netlify

WHITE PAPER

Build exceptional web experiences: An architectural guide to the modern web

CONTENTS

Introducing the Jamstack

Today's consumer demands an exceptional web experience. Whether it's an e-commerce store, a web app, a marketing site, or even a personal blog, the standards are high. It needs to be performant, consistent with the organization's brand, mobile-friendly, and – above all – engaging. But organizations building sites and apps with legacy technology, like monolithic CMSs and DXPs, often struggle to meet these demands. That's why today's web development teams are moving to a modern way of building for the web, known as Jamstack.

Jamstack is an architectural approach for the web that decouples the web experience layer from data and business logic, easing the pain points organizations experience with legacy technology. Jamstack architecture improves a site's:

PERFORMANCE

Page loading speeds have a huge impact on user experience and conversion. Jamstack sites remove the need to generate page views on a server at request time by instead generating pages ahead of time during a build. With all the pages already available on a CDN close to the user and ready to serve, very high performance is possible without complex infrastructure.

TIME TO MARKET

Because web monoliths such as WordPress, Drupal, and Sitecore tightly couple the frontend and backend, they come with significant amounts of backend code that's difficult to read, edit, and debug when something goes wrong. The Jamstack architecture makes it faster for developers to make changes that don't accidentally break other components on the site. As a result, web teams can move more quickly because they have more flexibility to create and iterate reusable components. No more scheduled weekly deploys; content teams can publish when they need to, without worrying about the site breaking.

SECURITY

The Jamstack serves pages and assets as pre-generated files, reducing the many attack vectors present in legacy tech. Meanwhile, dynamic tools and services can be provided by vendors with teams dedicated to securing their specific systems and providing high levels of service.

SCALE

Legacy monolithic web architectures deal with heavy traffic loads by requiring complex logic to cache popular views and resources, which takes valuable time away from developers. The Jamstack provides this by default. When sites can be served entirely from a CDN, there is no complex logic or workflow to determine which assets can be cached and when. With Jamstack sites, everything can be cached in a content delivery network with simpler deployments, built-in redundancy, and massive load capacity.

MAINTAINABILITY

When hosting complexity is reduced, so are maintenance tasks. A pre-generated site, being served directly from a simple host or directly from a CDN does not need a team of experts to "keep the lights on." The work was done during the build, so now the generated site is stable and can be hosted with less time spent patching, updating, and maintaining infrastructure.

DEVELOPER EXPERIENCE

Jamstack sites don't depend on proprietary technologies or exotic and little-known frameworks. Instead, they build on widely available tools and conventions. As a result, it's not hard to find and retain enthusiastic and talented developers who have the right skills to build with the Jamstack. Efficiency and effectiveness can prosper.

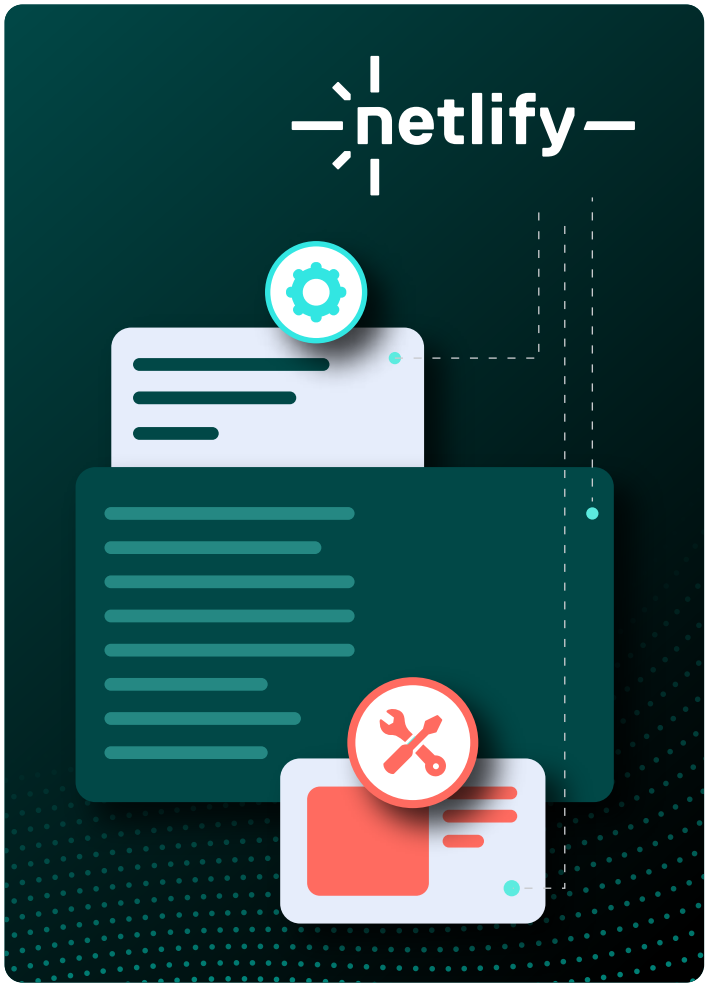
How to build a modern web stack

In addition to offering the aforementioned benefits, this approach offers teams the ability to select best-of-breed tools and only pay for the functionalities they need. This has huge cost-saving implications. In fact, a quarter of IT decision-makers said they spend over half of their IT budget on version upgrades for their legacy systems. On average, companies spend almost two-fifths of their budget on these upgrades.

The challenge to this approach is that, unlike the one-stop-shop that legacy technology offers, it requires managing different services. Every element of a Jamstack site needs to be selected and pieced together by the web team, or agency.

While Jamstack sites are infinitely customizable and easy to change elements over time, the main ingredients of a Jamstack site are:

- **The collaboration layer:** For engineers, this is the code that lives in a Git provider, and for non-technical contributors, this is often a project management tool like Trello or Jira.
- **The framework layer:** Jamstack sites require a static site generator that will pre-build as much of the site as possible into static assets, like 11ty or Next.jsAn abstracted source of content, often a headless CMS or e-commerce platform.
- **The integration layer:** This layer connects external APIs to the site, which are customizable to your team's needs. It might include a serverless database, a headless CMS, or an e-commerce platform.
- **Build and release management layer:** This is the platform and infrastructure that will help you build and host the site.

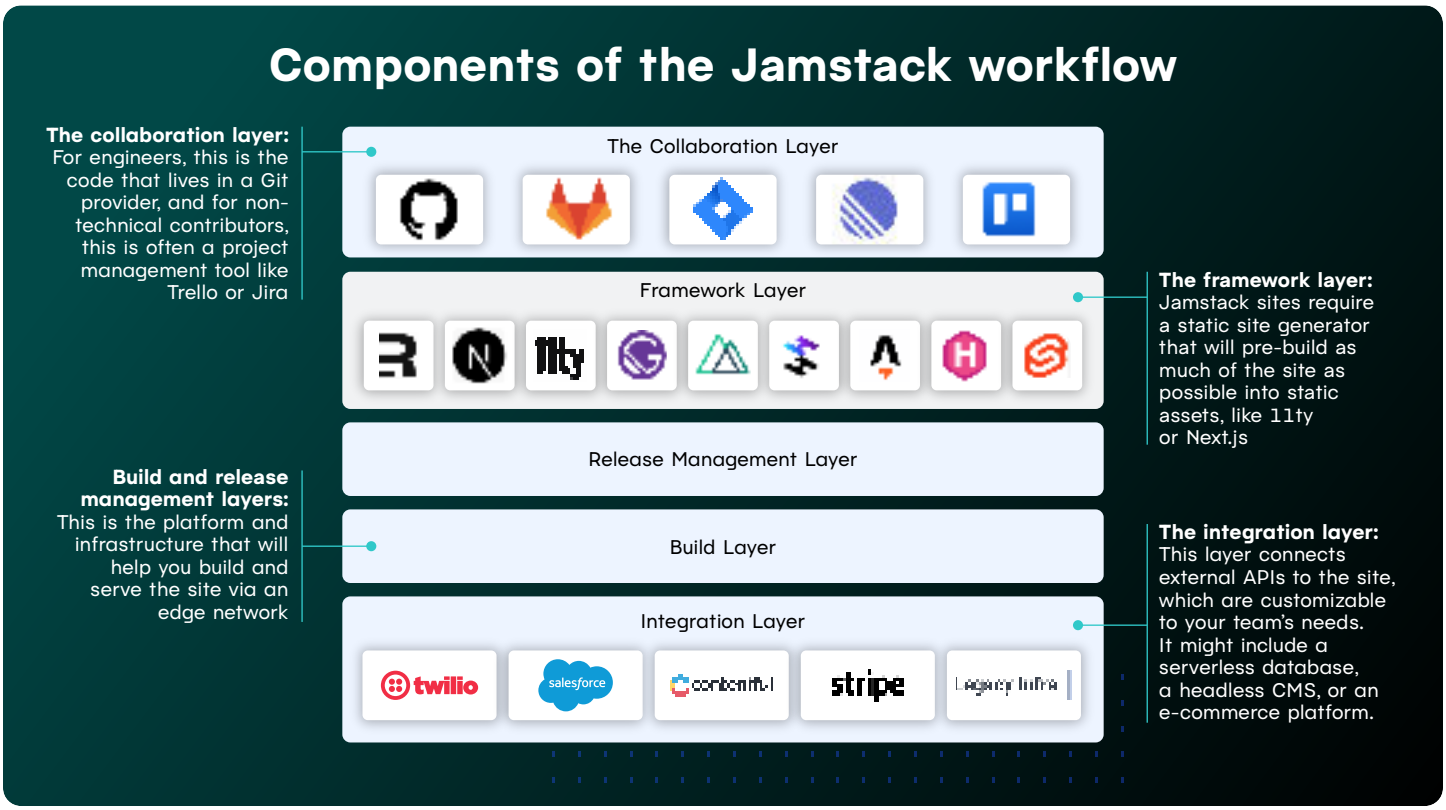


Without a streamlined workflow to manage these elements, this can create unnecessary complexity and sometimes, redundancies.

Bring the Jamstack ecosystem together with Netlify

Netlify is an all-in-one platform built to solve these challenges by helping developers connect, build, and run frontend web projects – from local development to full production.

Netlify provides the backbone to connect all the services your team needs, from integrating API-based tools, to building and running code, to collaborating with colleagues.

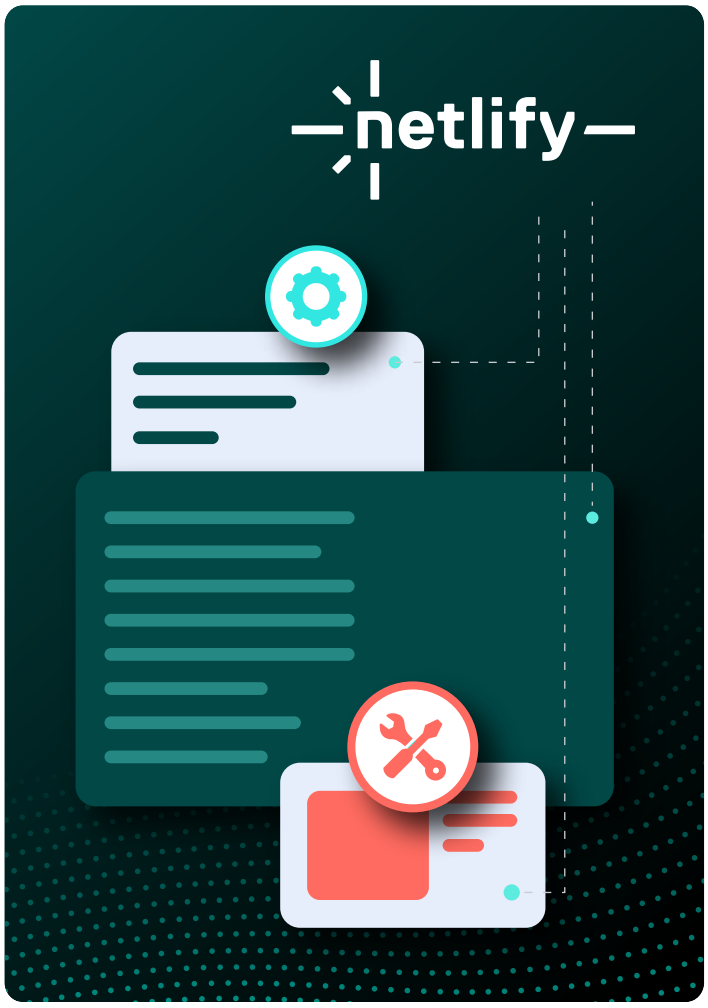


- Integration layer allows for one-click connections to the APIs your team needs to run the app.
- Collaboration layer helps all teams across the organization work together more efficiently.
- Build layer allows anyone to create build plug-ins to their favorite tools for an open, customizable build process.
- Runtime layer provides full serverless functions that transform content and responses using familiar web standard APIs.

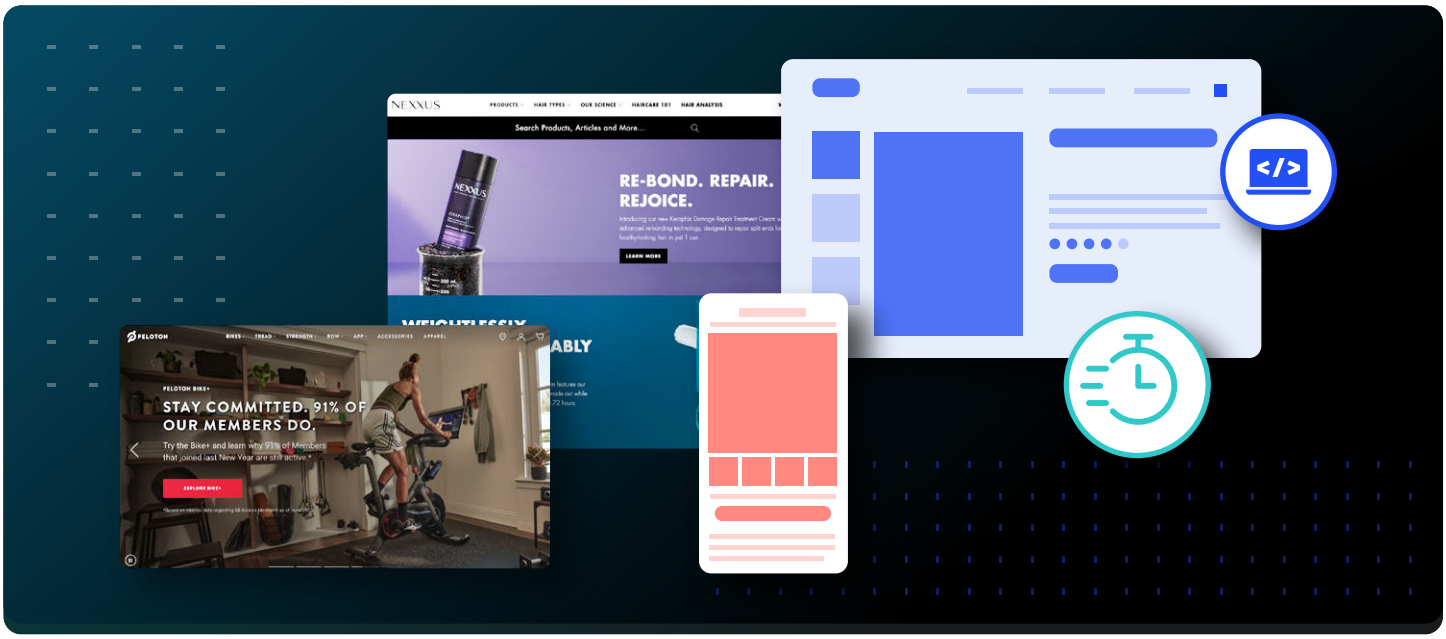
Read on to learn how Netlify helps enterprise teams connect, build, and run web applications at scale.

Connect all your tools, services, and code into one workflow.

- **Expand your build capabilities with one-click Build Plugins.** Choose from a directory of plugins or create your own. Use them to bring automated testing into your deploy process, enforce performance and accessibility standards, or kick off services such as search indexing after a deploy completes.
- **Build interactive and connected apps faster with Netlify Graph.** Start using APIs without spending time writing undifferentiated glue code, learning every API's unique structure, or composing endless amounts of endpoints to get data from various APIs into your app. Instead, you can abstract popular third-party services behind a secure, convenient, and easy to explore GraphQL endpoint with the click of a button.
- **Integrate just about any service with an API.** Build on CMS changes or other events with Webhooks, which are URLs you can use to trigger new builds and deploys. Pull in data from APIs during request time or build time, ensuring the content is always up to date.



Build sites, stores, and apps in record time.



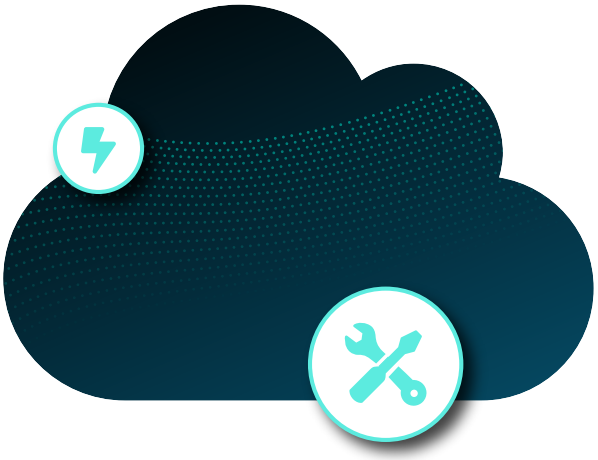
- **Configure continuous deployment straight from the command line with Netlify CLI.** Spin up a project in seconds, configure your build, test serverless and edge functions, run projects on a local dev server, and deploy globally – all from your command line.
- **Create a tight feedback loop with collaborative Deploy Previews.** Share progress early and often with preview links created automatically for every deploy. With no coding or setup, every preview enables reviewers to submit feedback complete with screenshots, videos, and annotations. Every annotation and comment automatically syncs to your favorite productivity tool.
- **Enjoy peace of mind with worry-free deployments.** With atomic deploys, phased rollouts, one-click rollbacks, and a totally customizable build environment, Netlify enables effortless continuous deployment with all the infrastructure and automation required.
- **Tackle even the most demanding web projects.** Additional build features include monorepo support, build prioritization, team build permissions, dependency caches, locked deploys, and support for enterprise Git workflows.

Run it all from a secure and performant global edge network.

- **Build, test, and deploy across one platform, from preview to production.**
  Ensure consistency and reduce the risk of error by keeping all your configuration, code, assets, edge logic, and serverless functions together in a single repository.

- **Enable full control, power, and performance with serverless runtime.**
  Leverage a fully automated web architecture with no servers to manage or clusters to configure. Key features include:

  + **Prerendered pages** – Pre-generate pages at build time so that even complex pages that pull content from multiple APIs can be distributed globally as fast, static assets.
  + **Netlify Edge Functions** – Run code at the edge, close to your users, to create custom authentication, personalize ads, localize content, intercept, and transform requests, perform split tests, and more.
  + **Netlify Functions** – Deploy on-demand, server-side code that works as API endpoints to process data, connect APIs, and build dynamic experiences.



  + **On-demand Builders** – Shorten build times by generating routes for your site on-demand (when a user visits them for the first time). Subsequent visits are served cached responses from the edge.
  + **Background Functions** – Perform longer tasks in the background, such as batch processing, web scraping, interfacing with slower APIs, or headless browser requests, with timing out.
  + **Scheduled Functions** – Ensure certain tasks — like generating reports or redeploying your site — happen on a consistent, repeating interval.

Why Netlify?

The Jamstack movement is rapidly gaining steam, and Netlify was built to enable that evolution. By uniting an extensive ecosystem of technologies, services, and APIs into one workflow, Netlify unlocks new levels of team productivity, while saving time and money.

We've onboarded more than 2.5 million developers and businesses and helped them deliver the best digital experiences and achieve valuable business benefits.

Unburden backend and ops teams

How?

- **LESS INFRASTRUCTURE –** Absorb huge traffic bursts without hammering your backend services. Never manage dev environments or CI/CD pipelines.
- **LESS DISTRACTION** – Empower frontend teams to self-service and deploy and manage projects, without infrastructure sprawl and with tight controls.
- **MORE CONFIDENCE** – Fully deploy automation, integrated testing, gradual rollouts, and instant rollbacks.

151% ROI

A Forrester Consulting analysis found that one Netlify customer achieved 151% ROI, with 9-month payback.

"With Netlify, now our teams deploy changes 10-15× a day."

– **HARRISON HARNISCH**, PRINCIPAL SOFTWARE ENGINEER, TWILIO

Empower frontend teams
How?

- **PICK THE RIGHT TECHNOLOGY** – Because the backend no longer dictates the frontend, you can select the technologies that are the best match for your team and each project.
- **BUILD, TEST, AND DEPLOY QUICKLY** – Just push to Git and it's deployed. Developers stay in the flow building, testing, and releasing features.
- **CREATE AMAZING EXPERIENCES** – Have all the tools to perfect app performance and use rich functionality from backend services.

Accelerate your entire organization
How?

- **SHIP FEATURES FASTER:** Developers can work faster with instant workflows, deploy previews, and easy collaboration.
- **GET THE RIGHT TOOLS FOR EACH TEAM:** Allow developers, marketing teams, data teams, and product teams to use best-in-class tools.
- **RECRUIT AND RETAIN TOP TALENT:** Recruit developers with the tools they want to use and the technologies they want to build careers around.

8x faster performance

One Netlify customer reported their site performance was eight times faster than before.

90x faster deploys

Netlify helped one customer accelerate site deploys by 90 times.

Ready to try Netlify? Get started Get in touch with