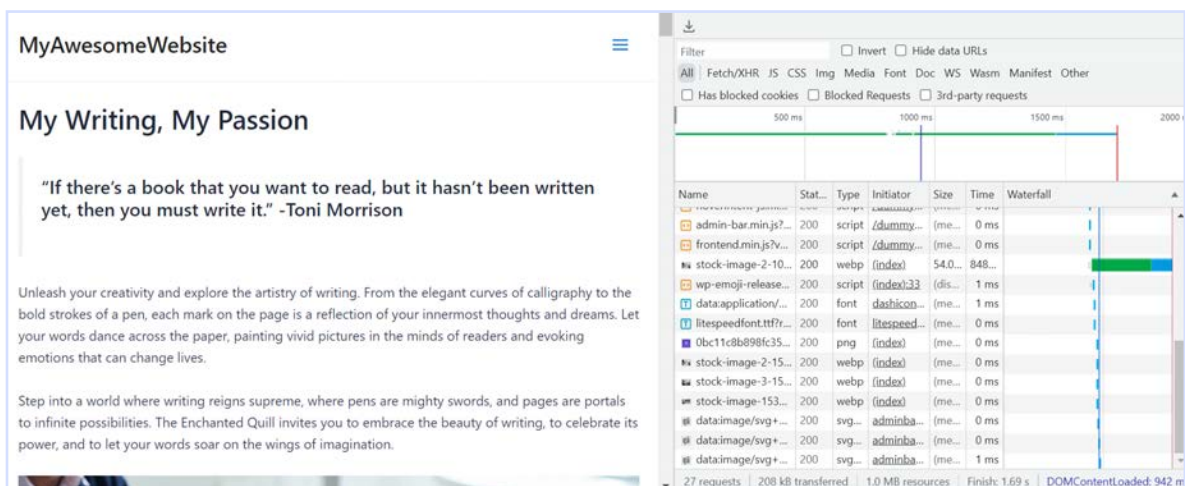
We know caching is one of the most effective techniques when it comes to optimizing website loading times. However, in practice, you're not making any changes to your website itself. Rather, you're telling visitors' browsers to hold on to part of its data in case they need it later on.

Unfortunately, this means that most website speed testing tools don't work all that well when it comes to measuring the impact of caching on your website. That's why we decided to run some local tests.

Here's a quick view of the Chrome Dev Tools' **Network** monitoring feature. You can use it to measure the impact of each element on a page's loading times. Before enabling caching, here's what our results looked like:



The first time we loaded the page, it took almost 4.13 seconds, which is not optimal. After enabling caching and reloading the content a couple of times, that number went down dramatically:



That same network test took just a little under two seconds to complete. As far as loading times go, those are significant savings.

Most websites you visit during your day-to-day browsing probably use caching in one way or another. Although loading times can vary depending on a lot of factors, using this simple technique is the best way to ensure they remain as smooth and fast as possible.



7

Integrating a Content Delivery Network



[Click here to return to the table of contents](#)

We've done a lot of work to speed up your website so far. That includes everything from choosing the right web host to on-site optimization.

Now, it's time to look outward to find more ways to serve your website to visitors even faster. That's where [Content Delivery Networks](#) (CDNs) come in.

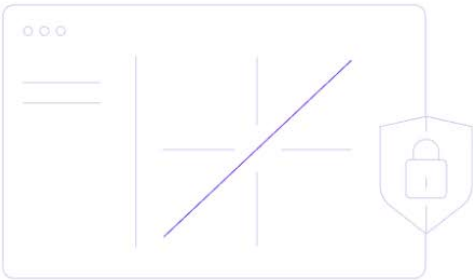
CDNs essentially function as middlemen that can help you cut loading times dramatically by using cached copies of your content stored across the globe. Whereas most hosting providers can only offer you a handful of server locations, CDNs do away with the limitation.

In this chapter, we're going to break down how CDNs work and what benefits they can bring to your website. Then we'll teach you how to integrate one with your WordPress site in four steps. Let's get down to it.

► An Introduction to Content Delivery Networks (CDNs)

Every website requires a physical server, which needs to be stored somewhere. Depending on your hosting provider, you might have several options when it comes to the location of yours.

For example, [Hostinger](#) has multiple data centers across Europe, Asia, North America, and South America:



DATA CENTERS

All Data – Secured

We protect your websites with multiple levels of fail-safes, RAID-10, daily or weekly backups. All of our 10 data centers are distributed globally to keep your websites fast in every country. Your visitors will enjoy the lowest latency and unmatched reliability as we deploy in globally connected Tier-3 datacenters.

Plans include:
In-house built hPanel, Access Manager, LiteSpeed web server, free SSL, in-house developed WAF.

Ultimately, a big part of website loading time comes down to server location. If your website's server is in Australia and someone from the United States tries to access your site, they will experience a delay of one or two seconds compared to a local user.

If you refer back to Chapter 2, you'll see we demonstrated this concept when we shared our initial Pingdom results. Our loading speeds varied depending on the location we selected for each round of testing.

On-site optimization, such as image compression, has its limitations. It doesn't address slow internet speeds or visitors located far away. While you can't improve the experience for users with dial-up connections, you can mitigate the impact of your server's location by utilizing a CDN.

CDNs store copies of your website in data centers all around the world, similar to how browsers cache your content as we described in the previous chapter. Requests are directed to the nearest server, resulting in faster loading times.

Additionally, CDNs help to reduce server load, allowing your website to handle high traffic without performance issues.

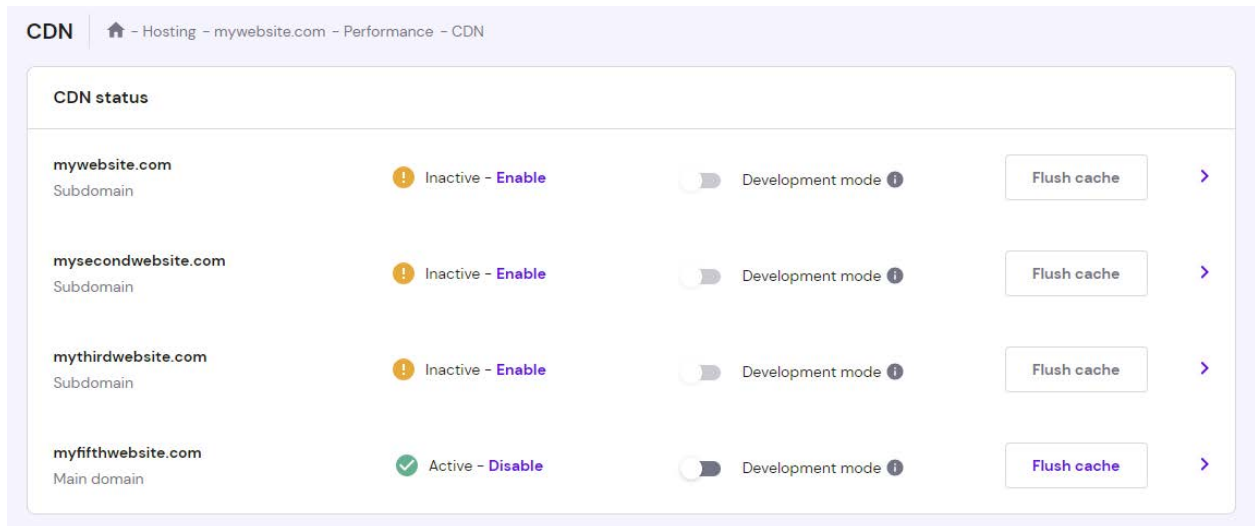
➤ How to Integrate a CDN With WordPress (In 4 Steps)

Integrating a [CDN with WordPress](#) is easier than you might imagine. However, a key factor in the process is which service provider you decide to use. Below, we'll start with our top suggestions, then walk you through the integration process.

1. Choose Which CDN to Use

CDNs provide an entirely different type of service than hosting. That means, in most cases, you'll need to sign up for a new account and possibly pay for that service in addition to storing your site with your web host. A few CDNs offer free plans, but those tend to be rather limited.

Hostinger eliminates the need for users to set up a CDN by offering a built-in solution. To enable [Hostinger CDN](#), simply go to **Performance** > **CDN** in your control panel. You only need to click **Enable** to turn it on, no configuration needed from your end.



This feature is currently available for servers located in the UK, US, France, Singapore, Brazil, and India, as it has recently exited the beta version. At the moment, our in-house CDN is integrated into the following hosting plans:

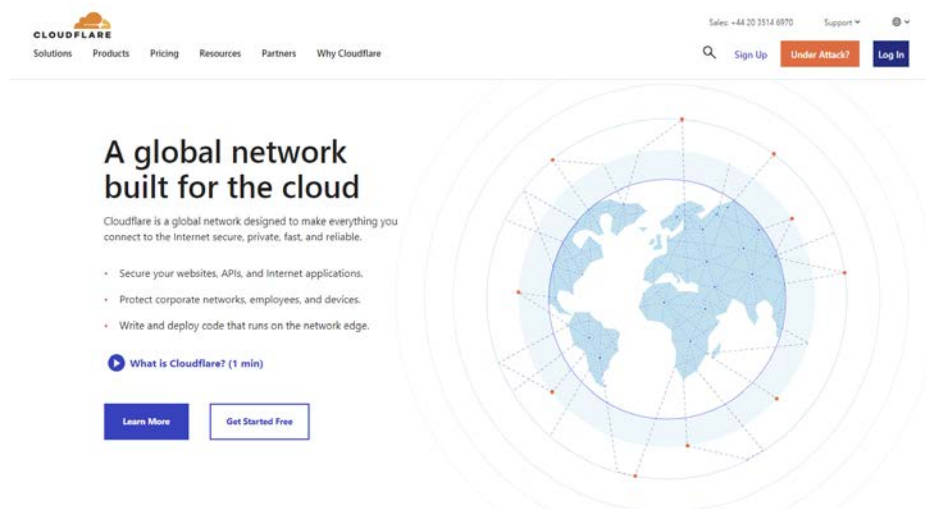
Business Web Hosting

Business WordPress

WordPress Pro

All cloud hosting plans

As far as which third-party CDN to use, we're partial to [Cloudflare](#) and [QUIC.cloud](#). The former is one of the most popular CDNs in the world, while the latter is an option built from the ground up with WordPress websites in mind.



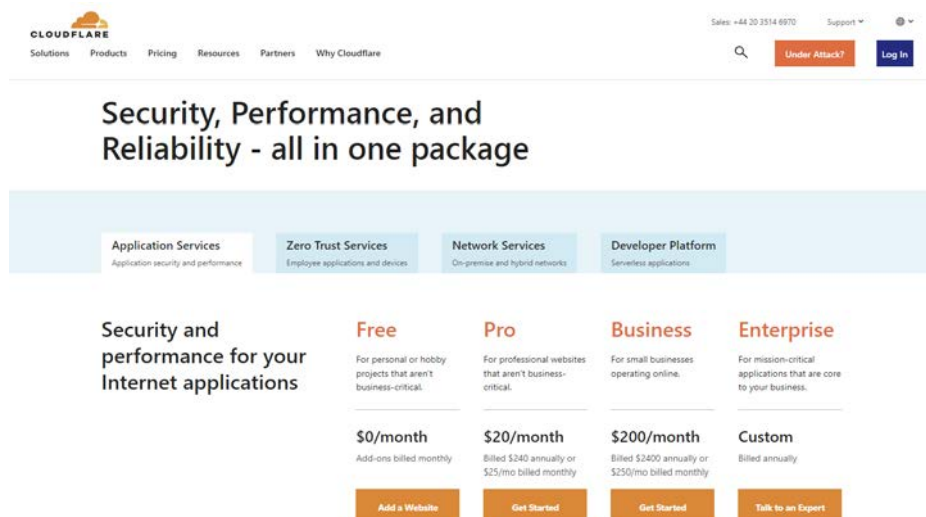
We'll guide you through setting up a CDN using Cloudflare, which has a free plan and is compatible with WordPress. However, we recommend researching other services before committing to a paid plan for your website.

2. Sign Up for Cloudflare and Retrieve Your API Key

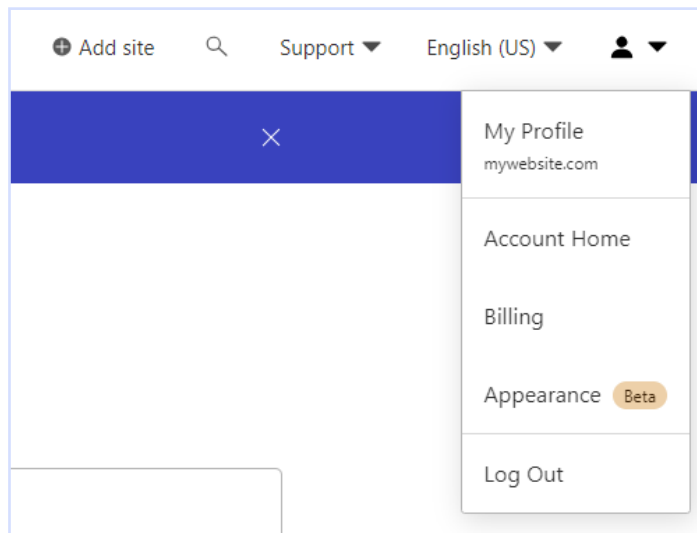
Signing up for a Cloudflare account is similar to any other online service. Fill out a few forms with your contact information and details, then select your desired plan.

For this tutorial, we're going to use the free plan to showcase that you don't need to have a massive budget to enjoy the benefits of using a CDN.

First, click on the **Sign Up** button at the top right corner of the page. When presented with the available options, pick the **Free** plan and proceed to creating a new account.



Once your account is verified, navigate to your Cloudflare dashboard and click on the profile icon at the top right corner. Select **My Profile** from the drop-down menu:



Open the **API Tokens** tab and click on the **View** button next to **Global API Key**. You don't need to set up a new token since we're going to use a plugin to connect your website to [Cloudflare](#).

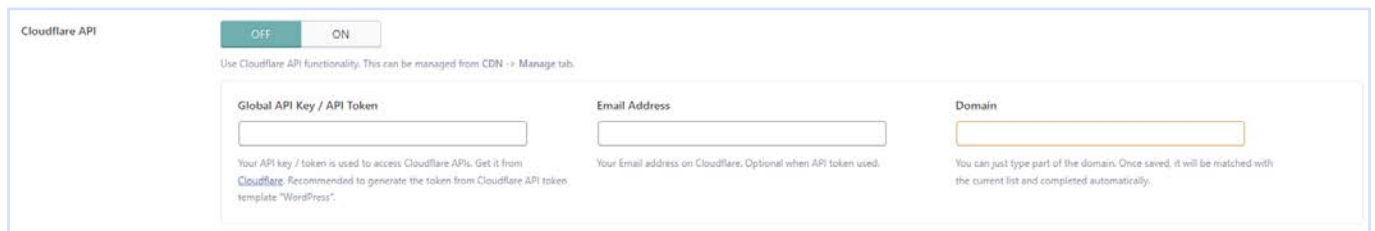
API Keys Keys used to access Cloudflare APIs.		
Global API Key	Change	View
Origin CA Key	Change	View
Help		

Cloudflare will ask you to confirm your account password before granting access to the API key. Once it displays your key, keep that tab open or copy it somewhere safe while you return to WordPress.

3. Connect Cloudflare With WordPress

As **LiteSpeed Cache** works with both QUIC.cloud and Cloudflare out of the box, we're going to use the plugin to set up the latter's CDN on your WordPress site.

Go to **LiteSpeed Cache > CDN** and scroll down until you see the **Cloudflare API** section.



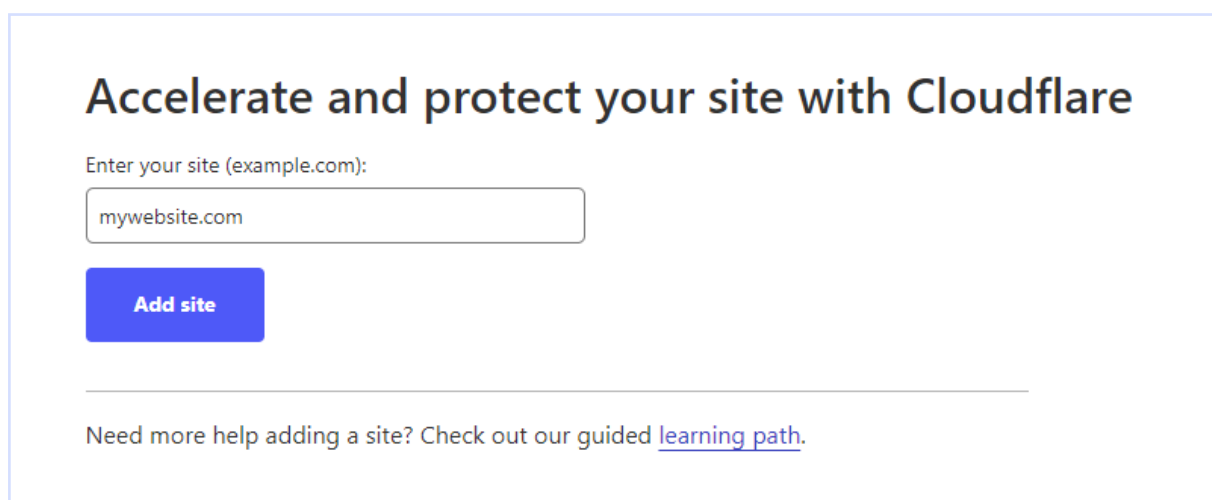
The screenshot shows the 'Cloudflare API' settings section. At the top, there is a toggle switch labeled 'OFF' and 'ON'. Below it, a note says 'Use Cloudflare API functionality. This can be managed from CDN -> Manage tab.' There are three input fields: 'Global API Key / API Token', 'Email Address', and 'Domain'. Each field has a small text box below it with instructions. The 'Global API Key / API Token' field has a note: 'Your API key / token is used to access Cloudflare APIs. Get it from Cloudflare. Recommended to generate the token from Cloudflare API token template "WordPress".' The 'Email Address' field has a note: 'Your Email address on Cloudflare. Optional when API token used.' The 'Domain' field has a note: 'You can just type part of the domain. Once saved, it will be matched with the current list and completed automatically.'

LiteSpeed Cache will ask for the email address used to register your Cloudflare account, as well as the API key you obtained in Step 2. Finally, you need to enter your website's domain name.

When you're done, turn on the **Cloudflare API** setting and save the changes.

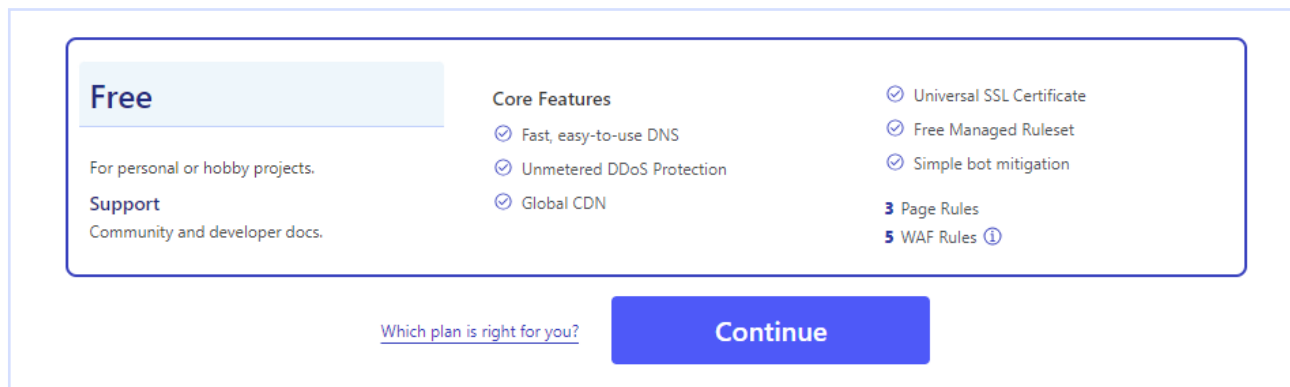
4. Add Your Website to Your Cloudflare Account

Before you can start seeing results from your new CDN, you need to return to your Cloudflare account dashboard and add your website there. On your Cloudflare **Websites** tab, click **Add Site** and enter your WordPress site's domain.



The screenshot shows the Cloudflare 'Accelerate and protect your site with Cloudflare' form. It has a title 'Accelerate and protect your site with Cloudflare' and a subtitle 'Enter your site (example.com):'. There is a text input field with 'mywebsite.com' entered. Below the input field is a blue button labeled 'Add site'. At the bottom, there is a link: 'Need more help adding a site? Check out our guided [learning path](#).'

Cloudflare will ask you to confirm the plan you've selected for your website. For our test website, we opt for the free tier, which is perfect for small sites.



Free

For personal or hobby projects.

Support
Community and developer docs.

Core Features

- ✓ Fast, easy-to-use DNS
- ✓ Unmetered DDoS Protection
- ✓ Global CDN

- ✓ Universal SSL Certificate
- ✓ Free Managed Ruleset
- ✓ Simple bot mitigation
- 3 Page Rules
- 5 WAF Rules ①

[Which plan is right for you?](#)

Continue




Cloudflare will scan your DNS records to make sure it can accelerate traffic to your website. Notice the Cloudflare icons below, which mean the service is working.



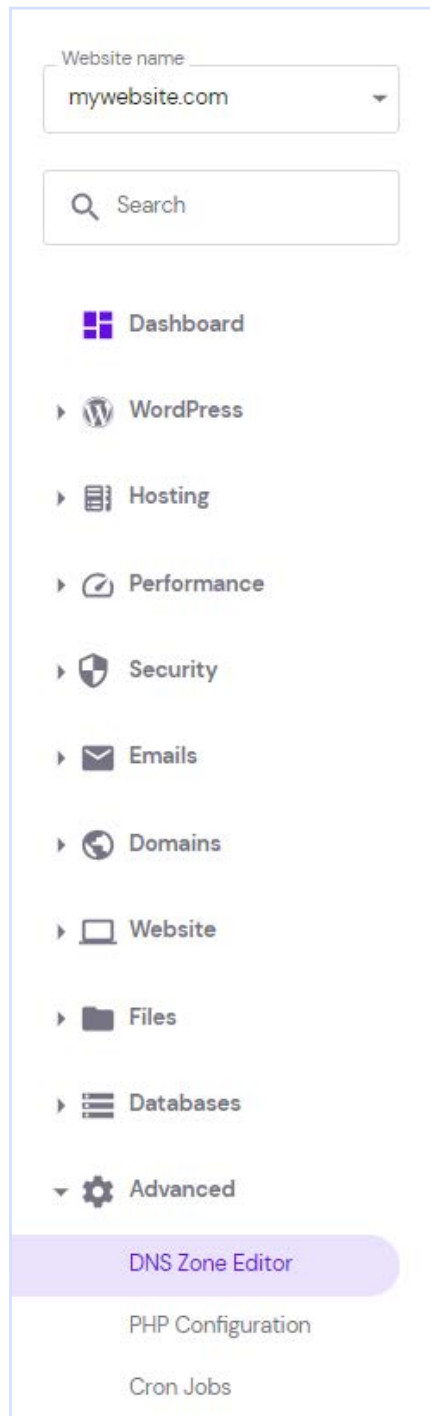
DNS management for **mywebsite.com** [Import DNS Records](#)

Search DNS Records

[Search](#) [+ Add record](#)

Type ▲	Name	Content	Proxy status	TTL	Actions
A	ftp		<input checked="" type="checkbox"/>  Proxied	Auto	Delete
A	staging		<input checked="" type="checkbox"/>  Proxied	Auto	Delete
AAAA			<input checked="" type="checkbox"/>  Proxied	Auto	Delete

To complete the setup process, Cloudflare will ask you to update your domain's nameservers. If you're a Hostinger customer, go to **Advanced > DNS Zone Editor** from your hosting account dashboard.



Use the search bar to look for “NS”, which represents your nameservers:

The image shows a search bar with the text 'NS' and a magnifying glass icon. Below it is a table of DNS records.

Type	Name	Priority	Content	TTL	
NS	@	0	ns2.dns-parking.com	86400	Delete Edit
NS	@	0	ns1.dns-parking.com	86400	Delete Edit

Replace your default nameservers with the ones provided by Cloudflare:

The image shows a guide titled 'Pointing to Cloudflare's nameservers is a critical step in activation and must be complete for Cloudflare to optimize and protect your site.' It includes a list of steps: 1. Determine your registrar via WHOIS. 2. Log in to the administrator account for your domain registrar. 3. Remove the following nameservers (ns2.dns-parking.com and ns1.dns-parking.com). 4. Add Cloudflare's nameservers (alexa.ns.cloudflare.com and christian.ns.cloudflare.com). 5. Save your changes. A note states that registrars can take 24 hours to process updates. A 'Done, check nameservers' button is at the bottom.

Pointing to Cloudflare's nameservers is a critical step in activation and must be complete for Cloudflare to optimize and protect your site.

① Nameservers are your primary DNS controller and identify the location of your domain on the internet.

- Determine** your registrar via [WHOIS](#).
- Log in** to the **administrator account** for your domain registrar
- Remove** the following nameservers
ns2.dns-parking.com
ns1.dns-parking.com
- Add** Cloudflare's nameservers
alexa.ns.cloudflare.com [Click to copy](#)
christian.ns.cloudflare.com [Click to copy](#)
- Save** your changes

Registrars can take 24 hours to process nameserver updates. You will receive an email when your site is active on Cloudflare.

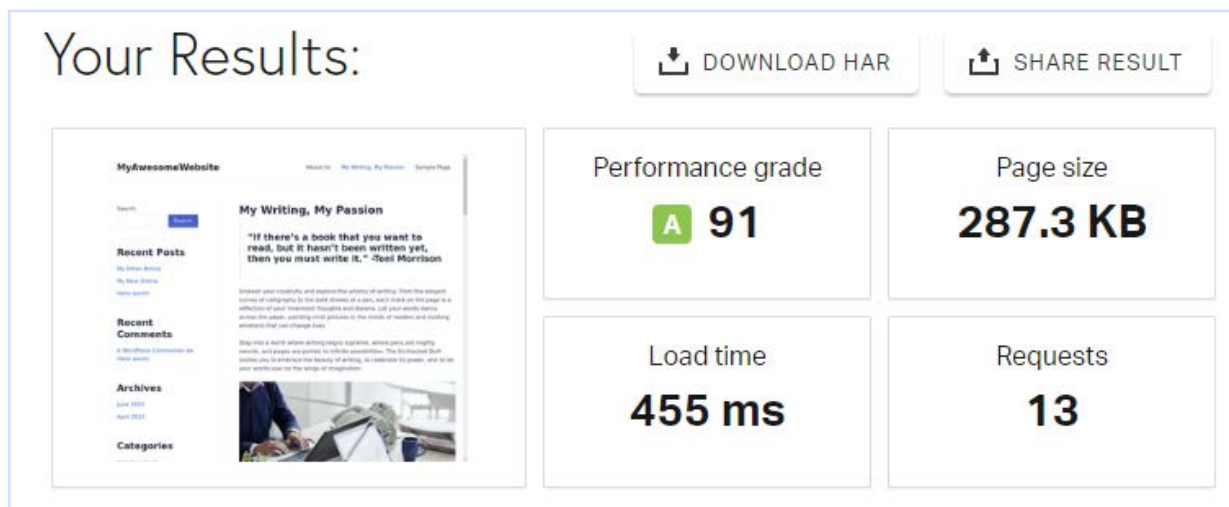
Done, check nameservers

This step enables Cloudflare to intercept traffic to your website and route it through its servers, cutting down your site's loading times. Once you're done, you can test your site to see the difference it makes.

➤ The Impact of Using a CDN on Site Speed

By the end of Chapter 6, our test website was loading in less than two seconds during Chrome's network throttling testing. Meanwhile, the Pingdom test clocked in at 453 ms at the end of Chapter 5.

After integrating a CDN, here's how our page is performing from Pingdom's Washington server:



We also tested our website using servers in various locations and compared those numbers with our results from Chapter 2. You'll see the initial result to the left and the latest numbers to the right:

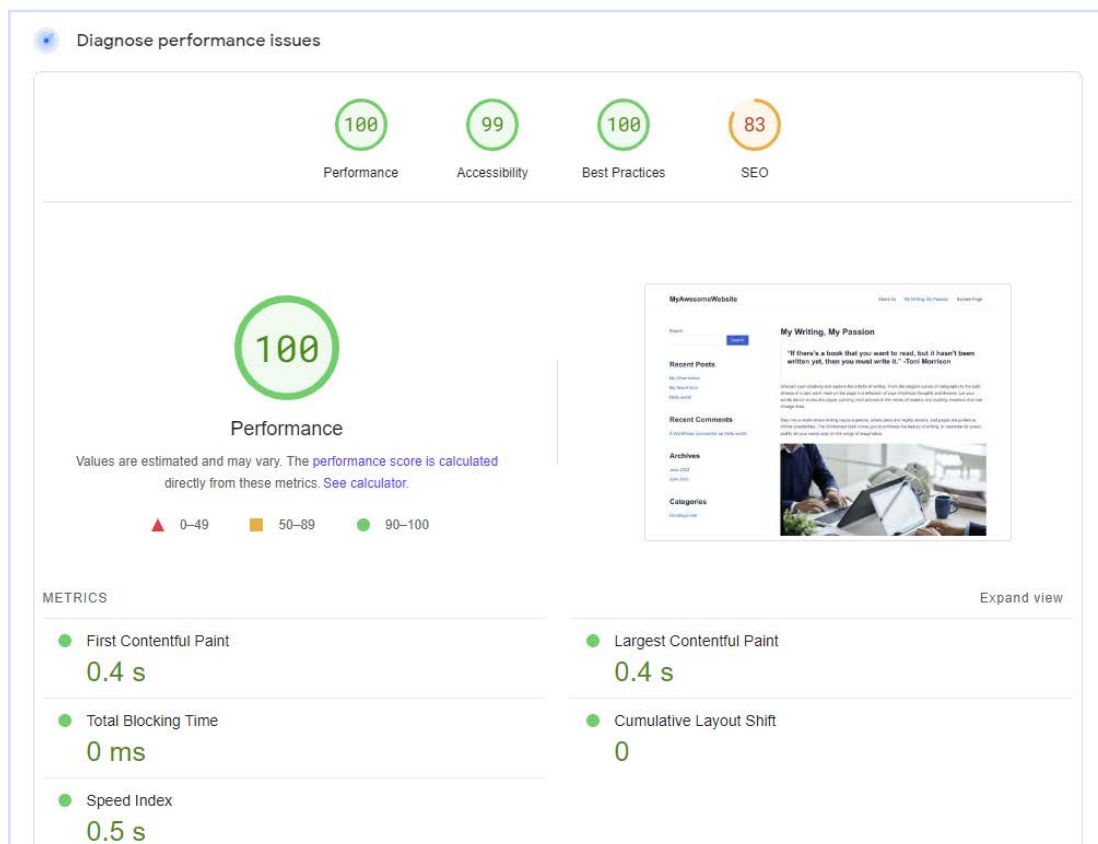
1. **San Francisco, USA:** 0.475 seconds / 0.438 seconds
2. **London, United Kingdom:** 0.434 seconds / 0.475 seconds
3. **Washington D.C., USA:** 0.375 seconds / 0.455 seconds
4. **Tokyo, Japan:** 1.2 seconds / 0.926 seconds
5. **Sydney, Australia:** 0.93 seconds / 1.46 seconds
6. **Frankfurt, Germany:** 0.14 seconds / 0.536 seconds

It's important to put these numbers into context. On the left, you have the loading times for a website that includes essentially zero content or images, weighing just 25 KB in our initial test. Now, after five chapters, it has grown to 287.3 KB, over ten times its original size.

Taking that into account, the minimal increase in loading times we see in these results are in line with what we'd expect from a well-optimized website. Without all the steps we've gone through so far, our website would be taking a much longer time to load.

► Testing Our Optimized Site With PageSpeed Insights

To wrap things up, let's check if PageSpeed Insights tells a similar story to Pingdom:



Our loading times, as per PageSpeed Insights, remain consistent with the results from the previous chapter. It's important to note that Pingdom is the preferred platform for tracking global loading times. Therefore, we anticipate a more substantial difference in results from that test compared to Google's.

Optimizing a young website can be challenging. However, as your website grows with more content and traffic, you'll experience the impact of improving its performance. Proper maintenance is often overlooked, resulting in slow-loading websites. Now that you've seen the results, you understand the competitive advantage it can provide.



8

Making Sure Your Website Remains Fast



[Click here to return to the table of contents](#)

You've put in a lot of work to improve your website's performance, so you want to make sure all that effort doesn't go to waste in the long run. That means keeping up with site maintenance. It shouldn't take much of your time, but it's an ongoing effort.

If you monitor your website's performance often, you'll be able to spot any dips in efficiency and identify what's causing them. Keeping your site fast will ensure your visitors always have a quality experience. Plus, maintenance work, in general, is a must for any growing site.

In this final chapter, we're going to walk you through some basic maintenance tasks to keep your website running fast. Let's get started.

► How to Make Sure Your Website Remains Fast (5 Key Tips)

Monitoring plays a crucial role in maintaining your website's performance. By keeping track of its loading speed for visitors and assessing the responsiveness of your admin dashboard, you can make necessary adjustments to ensure it works optimally.

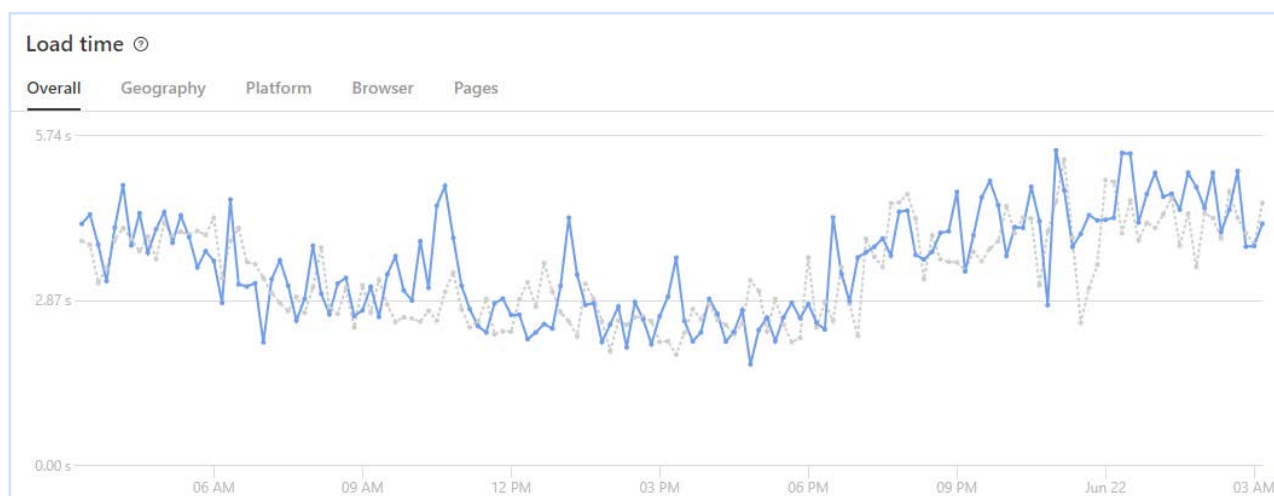
1. Monitor Your Loading Times Periodically

Over the past chapters, we've introduced you to several tools you can use to monitor your website's loading times. Specifically, we've dealt extensively with Pingdom Tools and PageSpeed Insights, which you can use to run simple performance tests at a moment's notice.

Here's the downside, however – you have to run those tests manually. They take less than a minute to execute, but since they can't be automated, it's easy to forget testing periodically. Your best bet if you want to make sure you don't miss anything is to set reminders to assess your website's speed at least weekly.

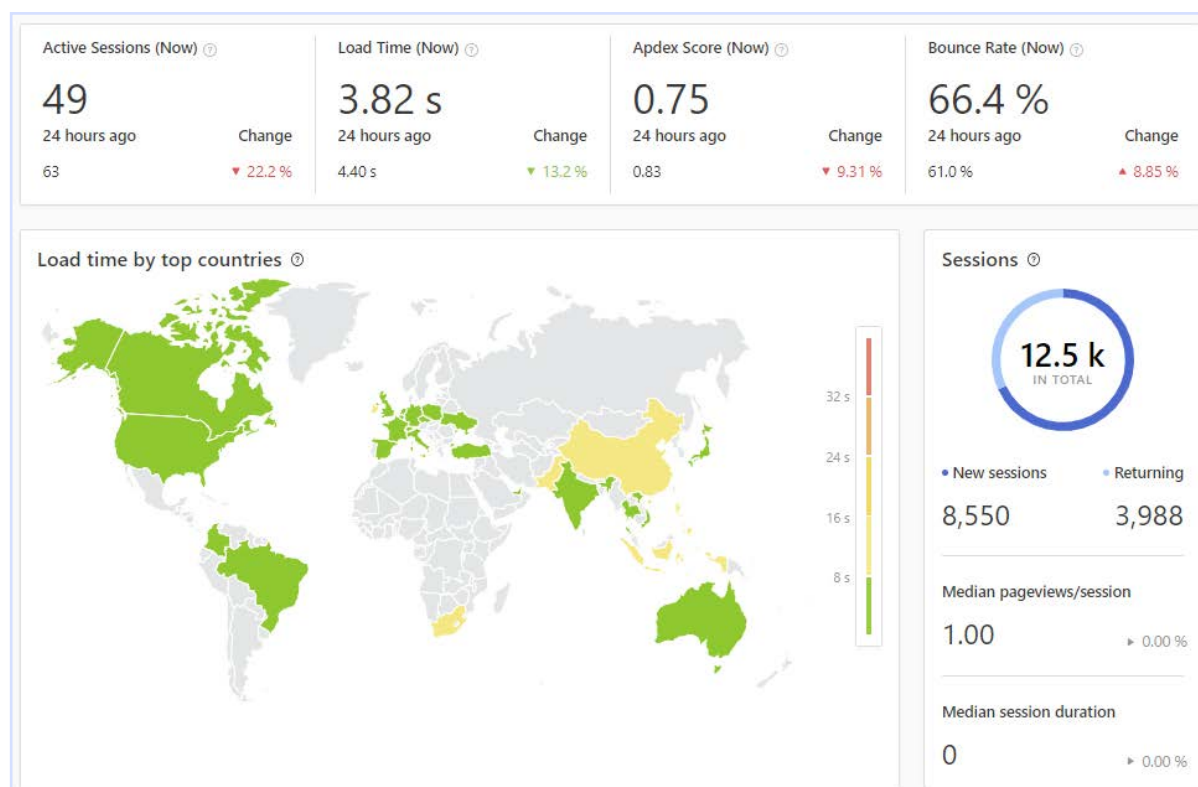
If you have extra budget to spend, you can sign up for premium website monitoring tools and leverage their automation feature. Pingdom Tools, for example, offers a suite of premium services to automate performance testing.

With its suite of services, you can also receive graphical data representation for easy analysis. Additionally, the tool can notify you if the load time exceeds your specified threshold.

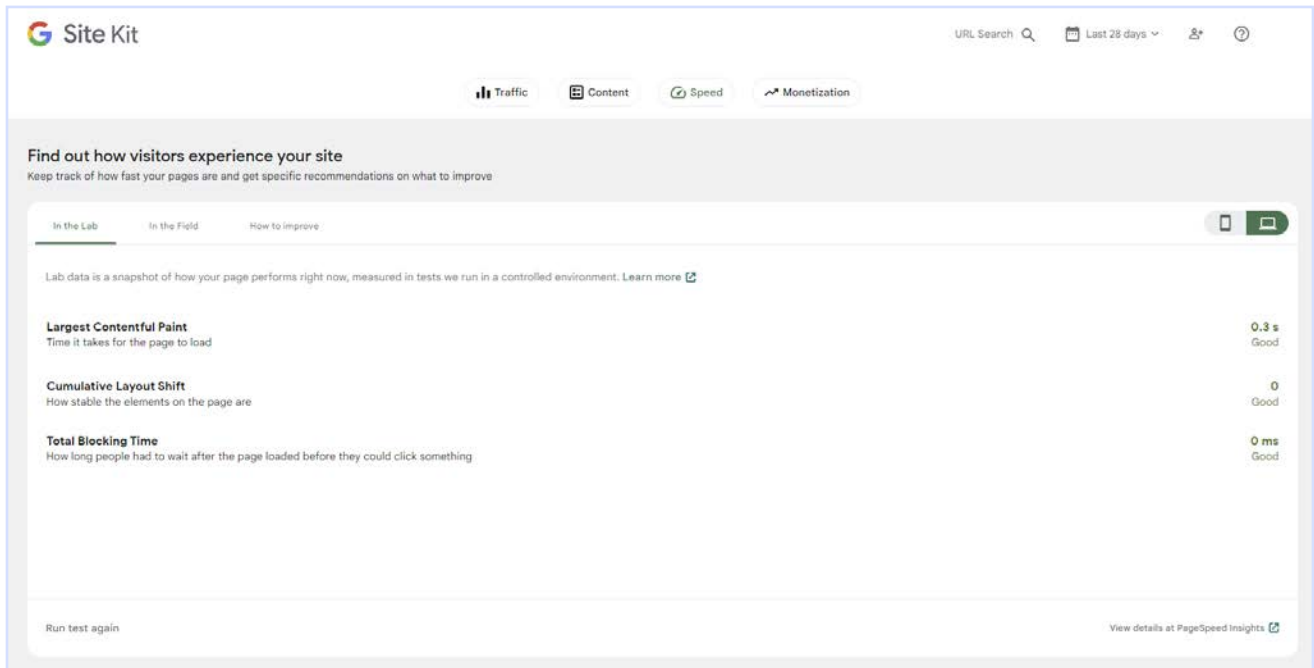


Many premium website maintenance platforms offer Real-User Monitoring (RUM), another premium feature that lets you monitor real-life user performance. That means you can know, at all times, whether users are experiencing slowdowns or if everything is working as intended.

Here's an example of Pingdom Tool's Real-User Monitoring dashboard, where you can see various useful insights like active sessions and current load time:



If you want to use PageSpeed Insights, you can use the [Google Site Kit](#) plugin to integrate the tool with your WordPress website. This way, you can note any drastic changes without having to leave your [WordPress dashboard](#).



However, it's important to note that you will still need to visit the PageSpeed Insights page to access detailed results and recommendations for improving performance.

2. Deactivate Plugins You're Not Using

As seen in Chapter 2, some plugins can have an impact on your website's performance. While their influence may be minimal in some cases, our goal is to optimize your website's performance down to the last millisecond.

If you're not using a certain plugin, there's no logical reason to keep it around. The correct approach is to uninstall or, at the very least, deactivate the plugin.

To ensure thorough removal of plugins from your site, don't forget to check your [WordPress database](#) and delete any leftover tables or rows. For instructions on accessing and managing your database, refer to the MySQL section of our [Knowledge Base](#).

3. Keep WordPress and All Its Components Up to Date

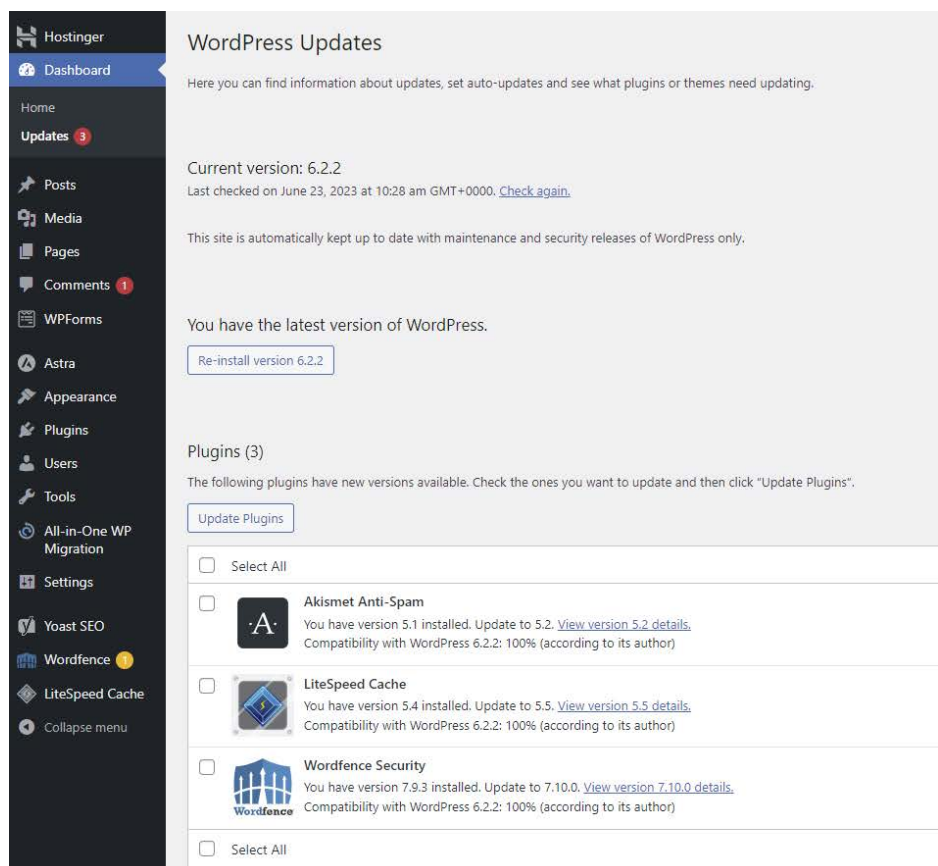
We've barely talked about updates so far, but that's not because they aren't important. Your WordPress website should always be **fully up-to-date**, which means:

Using the latest version of WordPress core.

Installing new versions of your plugins and the active theme.

On some occasions, updating to the latest version of WordPress immediately after it comes out may cause compatibility issues with your plugins or theme. However, those occasions are rare, and as long as you regularly **back up your website**, they shouldn't cause significant problems.

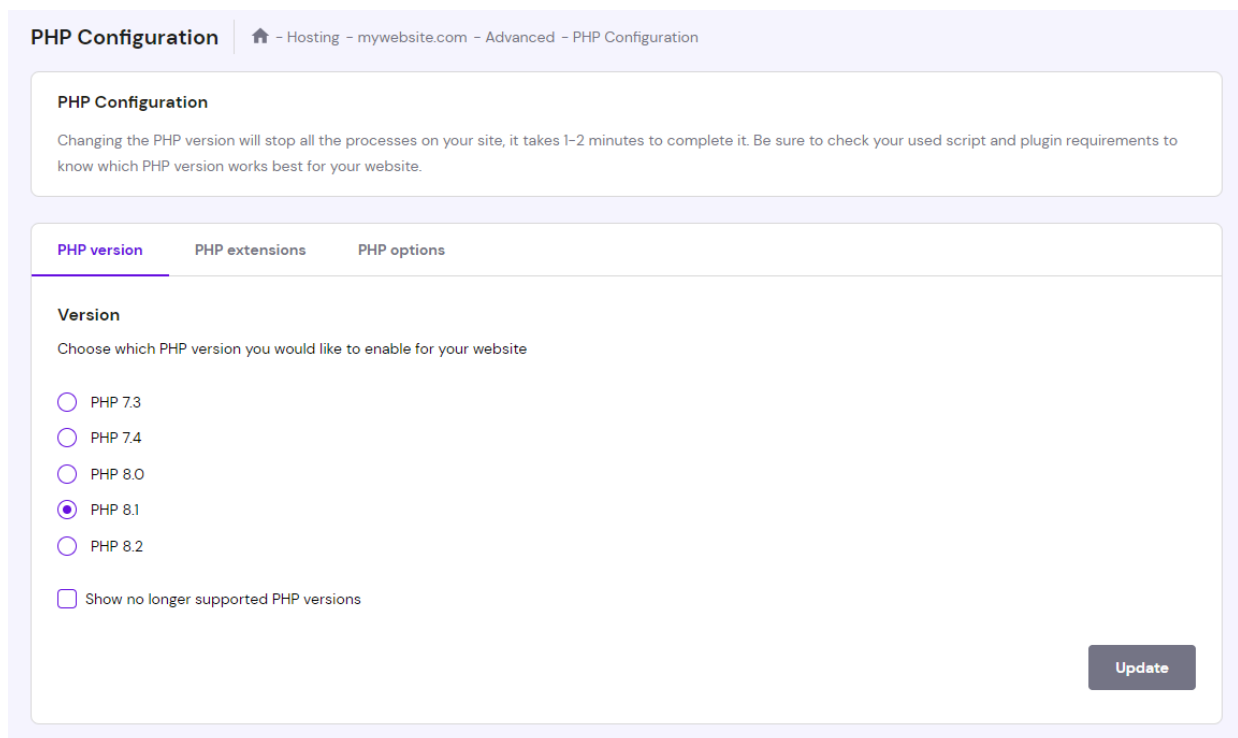
Any available updates for your plugins, theme, and WordPress core will be listed in the **Updates** tab in your dashboard. You'll also see red notifications in your sidebar. Simply click on the indicated buttons or links to run individual or bulk updates as needed.



4. Upgrade to the Latest Version of PHP

WordPress relies on PHP, and the version of PHP used by your server greatly influences your website's performance. For perspective, PHP 8 is three times faster than PHP 5.

Fortunately, some web hosts – [including Hostinger](#) – let you change PHP versions directly from your control panel. Navigate to **Advanced > PHP Configuration** and pick the PHP version you want to use:



The screenshot shows the 'PHP Configuration' interface in a web host control panel. At the top, there's a breadcrumb trail: 'Home - Hosting - mywebsite.com - Advanced - PHP Configuration'. Below this is a section titled 'PHP Configuration' with a warning: 'Changing the PHP version will stop all the processes on your site, it takes 1-2 minutes to complete it. Be sure to check your used script and plugin requirements to know which PHP version works best for your website.' Below the warning are three tabs: 'PHP version' (active), 'PHP extensions', and 'PHP options'. Under the 'PHP version' tab, there's a heading 'Version' and a prompt: 'Choose which PHP version you would like to enable for your website'. There are five radio button options: 'PHP 7.3', 'PHP 7.4', 'PHP 8.0', 'PHP 8.1' (which is selected), and 'PHP 8.2'. At the bottom left, there's a checkbox labeled 'Show no longer supported PHP versions' which is unchecked. At the bottom right, there's a dark 'Update' button.

If your host doesn't offer this feature you will have to [switch PHP versions](#) manually. Consider this factor when choosing your web host, as the manual PHP configuration requires technical expertise.

5. Consider Upgrading Your Hosting Plan Depending on Your Traffic Volume

Over the last seven chapters, we've covered pretty much everything you can do to ensure your website loads as quickly as possible. By now, you understand that although your choice of web host and plan matters a lot, so does the optimization work you do on your end.

As your website grows, its traffic will also increase. When your site outgrows its current hosting plan, consider upgrading to a hosting plan that offers more resources.

Virtual Private Server (VPS), for example, lets you scale server resources according to the growing traffic volume. Hostinger offers [VPS hosting plans](#) with Kernel-based virtualization (KVM), providing even better performance and stability.

Launch Your Own Website With Unlimited Hosting

Use **HOSTINGERTUTORIALS**
discount code and get an additional
10% OFF for any web hosting plan.

Use Code



➤ Conclusion

If you want your website to succeed and retain visitors, you must provide an enjoyable user experience. This involves creating high-quality content, engaging with your audience, and most importantly, ensuring fast loading times.

Ideally, your website should load in under two seconds, with a lower number being better. Once it reaches the three-second mark, users may lose interest and leave your pages.

Fortunately, improving page loading times doesn't have to be complicated. You can start by establishing a solid foundation for your website. In this eBook, we've gone over each step of the optimization process, including:

1. Selecting a fast hosting provider.
2. Choosing the right plugins and theme for optimal performance.
3. Compressing and optimizing images.
4. Minifying CSS and JavaScript files.
5. Enabling browser caching.
6. Integrating a Content Delivery Network (CDN).
7. Ensuring long-term website speed.

There are various factors that influence website loading times, but you have control over most of them. By investing time into optimizing your site's performance now, you will reap significant benefits in the long run. Moreover, your users will be happier, which is the ultimate measure of your website's success.