netlify

Lightning-fast Build Performance

Netlify Build Performance for the Enterprise

Building Better with Netlify's Programmatic Power

The way you build things affects the work you do.

When you use the right set of developer tools to build, maintain, and operate your suite of websites, you're helping to craft a solid foundation on which your business can grow and succeed. When enterprises use the wrong set of build tools, those tools end up weighing down the developer team tasked with using them, along with the business itself.

Netlify's build and performance tools give enterprises what they need to create rich, interactive websites and apps that scale seamlessly while offering enterprise-grade security.

Leading companies like Nike, Citrix, Peloton, Verizon, and Mailchimp all rely on Netlify's suite of developer tools to build, launch, and scale their websites so they can deliver an ideal experience for their customers.

If you trace a customer experience back to its origin point, you'll often end up in a Git repo of some sort. Code is the connective tissue that both defines and unites customer experiences across a company's portfolio of various websites.

When a developer can focus on what they're best at, whether that's designing a beautiful front-end environment or maintaining a company API, they're able to perform at a high level and produce better work. But, if they're using the wrong tools, developers are consistently pulled away from their mission-critical work to attend to tasks that are both menial and critical.

Developers are often stuck piecing together solutions from separate vendors to develop, build, test, deploy, host, scale, and distribute sites. With each vendor comes a different UX and a new integration for that developer to manage. This leaves teams of developers constantly context switching between various platforms just to keep their websites up and running.

Netlify believes in simplifying and streamlining that entire build workflow for developers so they can do their work faster, and build faster, more performant sites in a fraction of the time.

In this e-book, we'll show you how you can do just that.

The Netlify Difference

Netlify gives companies the ability to develop exceptionally fast, responsive sites in the blink of an eye using powerful developer tools, a globally distributed CDN, and high capacity build environments.

At the core of Netlify is the Jamstack, a modern web architecture that decouples the frontend from the backend, shortening the distance between your user and the content they desire. The Jamstack also enables companies to migrate away from legacy architecture and move towards more modern, flexible architecture at their own pace while delivering a consistent experience to users and maintaining mission critical operations.

The Jamstack also gives developers more freedom to choose the tools they prefer to use in each part of the stack, from front end to back end. Developers can pick the git-provider of their choice, the Headless CMS of their choice, and the third-party APIs they know and love and integrate them seamlessly.

Netlify itself operates as a one-stop shop where developers can tie in and manage the tools they love within the context of their build environment, without bouncing around between different cumbersome services. But, Netlify and Jamstack architecture don't just save developers time, they save users time, too.

In a traditional legacy architecture model, the content a user requests is handed off between several entities before it's delivered. This leaves the user waiting on their content as it runs the relay race from the browser, across a CDN, a load balancer, a webserver, and a database before making the trip back to the browser. Jamstack architecture leverages pre-built, static sites that are ready the instant a user requests them.

Developers can use static site generator tools like Hugo, Jekyll, or Gatsby to produce static sites before uploading them to Netlify with a simple drag and drop, or git command. Then, Netlify deploys the uploaded site across it's reliable, globally available CDN, ensuring that any user around the world has a consistent experience.

But before we dive into how a deployed site performs, let's explore how Netlify gives developers a proverbial superpower when it comes to building those sites.

The Netlify Build Effect: Before and After

What does your team's build workflow look like? What does your build itself look like?

Before switching over to Netlify, one customer found themselves in a proverbial build logjam that's all too common if you're using the wrong tools for your build process.

This customer had to ship several sites in a small window of time. But, they didn't have the capacity or concurrency on their local servers to build many sites at once. So, they were left waiting hours, watching one site slowly build minute by minute as it made way for one of the dozen sites in the queue behind it.

Now that they're using Netlify, this enterprise customer can deploy sites much, much faster using Netlify's High Performance Builds.

This type of experience is indicative of the night and day difference between a build architecture that's spread out across different legacy vendors, and one that's centralized on the Netlify platform.

When Netlify customer TunnelBear, moved from a server-side architecture dependent on a complex web of NGINX rules, to Jamstack architecture coupled with Netlify's HP Build environment, they were able to deploy sites ten times faster than before.

> **"It used to take us ten minutes to build and deploy. Now with Netlify, it takes us one minute...the moment of realization came really quick. I started and then realized this was going to be a lot easier than I thought."**
>
> **— Jared Krause,** Tech Lead for the web development team at TunnelBear.

Read the TunnelBear case study

Netlify's HP Build environment allows teams to run 12 concurrent builds, with 1000 uptime minutes and a 99.9% SLA, giving enterprises the scalability and bandwidth they need to build with speed.

Let's dive into just how that build process works for teams using Netlify.

99.9% SLA

12 CONCURRENT BUILDS

1000 UPTIME MINUTES

Finding the Right Team Workflow on Netlify

In a modern enterprise, there's constant interaction across different teams from marketing to engineering.

Let's say a content writer who sits in the marketing organization is working on a new post to be published on the company blog. Before they push the post live, they want to make sure everythings in order and see how their post will look outside of the internal CMS and in a production environment.

For companies using Netlify, that workflow is incredibly easy. The content writer can use Netlify's Deploy Preview feature to assess their work before it goes live, and even send their colleagues a link to preview their post.

Right when that content writer hits publish, Netlify registers that change in the CMS and pushes the new post live across its global CDN so any customer in any corner of the world can read it.

Now, let's imagine that the content writer isn't using Netlify. In this context, they might have to ping a developer to help them format their post correctly, or double check their work. They might be reliant on the developer to publish the post for them because their staging environment is divorced from a production environment.

These types of architectural inefficiencies add up and distract developers from their most important work.

Netlify is built to accommodate both editorial workflows and developer workflows so teams can work independently and effectively, whether they prefer to publish content through git-based commands or in a Headless CMS.

Using Build Hooks and the Netlify API to Make Programmatic Website Changes

When code or content changes, Netlify listens for those changes and responds based on commands you dictate in your Netlify build settings. Using Netlify's Build Hooks, you can trigger a specific content change on a particular site, giving you both granular control and programmatic command of your site.

Let's say you have an update you want to deploy to your company's features page once a new product launches. You don't have to leave the command line to push that change live. You can simply create a Build Hook called "Features", send a POST request to that webhook from your command line, and Netlify will deploy that new update to the branch you've selected.

Netlify knows that context switching can rob developers of valuable time and creative flow. That's why we made it easy for developers to stay within the context of code. Any action you can take in Netlify's visual UI, you can accomplish using Netlify's API.

This programmability pays off massively when you're shipping tens of sites at a time. Instead of spinning up each site manually and individually dictating its settings through Netlify's UI, you can build logic to ensure that every site is built uniformly based on your own custom logic and Netify build settings using Netlify's API. Think of it as moving from a hand-crafted, time intensive website workflow to a well-built programmatic assembly line. That's the type power Netlify gives developers.

The Horsepower of Netlify's HP Build

Remember that logjam referenced earlier? The one where a developer had to wait for hours as his local servers struggled to build his websites stuck in a build queue? Using Netlify HP build, you don't have to wait on your websites.

> **Netlify not only gives you the throughput you need to deploy sites fast, it uses baked-in intelligence to optimize your builds as they happen.**

Builds can get bogged down by duplicating the work of a previous deploy. This wastes valuable time and computational resources. It's almost as if every time you put down a book you were reading, you had to start from the beginning instead of picking up where you left off. That not only burns time, it taxes your energy. The same principle applies for builds, only with much higher stakes than your weekend reading.

Instead of starting from scratch with every build, Netlify caches content from your builds and intelligently finds content that's already been cached from previous builds. Netlify does this in the pre-build phase, before anything has been deployed.

Then, Netlify deploys only the new content to your site instead of blindly uploading every single file. Netlify only pushes out the new files you've uploaded. Eliminating this repetitive work means you can deploy more sites faster while saving build minutes.

Extend Netlify's Capabilities Using APIs and CI/CD Commands

Your build is yours to tailor as you see fit.

Maybe that means calling a Slack plugin `onSuccess` so your Slack engineering channel can celebrate a new successfully deployed site. Maybe that means calling a pager plugin if you encounter any hiccups in your build. You can easily define who in your team gets notified based on various states of the build cycle, and where they're notified.

> **Netlify makes it simple to implement and call third-party plugins and APIs at different parts of the build process by calling various event stages and dictating logic for a task to be accomplished once your build reaches that event stage.**

This allows developers to perform tasks as their builds are happening as opposed to running a build command, waiting on the site to be built, and then going through their logs to do things like sending Google or Bing a sitemap of their newly built website.

With Netlify, developers can cross off to dos as their builds are running, such as calling for their sitemap to be submitted to search engines 'onSuccess'.

But, how do developers make sure that their commands have been executed successfully?

Netlify gives developers granular information in their build logs, showing what dependencies were extracted from a cache, what actions Netlify took in deploying your site to a CDN, what build plugins were called, and what they accomplished.

These programmatic processes let developers get more done in less time, all without sacrificing one iota of flexibility or control. At scale, these processes let development teams move quickly while maintaining a high-degree of quality.

The Benefits of Building with Netlify

There are so many different ways to build your sites. We happen to think Netlify is the best platform you can pick to build your site the way you see fit.

Netlify’s High Performance infrastructure automates your building and hosting of sites so your teams focus on tailoring your site and tweaking the bespoke details that define a customer’s experience.

With extraordinary build concurrency, high processing power, and unlimited number of build plugins just a few clicks away, and asynchronous deploys, Netlify gives developers the tools they need to perform at their best, all using simple code-powered commands.

The distance between you and a new, incredible build process is just a git push away.

Contact Us →

netlify