

How the **JAMstack** enables  
enterprises to deliver web  
projects **10x faster**

SECTION 01

---

## Welcome to the JAMstack era

page 3

---

SECTION 02

---

## Build product not infrastructure: How the JAMstack is being used today

page 5

---

SECTION 03

---

## A deep dive into the JAMstack: Optimizing your toolchain for competitive advantage

page 7

---

SECTION 04

---

## With the JAMstack, the CTO is empowered to: Manage more teams, reduce complexity, and increase productivity

page 10

---

SECTION 05

---

## Collaborate on Content and Design At Scale: The JAMstack Simplifies New Content Creation for the CMO

page 13

---

SECTION 06

---

## Infrastructure-as-Code: The CIO's dream becomes reality

page 15

---

# Welcome to the JAMstack era

Over the past five to ten years, there has been a wholesale reorientation occurring for the enterprise. Digital has been the enabler for much of this action, but the real driver has been a change in business practices that puts the customer at the center of the business relationship. **In this new paradigm, being able to connect and engage with customers instantly becomes paramount, as does the need for fast feedback loops that allow product feature development and iteration.**

IT systems have been going through a modernization process to meet these requirements, resulting in new architectural design patterns and infrastructure models including use of multi-cloud providers and more composable, component-driven software development. On an organizational front, enterprises have employed more developers and are now competing for top talent, while also needing to reach new productivity gains rather than getting swamped in the complexity of too many autonomous teams.

This modernization process has also allowed businesses to connect to different systems through APIs, giving rise to an API economy in which external capabilities can be plugged in to a company's IT infrastructure. APIs also allow a business to create new mobile and digital products faster by reusing these components each time.

Customer expectations for product and feature development have also grown. **Businesses need to have responsive websites that load fast and offer instant functionality in the browser.** This has given rise to single page applications and edge networks that allow caching of content close to customers.

The JAMstack (a modern web development architecture based on client-side JavaScript, reusable APIs, and prebuilt Markup) leverages these modernized IT systems and increased developer workforces to create always-on, highly performant web products.

## Cornerstone OnDemand Delivers Web Projects 30% Faster with Netlify and the JAMstack

Cornerstone OnDemand, a talent management company whose software and services are used by over 42 million people in 192 countries, relies on its website with 12 international versions to attract potential buyers and convert them into sales leads. Determined to give their website visitors a fast and pleasant digital experience reflective of the quality of its own software products, the company migrated from an outdated, monolithic CMS architecture to the JAMstack, powered by Netlify, Gatsby, and Sanity.io.

As a result, Cornerstone's page loads are 25% faster and time to market with new web pages and content has improved 30%. As a bonus, the team has also greatly improved its ability to retain and attract new talent. [Read more about it.](#)



Netlify provides the missing workflow for the JAMstack. Using a Dev/Build/Edge platform approach, Netlify harnesses the power of the JAMstack to create a simplified and streamlined continuous delivery workflow so that there is always a single-point-of-truth for web products, including those in development and review. Content and code is drawn from git repositories during the build process and pre-rendered web products are then deployed to edge networks so that they are instantly accessible to customers as needed.

The JAMstack gives enterprises the superpowers they need to compete effectively and to meet customer demands in a digital era.

While Netlify provides the edge infrastructure you need, it is the JAMstack that gives your CTOs, CIOs, CMOs, and developers the superpowers to become force-10 multipliers driving your enterprise's customer relationships and revenue opportunities.

## The Top 3 JAMstack Superpowers for the Enterprise



### **Talent attraction:**

Being built on a modern stack, enterprises are better able to recruit and retain the best developer talent. Dreaded platforms like Drupal and WordPress are avoided and the complexity for frontend developers needing to manage DevOps concerns is eliminated so builders can concentrate on product with greater satisfaction.



### **Simplification:**

The JAMstack represents a turning point for enterprises that have gone through an IT modernization process, allowing companies to sunset antiquated software design patterns that were the vestiges of 20+ year old architectures and infrastructures. Adoption of the JAMstack model can be done without adding new steep learning curves to current developer best practices.



### **Velocity:**

Enterprises can outpace competitors. Digital innovation can come to market faster, and new ideas can be tested in production without impacting a monolithic code base. Changes are easily rolled back if requiring further iteration. New web products can be spun up quickly and successful features from one deployment can be instantly applied across a product suite.

# Build product not infrastructure: How the JAMstack is being used today

To compete and stay relevant, today's enterprises need developers with superpowers in scalability, security, and speed.

## The foundation to enabling 10X

Working with an infrastructure provider who can manage performance at the edge and can autoscale server workload is the foundation of enabling your developer team to deliver web projects 10x faster.

In other words, by working with a trusted provider, *you can focus on building product not infrastructure.*

**This gives rise to the JAMstack.**

In a JAMstack approach, developer teams are empowered to work to achieve business goals while the CTO creates a 'mission control' team to authorize common tools and to oversee the use of standard workflows that streamline continuous improvement and delivery across all digital products<sup>1</sup>. The JAMstack helps you make your products available to a global audience, and manage sites as products at the edge (see inset box: what is a product) to ensure around-the-clock, instant availability.

### Is this you?

- Your business is hiring more developers to help manage and run products for your enterprise's brands and services.
- Your business IT systems are now predominantly cloud-based.
- Your enterprise is needing to more quickly produce and maintain products that showcase new customer offerings.
- Your developer teams each do things slightly differently, and it is getting harder to know everything that is going on and to make changes to all products at once.
- You need to be able to track the revenue being generated from each product in real time.

## Netlify as your JAMstack infrastructure provider

Netlify is a new type of edge service infrastructure provider that replaces and enhances the role a content delivery network would usually play. Our Application Delivery Network is a continuous delivery build process and edge service delivery network that pulls content and data assets from your cloud services, deploys a build process to prerender web pages, and then pushes them to edge service nodes so that your web products are instantly accessible to your customers, wherever and whenever they want. Netlify replaces production servers, staging servers, deployment systems, and CDN caching.

Utilizing the JAMstack approach reduces complexity and cultivates a sense of dynamism and ownership amongst the entire workforce:

- Designers and writers can concentrate on creating engaging content.
- Developers can focus on building website products without managing servers or monolithic code bases.
- Architects and security managers can ensure streamlined workflows and focus on fewer security risk overheads.
- DevOps can focus on application monitoring rather than provisioning new servers and can play a greater business role in helping manage revenue and customer demand metrics.
- CTOs can focus on motivating and encouraging teams and on due diligence for a core set of tools (like identity management) that are used uniformly to ensure corporate IT standardization.

## What is a product?

Today, product management rather than project management thinking helps enterprises maintain their digital assets in ways that build ongoing conversations and relationships with customers. Viewing each of your web properties as a product brings in a different mindset to viewing them as projects.

---

<sup>1</sup><https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/the-platform-play-how-to-operate-like-a-tech-company>

## Citrix reduces costs by 65% using the JAMstack and Netlify

Citrix's docs.citrix.com is home to business critical product documentation for 15 products and all related components. Comprising more than 150,000 files in eight languages, the site serves more than 3.5 million users annually.

On Citrix's legacy CMS system, users were frustrated by slow response times, no real PDF support, and subpar search. The content contributors faced broken localization workflows, publishing challenges and diminished efficiency. And the platform team was challenged by a huge ongoing financial investment, no useful audit trail and frequent platform changes.

Citrix executed a massive migration from a legacy web management system to the JAMstack and Netlify with only one admin in mere months—ahead of schedule—resulting in a 65% cost savings and a significantly better user experience, measured in faster page loads, faster publishing times and improved customer satisfaction. [Read more about it.](#)



# A deep dive into the JAMstack: Optimizing your toolchain for competitive advantage

The JAMstack is a modern web development architecture based on client-side JavaScript, reusable APIs, and prebuilt Markup. It represents an evolution of best practices in building and delivering always-on, highly performant, instantly accessible digital products to a globally distributed customer base.

As a stack of technologies, the JAMstack is supported by a maturing ecosystem of service providers that allow businesses to manage sites and web applications as **products** that draw in traffic, create digital experiences, and convert visitors into customers. There are three components, and each component has its best-in-breed players:

- **JavaScript:** Any dynamic programming during the request/response cycle is handled by JavaScript, running entirely on the front-end, that is, your website or application.
- **APIs:** All backend processes are routed through APIs: either custom built ones to access your server-side components such as your databases, or to third parties to deliver specific services such as identity controls or form data collection.
- **Markup:** Markup is used to create prebuilt templates which are used with a static site generator or build tool to create content sites and web applications at deployment. This allows more instantaneous loading when a customer visits your webpages and applications, which in turn means it is rated more highly for search engines because of the quick response times, as well as keeping customers engaged with your product rather than jumping off to a faster site.

A JAMstack approach leverages the advantages of using the same toolchain across all projects so that repetitive work is diminished and integration of individual projects into a cohesive architecture is streamlined.

## What is the difference between a CDN and an ADN?

In order to bring many of the superpower benefits to the JAMstack, Netlify has created an Application Delivery Network, Netlify Edge, that has the capabilities necessary to deliver globally distributed web applications and products, going beyond a traditional Content Delivery Network. This infrastructure, trusted by Citrix, Verizon, Peloton, Atlassian, Samsung, and other leading enterprises, works with multi-cloud providers. It evolves current thinking of edge services and delivery technologies to provide the following additional functionality:

**Support for static sites:** Like a CDN, an ADN enables the management of static sites of prerendered content.

**Works without an origin:** In an ADN, pages are prerendered and site on edge servers to facilitate instant accessibility, without needing to be built from a database as the customer opens their browser. Because of instant cache validation, an ADN can eliminate the risk of delivering stale or superseded content and functionalities.

**Can prerender and run builds:** Not only is all content prerendered for a static site, but builds can be deployed continuously so that as new features or content is updated in git, new versions of the product are pushed live to the edge.

**Git-integrated CI/CD:** This build capability is integrated with the ADN so that as new features are added the build workflow is streamlined. This enables automatic deploys and rollbacks, and facilitates Infrastructure-as-Code models to creating and delivering web-hosted enterprise applications.

**Application Performance Monitoring:** Because of these functionalities, an ADN takes on the tasks of performance analytics and actions. APM becomes a service offered by the cloud management provider.

A JAMstack toolchain can consist of:

Tool	How used in the JAMstack	Best in breed suppliers	Current usage in business
<b>Git</b>	Stores all content: copy, images, design elements. Stores build processes (infrastructure as code)	<b>GitHub</b> <b>GitLab</b> <b>Bitbucket</b>	In use amongst enterprises
<b>JavaScript frameworks</b>	Enables developers to create web products that draw in content from Git and link to APIs	<b>Angular</b> <b>Vue</b> <b>React</b>	Used on a project-by-project basis
<b>Static site generators</b>	Allows developers to create pre-rendered web pages once templates and products have been designed and developed	<b>Jekyll</b> <b>Hugo</b> <b>Gatsby</b> <b>Middleman</b>	Project-by-project basis
<b>Headless CMS</b>	Provides a more user-friendly interface that makes it easier for editors, writers, and creatives to upload and share their content to Git for use in web products	<b>Contentful</b> <b>Forestry</b> <b>Sanity</b> <b>DatoCMS</b>	Limited use
<b>Enterprise grade API services</b>	Allows developers to include additional, non-core business functionality into web products such as site and product search, identity management, and additional security management	<b>Algolia</b> <b>Auth0</b> <b>Okta</b>	Enterprises are using these best-in-breed APIs, but implemented on a project-by-project basis
<b>Ecommerce APIs</b>	Allows developers to add best-in-breed ecommerce functionalities like shopping carts and payments capabilities	<b>Snipcart</b> <b>Stripe</b> <b>Commerce Layer</b>	Some enterprise use, some internal builds to replicate functionality
<b>Form building</b>	Enables developers to add form functionality without managing a database attached to the web product	<b>Netlify Forms</b> <b>Typeform</b> <b>Formspree</b>	Enterprises are doing server-side business logic on their web products to collect data and update databases directly from their websites
<b>CI/CD and ADN delivery</b>	After developing products using this JAMStack toolchain, developers can then generate a build process by defining a streamlined workflow and pushing their development to deployment from Git. This will create pre-rendered pages which can be delivered to an ADN for hosting and immediate access by customers. Further product development and additional features, or new content, can then be added via continuous delivery build processes.	<b>Netlify</b>	Use of complicated CI/CD tools and build processes Enterprise use of CDNs

Many of these component services are used regularly in enterprises today. But the CTO often has limited oversight on what tool choices developer teams are making.

By using the JAMstack model, infrastructure is decoupled from product development so that additional server load is automatically provisioned as needed, without developer team responsibility. CTOs can establish a common set of tools and component services that can be used by all developer teams. Prototyping of web products occurs as part of the workflow, rather than requiring multiple versioning for oversight processes, which holds up production flows as feedback from each version is reviewed and integrated into draft builds.

This generates greater developer satisfaction, which helps enterprises retain their best talent. As enterprises increase their digital service delivery, they need more highly skilled developers. This demand is generating a 15% growth rate over the next decade for the web developer workforce, much higher than in other professions<sup>2</sup>. Keeping the best developers as part of the enterprise team means giving those developers the opportunity to build, not to get weighed down by managing infrastructure or wrestling with legacy systems.

An enterprise's technology stack and the level of reliance on a monolithic IT infrastructure are two key factors that impact a business' ability to recruit and retain top developer talent.

---

<sup>2</sup> Bureau of Labor Statistics: <https://www.bls.gov/ooh/computer-and-information-technology/web-developers.htm>

## Three ways the JAMstack gives developers Superpowers:

### Why developers want to work in enterprises that use the JAMstack

The JAMstack turns enterprise developers into powerful members of your teams, who can be excited about their work on a daily basis as they solve problems and build new solutions. While enterprises often need to increase the size of their developer workforce, it is the JAMstack that then gives developers these productivity superpowers, including:

**Velocity:** Developers can use templates and streamlined workflows to deploy products in 3 minutes, not in 3-4 weeks.

**Collaborative:** The workflow hierarchy includes the opportunity to preview prototypes that share exact replicas of how the product is rendered online or on mobile and can be discussed before sign off. This occurs within the one workflow. Developers get real feedback about their work without having to incorporate comments made on multiple versions of their drafts in staging environments.

**Autonomy:** Developers can maximize their individual contribution and feel empowered to contribute towards business goals by using the JAMstack to get on with building products rather than managing operational infrastructure concerns arising between the frontend and backend.

**Together, developer teams are able to cultivate a 10x multiplier effect with these superpowered capabilities.**

# With the JAMstack, the CTO Gets Superpowers to Manage more teams, reduce complexity, and increase productivity

In recent years, IT reorientation started with a realization that in mobile and digital environments, legacy systems with monolithic code bases need to be broken down into component pieces (microservices). With microservices, APIs connect services into new compositions and more easily allow the introduction of new features without risking major downtime to deploy code and upgrade systems.

During this shift, LAMP stacks became common toolchains within IT departments. The LAMP stack relied on Linux, the Apache HTTP server, MySQL databases and the PHP programming language.

Enterprises would rely on a content management system (CMS) and plugins built on the LAMP stack to manage their web content, which (contradictory to the move to microservices) meant introducing and relying on another monolith.

## The evolutionary timeline to the JAMstack

**Monolithic systems:** All of the business operations in a single, monolithic code base became brittle as new feature additions risked breaking older code, and required production systems to come offline while being updated.

**Microservices:** A reorientation of monolithic systems into a series of components connected via API so that each component could be upgraded without the whole code base falling over or all operations needing to be taken offline during upgrades.

**LAMP stack:** Linux, Apache HTTP server, MySQL databases and PHP programming used to manage customer-facing digital engagement through websites and enterprise web applications. Often delivered through a CMS provider like Wordpress which builds and loads an initial HTML page from the server when required by the customer.

**MEAN stack:** MongoDB (or other NoSQL) database, Express, Angular and Node JavaScript frameworks. The MEAN stack leverages JavaScript to communicate between the frontend and backend. JavaScript frameworks allow sites to be more dynamic and include functionality in the frontend so that as customers engage with site content, delivery can be more responsive. Some webpage content can respond to inputs without requesting the backend server to reload pages.

**JAMstack:** Javascript, APIs and Markup allow delivery speed to be enhanced, as content is prerendered during the build process and deployed to edge networks so that it is instantly accessible to customer demand, rather than being built from the server each time.



## CMS and LAMP Stack Deficiencies

The CMS and the LAMP stack (and later, the MEAN stack, see inset box: *The evolutionary timeline to the JAMstack*) provided some benefits, but they also introduce new complexities:

**Speed constraints:** Initial page loads from the server still require rendering while the customer waits. Slow page loads drive customers away, with 53% of website visitors moving on if a page takes longer than 3 seconds to load<sup>3</sup>, while 79% of those who do stick around are reluctant to use and buy from the site in future<sup>4</sup>.

**Tightly coupled frontend and backend:** Frontend developers love their work because they love building things. The more they are required to monitor and manage performance of servers, the less actual building they get to do, and the less satisfied they are as members of your productive teams.

**Lack of version control:** CMS and LAMP stack architectures have more difficulty managing versioning. This often results in multiple versions of work being pushed to staging environments by developers who need sign off from multiple stakeholders. Frontend developer talent can be frustrated and overrun by draft edits as stakeholders each weigh in on different versions of staged content, and the need for developers to respond to multiple staging feedback can slow down the ability to push new changes to production.

**Increased repetitive work:** Many enterprises have many teams working on very similar development projects. Successful approaches learnt from deploying one product need to be rebuilt repeatedly in each new product, often manually. Since each web product ends up being a bespoke build, this reaffirms an organizational architectural culture where teams can choose their own toolchain, resulting in individual teams choosing different JavaScript frameworks based on their own preferences.

There is no doubt that with the move to CMS and LAMP stacks, enterprises have learnt new digital skills and been able to more rapidly create web products and apps for customer consumption, as well as create new web-based internal systems that act as enterprise apps in the browser.

But the complexities are now outweighing the advantages, and this is impacting on revenue:

- Slow load time for web content displayed in the browser and on mobile is driving away potential customers and slowing down internal efficiencies.
- Business logic introduced on the client-side is causing breaking changes that mean revenue generating elements of the business logic are not functioning correctly after a new feature is added.
- Costs are increasing for IT as developer teams are needing to focus resources to manage scaling of apps and server loads and website uptimes.
- And top talent end up looking for opportunities at other businesses where they can concentrate on building product.



## The Road Ahead: How the JAMstack Supercharges CTOs

The move towards putting more functionality into the frontend and creating single page apps has led enterprises to revisit the concept of microservices, focusing on decoupling services and using APIs to connect components. This component-driven mindset, alongside the growth in cloud managed IT systems, has encouraged enterprises to abstract their backends by using APIs to access components like databases.

One of the key enablers of the JAMstack model is the ability to manage all content assets and the build process via a Git repository. This creates a single-source of truth and introduces version control processes. Git enables code commenting and implements clear authorization chains for how pull requests are pushed to production.

This introduces the modern workflow in which an enterprise developer team can take content and create a build, drawing from the Git repo and using APIs to enable additional functionality. Pages are then pre-rendered and sent to a content or application delivery network so that when a customer requests a web page, there is no querying of a database and no creation by the server of the page. Instead, it is served as a cached static file from the CDN/ADN, instantly to the customer's browser.

---

<sup>3</sup> <https://developers.google.com/web/progressive-web-apps/>

<sup>4</sup> <https://skilled.co/resources/speed-affects-website-infographic/>

### Three ways the JAMstack gives CTOs superpowers

With larger teams to manage and more web products to be built in record time, the JAMstack gives CTOs the superpowers necessary to increase the productivity of their teams and deliver on business goals.

**Simplification:** CTOs are able to manage a streamlined workflow with version control where all developers are following the same CI/CD pipeline for their builds, creating prototypes for review that do not step out of the workflow into a staging environment, and are able to then push their changes to production without getting lost or held up along the way.

**Efficiency:** By pushing pre-rendered content to CDNs, scaling issues are minimized as infrastructure service providers take care of ensuring instant access to content to meet customer demand. Git repos allow the company's complete site to be centrally hosted, again cutting down on scaling and server load oversight. Serving a majority of content statically reduces costs and security risks from malicious data injection during production.

**Automation:** With central git repositories, use of a CDN, a common set of reusable components in place and a clear build and CI/CD pipeline in place, CTOs can concentrate on automating each stage of the process to speed up production and allow developers to focus on working with content producers and product designers to create engaging, revenue-generating web products, and internal enterprise apps and systems.

# Collaborate on Content and Design At Scale The JAMstack Simplifies New Content Creation for the CMO

How an enterprise enables content creation and delivery to customers through web products is a major focus of current reorientation efforts occurring in many companies. In exactly the same way as enterprises wrestled with monolithic code bases prior to adopting microservices, businesses around the world are now seeking ways to dismount their third party monolith, the Content Management System, often delivered via WordPress, Drupal, Sitecore or Adobe.

Many of the pain points described above have now reached critical mass (see in Section Four: *CMS and LAMP Stack Deficiencies*). Not only are they affecting the ability for enterprises to ship web products and engage with customers, they are also having an impact on developer retention and workforce morale. The 2019 Stack Overflow Developer survey<sup>5</sup> found

that 70% of respondents identified Drupal as their most dreaded web framework. (As a comparison, JavaScript was the most loved programming language amongst 67% of respondents and the JavaScript frameworks React and Vue were the top two ranked most loved web frameworks.)

In this same survey, WordPress was ranked number one as the “most dreaded” platform for developers to use, with 60% of respondents dreading having to use WordPress in their work.

That same dread with using WordPress, or systems like Adobe Experience Manager, is now also impacting on creatives. Many are frustrated by the complexity and confusion of new CMS “features” and struggle to adapt to overhauled approaches to how their content is uploaded and reviewed. This is frustrating enough for text additions, but for those

## Headless CMS: Making Life Easier for Creatives (and Frontend Developers, and CMOs!)

A headless CMS allows creatives to more easily share and upload their content in ways that mimic early Wordpress CMS interfaces, but without the reintroduction of monolithic code bases or the complexity of “user-friendly” drag and drop tools and feature overload. A headless CMS is an interface that CMOs can use to allow creatives to upload their content, without them having to use Git. A headless CMS enables live previews in a staging-like environment so that a creative can adjust their content after review, without impacting on the build process.

The design of content is separated from developer builds, so that developers handle the business logic and can ensure that revenue-generating components of a web product function properly and are not compromised by new design element additions.

working on design elements and images, the overbearing and overriding nature of many LAMP-based CMS systems that seek to reformat a creative's vision into what it thinks the creative wants to do is often now delaying delivery timelines.

**How the CMS is Now Holding CMOs Back from Delivery**  
In the new digitally savvy enterprise, CMOs rose to become a key stakeholder in the C-suite, being able to demonstrate revenue generation through their work by creating landing pages, managing campaigns, building engaging websites, and delivering a regular stream of content.

Part of that strength came from the CMO's user-centered design approach and use of data to identify and respond to customer needs<sup>5</sup>. But this integral role in the C-suite is losing some capital, as web products built using a CMS load slowly, reduce revenue generation opportunities and drive away potential customers<sup>7</sup>. In addition, key revenue generating business logic can easily falter as the addition of one component may unintentionally prevent sales-focused engagement events from triggering in a workflow.

Much like the decoupling of frontend and backend development and IT systems, a JAMstack enables the decoupling of content creation and design from development so that revenue risks are eliminated and product development is sped up. The JAMstack helps CMOs maintain a role as revenue generating engines for an enterprise.

## Three ways the JAMstack gives CMOs superpowers

CMOs are able to ensure their work has clear revenue impacts and fosters one of the rarest qualities in modern software development: the ability of the JAMstack to improve both developer and end customer experiences at the same time.

**Retention:** CMOs can help ensure top developer and creative talent do not seek other enterprise opportunities in order to avoid having to wrestle with WordPress or Drupal.

**Revenue generation:** CMOs can ensure web products are instantly accessible, speeding up purchasing behavior and strengthening customer relationships.

**Velocity:** CMOs are under intense pressure to constantly update digital experiences. The JAMstack architecture empowers a CMO's teams to rapidly deliver everything from punctuation changes to new campaigns with significant velocity improvements.

<sup>5</sup> <https://insights.stackoverflow.com/survey/2019>

<sup>6</sup> <https://www.constellationr.com/research/constellation-astrochart-new-c-suite>

<sup>7</sup> <https://skilled.co/resources/speed-affects-website-infographic/>



# Infrastructure-as-Code: The CIO's dream becomes reality (aka Automation as a Superpower)

For CIOs, the JAMstack represents a powerful new opportunity to build internal enterprise applications, and can give the CIO the superpowers necessary to shepherd both tech and the wider business along the road to automation.

The JAMstack is now being used to speed up building of enterprise internal facing apps, without introducing high maintenance costs. CIOs are finding the infrastructure-as-code model is fully realized with a JAMstack architecture design, and are creating secure, dynamic web apps for their workforces.

The concept of infrastructure-as-code promises reduced complexity through a single source of truth and the potential for more fully automated workflows.

## How the JAMstack makes Infrastructure-as-Code a reality

Infrastructure-as-code refers to the process of using code to manage server configurations and to automate provisioning of infrastructure. It is a process where code is written so that any potential performance issues or server overloads trigger alerts that initiate an event function in the code to respond to the alert and automate provisioning of necessary infrastructure.

This often involves maintaining version control and being able to rollback products to a previous build where necessary.

### Loblaw uses **Netlify** and **JAMstack** to increase digital productivity 3000%+

Loblaw is the largest retailer in Canada, with 2800 stores nationwide. Every week 85% of Canadians shop in one of Loblaw's stores, and every single day 7 million people engage with one of Loblaw's digital platforms.

Charged with building 150 digital experiences in two years, the Loblaw digital team knew it was time to change their current design system, which suffered from inefficiencies and burdensome bottlenecks. They opted to adopt the JAMstack, moving to Contentful, Storybook and NEXT.js, all bound together by Netlify. When Loblaw subsequently launched in its Mother's Day campaign, it experienced 92% better website performance, 26% better accessibility, and 11% better SEO. Lead time for the project was a month instead of the typical year, representing a 10x reduction in time to market, \$38,000 monthly cost savings, and an "infinitely happier team."

[Learn More](#)

With infrastructure-as-code, instead of an all-hands-on-deck culture for releasing new deployments, the business can move to a continuous delivery model where once a feature has been reviewed and approved, it can be deployed to production automatically, no matter what time the authorizing manager signed off on the latest preview.

The JAMstack makes this a low risk automation approach. Instead of a developer team spinning up a separate staging environment where they move their local build, the developer team instead uses a branch off the main git repo to build the new feature and issue a pull request when it is ready for production. The simplified workflow with a single-source-of-truth remains in effect, allowing an enterprise governance model to push the pull request to the appropriate manager to review and sign off. If all people in the approval chain sign off on the new feature pull request, it simply forms part of the continuous delivery pipeline and is moved to the build process and into production automatically, without needing any DevOps to help spin up new production servers or integrate code from a staged environment into a production environment. In addition, a build can include testing, which can be run as part of the continuous delivery pipeline and infrastructure-as-code will determine what actions should be taken if testing does not achieve the required results. This ensures that web products and enterprise applications are always available and not affected by breaking changes.

## Three ways the JAMstack gives CIOs superpowers

CIOs are empowered to unite the silos and connect the work of frontend developers with network edge service delivery.

**Abstraction:** Infrastructure is abstracted with the JAMstack and becomes a responsibility of the cloud provider. CIOs and developer teams only need to learn the workflows once, and do not have to create complex new models for CI/CD pipelines to fit into the cloud provider's way of doing things.

**Integrity:** The entire server tier is built into a Git push, which means there is only ever one version of the current code base for an enterprise application and features can be built as new sub-branches and only incorporated into the master branch once all approvals have been signed off, without needing any additional integration between staged and production environments.

**Control:** The JAMstack model gives full version control and the opportunity to create atomic builds that push new features live once they have been thoroughly reviewed, within a single CI/CD deployment pipeline.



# The JAMstack Era Begins Now - Supercharge your Enterprise with a 10X Multiplier

Enterprises need the JAMstack to make their investments pay off.

The IT modernization and reorientation process in itself has not been enough to improve the competitive strength of enterprises. The adoption of new C-suite skills and business management models that leverage user-focused design and data collection have not been enough to speed up product development to meet customer demand. Larger developer workforces in the enterprise have not been enough to increase productivity and streamline operations.

For many forward-thinking enterprises, IT modernization, digitally-oriented management models, and a developer workforce are now in place. The JAMstack is the new digital workflow that can now utilize these core fundamentals to generate 10X enhancements in product development velocity. And with Netlify as your JAMstack infrastructure provider, you can enforce a management model that harnesses:

- Consistent IT oversight on the use of developer tools, APIs, and frameworks
- A single-point-of-truth for all web applications (those in production and in development/staging)
- Simplified, automated workflows

Customers can engage with an enterprise at any time, wherever they are, through Netlify's application delivery network that serves up pre-rendered, dynamic, and secure web products instantly. Application monitoring and server performance is handled by the application delivery network and an infrastructure-as-code automated approach.

Data can inform product design and iteration, with new features previewed internally and pushed to production in a single workflow, using a common toolchain. Unsuccessful or outdated features can be rolled back easily, while revenue-enhancing features can be deployed instantly across an enterprise's entire product range.

Frontend developers are attracted to and want to work with Netlify and other JAMstack tools. That means enterprises using JAMstack technologies can recruit and retain talented developers who want to build product, not infrastructure.

**It's time to deliver better-performing web projects 10x faster, while reducing costs and overhead, with the JAMstack and Netlify.**

[Talk to an Expert ↗](#)