WPSHOUT!

Fastest WordPress Hosting With No Caching

2025 EDITION

Table of contents

Introduction

Have you ever stopped to think about how fast your WordPress site really is when the usual speed boosters - like caching - are taken out of the equation? 🤔

For many sites, especially ecommerce stores, membership sites, or more dynamic web creations, caching only helps a fraction of visitors. The rest are left relying entirely on the hosting provider's raw server performance. And if that performance isn't up to par, you can end up with sluggish load times, frustrated users, and lost conversions.

This report dives straight into a question many hosting providers would rather you not ask:

"Who is the fastest WordPress host when caching is disabled, and why does it matter?"

And it's not just a theoretical study of the topic that you're reading here. I've run detailed, real-world tests that strip away built-in optimizations to show you which hosts genuinely deliver speed, and which lean too heavily on caching to hide their weaknesses.

Why you should care

1. **Cut through marketing hype.** Hosting companies love bold speed claims, but how many hold up under real-world conditions when you prevent them from using things like caching? If your site relies on dynamic content or plugins, the answer might surprise you.

2. **Real-life scenarios.** Not every page you serve will be cached. Dynamic sites - like stores, membership sites, and custom apps - depend on raw server performance far more often than you think.

3. **Save time and money.** Choosing the right host now means fewer headaches later. The long-term price of hosting is usually relatively high[1], so it's in your best interest to make a good choice up front, find a provider that suits your site's unique needs, and avoid costly mistakes later on.

4. **Clear, actionable insights.** No fluff here - just clear results showing which hosts can handle the load and how you can boost your site's performance.

[1] For example, Bluehost - a popular host - will cost you $418 over the span of five years. SiteGround is even more - at $827 over five years. If you're interested in cheap hosting options, you'll find a link to an additional resource at the end of this report.

What you'll learn

This report is built to give you practical insights. First, I'll walk you through **how hosting performance changes** under various conditions. We'll test a fully loaded WordPress setup with lots of plugins, then a more streamlined setup, and finally, we'll compare those results to caching-enabled scenarios.

Along the way, I'll share **key takeaways and recommendations** to help you decide which host is right for you.

By the end, you'll know exactly how your host stacks up when it matters most. More importantly, you'll have the tools to make confident, informed hosting decisions that set your site up for success.

Hosting providers tested:

1. Hostinger
2. HostGator
3. Kinsta
4. Rocket
5. GreenGeeks
6. Namecheap
7. SiteGround
8. InMotion
9. A2 Hosting
10. Pressable
11. Bluehost
12. GoDaddy
13. Cloudways
14. WP Engine
15. DreamHost
16. Flywheel

Ready to find out who delivers speed when it counts? Let's dive in.

RESULTS: how hosting performance shifts *with* and *without* caching

Before we dive into the numbers, let's take a moment to understand the bigger picture. My testing focused on two crucial performance aspects for each host:

- **Forcing the server to show its raw performance.** This is about how a host handles traffic when caching and other speed-enhancing tricks are completely disabled. For sites with personalized features - like shopping carts, membership dashboards, or user-specific data - this is where the server's real strength is put to the test.
- **Having a baseline with built-in optimizations allowed.** I also evaluated how the host performs under "normal" conditions, with its caching and optimization features fully active. This reflects the experience most site owners get from their hosting provider if they never attempt to tune up the default setup.

By comparing these scenarios, we get a clearer view of each host's true capabilities. It reveals whether their speed comes from good infrastructure or clever caching, and how well they can handle demanding, dynamic sites versus simpler setups.

In the next sections, I'll break down the test results across different use cases:

- Scenario #1: Heavy plugins, no cache
- Scenario #2: Minimal setup, no cache

Scenario #1:
Heavy plugins, no cache

I'm first going to show you the numbers for the final case I've measured: when test sites have ~15 plugins installed and caching disabled. Next, I'll show the results when testing simpler sites with a minimal plugin stack (and caching still disabled).

I'm doing this because it's a more realistic scenario for most website owners. These days, very few sites run with just 2–3 plugins. It's far more common to see sites with 10, 20, or even more plugins installed.

Brace yourself; you're not going to enjoy this: 😬

Drum roll, please. 🥁

Not surprisingly, running a large plugin stack amplifies the pressure on servers, separating the stronger hosts from the weaker ones. These are the individual Times to First Byte (TTFB) when measuring from three locations, plus the average (ordered based on average TTFB):

Host	TTFB				VS. BASE				
	N. Virginia	California	Frankfurt	average	N. Virginia	California	Frankfurt	average	average %
Hostinger	0.35	0.43	0.24	0.34	-0.18	-0.47	0.09	-0.19	35%
HostGator	0.40	0.33	0.44	0.39	0.00	0.04	-0.30	-0.09	18%
Kinsta	0.42	0.46	0.48	0.45	0.31	0.38	0.40	0.36	-394%
Rocket	0.34	0.41	0.69	0.48	0.28	0.33	0.63	0.42	-649%
GreenGeeks	0.65	0.49	0.46	0.54	0.24	-0.05	-0.07	0.04	-8%
Namecheap	0.70	0.62	0.70	0.67	0.27	0.31	0.06	0.21	-46%
SiteGround	0.70	0.82	0.57	0.70	0.22	0.04	0.50	0.25	-57%
InMotion	0.91	0.74	0.72	0.79	0.86	0.46	0.19	0.50	-179%
A2	0.82	0.96	0.90	0.89	0.54	0.45	0.33	0.44	-98%
Pressable	0.87	0.94	1.03	0.95	0.75	0.81	0.53	0.70	-275%
Bluehost	1.27	0.87	0.83	0.99	1.20	0.79	0.29	0.76	-330%
GoDaddy	1.46	1.13	1.47	1.35	1.42	1.06	1.39	1.29	-2037%
Cloudways	1.52	2.04	0.65	1.40	1.20	1.48	0.58	1.09	-341%
WP Engine	1.40	1.80	2.07	1.76	0.91	1.65	1.34	1.30	-285%
DreamHost	3.60	5.30	1.88	3.59	2.97	4.80	0.92	2.90	-417%
Flywheel	4.75	2.67	4.28	3.90	3.56	1.91	4.24	3.24	-489%
								AVERAGE	-347%
								MEDIAN	-280%

What am I looking at?

- The table on the left shows the current TTFB measurements for the test setups - ~15 plugins installed, no cache.
- The table on the right shows the difference between the current measurements and the baseline tests - with minimal plugins installed and caching not blocked. 🟥 In general, the more red you see here, the worse the TTFBs have gotten vs the baseline.

📈 **The average TTFB increased by 347% across all hosts once caching was disabled, while the median increase was 280.**

Top performers

Some hosts actually performed better without caching than with it. I expected to see a few outliers like this, which could be due to testing glitches, network conditions, or other factors. Still, it's data, so I chose not to ignore these results and included them in the overall averages:

- **Hostinger** - 0.19s faster without caching - 35% improvement
- **HostGator** - 0.09s faster - 18%
- Right after that, we have **GreenGeeks** who has dropped off by only 0.04s with caching disabled - 8% setback.

I can also highlight **Namecheap** and **SiteGround** for having only slight drop offs in terms of raw numbers - respectively: 0.21s, and 0.25s worse.

Biggest losers of this test

You should generally expect load times to be longer with no caching and with each additional plugin installed. However, some hosts still look much worse than the others here:

- **Flywheel** recorded the worst drops in performance of all the hosts - on the average, the TTFB increased by 3.24s when comparing the baseline (with caching not disabled) to the final setup with heavy plugins installed on the site.
- **DreamHost** was the second worst at 2.90s.
- **WP Engine** and **GoDaddy** both hold the third place at 1.30s and 1.29s respectively. Staying on the topic of GoDaddy for a second; they experienced the worse drop in terms of percentage of all the hosts - 2037%.

Flywheel and WP Engine seem interesting to me since they are of course run by the same entity. While Flywheel remains the more approachable and easier to use of the two, it seems that when it comes to performance, there's no significant difference between it and WP Engine.

Other managed WordPress hosting?

Three interesting cases I noticed were Kinsta, Rocket, and Pressable. While the raw drops for all still look good, the percentages look huge:

- **Kinsta** increased their TTFB by only 0.36s, but it is a 394% drop compared to the cached setup.
- **Rocket** increased their TTFB by 0.42s, with it being a 649% drop.
- **Pressable** increased their TTFB by 0.7s, with it being a 275% drop.

While the no-cache results are still great, it suggests that these hosts rely heavily on aggressive caching in their default setups.

Scenario #2:
Minimal setup, no cache

Now, let's look at a more favorable scenario for web hosts: test sites with only 2–3 plugins installed, a minimal setup overall, and no caching. This helps highlight the impact plugins can have on load times (ordered based on average TTFB):

Host	TTFB				VS. HEAVY				
	N. Virginia	California	Frankfurt	average	N. Virginia	California	Frankfurt	average	average %
Hostinger	0.19	0.28	0.19	0.22	-0.16	-0.15	-0.05	-0.12	36%
Kinsta	0.27	0.32	0.36	0.32	-0.15	-0.13	-0.12	-0.14	30%
HostGator	0.35	0.27	0.34	0.32	-0.05	-0.06	-0.10	-0.07	18%
GreenGeeks	0.28	0.36	0.34	0.33	-0.37	-0.13	-0.12	-0.21	39%
Rocket	0.43	0.28	0.32	0.34	0.09	-0.13	-0.37	-0.14	29%
Namecheap	0.36	0.34	0.44	0.38	-0.34	-0.28	-0.26	-0.29	44%
WP Engine	0.48	0.47	0.67	0.54	-0.92	-1.34	-1.40	-1.22	69%
SiteGround	0.52	0.67	0.46	0.55	-0.18	-0.15	-0.11	-0.15	21%
InMotion	0.76	0.63	0.54	0.65	-0.15	-0.11	-0.17	-0.14	18%
A2	0.44	0.76	0.85	0.68	-0.38	-0.20	-0.05	-0.21	23%
Bluehost	0.77	0.62	0.79	0.73	-0.50	-0.25	-0.04	-0.26	27%
GoDaddy	0.83	0.73	0.63	0.73	-0.63	-0.40	-0.84	-0.62	46%
Pressable	0.74	0.72	0.76	0.74	-0.14	-0.22	-0.27	-0.21	22%
Cloudways	0.97	0.80	0.77	0.85	-0.55	-1.24	0.11	-0.56	40%
Flywheel	1.11	1.52	0.52	1.05	-3.65	-1.15	-3.75	-2.85	73%
DreamHost	2.07	0.99	1.35	1.47	-1.52	-4.31	-0.53	-2.12	59%
								AVERAGE	37%
								MEDIAN	33%

What am I looking at?

- The table on the left shows the TTFBs for the current test setups - minimal plugins installed, no cache.
- The table on the right shows the difference between the current measurements and the previous setup with heavy plugins installed. 🟩 Lots of green in this table tells us that the results for minimal plugins are better across the board for all setups.

📈 **The average and median TTFB improved by around 33-37%.**

⚠️ **The main takeaway is that no host performed better with more plugins installed.** In other words, if you can limit the number of plugins you have installed on your WordPress site, do it. **Every additional plugin impacts your load times.**

That said, we're splitting hairs at this point a bit, given that the exact increase in TTFB is rather minimal for most hosts.

Hosts where plugins have the most impact

A few hosts showed less resilience when handling additional plugins, performing significantly better with only a few installed:

- **DreamHost** - 2.12s faster TTFB with less plugins - 59% better
- **Flywheel** - 2.85s faster - 73% better
- **WP Engine** - 1.22s - 69% better

The rest of the pack

For the other hosts, the impact was rather marginal - usually in the range of sub 0.5s in reduced TTFB.

> 💡 **Bonus tip:** Did you know that images can make up anywhere from 30% to 85% of any web page's total size? You can learn more on how to optimize your images here.

What was the baseline with cache not disabled?

Let's zoom back out and look at what the baseline was for the testing: i.e. how hosts performed with caching not disabled, which is the default setup most users experience.

The average TTFB varied widely (ordered based on average TTFB):

Host	TTFB (s)				
	N. Virginia	California	Frankfurt		average
GoDaddy	0.04	0.07	0.08		0.06
Rocket	0.06	0.08	0.05		0.06
Kinsta	0.11	0.08	0.09		0.09
Bluehost	0.07	0.08	0.55		0.23
Pressable	0.13	0.13	0.50		0.25
InMotion	0.05	0.28	0.53		0.28
Cloudways	0.32	0.56	0.08		0.32
SiteGround	0.49	0.78	0.07		0.45
A2	0.28	0.50	0.57		0.45
WP Engine	0.49	0.15	0.73		0.46
Namecheap	0.44	0.31	0.64		0.46
HostGator	0.41	0.28	0.74		0.48
GreenGeeks	0.41	0.55	0.53		0.50
Hostinger	0.53	0.90	0.15		0.53
Flywheel	1.19	0.77	0.03		0.66
DreamHost	0.62	0.50	0.96		0.69

What am I looking at?

The table shows the baseline TTFB measurements for minimal website setups with caching not blocked.

Top performers with cache enabled

The best results apparently came from hosts with robust default optimizations and advanced caching systems:

- **GoDaddy** and **Rocket.net** led the pack with an average TTFB of just 0.06s. These hosts showed remarkable consistency across all tested regions.
- **Kinsta** followed closely at 0.09s, benefiting from their premium infrastructure designed for optimized WordPress performance.

The rest

The baseline scenario didn't pose much of a challenge for the other hosts either. No host on the list went above the 0.7s mark when it came to TTFB.

🚓 **Bonus:** If you're interested, we've done a huge study looking into various hosts' baseline performance over the span of 3+ years. The results have been fascinating. You can find them here .

Conclusions and what all this means

These results highlight the importance of understanding how your host performs under stress. While caching-enabled setups often make hosts appear fast, real-world scenarios involving dynamic pages, user interactions, or large plugin stacks can reveal performance bottlenecks.

Some hosts can swallow the load better than others and provide more consistent server performance, while others hide behind strong caching optimizations to make it appear as if they're top class.

Can I recommend any of these hosts?

Yes.

The groupings below are based on how well each host performed both with and without caching, and how much their raw server power contributed to consistent load times:

- Tier 1: Stable performers with minimal drop
- Tier 2: Managed WordPress hosts you can count on
- Tier 3: Better left alone

Tier 1: Stable performers with minimal drop 👌

These hosts showed relatively small performance drops when caching was disabled, even with a more complex plugin stack. Their consistent results make them reliable choices for sites that might not always benefit from caching. They're not necessarily the flashiest, but they're solid and dependable:

- **A2 Hosting**: Moderate performance drop and steady TTFB across all scenarios.
- **HostGator**: A surprise performer, maintaining low TTFB in both uncached scenarios.
- **Hostinger**: Delivered excellent TTFB even under stress, making it a budget-friendly powerhouse.
- **Namecheap**: A stable option with consistent results and low TTFB in uncached tests.
- **SiteGround**: Balanced performance and minimal drop-off, ideal for a variety of setups.

Tier 2: Managed WordPress hosts you can count on 💪

These premium hosting providers delivered excellent baseline performance and handled more demanding scenarios with ease. While they did see some performance declines in uncached setups when looking at things percentage-wise, the raw numbers looked much better:

- **Kinsta**: Maintains strong performance with exceptional baseline TTFB, even with caching disabled. WPShout is hosted on Kinsta, btw.
- **Rocket.net**: Top-tier results in baseline tests and robust infrastructure for dynamic needs.

Tier 3: Better left alone 🤷

These hosts struggled significantly when caching was disabled, with more significant performance drops and higher TTFB in uncached scenarios. While they might be sufficient for sites that don't present too much non-cacheable output, their results in raw server performance make them less appealing for more dynamic use cases:

- **DreamHost**: Significant performance degradation across all uncached scenarios.
- **Flywheel**: Heavy reliance on caching, with sharp performance declines in uncached setups.
- **GoDaddy**: A sharp decline in performance, suggesting that caching is critical for maintaining acceptable speeds.

As always when selecting a host, consider the type of site you're running and how often caching will realistically come into play.

> 👆 For stable, cost-effective performance, **Tier 1 hosts** are a safe bet. If you're managing a high-traffic, complex site, **Tier 2 hosts** provide the infrastructure and support needed for exceptional performance.

Let's zoom back out 🔭

My testing methodology - all about building a reliable testing setup

I've been using WebPageTest for years to measure hosting performance, and while I'd always stuck to its out-of-the-box features, I knew it had a scripting capability for more advanced scenarios.

With WebPageTest's scripting, I realized I could simulate a real-world-like interaction between a user and a WordPress website that would naturally avoid caching.

Just like every other step in this process, though, there were nuances I hadn't anticipated. It wasn't enough to simply script a set of actions; I needed to ensure that the tests were consistent across all hosts and reflected the same conditions, down to the smallest detail.

The first challenge was figuring out how to create a controlled environment. I couldn't just throw together a quick WordPress site and start testing. To make fair comparisons, every host needed the same setup: the same theme, the same plugins, and the same (minimal viable) WooCommerce configuration.

I also needed to confirm that none of the hosts were sneaking caching back into the mix. That meant diving into response headers to ensure no caching rules were applied behind the scenes. As my former manager once said, *"control is the highest form of trust."*

The test scenario: Simulating a real-world user journey

Next came designing the test itself. My goal was to recreate a typical ecommerce interaction that would bypass caching naturally, so I wouldn't have to fight the server every step of the way. After experimenting with a few ideas, I landed on a scenario that felt realistic and repeatable:

1. **Load a product page.** Start on a standard WooCommerce product page, simulating a user browsing an item.
2. **Add the product to cart.** Perform an action that's inherently uncached: adding the product to the shopping cart. Most ecommerce setups can't cache this step because it's tied to individual user sessions.
3. **View the cart page.** Navigate to it by leaving the product page and going to the cart..
4. **Measure the performance of the cart page.** Measure the load time there. The cart relies on dynamic data specific to the user, making it a natural bypass for caching.

This sequence hit the sweet spot for what I wanted to test, finally! It was realistic, represented a common user experience, and ensured that caching systems couldn't interfere without breaking functionality.

The exact script used

The real magic here was in the WebPageTest script. It let me define step-by-step instructions for the testing tool to follow, creating the exact user journey I needed.

The result was a short but powerful script. Here's the exact code:

```
logData   0
navigate   %URL%product/test/
clickAndWait name=add-to-cart
logData   1
addHeader  Cache-Control: no-cache
addHeader  Pragma: no-cache
navigate   %URL%cart/?nocache=%TEST_ID%
```

Here's what's going on:

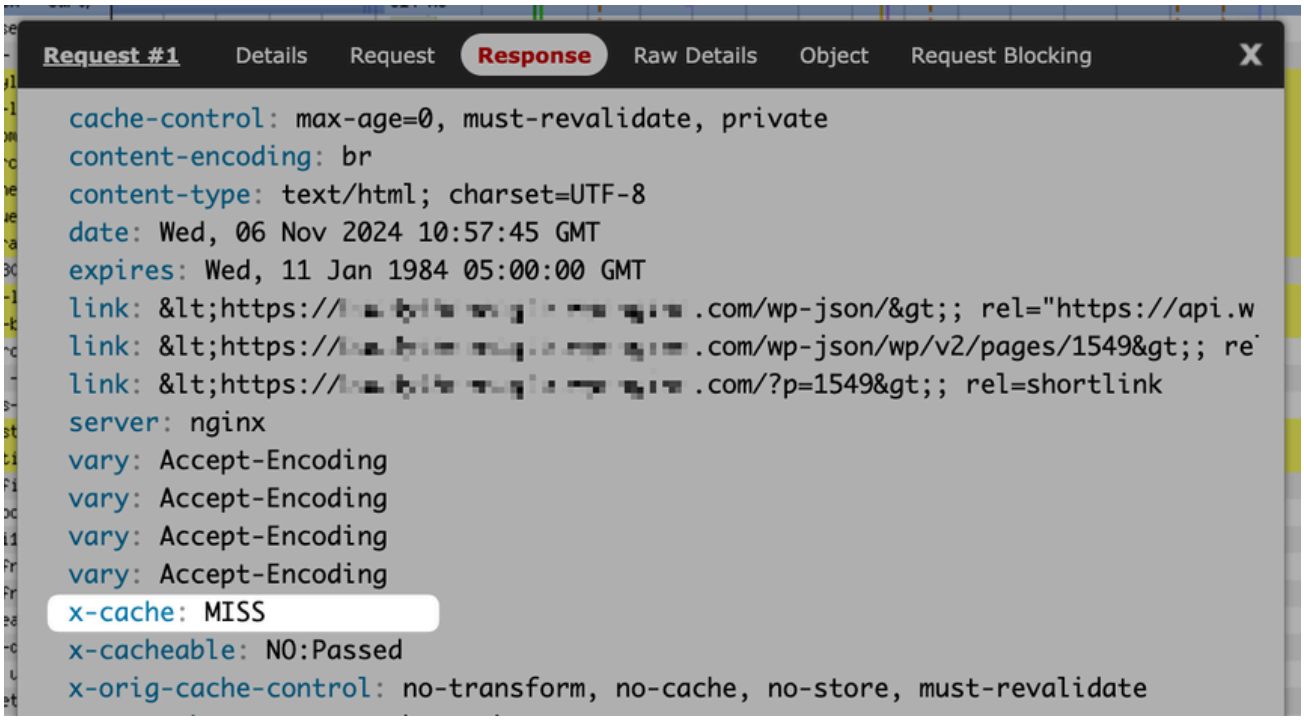
- logData 0 - tells WebPageTest not to do any measurements from this point on
- navigate %URL%product/test/ - goes to the product page; in this scenario, the product page URL has to be the same on all sites - /product/test/
- clickAndWait name=add-to-cart - this line clicks on the "add to cart" button and waits for the page to reload
- logData 1 - at this point, I can start tracking the performance
- addHeader ... - some no-cache headers for good measure
- navigate %URL%cart/?nocache=%TEST_ID% - visit the cart page and test it

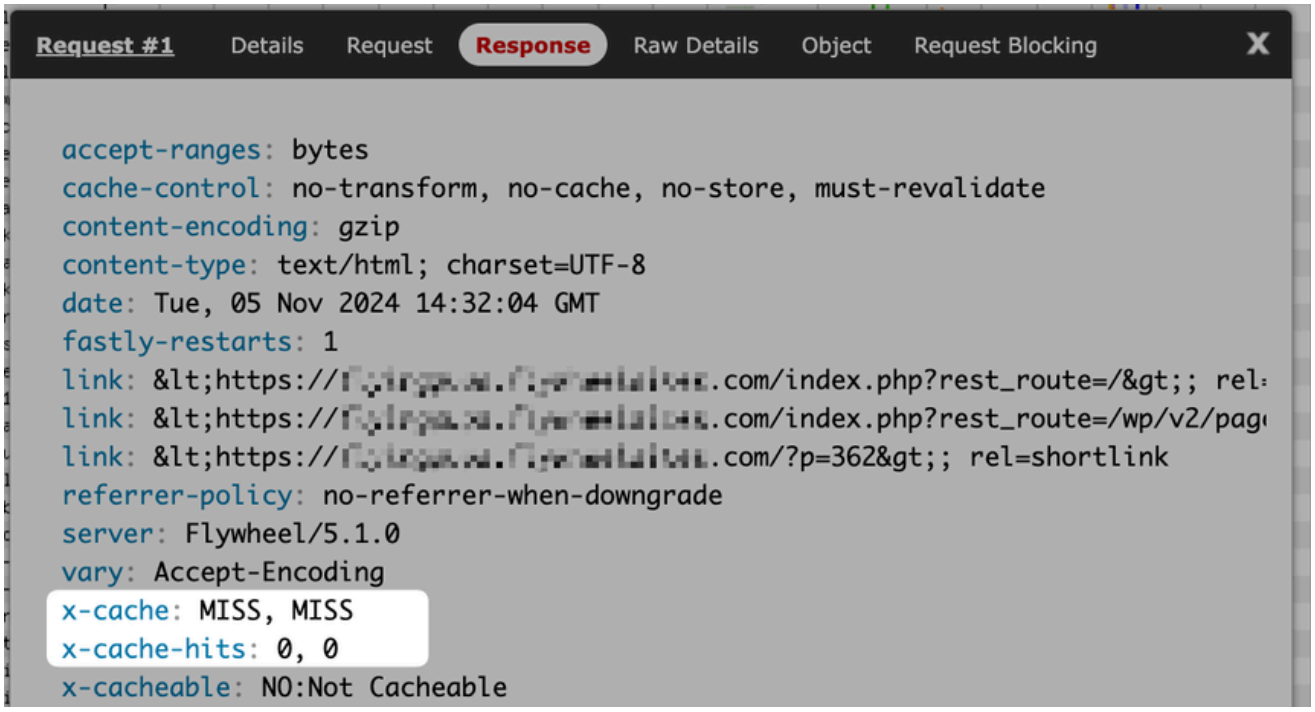
Verifying the results - did caching really stay out?

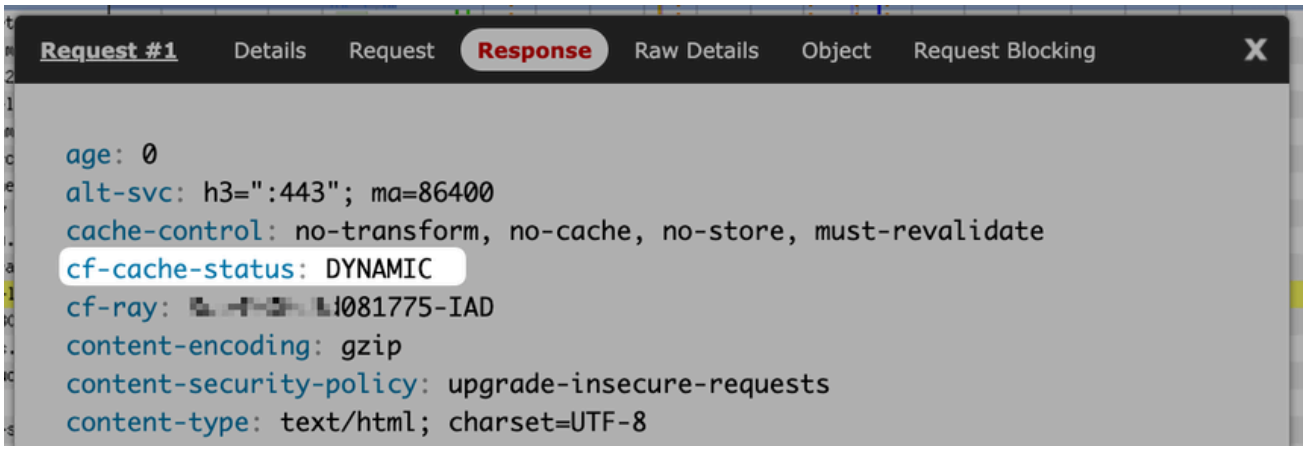
Even with this setup, I needed to double-check that the tests were truly cache-free. WebPageTest offers detailed response headers for every test, which turned out to be my best friend here.

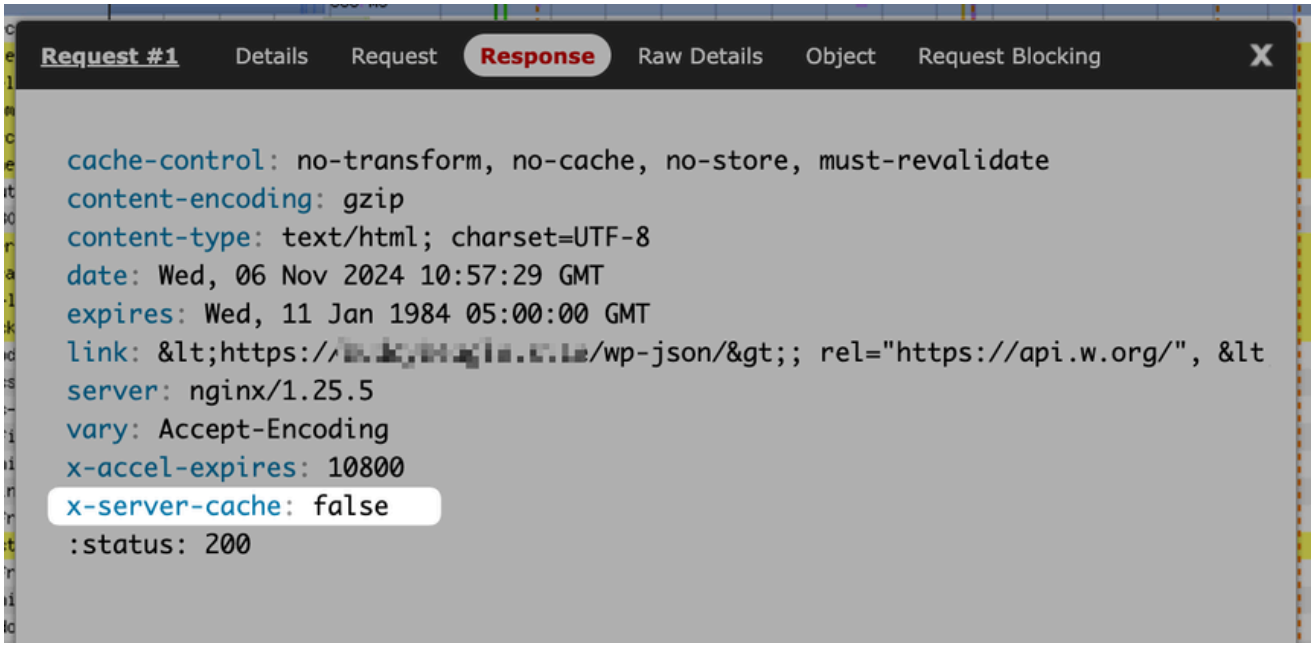
I browsed the results one by one to confirm that no cached assets were served during the test runs. For hosts with aggressive caching layers, this verification step was crucial - especially for those using Cloudflare.

Here are some example screenshots of the response headers I got:


Request #1
Details
Request
Response
Raw Details
Object
Request Blocking
cache-control: max-age=0, must-revalidate, private
content-encoding: br
content-type: text/html; charset=UTF-8
date: Wed, 06 Nov 2024 10:57:45 GMT
expires: Wed, 11 Jan 1984 05:00:00 GMT
link: <https://[redacted].com/wp-json/>; rel="https://api.w
link: <https://[redacted].com/wp-json/wp/v2/pages/1549>; re
link: <https://[redacted].com/?p=1549>; rel=shortlink
server: nginx
vary: Accept-Encoding
vary: Accept-Encoding
vary: Accept-Encoding
vary: Accept-Encoding
x-cache: MISS
x-cacheable: NO:Passed
x-orig-cache-control: no-transform, no-cache, no-store, must-revalidate


Request #1
Details
Request
Response
Raw Details
Object
Request Blocking
accept-ranges: bytes
cache-control: no-transform, no-cache, no-store, must-revalidate
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Tue, 05 Nov 2024 14:32:04 GMT
fastly-restarts: 1
link: <https://[redacted].com/index.php?rest_route=/>; rel=
link: <https://[redacted].com/index.php?rest_route=/wp/v2/pag
link: <https://[redacted].com/?p=362>; rel=shortlink
referrer-policy: no-referrer-when-downgrade
server: Flywheel/5.1.0
vary: Accept-Encoding
x-cache: MISS, MISS
x-cache-hits: 0, 0
x-cacheable: NO:Not Cacheable


Request #1
Details
Request
Response
Raw Details
Object
Request Blocking
age: 0
alt-svc: h3=":443"; ma=86400
cache-control: no-transform, no-cache, no-store, must-revalidate
cf-cache-status: DYNAMIC
cf-ray: [redacted]081775-IAD
content-encoding: gzip
content-security-policy: upgrade-insecure-requests
content-type: text/html; charset=UTF-8


Request #1
Details
Request
Response
Raw Details
Object
Request Blocking
cache-control: no-transform, no-cache, no-store, must-revalidate
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Wed, 06 Nov 2024 10:57:29 GMT
expires: Wed, 11 Jan 1984 05:00:00 GMT
link: <https://[redacted]/wp-json/>; rel="https://api.w.org/", &lt
server: nginx/1.25.5
vary: Accept-Encoding
x-accel-expires: 10800
x-server-cache: false
:status: 200

How about adding more complexity to the test sites?

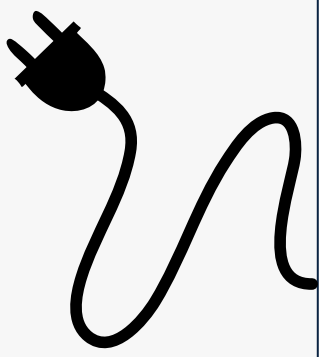
Once I nailed down the base test setup, I decided to push it further. After all, most WordPress sites don't stop at a couple of plugins or a barebones WooCommerce setup.

I wanted to see how additional plugins - think SEO tools, analytics trackers, and other common add-ons - affected performance. Would the load times get noticeably worse? How much strain would more plugins put on each host when caching wasn't there to smooth things over?

So, I repeated the process with an expanded plugin stack, adding popular tools that many site owners rely on. This extra layer of complexity provided an even clearer picture of how each host handled real-world conditions, where raw server performance matters most.

If you're curious, the plugins I installed:

- Akismet
- All In One WP Security
- CoBlocks
- Cookie Notice
- Elementor
- Jetpack
- Otter
- Secure Custom Fields
- Templates Patterns Collection (for Neve)
- UpdraftPlus
- Visualizer
- WooCommerce
- WP Statistics
- WPForms Lite
- Yoast SEO



With the testing environment locked in and results starting to take shape, I could finally start seeing the trends emerge. As you've seen above, some hosts thrived, even when juggling uncached requests and multiple plugins. Others...not so much. But the deeper I went, the more interesting the patterns became for sure.

I'd love to hear your thoughts! Have you ever tested your host without caching? Did the results surprise you? Or are you considering switching hosts based on these findings? Share your experiences, questions, or recommendations with us on social media @WPShout

Additional resources

1. We've done a huge study looking into various hosts' baseline performance over the span of 3+ years. Learn more here.
2. If you're instered in cheap hosting solutions this resource looks into the long-term cost of hosting - over the span of 5 to even 10 years.
3. Every 1-2 years, we conduct a hosting survey to find out what our readers think of their WordPress hosts. See more info here.
4. We track the uptime of each of our test setups continuously. See our live status page here.

About the author

Karol Krol is a writer, content strategist, and WordPress figure-outer with over 20 years of experience rooted in website building and web technologies.

With his expertise underpinned by a master's degree in computer science, he authored "WordPress Complete" - the ultimate WordPress handbook for newbies.

His work has been published across numerous industry websites.

He leads the editorial team at **WPShout**.