

Speed Up Your Website

8 Practical Tips that Work



Table of Contents

Click or tap to jump to any section:

2 Choosing a Fast
Hosting Provider

15 Setting Up Our
Test Website

28 Selecting the Right Plugins
and Theme for Optimal
Performance

40 Optimizing Your
Images

52 Minifying Your Website's
CSS and JavaScript

60 Enabling Browser
Caching

71 Integrating a
Content Delivery
Network

83 Making Sure Your
Website Remains Fast

Introduction

It's easy to overlook website performance, especially when you're first starting out. However, page loading times are vital to Search Engine Optimization (SEO) as well as for providing a quality User Experience (UX) to your visitors.

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

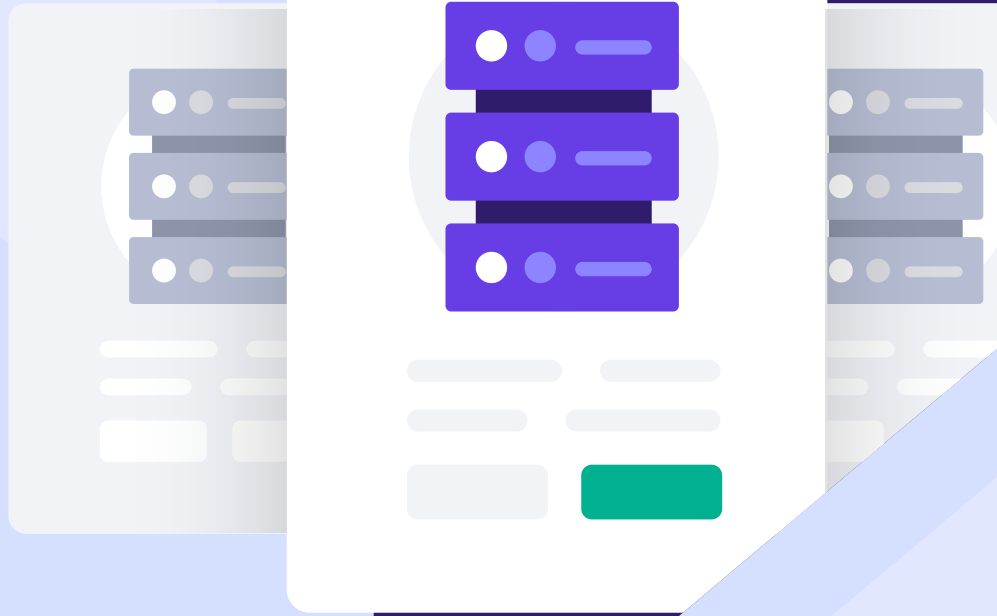
Internet users have high standards when it comes to site speed. Meeting or exceeding their expectations will make your site a much more attractive prospect.

Throughout this e-book, 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. Optimizing your images.
4. Minifying your website's CSS and JavaScript.
5. Enabling browser caching.
6. Integrating a Content Delivery network.
7. Ensuring your website remains fast.

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

Speed Up Your Website;
8 Practical Tips that Work



1

Choosing a Fast Hosting Provider



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 than their peers.

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're going to teach you how to pick a web host that will 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 for your website, and how to sign up for a Hostinger account. Let's get to work.

➤ What to Look for in a Hosting Provider

There are a lot of factors that set different 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** per month.

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

A decent variety of plans: Since you want your website to grow in the future, the web host you sign up for needs to offer a broad range of plans to choose from.

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

Features that make your life easier: This may include backups, easy WordPress setup, heightened security, and more.

Great performance out of the box: Some providers offer outright better performance than others.

You may be wondering how you can tell if a web host offers excellent performance without purchasing a plan. Looking at reviews is one route. However, you can often get an idea of how well-optimized a provider is by examining their tech stack.

Hostinger, for example, has [data centers around the world](#). You can pick whichever one is closest to your audience to cut down on loading times for most of your users. Beyond that, all of our plans support integration with Content Delivery Networks (CDNs), customizable server-level caching, Solid-State Drive (SSD) storage, and more.

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

➤ How to Choose the Right Hosting Plan for Your Website

As we mentioned before, a quality web hosting provider will give you plenty of options when it comes to plans. However, you still need to know which one is best suited to your needs.

Every hosting provider offers unique plans you can pick from, so the [options available](#) to you will vary based on the companies you're looking into. However, for the vast majority of websites, these are the five you should focus on choosing between:



WordPress hosting. A specialized hosting plan, specifically optimized for WordPress websites. It shares a lot of technical similarities with shared hosting.



Shared hosting. This type of plan tends to be the cheapest since you share server resources with other users. This comes with the downside of becoming dependant on the other websites on the server.



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



Cloud hosting. This type of hosting combines the simplicity of shared plans with superior scalability. You can gain access to more resources as you need them since your website will be distributed across many servers.



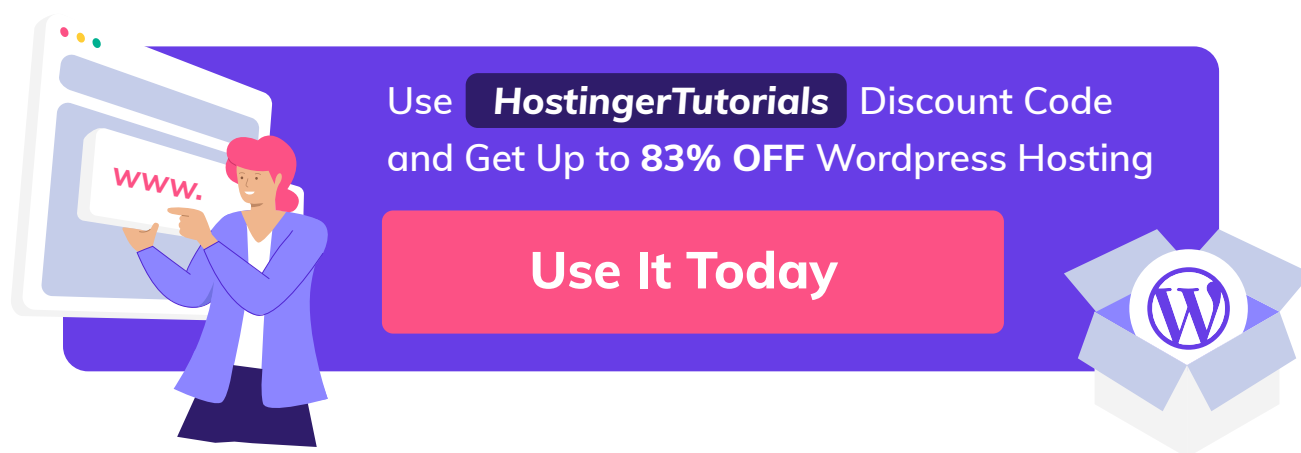
Managed hosting. With a managed plan, your provider takes care of performance optimization and maintenance, so you can focus on running your website. Naturally, that often comes at a more premium cost.

The more powerful the hosting plan, the more it is likely to cost. As a rule of thumb, if you're starting a new WordPress website, we recommend you go with either WordPress hosting or a VPS. The latter usually provides more room for growth and better performance out of the gate.

However, if your web host offers excellent performance, even shared plans can be pretty fast. With that in mind, let's check out how to proceed.

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

Now that we've gone over the basics of web hosting, we're going to show you what it looks like to sign up for a new plan. For this tutorial, we'll use one of our own Hostinger packages. However, the steps remain much the same regardless of which company you choose.



1. Select and Purchase a Hosting Plan

Signing up for web hosting tends to be pretty straightforward. With Hostinger, all you need to do is identify the plan you want and click on the **Add to cart** button. For our test website, we're going to use the **WordPress Starter Hosting** plan.

Choose Your WordPress Hosting Plan

30-day money-back guarantee.

Single WordPress	WordPress Starter	Business WordPress	WordPress Pro
Ideal solution for beginners	Perfect package for personal websites	Optimized for small and medium businesses	Advanced solution for revenue generating businesses
\$7.99 SAVE 75%	\$14.99 SAVE 80%	\$15.99 SAVE 50%	\$29.00 SAVE 60%
\$1.99 /mo	\$2.99 /mo	\$7.99 /mo	\$11.59 /mo
USD	USD	USD	USD
✓ 1 Website ✓ 1 Email Account ✓ 10 GB SSD Storage ✓ Limited Bandwidth (100 GB) ✓ Free SSL ✓ Managed WordPress ✗ Jetpack Free	✓ 100 Websites ✓ 100 Email Accounts ✓ 20 GB SSD Storage ✓ Unlimited Bandwidth ✓ Free SSL & Domain ✓ Managed WordPress ✓ Jetpack Free	✓ 100 Websites ✓ 100 Email Accounts ✓ 100 GB SSD Storage ✓ Unlimited Bandwidth ✓ Free SSL & Domain ✓ Managed WordPress ✓ Jetpack Free	✓ 300 Websites ✓ 100 Email Accounts ✓ 100 GB SSD Storage ✓ Unlimited Bandwidth ✓ Free SSL & Domain ✓ Managed WordPress ✓ Jetpack Personal
Add to cart	Add to cart	Add to cart	Add to cart
\$3.99/mo when you renew	\$7.99/mo when you renew	\$11.99/mo when you renew	\$19.99/mo when you renew
See all features	See all features	See all features	See all features

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

You're almost there! Complete your order

1. Choose a period

Selected plan: **WordPress Starter**

30-DAY MONEY-BACK GUARANTEE

You save the most by choosing this plan!

1 month	12 months	24 months	48 months
☐	☐	☐	☒
\$14.99	\$14.99 \$5.99	\$14.99 \$3.99	\$14.99 \$2.99
USD / month	USD / month	USD / month	USD / month
Plan renews at \$14.99/month on 0/10/2021	Plan renews at \$9.99/month on 11/10/2021	Plan renews at \$8.99/month on 11/10/2022	Plan renews at \$7.99/month on 11/10/2024

You save: **\$576.00**

Subtotal: **\$143.52**

If you've reviewed your order and everything appears correct, you can create an account as well as select your payment method on the same page. You just need to scroll down.

2. Create your account

Email address you will use to log in

 or







We will not share your information or send spam.

3. Select payment

☒ **Credit Card**


☐ **PayPal**


☐ **CoinPayments**


☐ **Google Pay**


WordPress Starter - 48 Months Plan **\$719.52**

SSL Certificate Activation **FREE** **\$0.00**

Domain Name **FREE** **\$0.00**

Discount -80% **-\$576.00**

Taxes & Fees **\$30.14**

Total ~~\$879.62~~ **\$173.66**

[Have a coupon Code?](#)

Name on card

Card number

Expiry date

CVC code

Submit Secure Payment

 30-Day Money-Back Guarantee

 Encrypted and Secure Payments

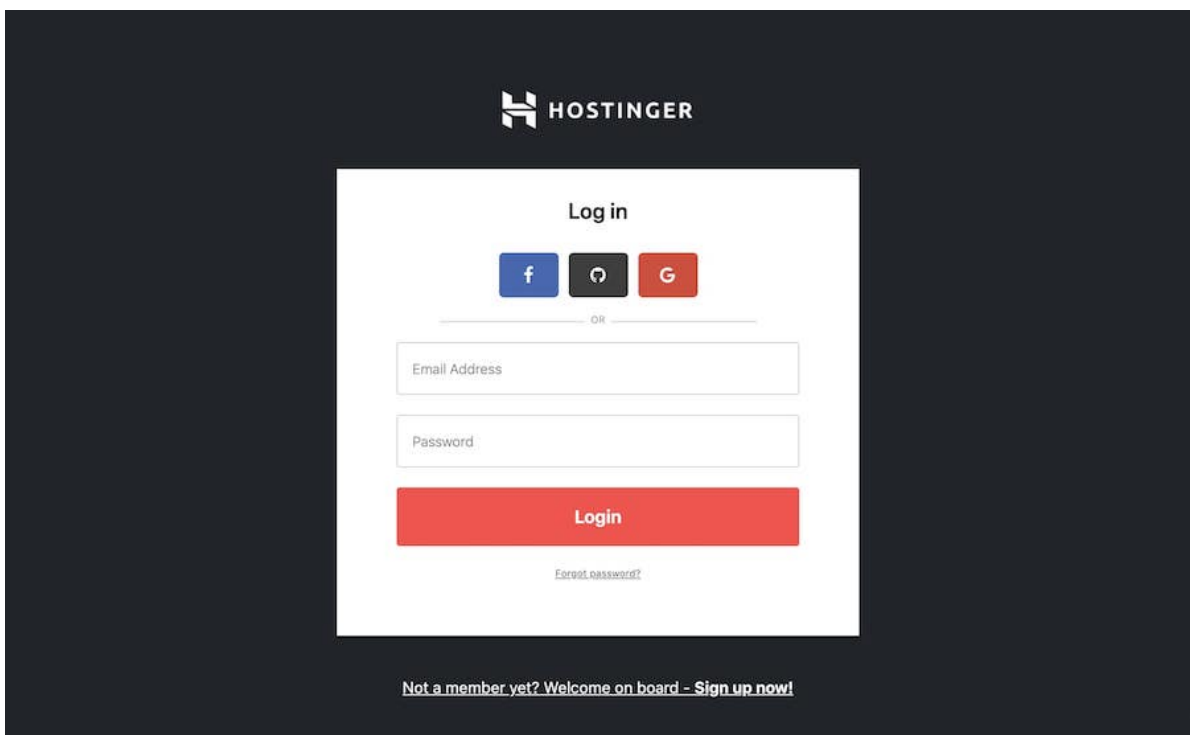
Features:

-  24/7/365 Chat Support
-  Excellent Customer Reviews
-  Optimized for WordPress
-  TOP Speed and Performance

Once you've entered your credentials, click on the **Submit Secure Payment** button.

2. Log in to Your New Hosting Account

Accessing your new hosting account is simple. Return to the Hostinger homepage and look for the **Client area** button in 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.

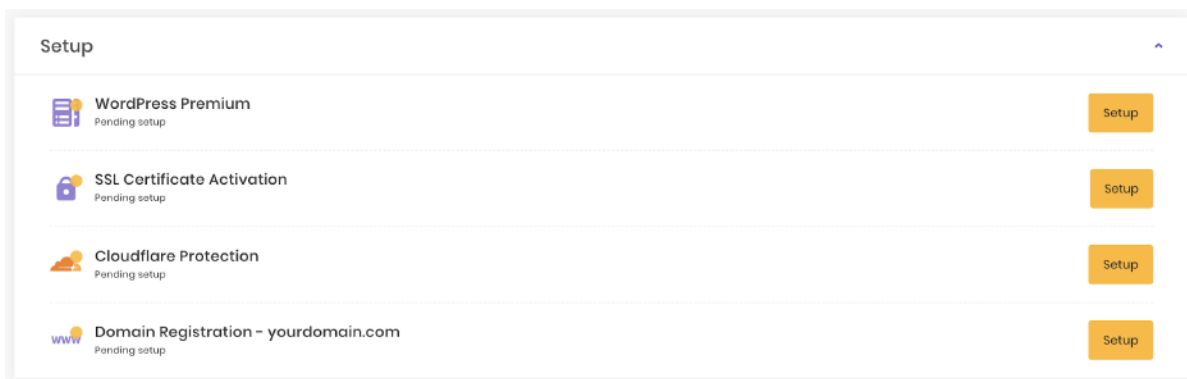


You'll then be redirected to your new hosting control panel. This is where you can view all the details about your current services, including billing information. However, before you worry about that, you'll need to finish configuring your plan.

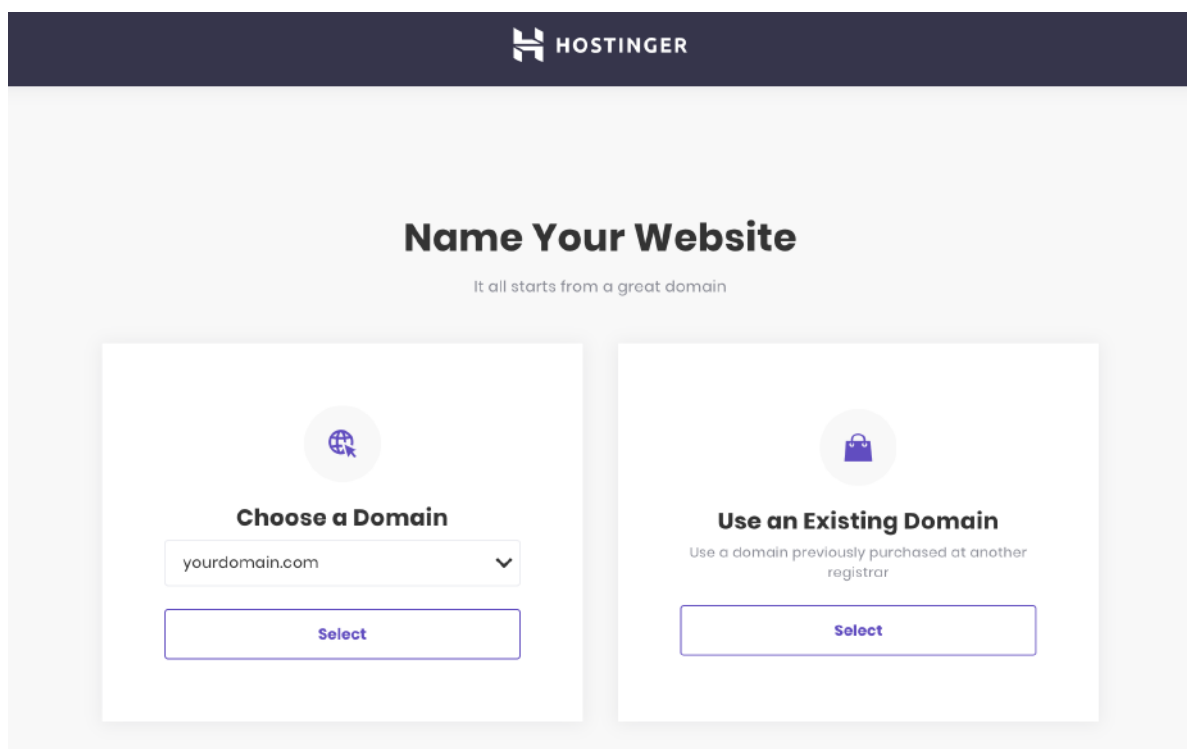
3. Install WordPress on Your Hosting Plan

The front page of your hosting control panel will provide an overview of your available plans and other features. Any elements that require further atten-

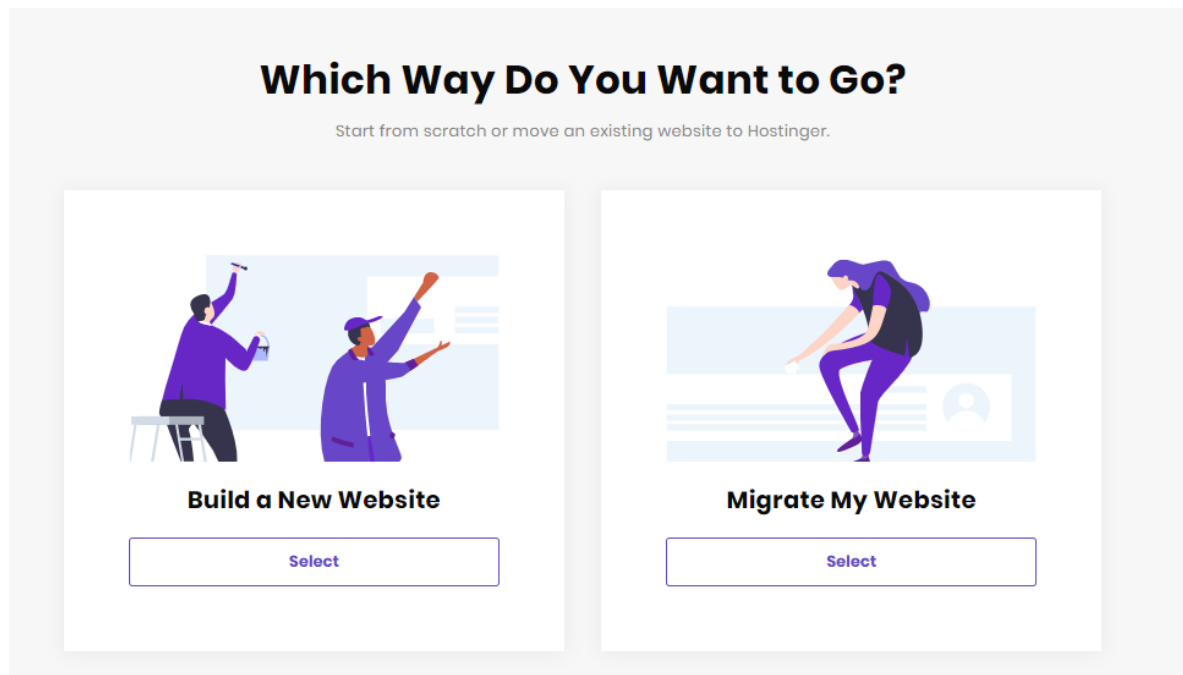
tion are indicated by a **Setup** button like with our **Premium WordPress Hosting** plan below:



First, Hostinger will ask whether you want to use the free domain name you got when choosing the WordPress Starter hosting plan or use an existing one. We will go with a free domain name.



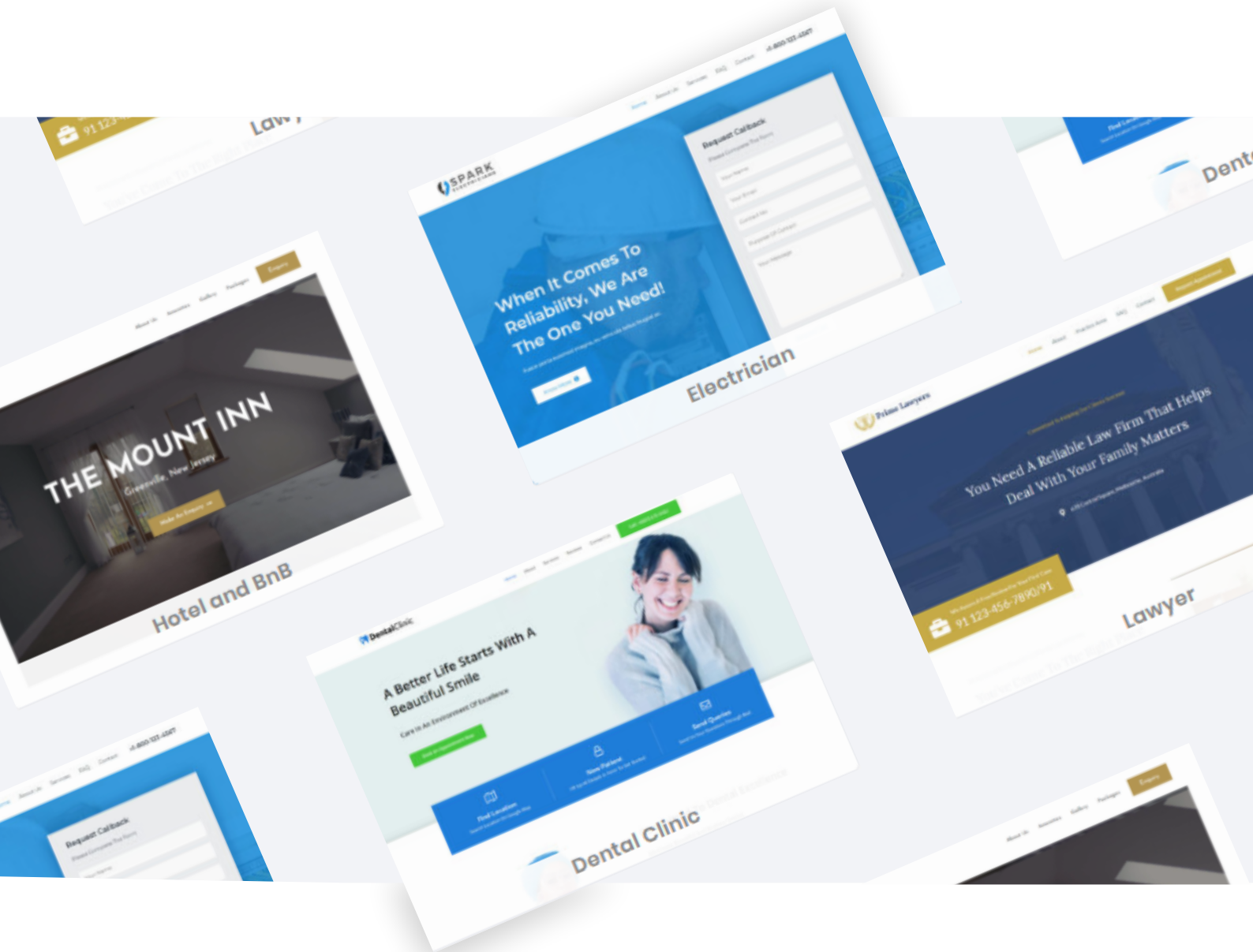
Once you click on the **Select** button you'll be guided through the steps to finish configuring your plan. First, decide whether to build or migrate your site. We'll be taking the former route.



Next, you'll need to provide some login details for your WordPress admin account at this stage. Keep these somewhere safe, as you will need them to access your WordPress admin area later.

A screenshot of the Hostinger "Create an Account" form. The title "Create an Account" is in bold black text, with a subtitle "You will use these details to access the admin area" below it. The form has two input fields: "Email address" with the placeholder "john.doe@domain.com" and a "Password" field with the placeholder "8 characters min.". Both fields have a small icon of a document with a checkmark. To the right of the password field is a "Continue" button in a purple box.

Additionally, Hostinger can set up a new theme for your WordPress site:



We're going to skip this step and start with one of WordPress's default options instead.

Hostinger then will take a minute or two to install WordPress in the background. If you registered a domain with us, you don't need to take any further action. If your domain was registered with a different registrar, however, make sure to point your domain to our nameservers.



You're almost there!

To view your website and make any changes you will need to point your domain to our nameservers from your current registrar. Our nameservers are:

ns1.dns-parking.com



ns2.dns-parking.com



Keep in mind that DNS changes can take up to 24 hours to reflect globally. You can learn more about domain pointing [here](#).

Click on the **Manage WordPress** button once your installation is complete for easy access to the hPanel WordPress dashboard. Here you can control various features such as your PHP version, enabling/disabling LiteSpeed Click, and flushing your cache.

Alternatively, you can return to your control panel to configure any other aspects of your account, such as your domain or SSL certificate.

The screenshot displays the hPanel WordPress dashboard for a site titled "A WordPress Site". The interface includes a top navigation bar with "Dashboard" and "Plugins" tabs. The main content area is divided into several sections:

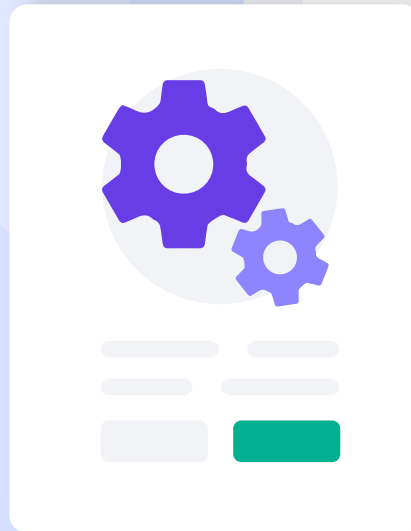
- Core**: Contains settings for "Force HTTPS" (Inactive), "Flush Cache" (with a "FLUSH CACHE" button), "Maintenance mode" (Inactive), and "WordPress Updates" (with a "CHANGE" button).
- Edit Website**: Includes a description "Create and update your website with WordPress Admin" and an "EDIT WEBSITE" button.
- Database**: Shows a database name "u638613412_ALn3q" and a "Manage database" link.
- Daily Backup**: Displays a backup status of "80.95 / MONTH" and a note "Make sure your files are safe and sound."

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

Keep in mind that the signup process may vary depending on your hosting company. It's vital to look closely at the available plans, support, features, and performance in order to make the right choice. Making an informed decision will save you trouble down the road.



Speed Up Your Website;
8 Practical Tips that Work



2

Setting Up Our Test Website

If you choose the right web host, a website running on a fresh WordPress installation should be blazing fast. It should take less than two seconds for pages to load, both on the front end and in the admin dashboard. However, loading often starts to take longer as you start to add content to your website.

The more complex and “heavy” your pages become, 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’re going to show how a brand new WordPress website feels in terms of performance. Then we’ll set up test content so you can see the difference various optimization techniques we explore in this e-book have on page speed.

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

There are many online guides you can follow to [speed up](#) your WordPress website. However, they almost never provide you with a baseline to help you understand the impact of the practices they outline.

While these articles are helpful resources, they don’t paint a full picture of WordPress performance optimization. With that in mind, we’re going to show you what affects the loading times of your website and how much of a difference best practices can make.

To do that, we’re going to set up a fresh installation of WordPress on the hosting plan we 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's been customized.

Once we have those statistics, we'll incorporate content and add-ons over the course of several chapters to mimic a real-life site. Then we'll run the same tests over again to show you the impact these elements have on site speed.

To give you an idea of what we're aiming for, a well-optimized website should load in under one second. Anything between one and three seconds is decent. Anything beyond that, and your site's performance begins to significantly affect UX.

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

Throughout the rest of this book, we're going to 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 Premium Shared plan. We also used the auto-installer to set up WordPress on our server.

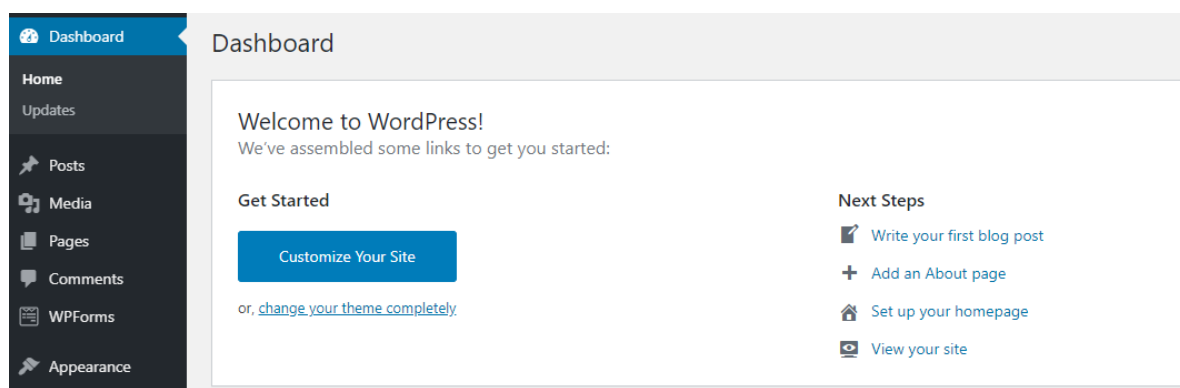
Take note, however, that how you should install your own WordPress package will vary depending on your hosting provider and the plan you chose. Overall, there are three approaches you can take:

Use an auto-installer. Many hosting companies will provide you with the option to set up WordPress using an automatic installer. This is ideal for beginners as all you have to do is configure a few settings and your host will take care of the rest.

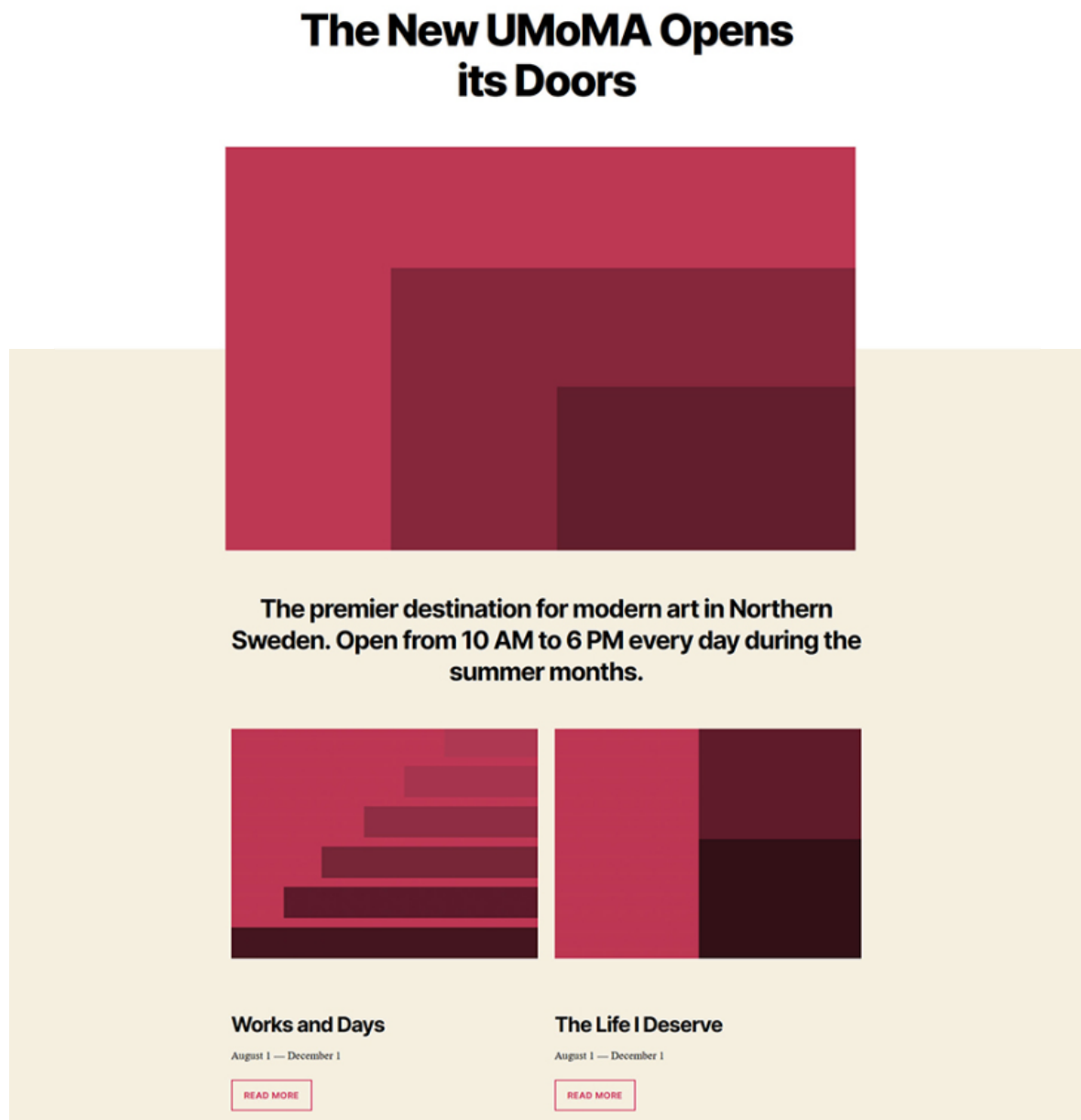
Set up WordPress manually. WordPress is an open-source platform, which means you're free to download its files from [WordPress.org](https://wordpress.org), upload them to your server, and use the CMS' built-in installer. This process requires a small amount of technical know-how.

Install WordPress through the command line. If you're using a VPS hosting plan, you'll have access to the command line, which you can use to install WordPress. This method is a bit more advanced and requires multiple commands.

Regardless of which method you want to use, we've got you covered. We have a tutorial you can check out that covers [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 view your website's front end, you should see a barebones page or some dummy content. Here's one of the sample pages that ship with the default [Twenty Twenty WordPress theme](#):



At this point, our WordPress site is ready for us to start building and customizing it. Before we get to work, however, 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 Tools

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

Google PageSpeed Insights

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

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

► Performance Tests With Pingdom Tools

Let's start with Pingdom Tools. As soon as you visit its homepage, it'll ask you for the URL you want to analyze and which 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

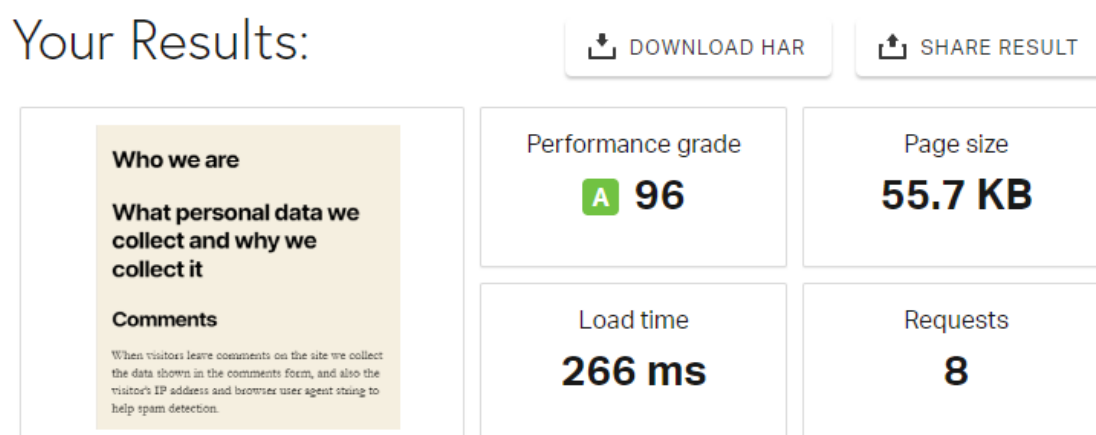
www.example.com

Test from

North America - USA - San Francisco ▼

START TEST

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



Our initial result is fantastic. 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 webpage that's only 55.7 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 more complex elements, page size climbs and so do the number of requests and overall loading times.

We'll show you how much of a difference this makes later. For now, here are the results of our Pingdom tests from five different server locations, including the results above:

1. **United Kingdom, London:** 0.565 seconds
2. **USA, Washington D.C.:** 0.266 seconds
3. **Japan, Tokyo:** 1.59 seconds
4. **Australia, Sydney:** 1.42 seconds
5. **Brazil, Sao Paulo:** 1 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 we ran naturally got the best result.

In Japan and Australia, however, our page is slow enough to cause UX problems and increase our bounce rate. 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 Tools, this platform asks for the URL you want to test. We'll use our default homepage once more.



PageSpeed Insights

HOME

DOCS

Improve the speed of your web pages on all devices

ANALYZE

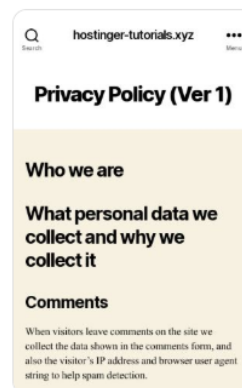
Right away, you'll notice the numbers you see below are in line with the slowest page loading times we got using Pingdom. That's because PageSpeed Insights breaks down performance using different metrics.

Field Data — The "Chrome User Experience" report [does not have enough real-time data on the speed](#) of this page.

Origin Summary — The Chrome User Experience report [does not have enough real-time data on the speed](#) of this origin.

Experiment data

● First rendering with content	1.4 s	● First significant rendering	1.5 s
● Speed index	1.4 s	● First CPU idle time	1.5 s
● Time until interactive	1.5 s	● Maximum potential latency of the first interaction	60 ms



PageSpeed Insights 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 **first significant rendering** time. Meaning, how long it takes for the main content of the page to display, even if there are still some elements working in the background.

The numbers you see above are not bad – we're still below that two-second mark. Plus, we're getting a near-perfect optimization score, which means our servers are configured for speed right out of the gate.

Although each tool provides slightly different numbers, the results tell us the same thing – 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 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 that we'll be referencing throughout this e-book, we created a unique page filled with dummy content and several images. Here's a quick look at it:



Comments

When visitors leave comments on the site we collect the data shown in the comments form, and also the visitor's IP address and browser user agent string to help spam detection.

An anonymized string created from your email address (also called a hash) may be provided to the Gravatar service to see if you are using it. The Gravatar service privacy policy is available here: <https://automattic.com/privacy/>. After approval of your comment, your profile picture is visible to the public in the context of your comment.

Media



You can only see a fraction of the page, but it weighs 602.4 KB, which is about ten times the size of the default homepage we used for our baseline test.

The idea behind this page is to set up a test that includes a variety of elements, such as images as well as lengthy text. 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 features later. For now, it's important to see the difference they make for your site's performance.

4. Re-Test Our New Page

Now that we have our content set up, we're going to run our initial tests again to illustrate how heavier pages influence loading times. You already know how Pingdom Tools and PageSpeed Insights work, so let's skip to the results. Here are the results of the Pingdom Tools test from Washington D.C. for our new test page:

Your Results:

[📄 DOWNLOAD HAR](#)[📄 SHARE RESULT](#)

	Performance grade A 94	Page size 587.7 KB
	Load time 677 ms	Requests 15

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

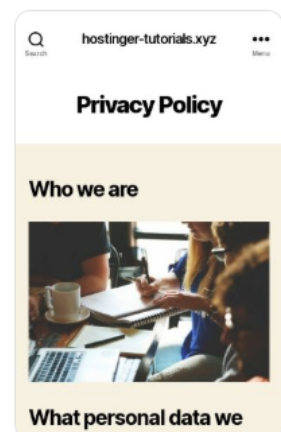
Once more, PageSpeed Insights tells a slightly different story. Although the time to first render remains pretty much the same, our website is taking almost one extra second to load before it's fully interactive. That is to say, the resources we've added so far have a measurable impact on performance:

Field Data — The "Chrome User Experience" report [does not have enough real-time data on the speed](#) of this page.

Origin Summary — The Chrome User Experience report [does not have enough real-time data on the speed](#) of this origin.

Experiment data

● First rendering with content	1.3 s	■ First significant rendering	2.4 s
● Speed index	1.3 s	● First CPU idle time	2.4 s
● Time until interactive	2.4 s	● Maximum potential latency of the first interaction	20 ms



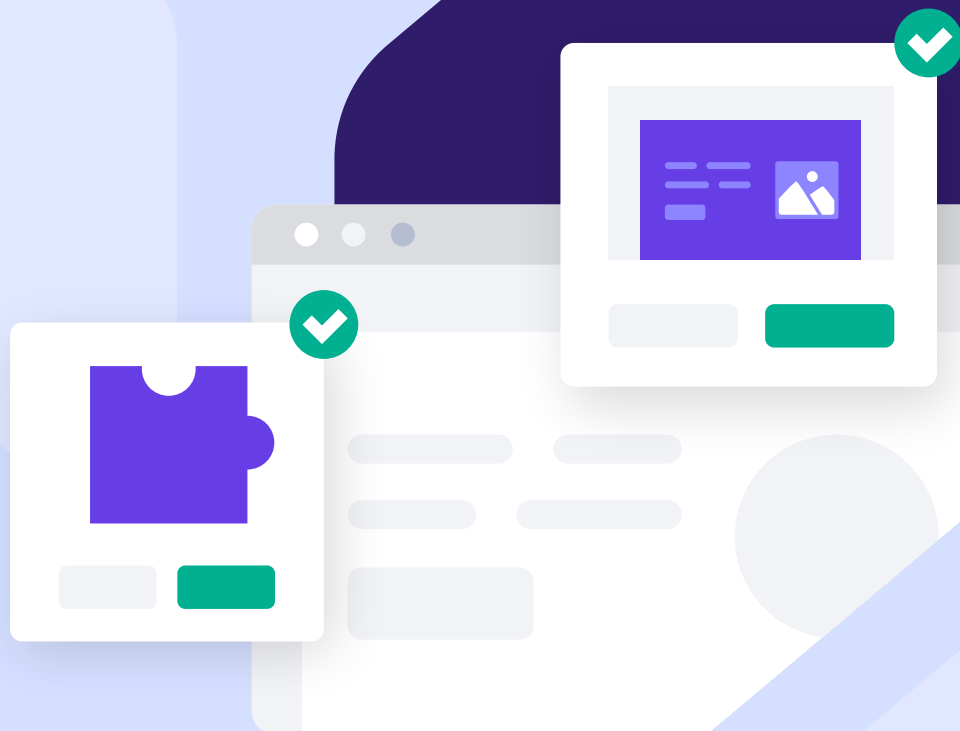
In practice, you should never rely on a single test to gauge your website's performance. It's only by aggregating data from multiple sources that you can replicate what your users experience.

After all, you'll receive visitors from across the world with access to 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 is something we're going to address in further chapters.



Speed Up Your Website;
8 Practical Tips that Work



3

Selecting the Right Plugins and Theme for Optimal Performance

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

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 work.

In this chapter, we're going to break down what you should look out for in a WordPress theme as well as the plugins you use. We'll also offer you some suggestions that will provide you with amazing performance along the way. 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 aesthetics you have in mind 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 improve quality and security.



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



Proper optimization to avoid adding unnecessary code to your site that will weigh it down.

It's fairly easy to spot themes that have features you want for your site and receive regular updates. On WordPress.org, you can find this information in the short description and listed under the **Download** button:

Preview Download

Version: 2.3.1
 Last updated: February 18, 2020
 Active Installations: 600,000+
 PHP Version: 5.3 or higher
[Theme Homepage →](#)

Ratings [See all >](#)

★★★★★ 5 out of 5 stars.

5 stars 3305
 4 stars 45

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
	@125697	3 days	themedropbox	New version of Astra - 2.3.1
	@125372	9 days	themedropbox	New version of Astra - 2.3.0
	@123418	6 weeks	themedropbox	New version of Astra - 2.2.1
	@123151	7 weeks	themedropbox	New version of Astra - 2.2.0
	@122222	2 months	themedropbox	New version of Astra - 2.1.4
	@120516	3 months	themedropbox	New version of Astra - 2.1.3
	@119326	4 months	themedropbox	New version of Astra - 2.1.2
	@118806	5 months	themedropbox	New version of Astra - 2.1.1
	@118753	5 months	themedropbox	New version of Astra - 2.1.0

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

Premium themes often cost between \$20-60 for a one-year license. You don't want to pay that much money for a tool that's going to slow your site down.

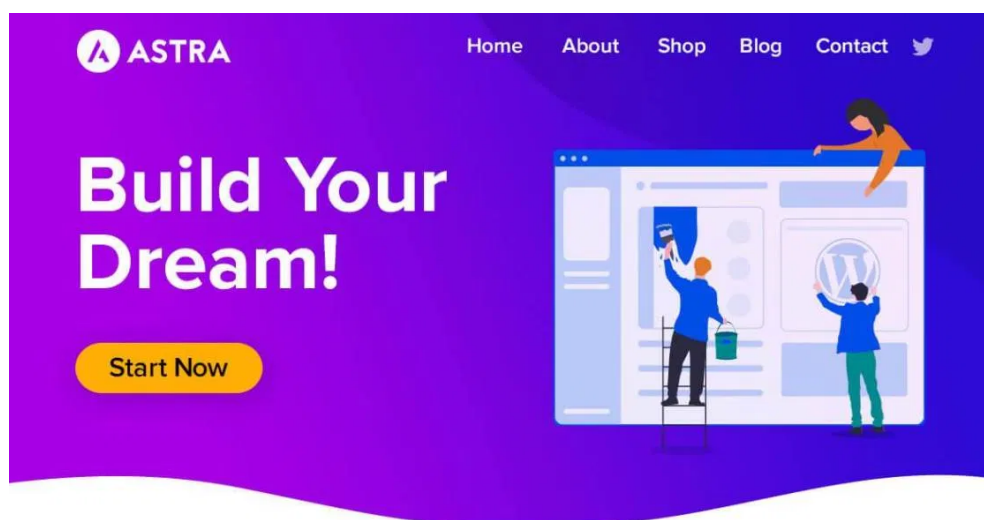
There's no obvious solution to this problem. Most theme developers will tell you their products are fast and try to persuade you with high performance scores from services such as PageSpeed Insights.

However, as we demonstrated in the previous chapter, a high score 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:

Privacy Policy

Who we are



What personal data we collect and why we collect it



Comments

When visitors leave comments on the site we collect the data shown in the comments form, and also the visitor's IP address and browser user agent string to help spam detection.



Recent Posts

[Hello world!](#)

Recent Comments

[A WordPress Commenter](#) on [Hello world!](#)

Archives

[February 2020](#)

Categories

[Uncategorized](#)

Meta

[Site Admin](#)

[Log out](#)

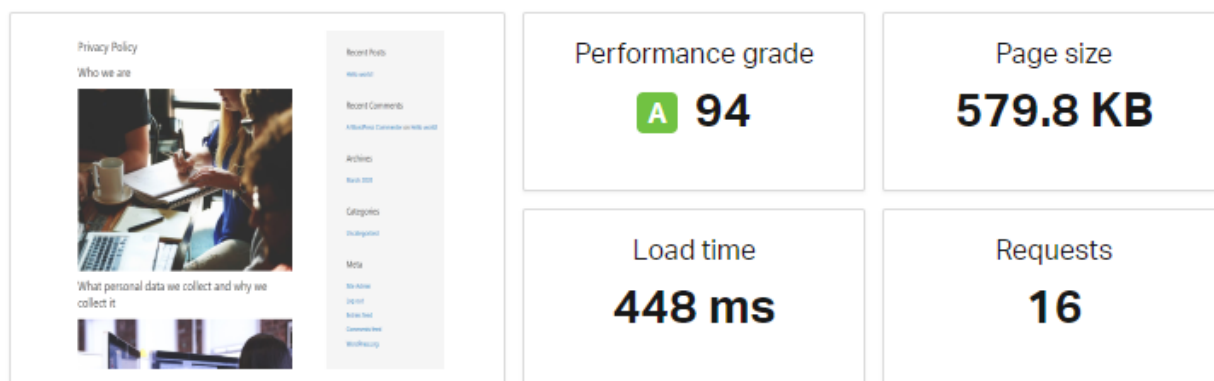
[Entries feed](#)

[Comments feed](#)

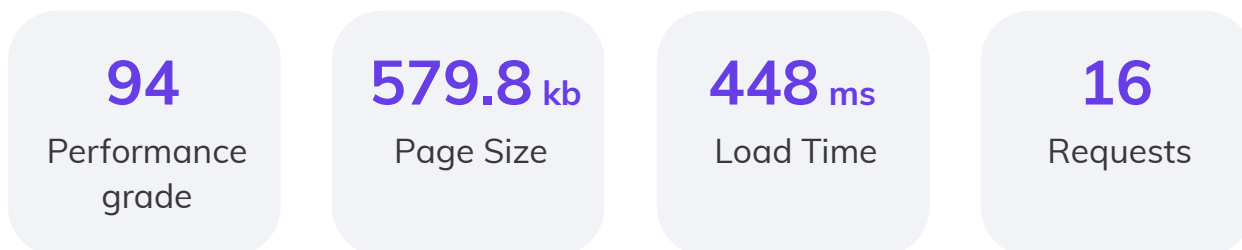
[WordPress.org](#)

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

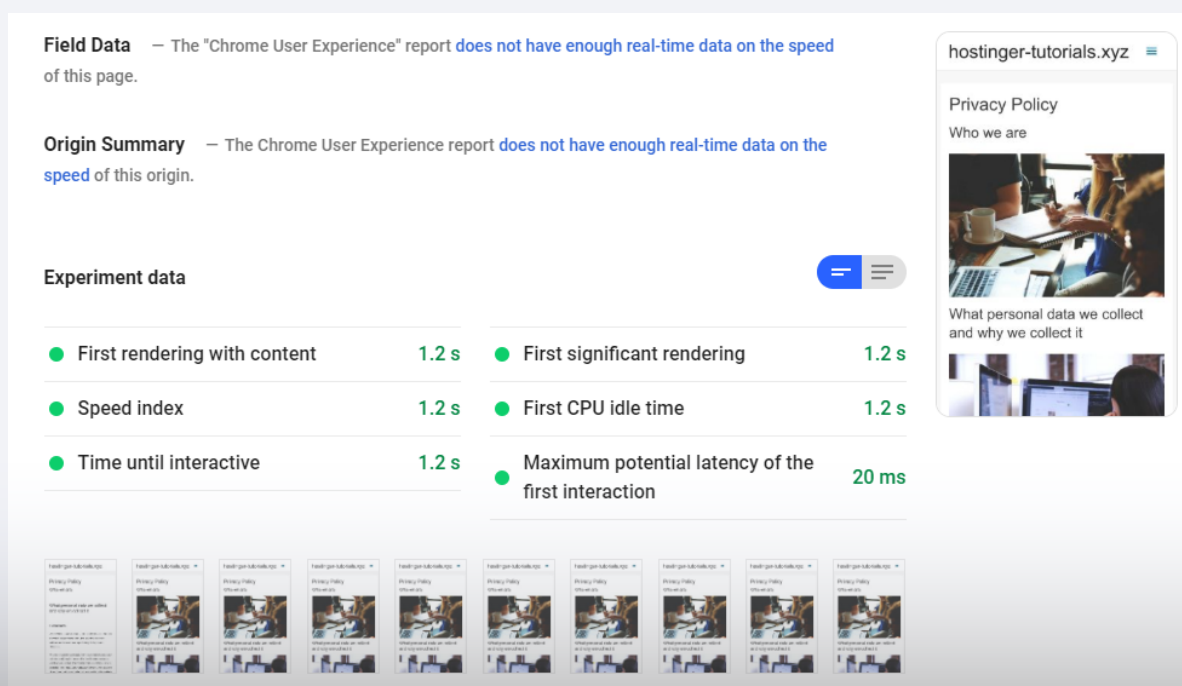
Your Results:

[DOWNLOAD HAR](#)[SHARE RESULT](#)

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



Our overall page size decreased by a small margin. Loading times, on the other hand, went down by over 100 ms without us doing any additional changes to the website. PageSpeed tells a similar story, with the Astra version of our test page loading faster across the board:



Recall that our time to first significant rendering for this page was 1.3 seconds in Chapter 2, the same as its **speed index**. The speed index is what [Google defines](#) as “how quickly the contents of a page are visibly populated” minus .1 seconds.

Keep in mind – the default WordPress themes are already fast. However, these tests show you just how much of a difference a theme that is optimized for performance can make without any additional work on your end. By the time we’re done with this book, those numbers should be even lower.

While our test website is pretty lean by modern standards, there are several plugins working in the background that can impact overall performance as well.

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

Recommending which plugins are a must-use is a hard task. At the time of this writing, there are over 55,000 available for free in the [WordPress Plugin Directory](#):

Plugins

Extend your WordPress experience with 55,755 plugins.



If you factor in premium options, that number skyrockets. Which ones you should install depends on the functionality your site requires as well as your personal preferences. That said, there are some best practices for smart plugin selection:

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

Avoid overlapping functionality. Try not to install multiple plugins that do the same thing. Also consider any features your theme includes and avoid overlaps there, too.

Consider multi-functional vs. single-use plugins. If you only use one feature of a multi-functional plugin, consider looking for an alternative that specializes in the element you need. On the other hand, it's more efficient to use one plugin that covers many key features you'll implement than to install several single-use plugins.

Deactivate or delete unused plugins. This not only eliminates unnecessary code from your site but is an important security consideration as well.

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

Unlike themes, you can have dozens of plugins working side by side on your website. In practice, that means most people aren't as selective when it comes to choosing which ones to install.

Unfortunately, this tends to spell bad news for performance. Even the most straightforward plugins add some background scripts or functions to your website. That means there's more code that needs to load in the background, even if you can't see the changes with your naked eye.

► 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 helps you backup and move your website.

Yoast SEO

The most popular Search Engine Optimization (SEO) plugin.

Akismet Anti-Spam

A comment spam prevention tool.

WPForms Lite

A simple plugin that helps you set up contact forms.

Wordfence Security

A comprehensive WordPress security plugin.

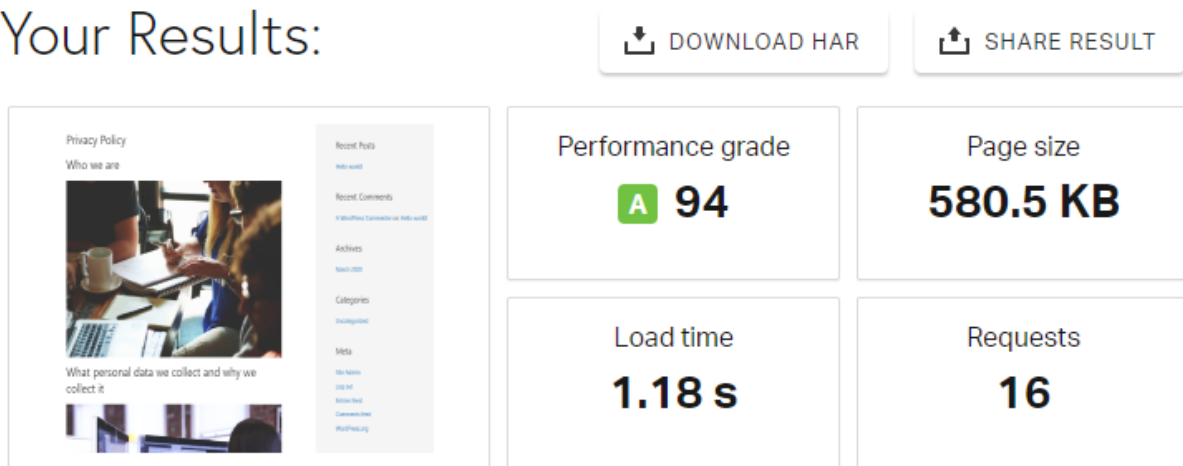
For guidance installing WordPress plugins, you can read our [beginner's tutorial on this skill](#). Once we added the five listed above, we activated them:

☐ Plugin	Description
☒ 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 from comment spam. Version 4.1.3 By Automattic View details
☒ All-in-One WP Migration Deactivate	Migration tool for all your blog data. Import or Export your blog content with a single click. Version 7.17 By ServMask View details Get Support
☐ 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. Version 1.7.2 By Matt Mullenweg View details
☐ LiteSpeed Cache Activate Delete	High-performance page caching and site optimization from LiteSpeed Version 2.9.9.2 By LiteSpeed Technologies View details
☒ Wordfence Security Upgrade To Premium Deactivate	Wordfence Security - Anti-virus, Firewall and Malware Scan Version 7.4.6 By Wordfence View details
☒ WPForms Lite Settings Deactivate	Beginner friendly WordPress contact form plugin. Use our Drag & Drop form builder to create your WordPress forms. Version 1.5.8.2 By WPForms View details
☒ Yoast SEO FAQ Premium Support Settings Deactivate	The first true all-in-one SEO solution for WordPress, including on-page content analysis, XML sitemaps and much more. Version 13.1 By Team Yoast View details
☐ Plugin	Description

As you can see, there are two other plugins listed in our dashboard. [Hello Dolly](#) comes pre-installed on every WordPress site, and [LiteSpeed Cache](#) is included with our Hostinger plan. We'll discuss it more in a later chapter.

For now, we're going to leave them deactivated. Since they're not running, they shouldn't affect loading times. With that in mind, let's go through our standard tests once more, starting with Pingdom Tools:

Your Results:



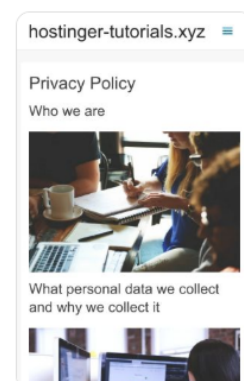
Right away we see an increase in loading times of about 0.6 seconds, although our performance score didn't change. For the first time during our tests, Page-Speed Insights also returned results that are in line with Pingdom. Compared to the last round of tests, here you can see increases of around .1 in all results:

Field Data — The "Chrome User Experience" report [does not have enough real-time data on the speed](#) of this page.

Origin Summary — The Chrome User Experience report [does not have enough real-time data on the speed](#) of this origin.

Experiment data

● First rendering with content	1.2 s	● First significant rendering	1.3 s
● Speed index	1.5 s	● First CPU idle time	1.3 s
● Time until interactive	1.3 s	● Maximum potential latency of the first interaction	20 ms



These numbers only confirm what we already knew – the more scripts you have running in the background and loading with your pages, the higher loading times should be. However, that doesn't mean that you can't use a broad range of plugins on your website.

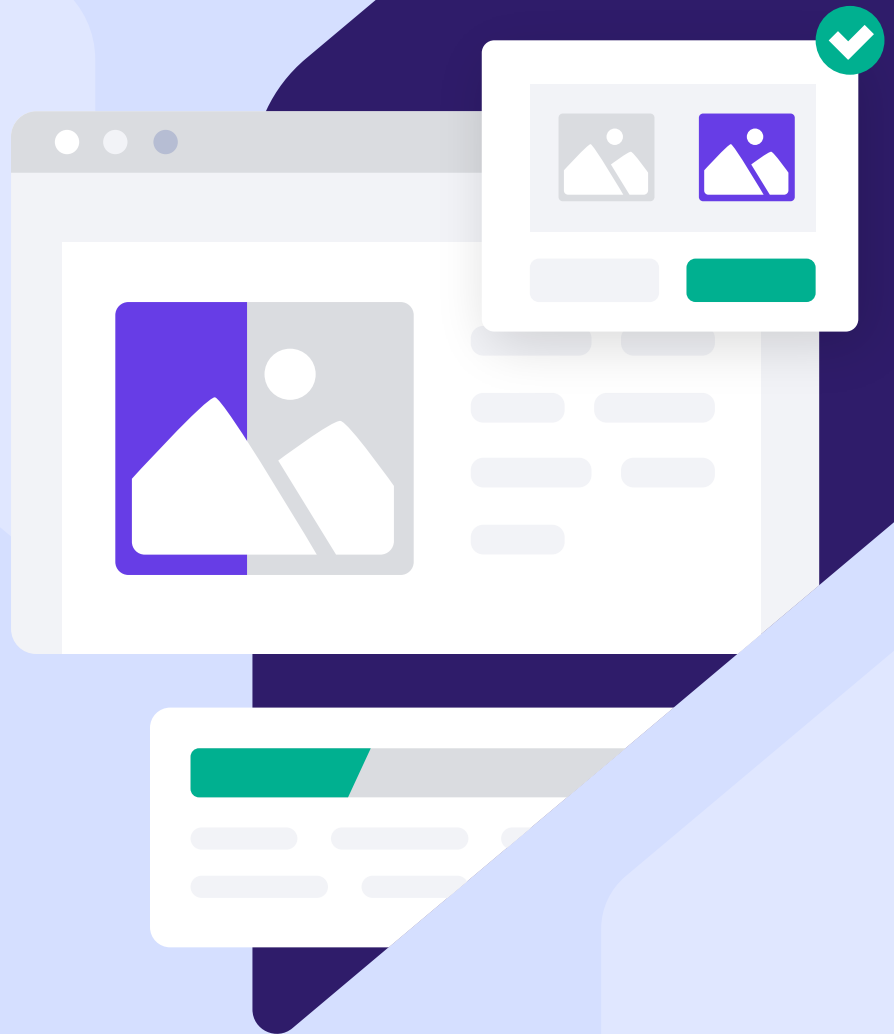
Usually, performance dropoff from plugins stabilizes after a while. If your website uses around ten plugins or less, you should be fine, as long as you make sure you optimize the rest of your website.

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.



Speed Up Your Website;
8 Practical Tips that Work



4

Optimizing Your Images

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

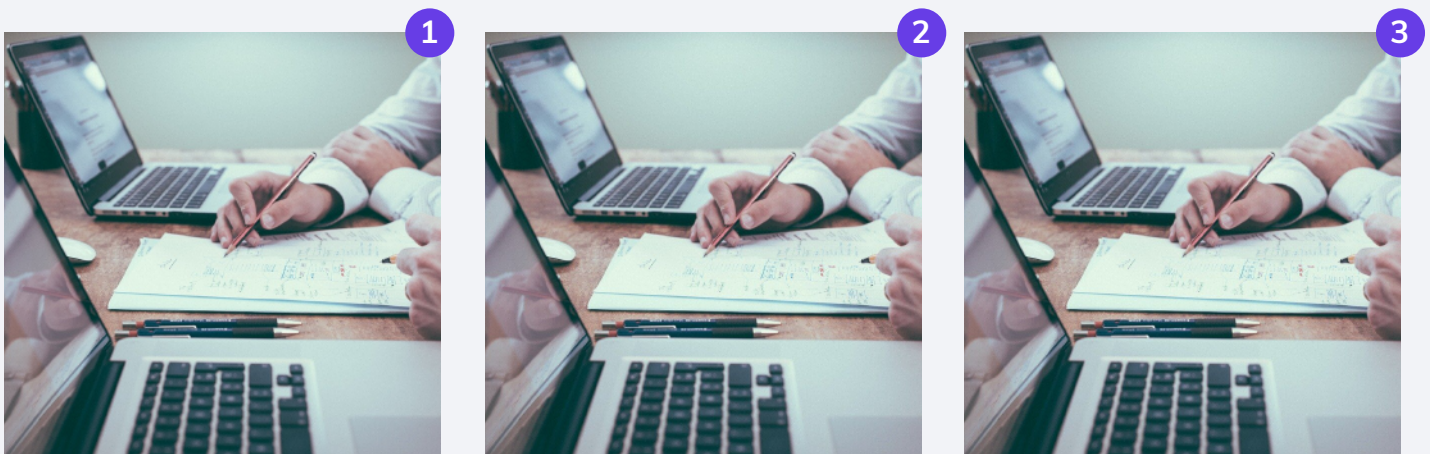
You've probably dealt with compressed files at some point. Essentially, compression software reduces the size of a file without affecting the data itself. This enables you to transfer it much faster.

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.

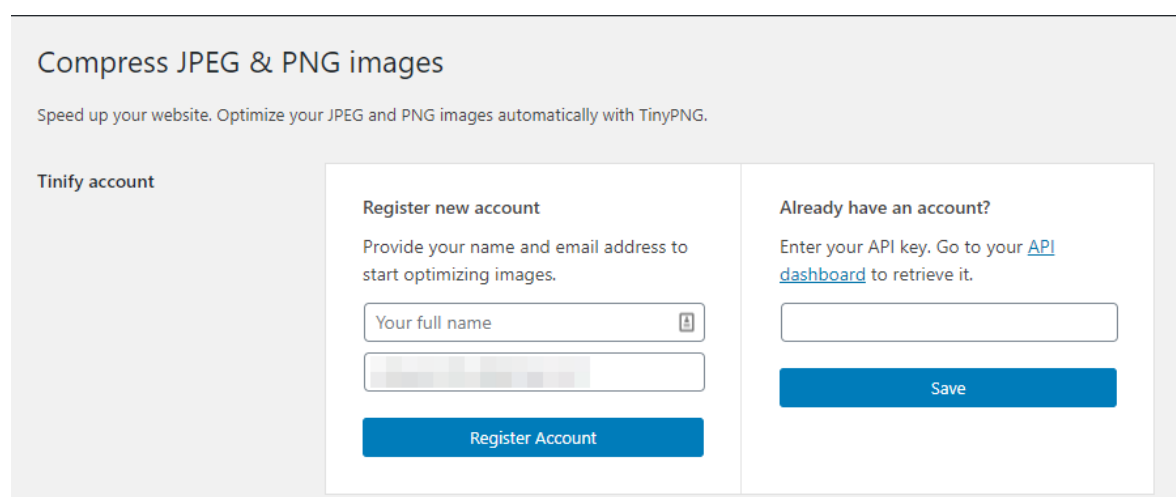
Putting aside premium options, a handy solution is [Compress JPEG & PNG Images](#). As the name implies, it supports both popular image file formats and optimizes them automatically when you upload them to your site. It can also handle the existing ones in your Media Library:



Beyond that, Compress JPEG & PNG Images 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 Compress JPEG & PNG Images 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 is enough for most medium-sized websites.

You can register or connect an existing account from the **Settings > Compress JPEG & PNG Images** 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) <i>This is the fastest method, but can cause issues with some image related plugins.</i> ☒ Compress new images during upload <i>Uploads will take longer, but provides higher compatibility with other plugins.</i> ☐ Do not compress new images automatically <i>Manually select the images you want to compress in the media library.</i>

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 > Compress JPEG & PNG Images** tab, you'll see a section that reads **Image sizes**. Here, you can configure which image resolutions the plugin should automatically optimize.

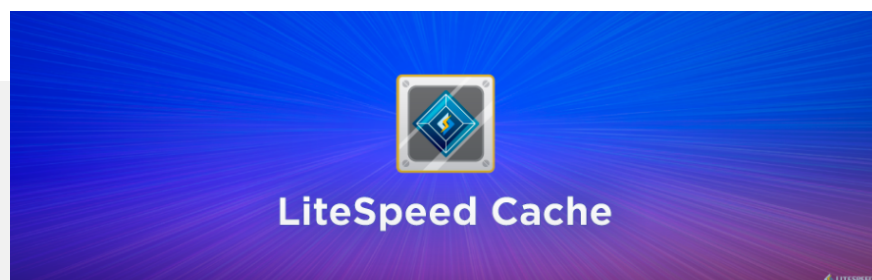
Image sizes	Select image sizes to be compressed
	Wordpress generates resized versions of every image. Choose which sizes to compress.
	☒ Original image (overwritten by compressed image) ☒ Thumbnail - 150x150 ☒ Medium - 300x300 ☒ Medium large - 768x? ☒ Large - 1024x1024 ☒ 1536x1536 - 1536x1536 ☒ 2048x2048 - 2048x2048 ☒ Post-thumbnail - 1200x9999 ☒ Twentytwenty-fullscreen - 1980x9999
	Remember each selected size counts as a compression.
	With these settings you can compress at least 55 images for free each month.

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.

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:

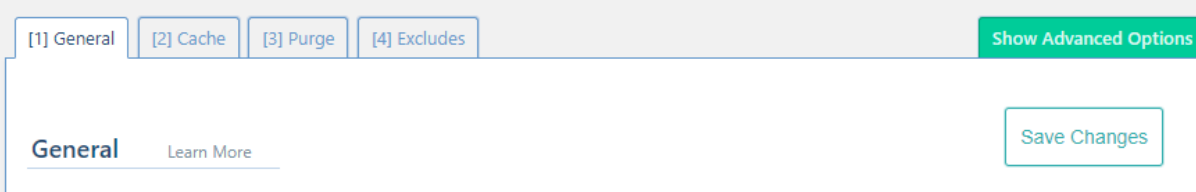


We're going to touch on many of those features throughout this e-book. For this chapter, we want to focus on a specific option called **lazy loading**.

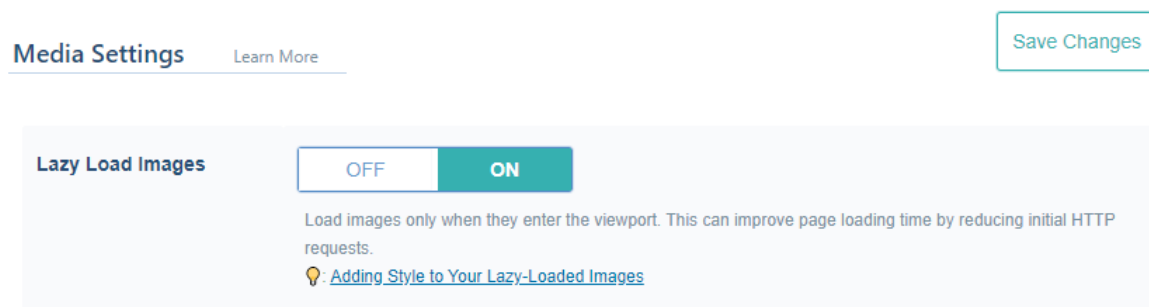
Usually, when you visit a website, your browser will attempt to load all its elements at once. If you visit a page that requires a fair bit of scrolling, your browser prioritizes elements at the top of the screen and at the bottom equally.

Lazy loading tells browsers to hold off on loading images until users scroll down and reach them. It's a subtle difference, 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 > Settings** and clicking on **Show Advanced Options** to the right:



This will reveal multiple new tabs, including one labeled **Media**. Click on it and turn on the **Lazy Load Images** option:



If you scroll down, you'll see that LiteSpeed Cache enables you to disable lazy loading for specific images and CSS classes. Setting up a class for files you want to exclude can be a smart option, particularly for smaller graphics such as logos:

Lazy Load Image Excludes

Listed images will not be lazy loaded. Both full URLs and partial strings can be used. One per line.

API: Filter `litespeed_cache_media_lazy_img_excludes` is supported. Elements with attribute `data-no-lazy="1"` in html code will be excluded.

Lazy Load Image Class Name Excludes

Images containing these class names will not be lazy loaded. Both full and partial strings can be used. One per line.

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:

Available Images

Here you can start optimizing your entire library. Press the big button to start improving your website speed instantly!

UPLOADED IMAGES	UNCOMPRESSED IMAGE SIZES	ESTIMATED COST
7	14 i	\$ 0.00 USD

Remember For the plugin to do the work, you need to keep this page open. But no worries: when stopped, you can continue where you left off!

0 / 14 (0%)

Start Bulk Optimization

Total Savings

Statistics based on all available JPEG and PNG images in your media library.

0% savings

0 image sizes optimized

1.95 MB initial size

1.95 MB current size

File	Initial Size	Current Size	Savings	Status
------	--------------	--------------	---------	--------

Remember that with your free account, you can optimize up to 500 images for free per month, so don't 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 Tools has to say about our sample page now.

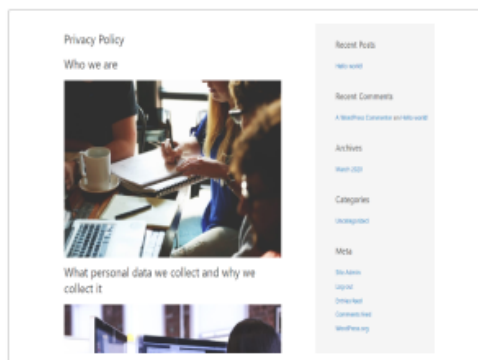
Your Results:



DOWNLOAD HAR



SHARE RESULT



Performance grade

A 94

Page size

590.0 KB

Load time

628 ms

Requests

16

Let's start with the **Page Size** section. During the previous chapter it clocked in at 580.5 KB and now it's **up** by almost 10 KB. 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.

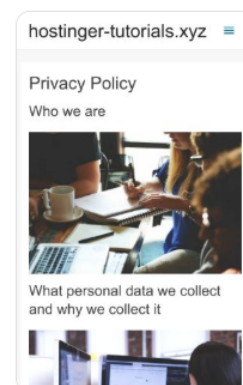
All in all, although the test page's size is larger, it actually loaded a lot faster. During Chapter 3, Pingdom returned a result of 1.18 seconds. It's now down to 0.628, which is fantastic. Once more, however, we need to compare those results with PageSpeed Insights.

Field Data — The "Chrome User Experience" report [does not have enough real-time data on the speed](#) of this page.

Origin Summary — The Chrome User Experience report [does not have enough real-time data on the speed](#) of this origin.

Experiment data

● First rendering with content	1.6 s	● First significant rendering	1.6 s
● Speed index	1.7 s	● First CPU idle time	1.6 s
● Time until interactive	1.6 s	● Maximum potential latency of the first interaction	60 ms



The results here aren't as positive. With the additional scripts from Compress JPEG & PNG Images, our loading times went up slightly across the board – although not much, and we're still achieving 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.

Speed Up Your Website;
8 Practical Tips that Work

5

Minifying Your Website's CSS and JavaScript

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}
```

In essence, you take out spaces, comments, and sometimes replace variable names. While it may seem like such drastic changes to your site's files should impact it in some way, that's not the case. Although minified code is more difficult for humans to read, to a computer it all looks the same.

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.

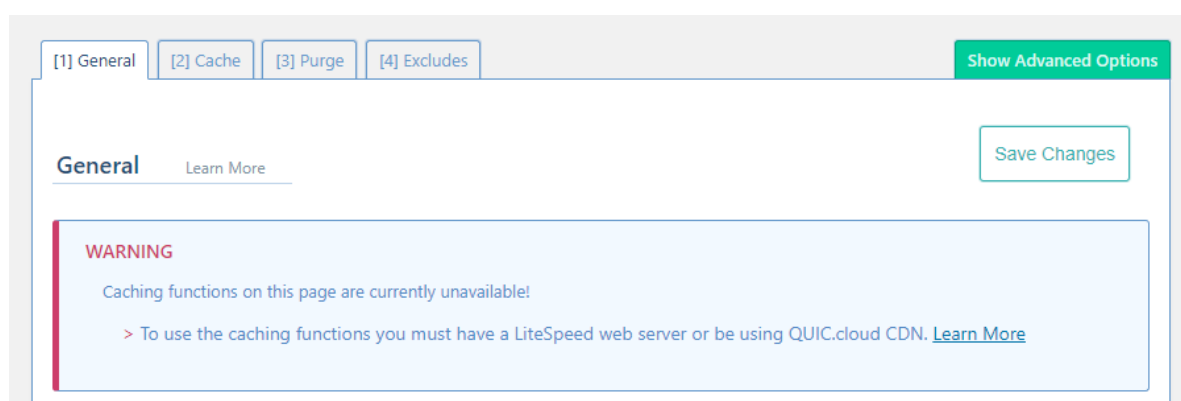
Usually, you want to focus on JavaScript and CSS, which are among the most common types of scripts for most websites. You can also try to minify your HTML files, but in most cases, that's not necessary if your web host uses **GZIP** compression.

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

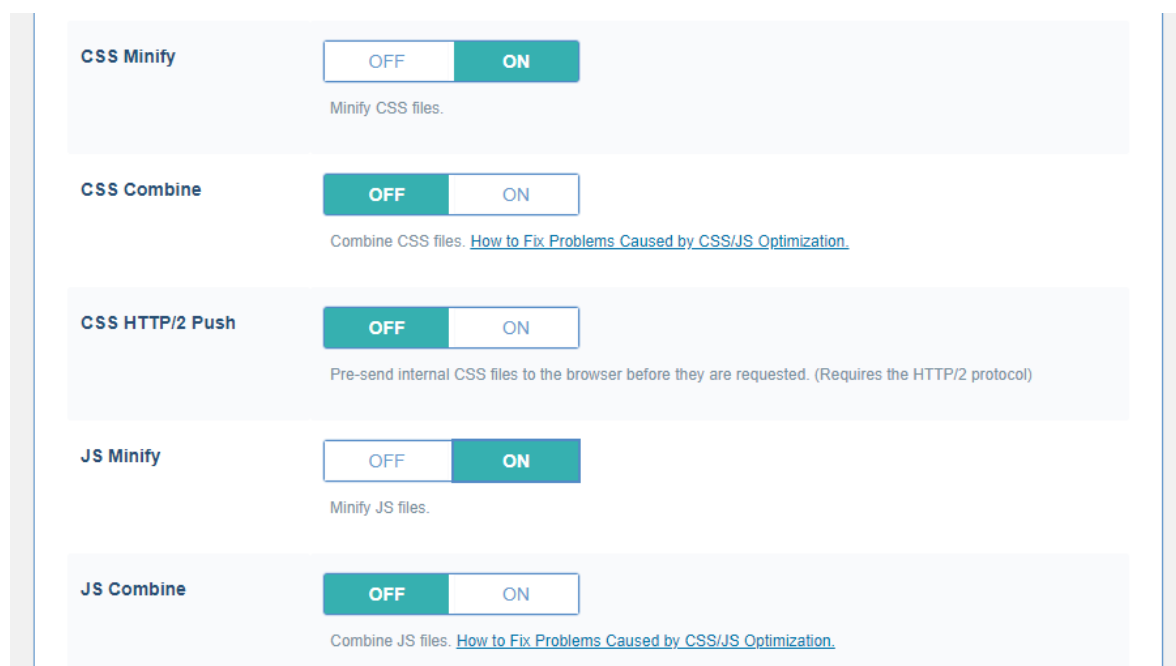
In the last chapter, we introduced you to the [LiteSpeed Cache](#) plugin as our preferred tool for implementing lazy loading. As we've mentioned, it can do a lot more to help speed up your website. In the following sections, we'll show you how to use it to minify key files on your website.

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

Enabling minification using LiteSpeed Cache is very straightforward. In the last chapter, we turned on the plugin's advanced settings. If you skipped that step, go to **LiteSpeed Cache > Settings** and click on the **Show Advanced Options** button:



Now, move to the **Optimize** tab and scroll down until you find the **CSS Minify** setting. Turn it on right away and do the same for **JS Minify** further down.



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.

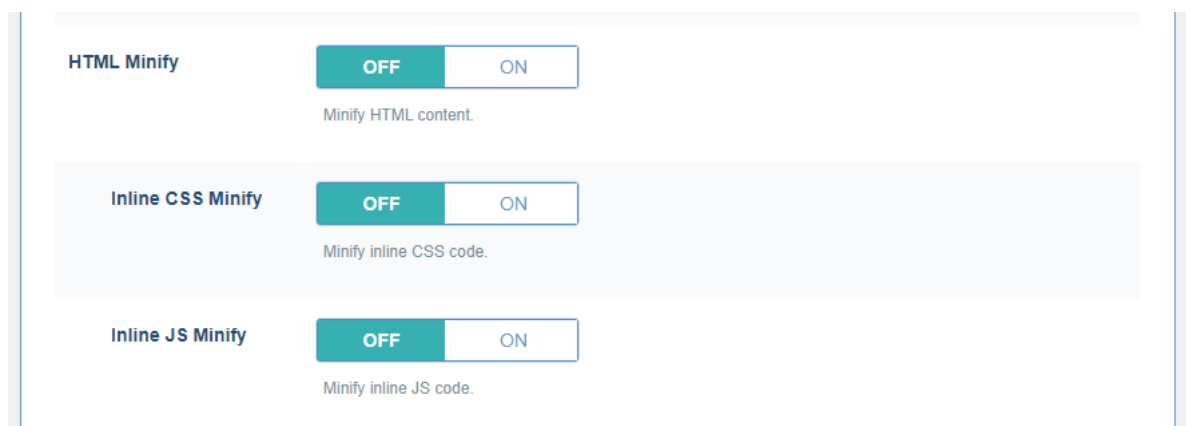
That sounds fantastic in practice, but in our experience, the improvements are often marginal. Despite combining files, your page size won't change, and in most cases, merging scripts can prompt errors on your website. However, there are further steps you can take to optimize your code.

2. Enable Additional Code Optimization Settings

Another setting you might want to enable in LiteSpeed Cache is **CSS HTTP/2 Push**. This enables parallel requests for your website's resources.

You can think of this technique as adding new lanes to a highway, where there previously was only one. If your web host and your visitors' browsers support HTTP/2, then enabling this feature can help you decrease loading times.

Moving on, you'll see the setting to minify [HTML files](#) as well. As we mentioned before, we want to skip this option, as well as minifying inline CSS and JS:



The last two settings we care about in this section are 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 rendering a page, cutting down on initial loading times.

The process is somewhat similar to lazy loading, which we described in Chapter 3. With that in mind, let's go ahead and enable both of those settings.

Load JS Deferred

OFF

ON

Doing so can help reduce resource contention and improve performance. This can improve your speed score in services like Pingdom, GTmetrix and PageSpeed.

Exclude JQuery

OFF

ON

Improve compatibility with inline JS by preventing jQuery optimization. (Recommended Setting: ON)
⚠ If there is any JS error related to jQuery when enabled JS Combine, please turn on this option.

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.

Your Results:

Privacy Policy

Who we are



What personal data we collect and why we collect it



Recent Posts

[Hello world!](#)

Recent Comments

[A WordPress Comment on the world!](#)

Archives

[March 2022](#)

Categories

[Uncategorized](#)

Meta

[Site Name](#)

[Site URL](#)

[WordPress](#)

[Contact Us](#)

[About Us](#)

DOWNLOAD HAR

SHARE RESULT

Performance grade

A 94

Page size

583.4 KB

Load time

535 ms

Requests

16

During Chapter 4, this same page weighed in at 590 KB. That means minifying its scripts and styles led to a reduction in file size of about 7 KB.

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 100 ms faster than our last test.

Seven kilobytes 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:

Passes audited (19) ^

● Eliminate resources that block rendering ^

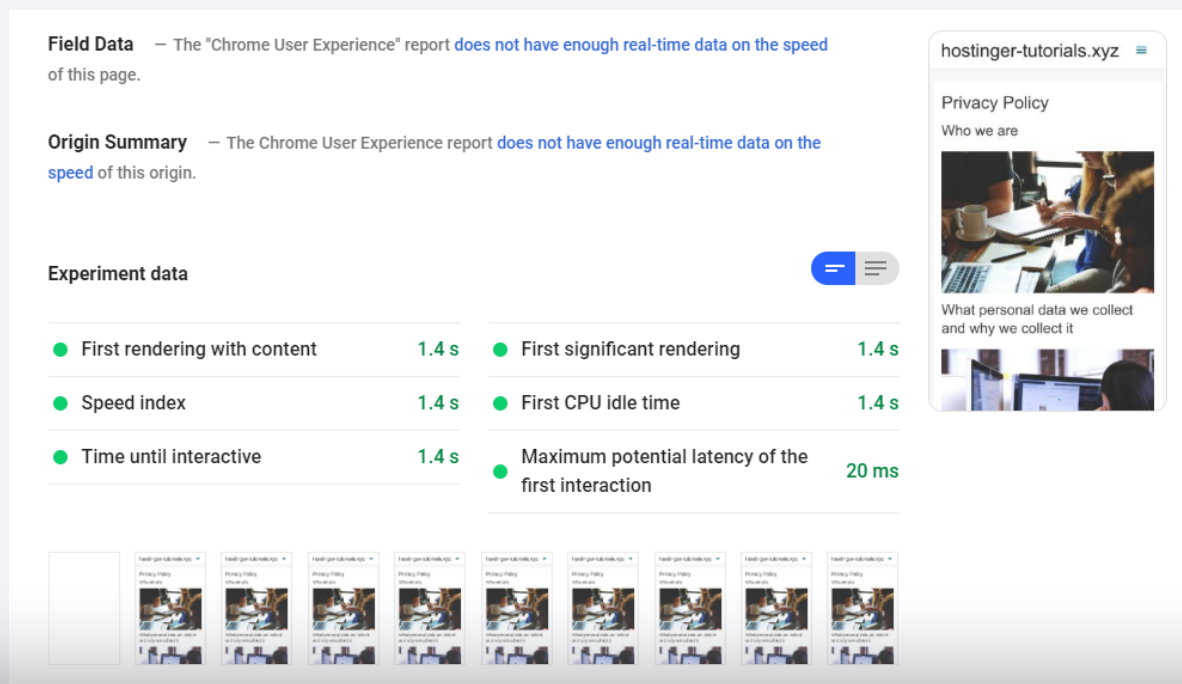
There are resources that block the first rendering of the page. We recommend that you display the inserted critical JavaScript and CSS elements and postpone all the non-essential ones.

[More information](#)



There are several WordPress plugins that can help you [insert critical resources](#) or [postpone less important resources](#). Keep in mind that the optimizations offered by these plugins can block functions of your theme or your plugins, so you probably have to make changes to the code.

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



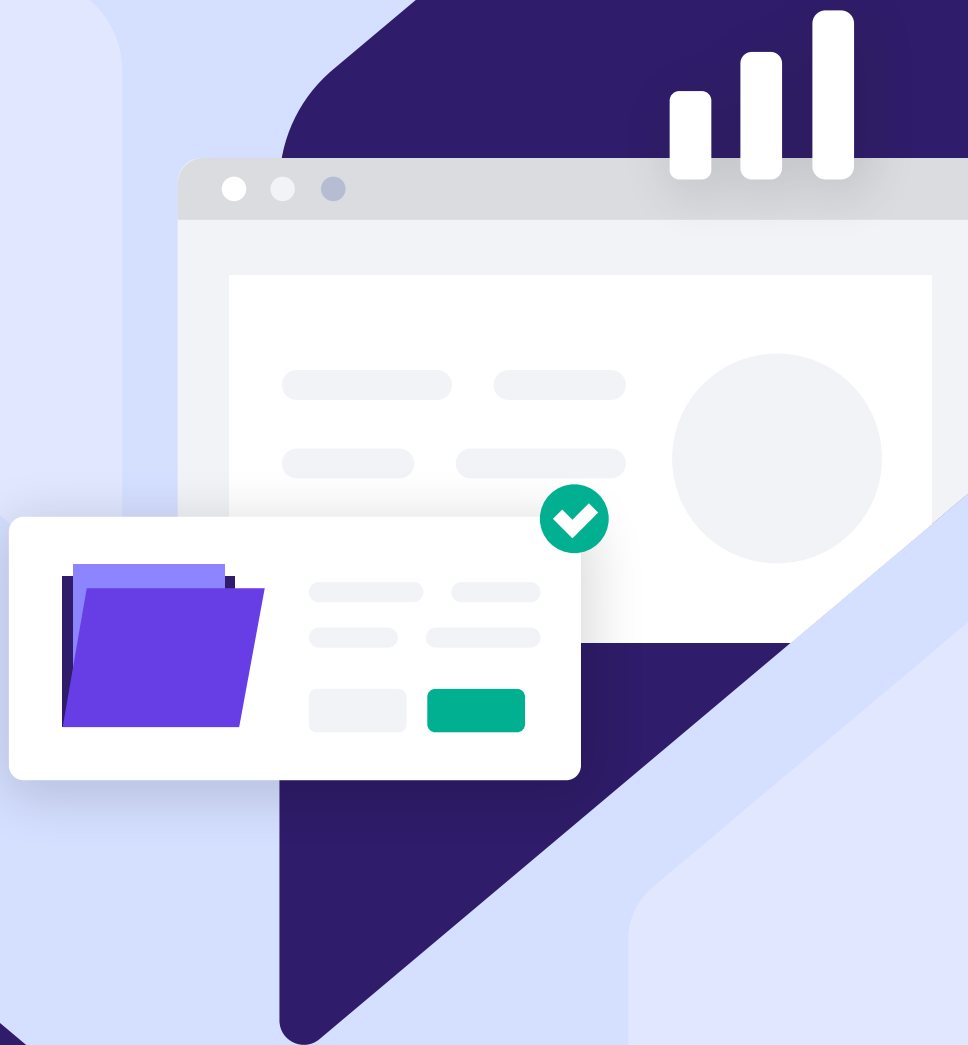
Comparing these results with those from the previous chapter, there's a reduction of about .2 seconds across all metrics. This round of performance optimization offers the fastest overall loading times we've seen so far, including our initial tests in Chapter 2.

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 by a whole second 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](#).

Speed Up Your Website;
8 Practical Tips that Work



6

Enabling Browser Caching



Every time someone visits your website, they load a lot of 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 have an impact on your server's performance.

The solution to this problem is called **caching**. This process enables you to store parts of your website in visitors' browsers. That way, when they return, 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 know by now, a lot goes on in the background every time someone visits your website. In most cases, your browser will load dozens of elements behind the scenes before users see anything appear on their screens. That process, ideally, 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, scripts, and more.

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

Reducing the number of elements that visitors have to load.

Usually, option number three is the last resort unless we're talking about website components that don't fulfill any purpose. In most cases, for example, you wouldn't remove all the images from your pages, or key elements such as menus and contact forms, just to reduce loading times.

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

In short, this process involves your visitors temporarily storing some parts of your website on their computers. That way, they don't need to load entire pages from scratch over and over again. Instead of making a request to your site's server, their browsers simply retrieve your content from their caches.

All of the most popular modern browsers support caching. However, it's up to you to configure it for your website.

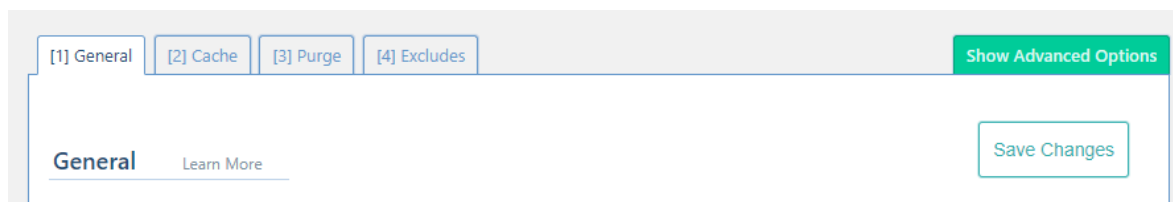
Most importantly, you have to set timeout rules. This ensures visitors will load new content occasionally so that they can see changes to your website. Otherwise, they would repeatedly view the outdated versions of your content stored in their browsers.

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

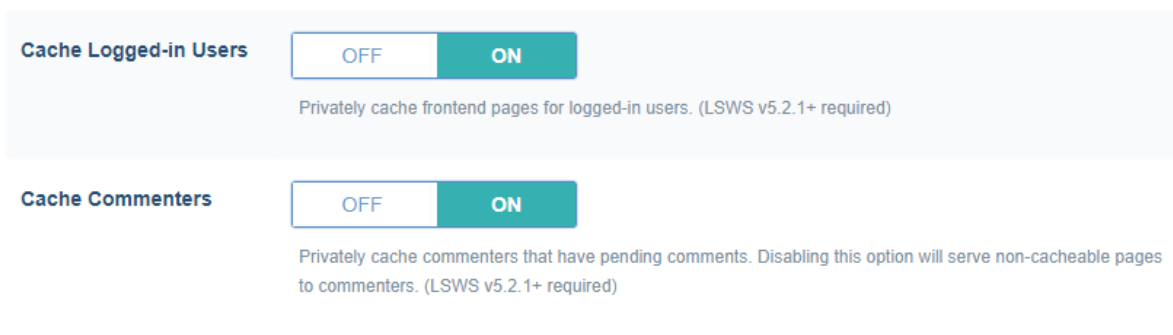
Once more, we're going to rely on the [LiteSpeed Cache](#) plugin to help us implement this feature. While it's not the only option as far as caching plugins go, it does include a lot of settings that enable us to finetune our website's local storage configuration. Let's take a look at them.

1. Enable Caching and Configure Your Timeout Settings

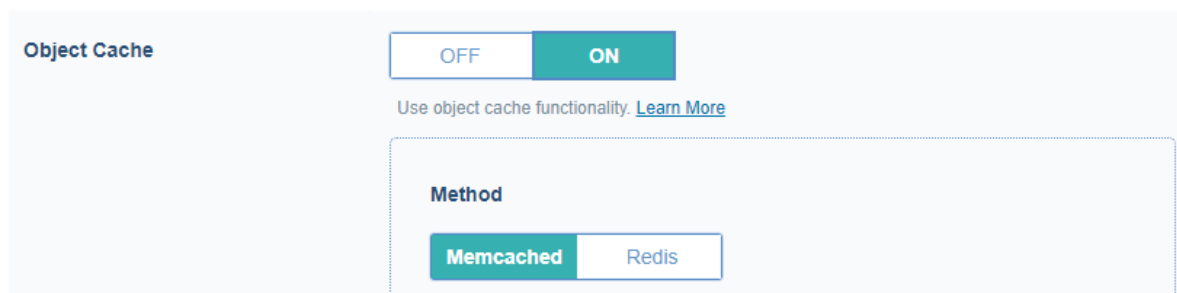
Once more, we need to display LiteSpeed Cache's advanced options. Go to **LiteSpeed Cache > Settings** and click on **Show Advanced Options** in the top right corner of the page:



The first thing you want to do is go to 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.



Then, navigate to the **Advanced** tab. Here, you get access to more complex settings, such as enabling object caching, which is different from browser caching. With object caching, you store database queries so users don't have to process them individually, which can help reduce loading times further. The **Object Cache** setting is turned off by default, so let's enable it.



Object Cache

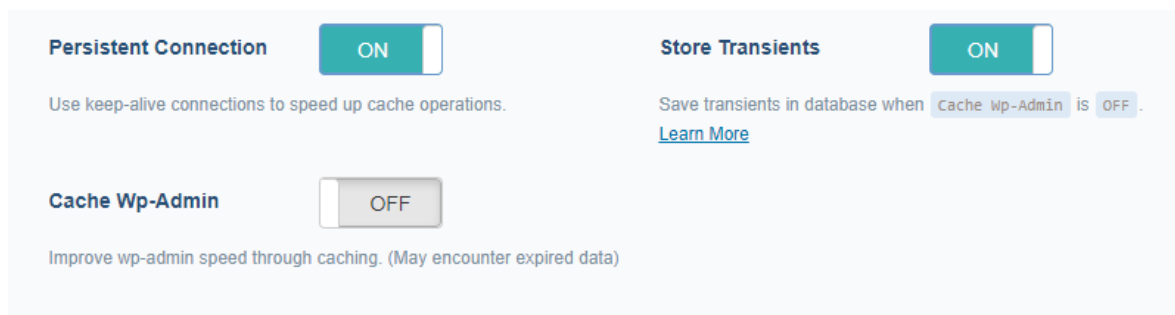
OFF ON

Use object cache functionality. [Learn More](#)

Method

Memcached Redis

If you turn this setting on, we recommend you turn off the **Cache Wp-Admin** option so it doesn't apply to your dashboard. That way, you'll avoid running into errors when updating your website.



Persistent Connection ON

Use keep-alive connections to speed up cache operations.

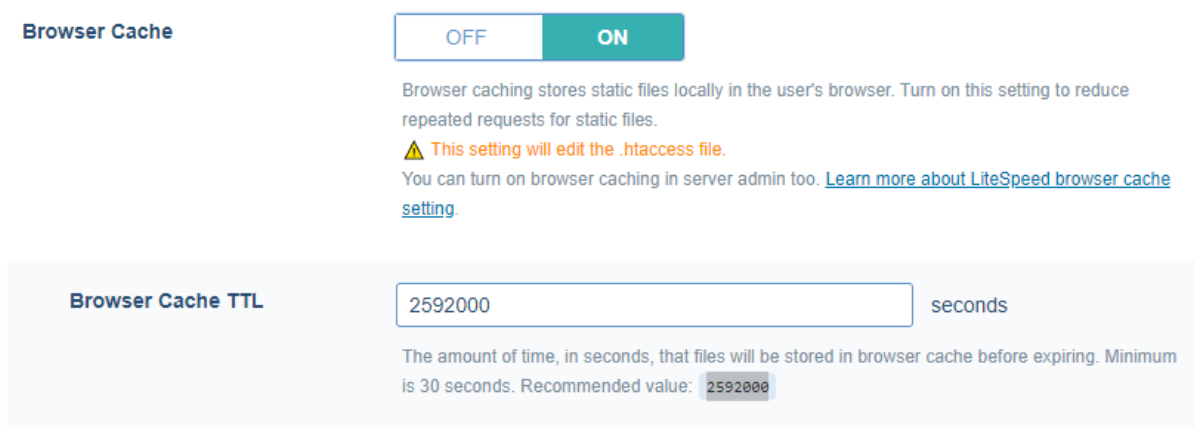
Store Transients ON

Save transients in database when **Cache Wp-Admin** is OFF. [Learn More](#)

Cache Wp-Admin OFF

Improve wp-admin speed through caching. (May encounter expired data)

Now scroll down further and enable the **Browser Cache** setting, which is what will speed up loading times for all your visitors. While you're at it, review the **Browser Cache TTL** value right below it.



Browser Cache

OFF ON

Browser caching stores static files locally in the user's browser. Turn on this setting to reduce repeated requests for static files.

⚠ This setting will edit the .htaccess file.

You can turn on browser caching in server admin too. [Learn more about LiteSpeed browser cache setting.](#)

Browser Cache TTL

2592000 seconds

The amount of time, in seconds, that files will be stored in browser cache before expiring. Minimum is 30 seconds. Recommended value: 2592000

The number you see above governs your cache expiry settings. We set ours to 2592000 seconds, which translates to 30 days.

This means that once 30 days have gone by since a user's initial visit to our website, their browser will need to reload every file. That way, we can prevent situations where they load cached elements after we've updated them. The process then repeats.

Now that we've enabled caching for visitors and logged-in users, and disabled it for your admin dashboard, 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 specific 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:

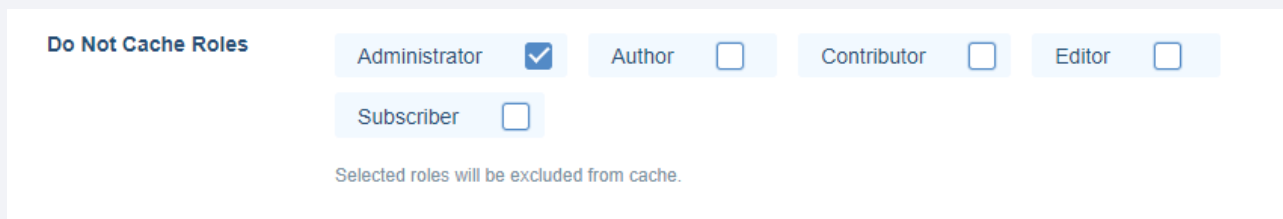
Do Not Cache URIs

Paths containing these strings will not be cached. The URLs will be compared to the REQUEST_URI server variable. For example, for `/mypath/mypage?aa=bb`, `mypage?aa=` can be used here.

To match the beginning, add `^` to the beginning of the item. To do an exact match, add `$` to the end of the URL. One per line.

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.

Earlier, we disabled caching for the WordPress admin area. If you scroll down within the **Excludes** tab, you can also select user roles that will be excluded from caching:



Do Not Cache Roles

Administrator	☒	Author	☐	Contributor	☐	Editor	☐
Subscriber	☐						

Selected roles will be excluded from cache.

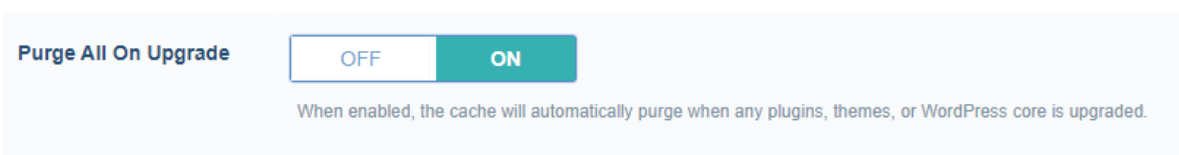
If you work with others on your website, it can be a smart idea to disable caching for them as well. That way, they'll always have access to the latest versions of your pages and you can avoid any mix-ups.

3. Tweak Your Cache Purge Settings

During 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 enable you to 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.

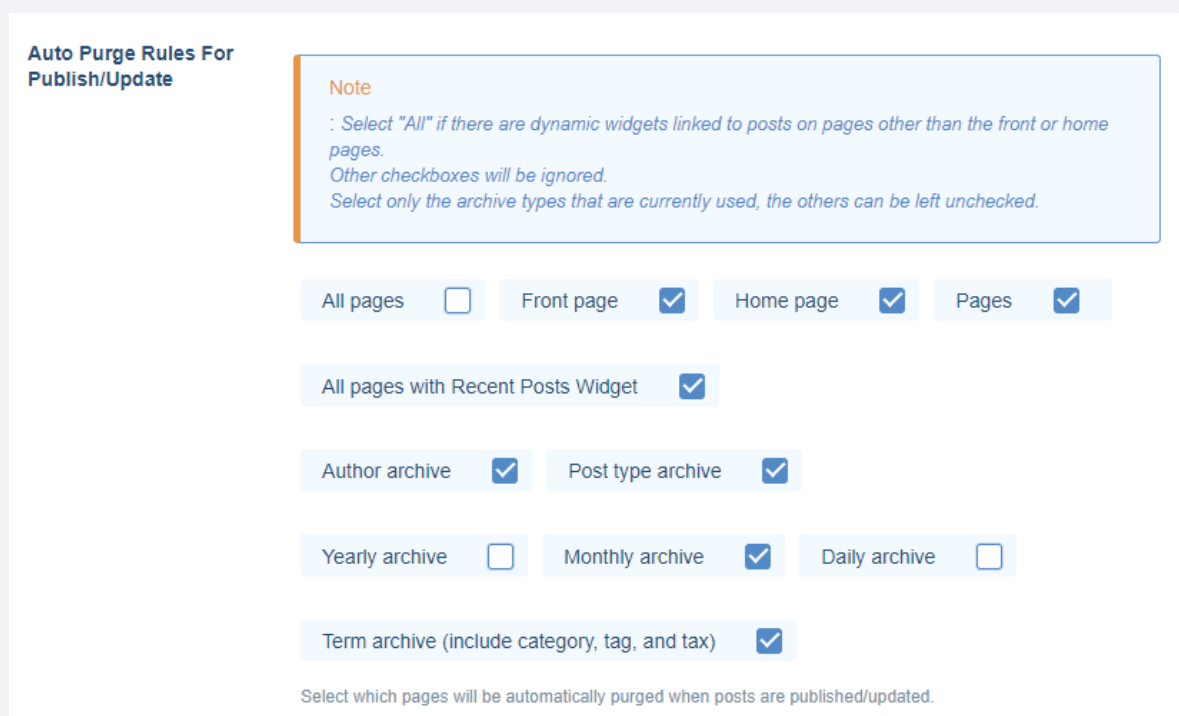
First off, you'll notice a setting labeled **Purge All On Upgrade**. This option will clear your users' caches whenever you update plugins, themes, or the WordPress core:



Purge All On Upgrade

When enabled, the cache will automatically purge when any plugins, themes, or 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, forcing users to reload it so they don't miss out on your updated content.

By combining 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. However, PageSpeed Insights' unique methods for assessing performance are still helpful. Here's what our latest round of testing on this platform shows:

Field Data — The "Chrome User Experience" report [does not have enough real-time data on the speed](#) of this page.

Origin Summary — The Chrome User Experience report [does not have enough real-time data on the speed](#) of this origin.

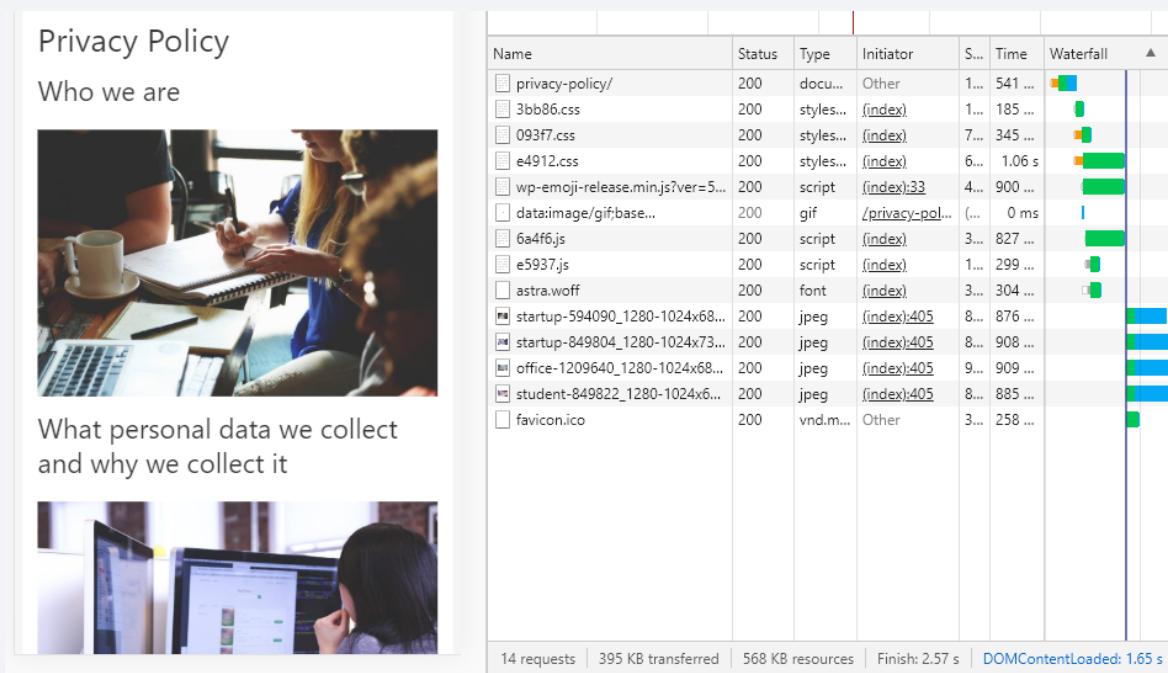


Experiment data

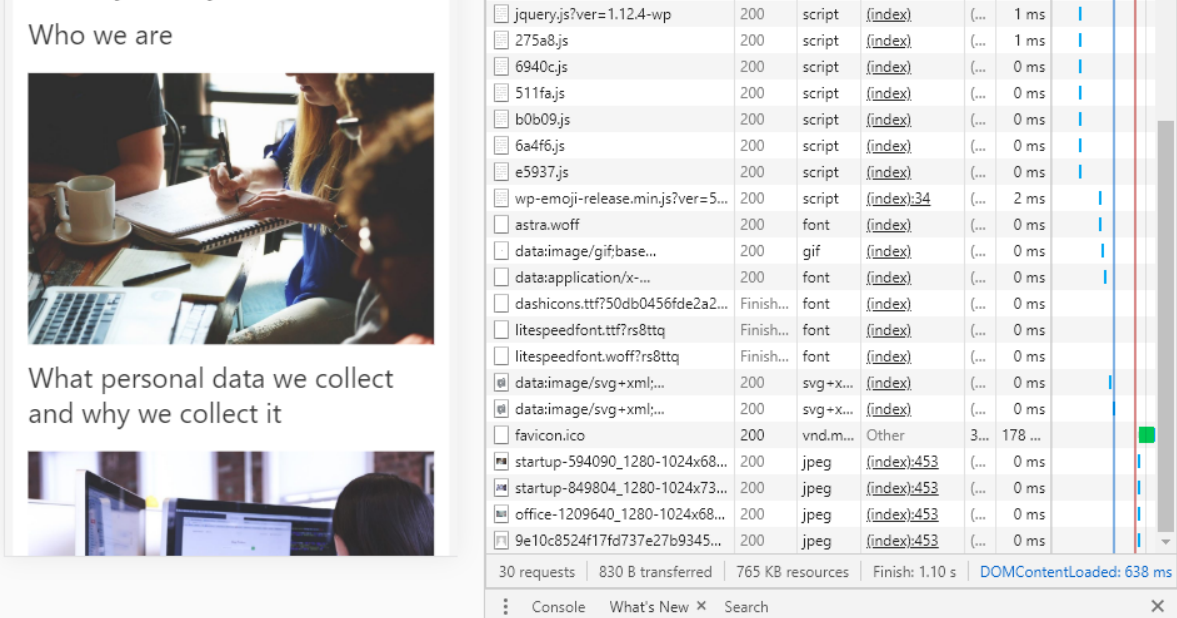
● First rendering with content	0.4 s	● First significant rendering	0.4 s
● Speed index	0.5 s	● First CPU idle time	0.4 s
● Time until interactive	0.4 s	● Maximum potential latency of the first interaction	20 ms

According to the service, enabling caching cut down our page speed to about one-third of our previous loading times. During the last chapter, our website was taking an average of about 1.4 seconds to load. That's a massive change for any metric.

Those numbers looked too good, so we decided to run a couple of 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 2.57 seconds, which, of course, is not optimal. After enabling caching and reloading the content a couple of times, that number went down dramatically:



Who we are

What personal data we collect and why we collect it

Request	Status	Type	Size	Time
jquery.js?ver=1.12.4-wp	200	script	(index)	1 ms
275a8.js	200	script	(index)	1 ms
6940c.js	200	script	(index)	0 ms
511fa.js	200	script	(index)	0 ms
b0b09.js	200	script	(index)	0 ms
6a4f6.js	200	script	(index)	0 ms
e5937.js	200	script	(index)	0 ms
wp-emoji-release.min.js?ver=5...	200	script	(index):34	2 ms
astra.woff	200	font	(index)	0 ms
data:image/gif;base...	200	gif	(index)	0 ms
data:application/x-...	200	font	(index)	0 ms
dashicons.ttf?50db0456fde2a2...	Finish...	font	(index)	0 ms
litespeedfont.ttf?rs8ttq	Finish...	font	(index)	0 ms
litespeedfont.woff?rs8ttq	Finish...	font	(index)	0 ms
data:image/svg+xml;...	200	svg+x...	(index)	0 ms
data:image/svg+xml;...	200	svg+x...	(index)	0 ms
favicon.ico	200	vnd.m...	Other	3...
startup-594090_1280-1024x68...	200	jpeg	(index):453	0 ms
startup-849804_1280-1024x73...	200	jpeg	(index):453	0 ms
office-1209640_1280-1024x68...	200	jpeg	(index):453	0 ms
9e10c8524f17fd737e27b9345...	200	jpeg	(index):453	0 ms

30 requests | 830 B transferred | 765 KB resources | Finish: 1.10 s | DOMContentLoaded: 638 ms

Console | What's New | Search

That same network test took just a little over one second 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.

Speed Up Your Website;
8 Practical Tips that Work

7

Integrating a Content Delivery Network

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)

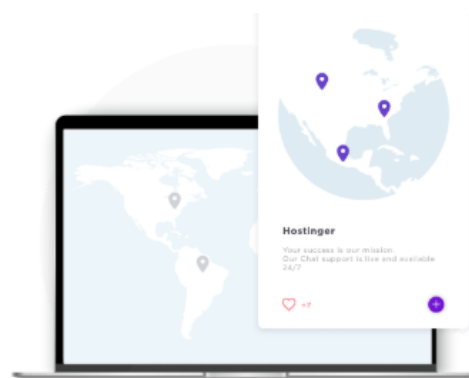
Every website requires a physical server, which needs to be stored somewhere. Depending on your hosting provider, you might have several options when it comes to the location of yours.

For example, at [Hostinger](#), we have servers in seven different countries around the world:



Hostinger: Global Data Centers

Our servers are deployed in 7 different regions: UK, US, Brazil, Netherlands, Singapore, Indonesia and Lithuania. Customers can enjoy lowest latency and unmatched reliability as we deploy in globally connected Tier-3 datacenters.



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'll have to wait for an additional second or two when compared to a local user.

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

On-site optimization such as image compression only goes so far. It also doesn't account for slow internet speeds or visitors located on the other side of the world. You can't do anything about users who are stuck with dial-up connections, but you can minimize the impact of your server's location by using 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. Visitors' requests are automatically routed to the nearest server, which returns its copy of your content.

In theory, this leads to faster loading times. Another benefit is that CDNs take some of the load off your server's shoulders. That means your website will be able to handle larger volumes of traffic without impacting performance negatively.

➤ 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.

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



Products Solutions Resources Developers For Enterprise Pricing

Support | Sales: +1 (888) 99 FLARE | @ English +

Log In

Sign Up

Under Attack?

The Integrated Global Cloud Platform

Trusted by over 26 million Internet properties. Cloudflare provides a scalable, easy-to-use, unified control plane to deliver security, performance, and reliability for on-premises, hybrid, cloud, and SaaS applications.

 [Learn how Cloudflare works](#)



For the following steps, we're going to show you how to set up a CDN using Cloudflare since it offers a free plan that's easy to integrate with WordPress. However, we encourage you to research additional services before you sign up for a paid plan for your own website.

2. Sign Up for Cloudflare and Retrieve Your API Key

Signing up for a Cloudflare account works much the same as with any other online service. You fill out a few forms with your contact information and other details, then choose which plan you want to use.

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 perks of a CDN.

Get started with Cloudflare

Email

Required. Must be an email address.

Password

[Show](#)



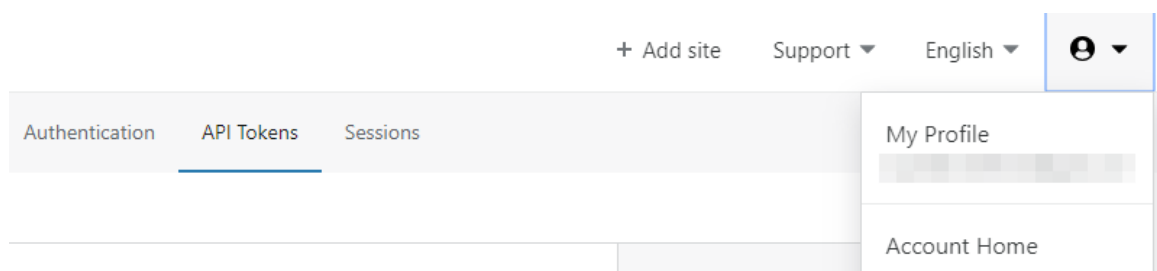
Performance
Global CDN and web optimization



Reliability
Always available and online



Once your account is ready and you have access to the Cloudflare dashboard, click on the profile icon on the top right corner of the screen and select the **My Profile** option from the drop-down menu:



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

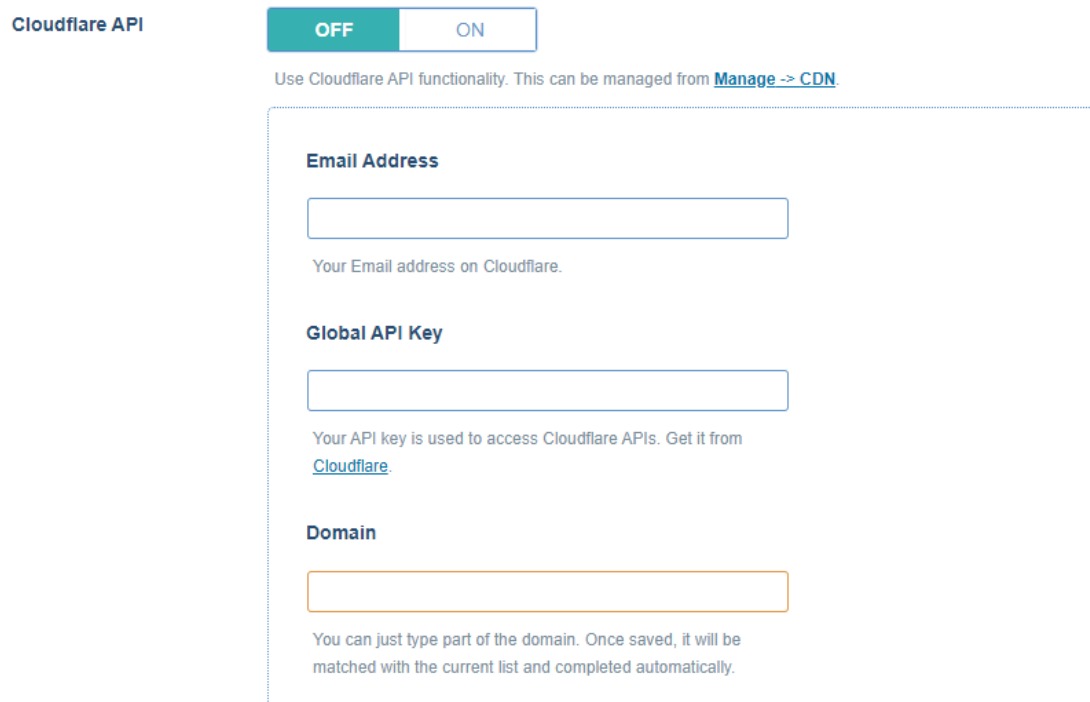
API Keys	
Keys used to access Cloudflare APIs.	
Global API Key	Change View
Origin CA Key	Change View
Help	

Cloudflare will ask you to confirm your account password before granting you 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

Once more, we're going to rely on [LiteSpeed Cache](#) to help us achieve our next performance optimization technique. It works with both Quic Cloud and Cloudflare out of the box and it enables you to integrate either option in a matter of minutes.

To get started, go to **LiteSpeed Cache > Settings** and move to the **CDN** tab. Scroll down until you see the section labeled **Cloudflare API**.

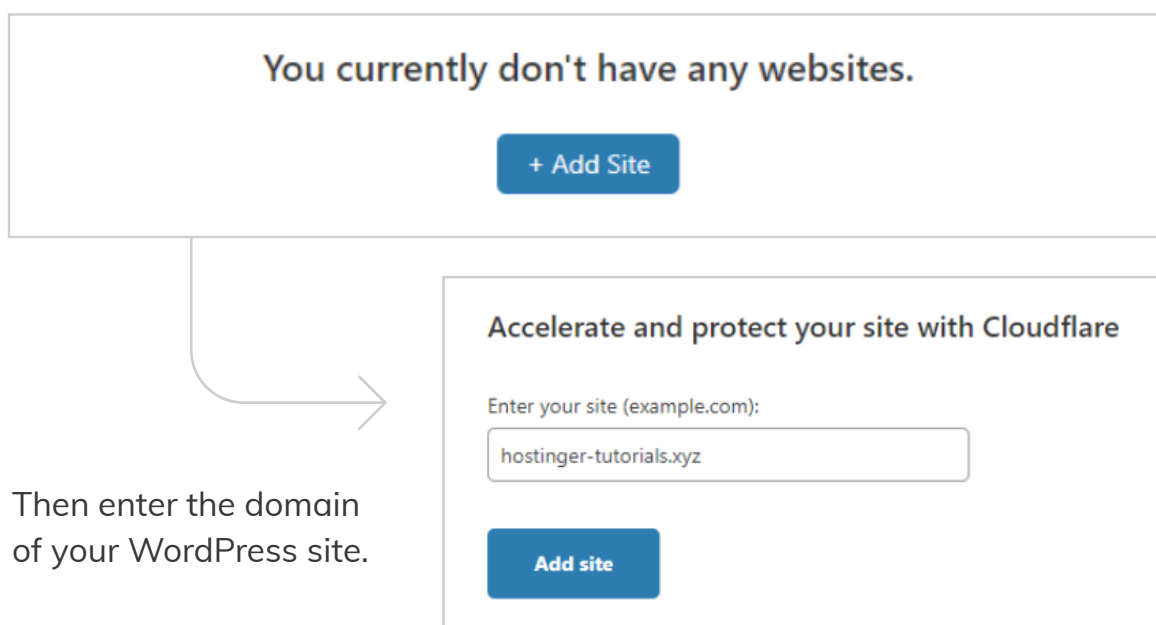


The screenshot shows the 'Cloudflare API' settings section. At the top, there is a toggle switch labeled 'OFF' (in a green box) and 'ON' (in a white box). Below the toggle, a note states: 'Use Cloudflare API functionality. This can be managed from [Manage -> CDN](#).' The main settings area is enclosed in a dashed blue border and contains three fields: 'Email Address' with a text input box and a note 'Your Email address on Cloudflare.', 'Global API Key' with a text input box and a note 'Your API key is used to access Cloudflare APIs. Get it from [Cloudflare](#).', and 'Domain' with a text input box and 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 you for the email address you used to register your Cloudflare account, as well as the API key you obtained during Step 2. Finally, you need to enter your website's domain name. When you're done, turn the **Cloudflare API** setting to **ON** and save your 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 as well, to finish integrating the two platforms. On your Cloudflare **Home** tab, click on **Add Website**.



You currently don't have any websites.

+ Add Site

Accelerate and protect your site with Cloudflare

Enter your site (example.com):

hostinger-tutorials.xyz

Add site

Then enter the domain of your WordPress site.

At this stage, Cloudflare will ask you to confirm the plan you've selected for your website. For our test, we'll stick with the free tier, which is perfect for small sites.

Select a plan

Free	Pro	Business	Enterprise
\$0/month	\$20/month per domain	\$200/month per domain	Get in touch
For personal websites, blogs, and anyone who wants to explore Cloudflare.	For professional websites, blogs, and portfolios requiring basic security and performance.	For small eCommerce websites and businesses requiring advanced security and performance, PCI compliance, and prioritized email support.	For companies requiring enterprise-grade security and performance, prioritized 24/7/365 phone, email, or chat support, and guaranteed uptime.

[Learn more about our plans](#)

Confirm plan

Now [Cloudflare](#) will scan your DNS records to make sure that it can “accelerate” traffic to your website. Notice the Cloudflare icons below, which mean the service is working.

 Proxied: Accelerates and protects traffic

 DNS resolution only: Bypasses Cloudflare

Note: Records with no cloud icon use DNS resolution but cannot be proxied.

A

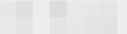
Name

IPv4 address

Automatic TTL



Add Record

Type	Name	Value	TTL	Proxy status
A	ftp	points to 	Automatic	 
A	hostinger-tutorial...	points to 	Automatic	 

To complete the setup process, Cloudflare will ask you to update your domain’s nameservers. If you’re a Hostinger customer, you can do this from your hosting control panel by selecting the **Manage** option next to your domain.

On the next page, you can view the nameservers you’re using now.

Update nameservers ?

Nameserver #1

Nameserver #2

Nameserver #3

Nameserver #4

Use Default

Update

Here's a quick look at what your Cloudflare nameservers might look like. However, keep in mind that it doesn't always use the same ones, so check your account and copy your specific information from there.



Complete your nameserver setup

hostinger-tutorials.xyz is not yet active on Cloudflare.

1. Log in to your registrar account

Determine your registrar via [WHOIS](#).

Remove these nameservers:

```
ns1.dns-parking.com  
ns2.dns-parking.com
```

2. Replace with Cloudflare's nameservers



Nameserver 1

```
elliott.ns.cloudflare.com
```

Click to copy



Nameserver 2

```
gwen.ns.cloudflare.com
```

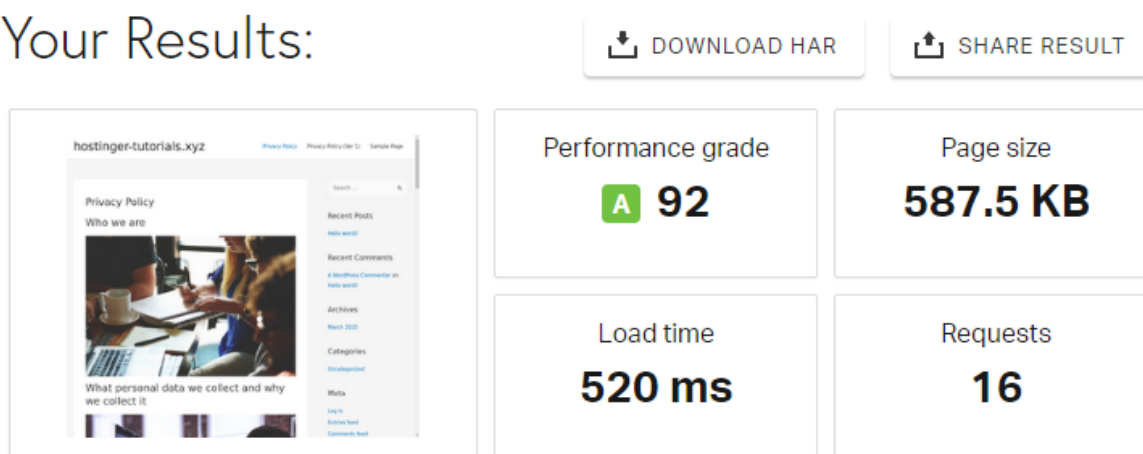
This step is important because it enables Cloudflare to intercept traffic to your website and route it through its servers. That's what's going to cut down your site's loading times. Once you're done, you can test your site to see what a difference it makes.



The Impact of Using a CDN on Site Speed

By the end of Chapter 6, our test website was loading in less than half a second according to PageSpeed Insights. With that in mind, let's see how our page is performing after integrating a CDN, starting with Pingdom Tools.

Your Results:



The last time we ran a Pingdom test was two chapters ago. On that occasion, we received a result of 535 milliseconds. With our CDN enabled, we're getting slightly lower loading times consistently.

Keep in mind, however, that the result above is for a Washington, DC-based testing server, which was already providing us fantastic numbers from the get-go.

One of the perks of using a CDN is that it should improve loading times across the globe. With that in mind, we re-tested our website using Pingdom 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:

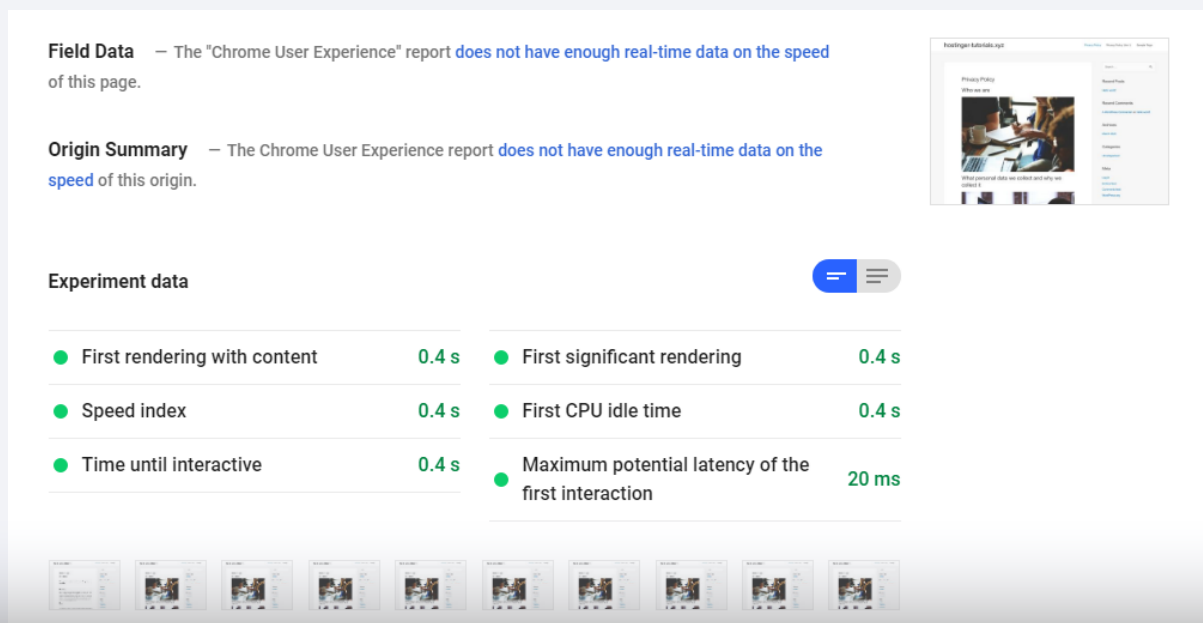
1. **United Kingdom, London:** 0.565 seconds / 0.912 seconds
2. **USA, Washington D.C.:** 0.266 seconds / 0.520 seconds
3. **Japan, Tokyo:** 1.59 seconds / 1.50 seconds
4. **Australia, Sydney:** 1.42 seconds / 1.64 seconds
5. **Brazil, Sao Paulo:** 1 seconds / 1.14 seconds

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. The test page we were using back then weighed a mere 55.7 KB. Now, five chapters later, it weighs more than ten times that amount – 587.5 KB.

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:



According to PageSpeed Insights, our loading times have decreased slightly compared to our results from the last chapter, which is precisely what we want to see.

Keep in mind that Pingdom is the superior platform when it comes to global loading times. It's expected that we would see a more significant difference in our results for that test than in Google's.

At this stage, optimizing this website even further becomes difficult because it's so young. The more content, files, and traffic your website has, the more you'll see and feel the impact of taking action to improve its performance.

Normally, it's not uncommon to run into websites that take ages to load. That's because people forget the impact of proper maintenance. Now that you've seen the results firsthand, however, you should understand how much of an edge it can give you.



Speed Up Your Website;
8 Practical Tips that Work



8

Making Sure Your Website Remains Fast



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 you'll want to carry out periodically to make sure your website remains speedy. Let's get started.

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

A significant part of maintaining your website's performance depends on monitoring. You need to keep an eye on how it's loading for visitors and whether you feel the admin dashboard is sluggish. Then, you'll be able to make appropriate adjustments to keep your site in tip-top shape.

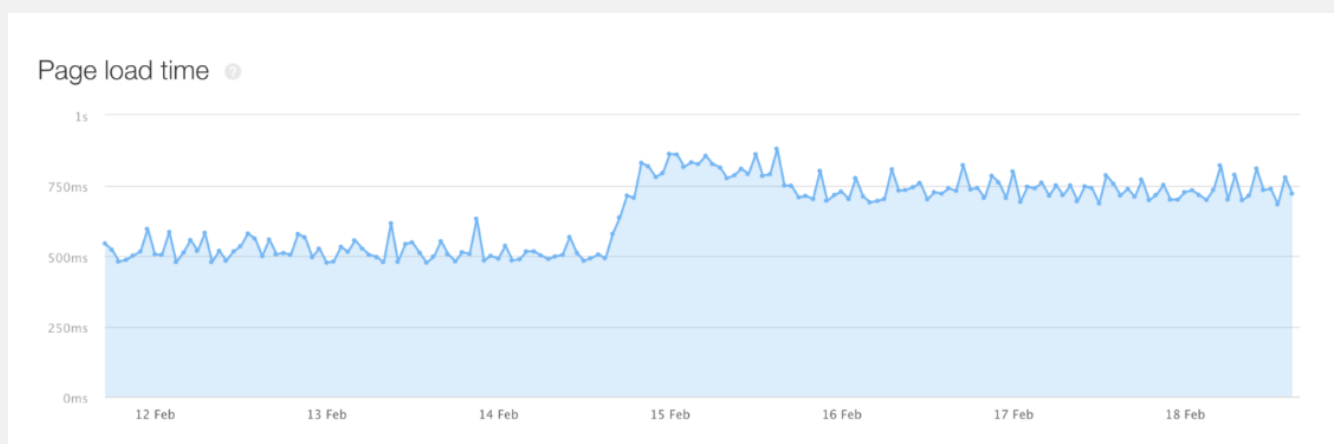
1. Monitor Your Loading Times Periodically

Over the past chapters, we've introduced you to several tools you can use to monitor your website's loading times. Namely, 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.

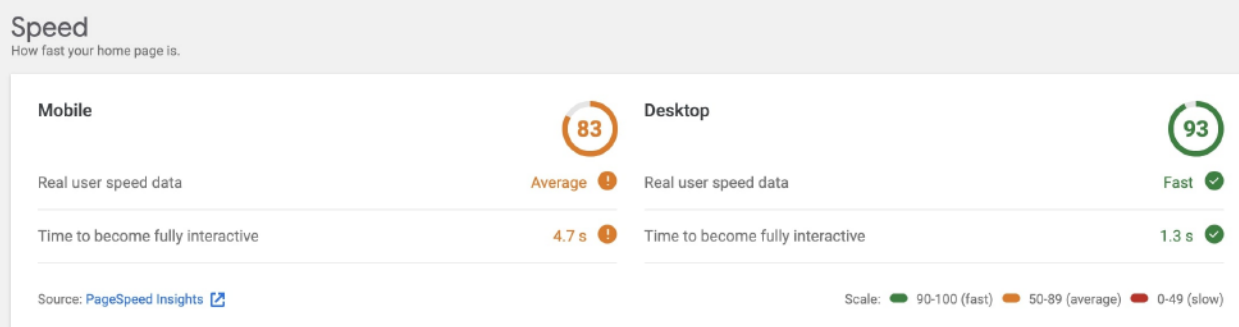
If you want to take it to the next level, you can sign up for paid website monitoring tools. Pingdom itself offers a suite of services available through premium plans that enable you to automate performance testing:



One of our favorite aspects of this particular service is it enables you to monitor performance for real-life users, instead of depending solely on the data from hypothetical tests. That means you can know, at all times, whether users are experiencing slowdowns or if everything is working as intended.

Real-User Monitoring (RUM) is a feature that many website maintenance platforms offer. However, this type of service usually doesn't come cheap, so keep that in mind.

As for PageSpeed Insights, there's a relatively new plugin that can help you keep an eye on your overall score and loading time. [Google Site Kit](#) integrates several different platforms with WordPress, including its performance monitoring tool:



This way, you can note any drastic changes without having to leave your [WordPress dashboard](#). However, you will still need to visit the PageSpeed Insights page to view your detailed results and recommendations for performance improvements. It's worth noting that it only tests your Home page as well.

2. Deactivate Plugins You're Not Using

As we tested during Chapter 2, some plugins can have an impact on your website's performance. Depending on which ones you're using and your web host, their influence may be minimal. However, we're trying to squeeze every last millisecond of performance we can out of your website.

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

To be really thorough when it comes to removing plugins from your site, you should also check out your [WordPress database](#) and remove any leftover tables or rows. You can learn more about how to access and manage your database in the MySQL portion of [our Knowledge Base](#).

3. Keep WordPress and All Its Components Up to Date

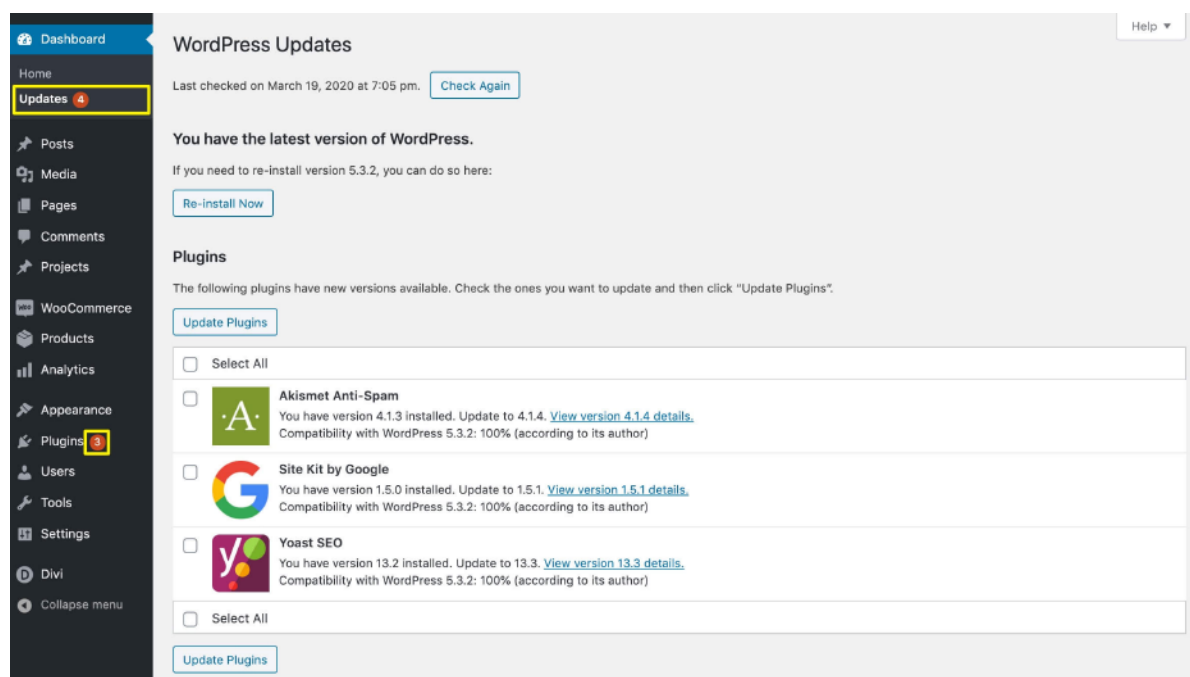
We've barely talked about updates so far, but that's not because they aren't important. Your WordPress website should always be **fully up-to-date**, which means:

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 should never pose a major problem.

Any available updates for WordPress or your plugins and theme will be listed in the **Updates** tab in your dashboard:

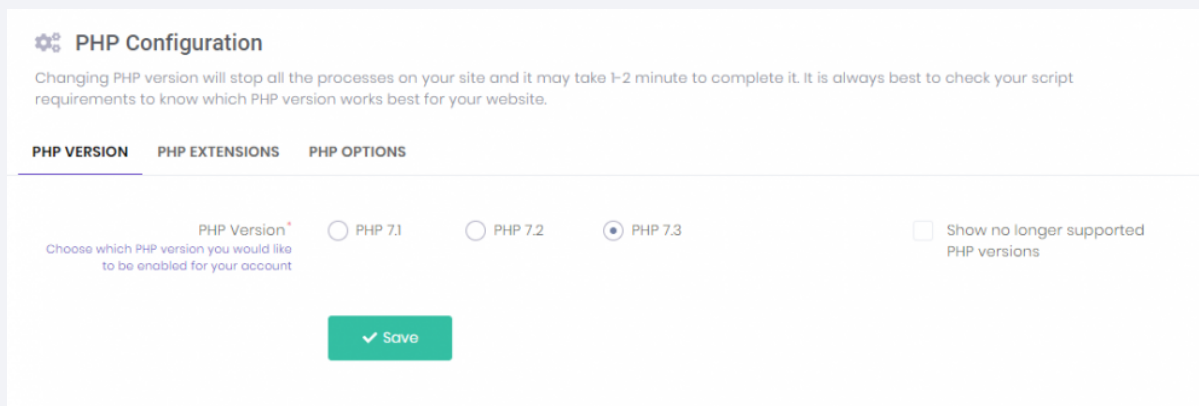


You'll also see red notifications in your sidebar. Simply click on the indicated buttons or links to run individual or bulk updates as needed.

4. Upgrade to the Latest Version of PHP

WordPress is built on PHP. That means the version of the programming language your server uses can have a major impact on your website's performance. To give you an idea of just how significant this factor is, PHP 7 is three to four times faster than PHP 5.

Those performance gains translate directly to your website. However, updating PHP isn't as simple as upgrading plugins or WordPress itself. However, some web hosts – [including Hostinger](#) – will enable you to change PHP versions directly from your control panel:



The screenshot shows a 'PHP Configuration' panel with a gear icon. Below the title is a warning: 'Changing PHP version will stop all the processes on your site and it may take 1-2 minute to complete it. It is always best to check your script requirements to know which PHP version works best for your website.' There are three tabs: 'PHP VERSION', 'PHP EXTENSIONS', and 'PHP OPTIONS'. The 'PHP VERSION' tab is active. It contains a section 'PHP Version*' with the instruction 'Choose which PHP version you would like to be enabled for your account'. Below this are three radio buttons: 'PHP 7.1', 'PHP 7.2', and 'PHP 7.3'. The 'PHP 7.3' option is selected. To the right of these buttons is a checkbox labeled 'Show no longer supported PHP versions', which is currently unchecked. At the bottom center of the panel is a green 'Save' button with a checkmark icon.

If your host doesn't enable you to update to the latest version of PHP directly, then you will have to [do so manually](#), which requires technical knowledge, or return to the market and look for a new provider.

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.

At some point, however, you will likely outgrow shared hosting if your site starts to attract more traffic. In our experience, if you're seeing daily visitors in the hundreds, then it's probably time to consider a plan with a bit more firepower, such as a Virtual Private Server (VPS).

There are a lot of accessible hosting plans beyond shared options. VPSs, for example, come in all shapes and sizes, with some well below the **\$10** per month mark:

VCPU	MEMORY	STORAGE	GEEKBENCH SCORE	BANDWIDTH	PRICE	
1 vCPU	1 GB	20 GB	3513	1000 GB	SAVE 60% \$9.95/mo \$ 3.95 /mo	ADD TO CART
2 vCPU	2 GB	40 GB	5363	2000 GB	SAVE 55% \$19.95/mo \$ 8.95 /mo	ADD TO CART
3 vCPU	3 GB	60 GB	7545	3000 GB	SAVE 68% \$39.95/mo \$ 12.95 /mo	ADD TO CART

Before you consider upgrading your hosting plan, however, you want to be sure you've tried everything you can to optimize performance on your end. If you're confident that you've reached the limits of your current plan, then go ahead and move up to the next level.

➤ Conclusion

If you want your website to succeed and retain visitors, you need to make the experience of using it as enjoyable as possible. That means creating quality content, engaging with your audience, and of course, ensuring your site always loads as fast as possible.

Ideally, you want your website to load in under two seconds. The lower that number is, the better. Conversely, once you reach the three-second mark, users may start to lose interest and navigate away from your pages.

Fortunately, reducing your page loading times is not as complicated as it might sound. All you need is a solid foundation for your website. In this e-book, 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 your images.
4. Minifying your website's CSS and JavaScript.
5. Enabling browser caching.
6. Integrating a CDN.
7. Ensuring your website remains fast over time.

There are a lot of factors that influence website loading times, but none of them are beyond your control. If you're willing to invest some time now into optimizing your site's performance, it will pay off dramatically over the long run. Plus, your users will be happier, which is the ultimate metric of your website's success.