

# Build faster, safer, and smarter client websites with Jamstack

Deep dive into the issues that come with conventional CMS-based websites and learn how you can use Jamstack for leaner workflows and better client websites.

## Table of contents

<b>Sound familiar?</b>	<b>3</b>
Code recycling, live testing, feedback loops — the only way to develop for the web?	
<b>A look back on the development of the CMS</b>	<b>3</b>
Websites are becoming more dynamic and interactive.	
<b>Is the CMS really a model for the future?</b>	<b>4</b>
Complex database structures and high resource loads demand a rethink of static websites.	
<b>Static websites bring caching, high fault tolerance and security</b>	<b>5</b>
Fast loading times and high availability are critical to the success of today's websites.	
<b>Jamstack redefines static website builds for the better</b>	<b>6</b>
Decoupling and prerendering help Jamstack redefine the static website as a viable alternative for client websites.	
<b>Built for devs: Jamstack redefines workflows</b>	<b>9</b>
The close integration of GitHub etc. makes for smooth development processes with easy version control.	
<b>Jamstack's advantages for agencies</b>	<b>10</b>
Faster, more scalable and with less maintenance — Jamstack sets the benchmark for future web projects.	
<b>Adapt the Jamstack toolchain to your use case</b>	<b>12</b>
Lots of ways to create engaging website with a growing number of tools.	
<b>Performance Jamstack hosting with Deploy Now</b>	<b>13</b>
Launch your project via your GitHub repository on secure, georedundant IONOS infrastructure.	
<b>Client projects with Deploy Now in just a few steps</b>	<b>13</b>
Pick your favorite framework and set up your Git repository — Deploy Now takes care of the rest.	

## Sound familiar?

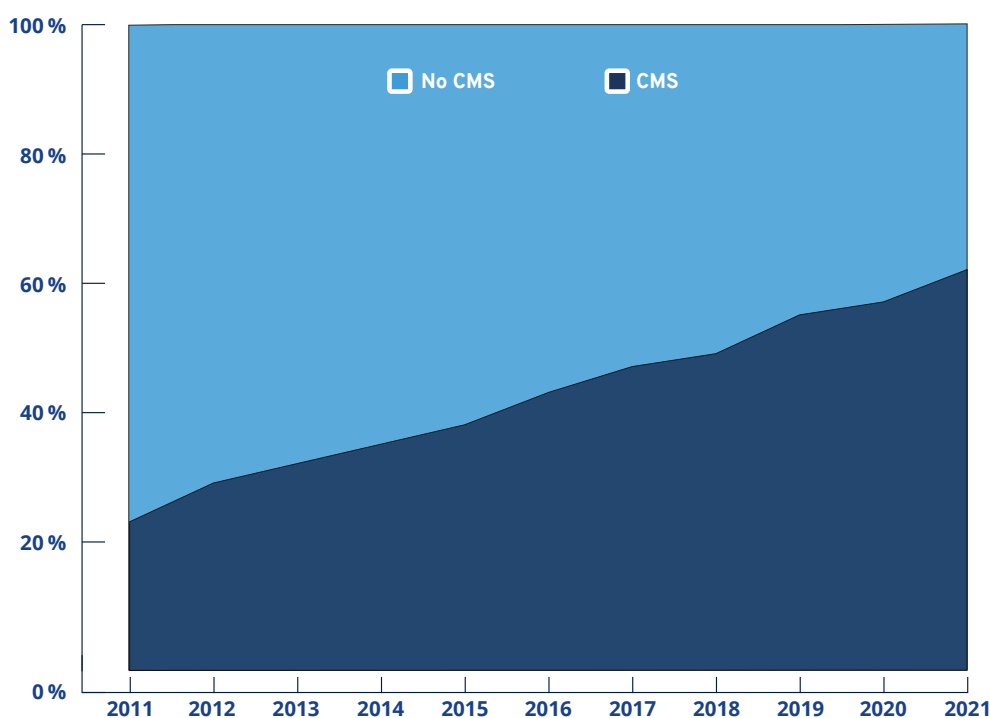
Building web projects for clients means you face a variety of jobs, from simple to complex. Typically though, you use existing code, tailoring it to the project using a content management system (CMS) like WordPress, Joomla or Drupal. The development process is characterized by a series of live tests, followed by client feedback — necessary steps that can put an unnecessary strain on your time and resources.

**Sound familiar?** Consider making the switch to Jamstack and start paving the way for leaner, faster client websites.

## A look back on the development of the CMS

When the world wide web was born with the [first website](#) over 30 years ago, static websites defined how we experienced the internet. Every page started with a little HTML, then was garnished with a pinch of CSS, and served up to every user.

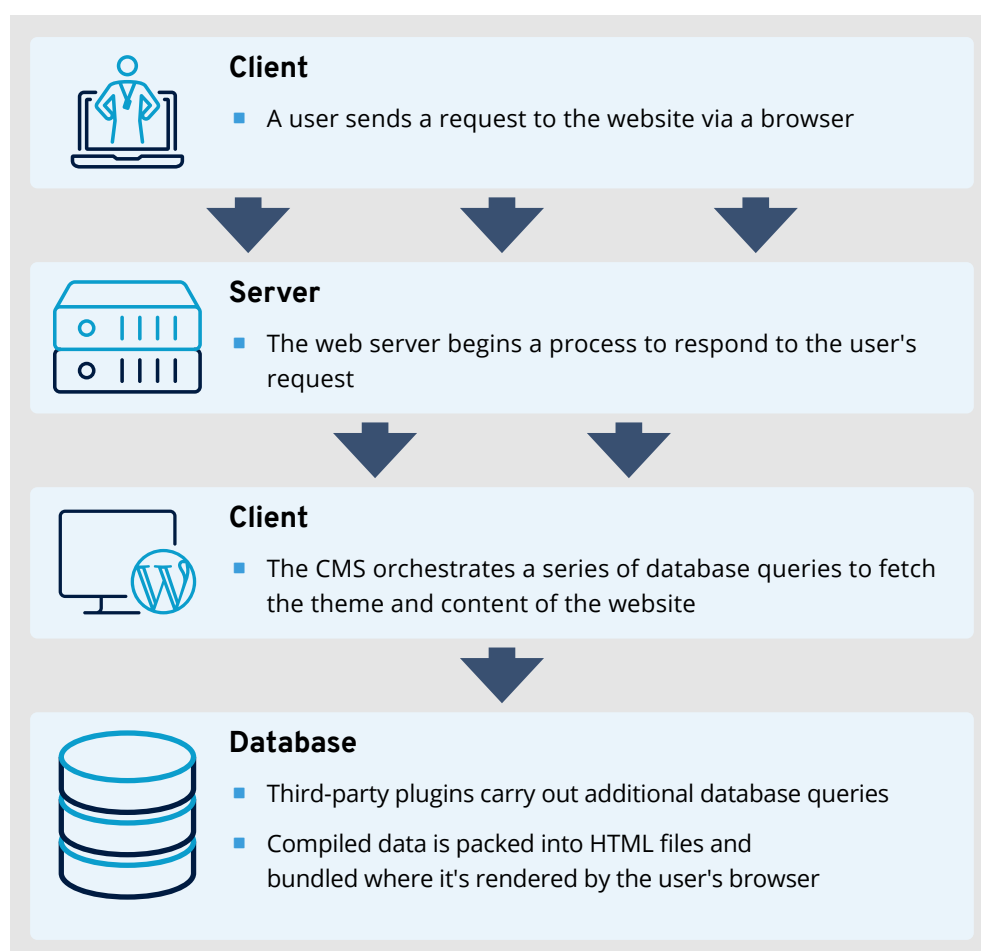
This changed at the turn of the millennium — enter Web 2.0. With this trend came increased emphasis on interactivity, usability, and user-generated content. The web became dynamic with content management systems like WordPress. Users now had an easy way to build their websites and manage their content without any coding or programming.



**The result?** The number of websites grew exponentially as small businesses, organizations, and individuals found a way to get online. As competition for visitors increased with the number of websites, attention shifted to keywords and SEO.

## Is the CMS really a model for the future?

Unlike static websites, which have a simple data structure that's always available, conventional content management systems utilize a complex architecture that relies on time-consuming response processes.



This process has gone on virtually unchanged for the last 20 years. Does this mean that LAMP stack-based content management systems will continue to be the way to build future websites? Given that performance (low load times, scalability, etc.) is the primary factor for a website's success, there's some room for skepticism.

## Loading times

The total number of required database queries increases the loading time of a CMS-based website. To offset this, caching mechanisms are developed, but these add complexity to the architecture, further slowing down the site. And today's users rarely tolerate slow loading pages. Not only do they cause more bounces, but they can also impact search engine rankings.

## Resource consumption

Setting up a CMS usually takes two resources: a web server with a file system and a database. Since requests to a CMS force frequent database queries, the database often becomes a critical bottleneck. With high-traffic or popular websites, this hunger for resources can quickly translate into higher hosting costs.

## Scalability

Today's social media marketing, viral campaigns, and even email marketing can result in sudden peaks in website traffic. Peaks like these can quickly overwhelm content management systems. As a consequence, requests are queued or immediately dropped.

## Maintenance and security

CMS versions and their plugins need to be updated regularly, in order to close known vulnerabilities. The underlying architecture of a CMS also offers multiple attack vectors through its server-side processes and database.

## Static websites bring caching, high fault tolerance and security

Static websites are 100% cacheable, making Content Delivery Networks, i.e. geographically distributed caches that minimize loading times, a very attractive option for you. Caches, however, are not an absolute necessity for static websites. This is because the web server only serves the existing HTML and CSS files to the user, keeping loading times negligible. Static sites also work without scripts or server-side logic, so their fault tolerance is high. They're also less vulnerable to attack, due to simpler architecture.

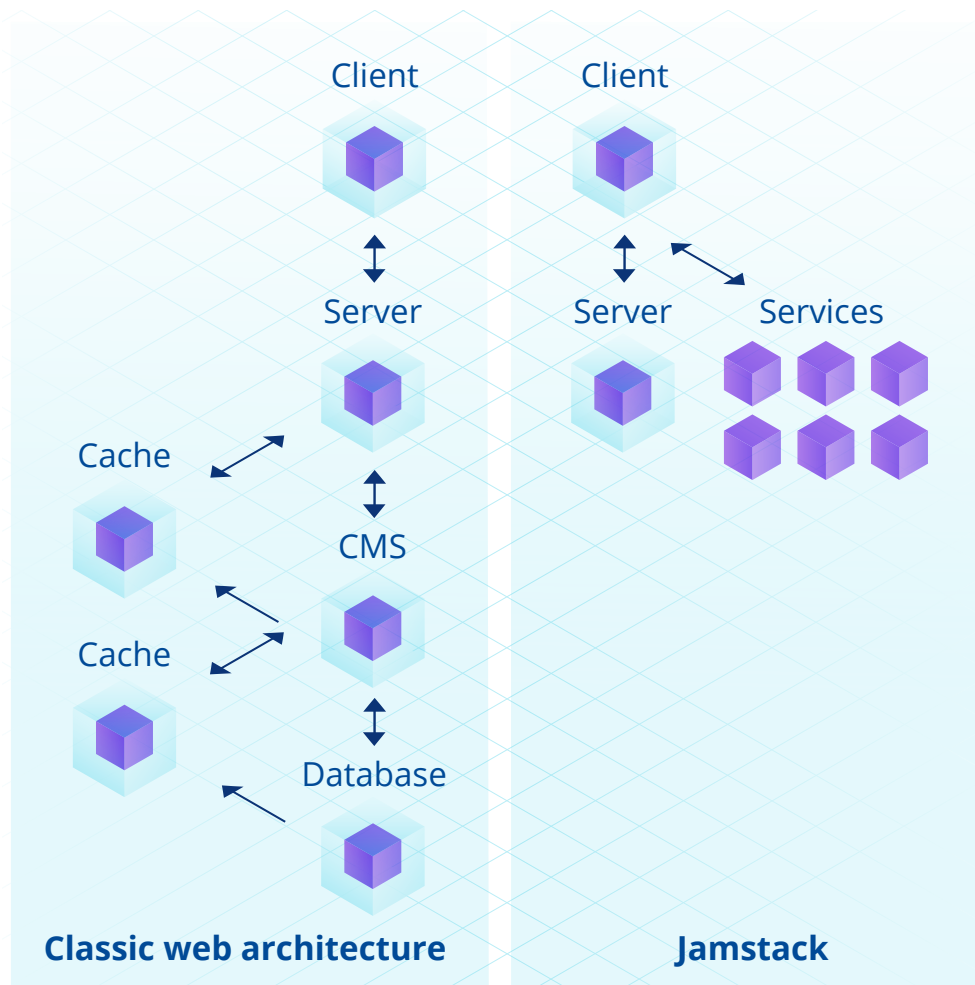
Despite these advantages, static websites do have some drawbacks, which help explain why CMS-based websites are so popular. Static sites by themselves don't have an interface that allows for easy editing without going into the code. Code often ends up duplicated, where it can be found unchanged in a multitude of HTML files. And there is no way to incorporate outside, variable data into these websites.

## Jamstack redefines static website builds for the better

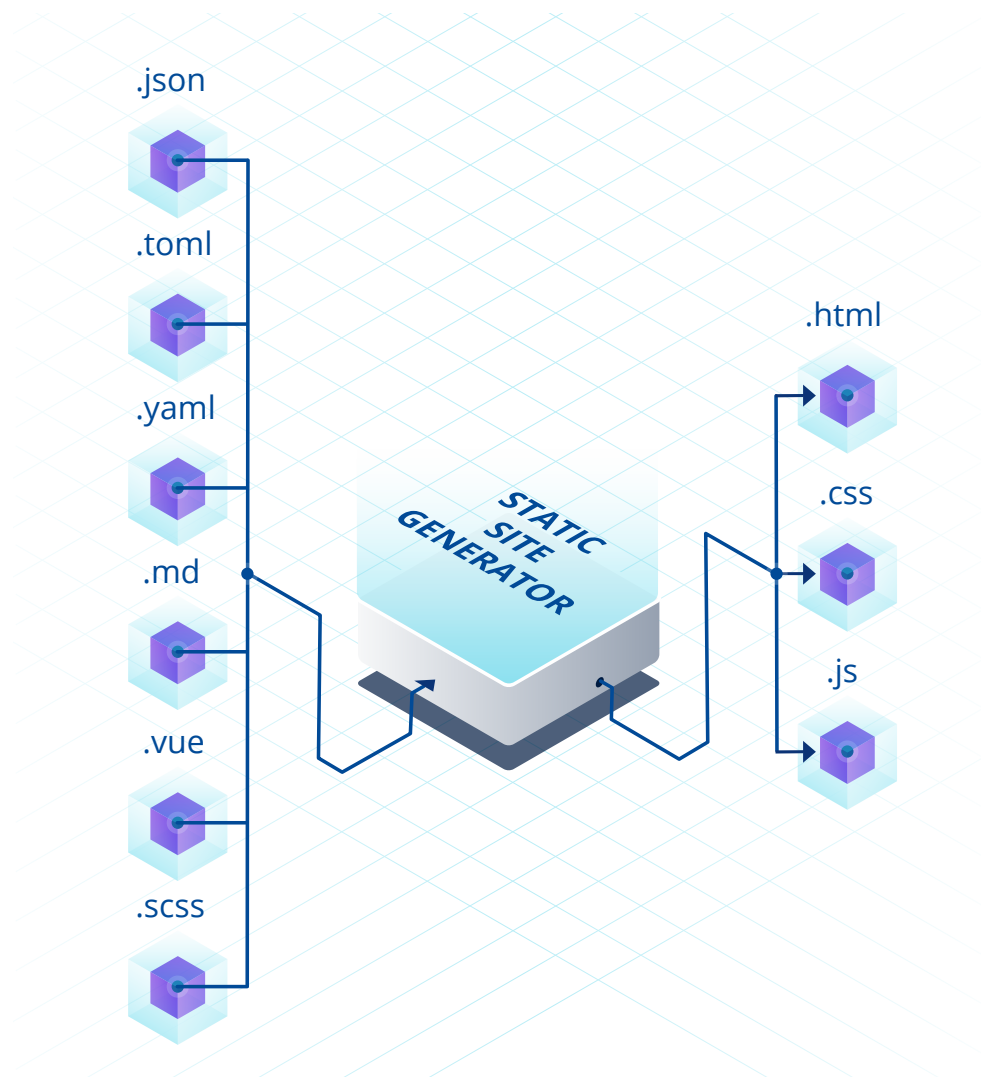
Jamstack has made the static website a viable option again. Client-side JavaScript, reusable APIs and static Markup make up the Jamstack. And in this respect, the Jamstack represents a new web architecture rather than a concrete technology stack. Decoupling and prerendering are the two core principles behind Jamstack.



## Decoupling:

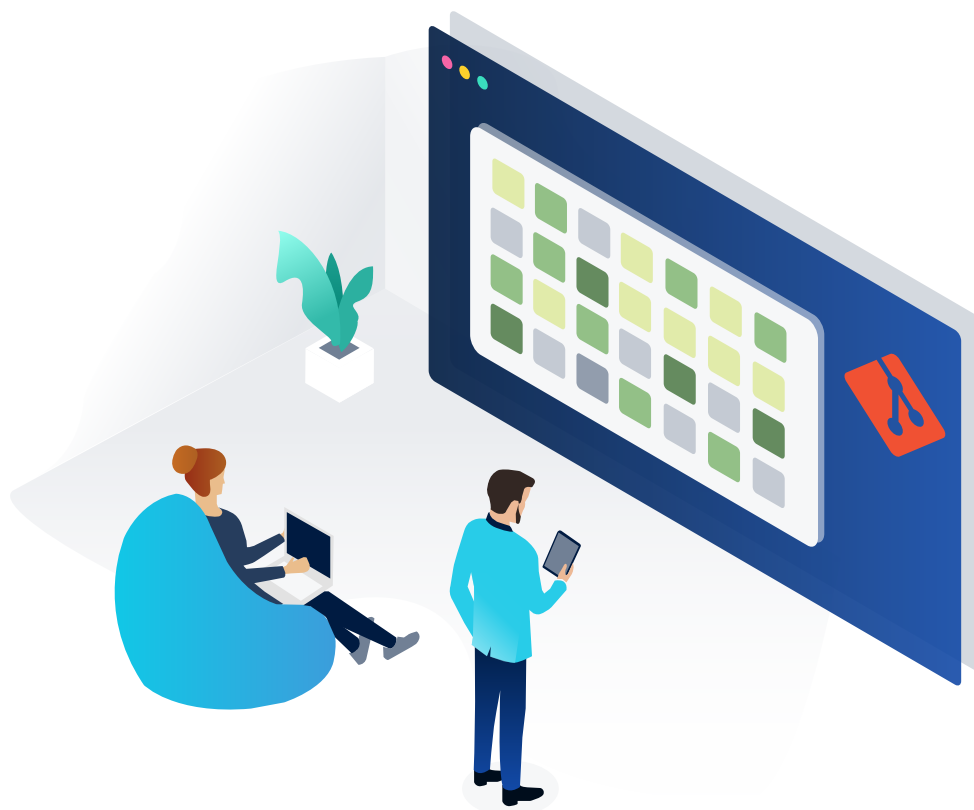


While conventional web architecture relies on a multi-layered request-response process, Jamstack aims to keep the response process as lean as possible. The web server delivers HTML and CSS files directly to the user on request, eliminating the need for database queries altogether. Instead, the web server delivers static content and doesn't need to compute responses to the client at runtime. Dynamic content is fetched by the user's browser once the frontend renders. This is done using client-side JavaScript that connects via APIs to a variety of proprietary or third-party services.

**Prerendering:**

A core Jamstack component is the static site generator. This script generates static HTML files every time it receives new data, content or templates. These newly generated files are then updated and made available on the web server after each content change. The update process takes longer compared to saving a content change using a CMS. However, the response time is reduced considerably, since static files aren't regenerated for each request. Nowadays, there are static site generators for just about every popular scripting language.





## Built for devs: Jamstack redefines workflows

One major advantage of Jamstack as a website building solution for agencies is that it's very much oriented to the needs of developers. So much so, in fact, that the integrated version control systems like GitHub, Bitbucket or GitLab have become a standard among Jamstack hosting providers. Going live with web projects no longer disrupts the workflow.

Instead, you can deploy websites directly from a Git repository to a live environment while still in your testing and feedback phase. Every time you make a change to code or content, the static site generator triggers an update process and you can review the changes live a few seconds later. Most vendors allow you to deploy multiple code branches on separate preview URLs in case you don't want changes visible on your primary domain.

As a result, teams of developers can collaborate more easily. At the same time the versioning function of GitHub and related systems replaces the need for website backups.

“  
*I want to just focus on the build and the development, link a repository and just let my host do the rest.*  
 ”



**Manuel Kling**  
 CEO  
 Fachwerk Media GbR

## Jamstack's advantages for agencies

### Agency benefits

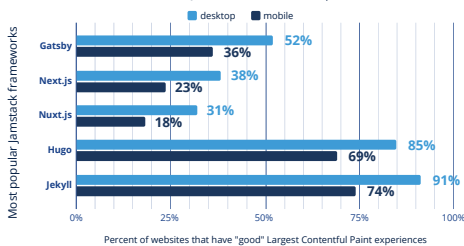


#### Websites load up to 10 times faster

- Higher probability of quicker SEO success
- More conversion through lower bounce rates

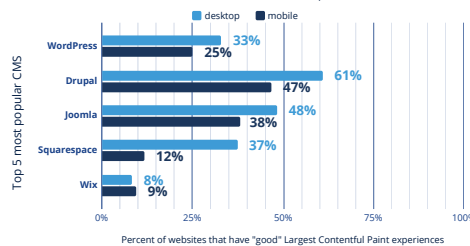
#### Real-user Largest Contentful Paint experiences

Web Almanac 2020: Jamstack (Chrome UX Report 202007)



#### Real-user Largest Contentful Paint experiences

Web Almanac 2020: CMS (Chrome UX Report 202007)



The Largest Contentful Paint gauges how fast primary website content loads. The static site generators Hugo and Jekyll show some compelling results.

Sources: <https://almanac.httparchive.org/en/2020/cms>, <https://almanac.httparchive.org/en/2020/jamstack>

#### Fewer vulnerabilities, less maintenance



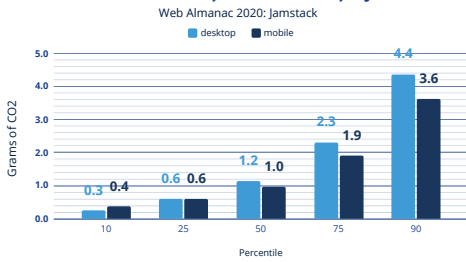
- Each plugin added to a CMS-based website brings a possible attack vector. Jamstack reduces these to the bare minimum.
- 56% of malware incidents in 2019 were traced back to out of date CMS applications** (<https://sucuri.net/reports/2019-hacked-website-report/>). Because it's not a CMS, Jamstack doesn't require frequent updates.

#### Lower costs, more scalability, and a lower environmental impact

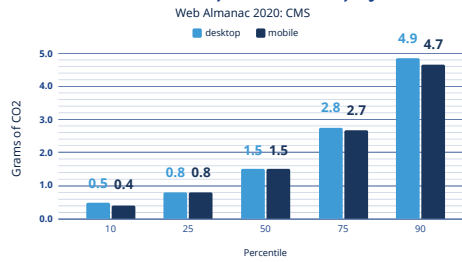


- Since the web server doesn't require much processing power, server costs fall considerably. Jamstack has high traffic scalability built in, especially when paired with a reliable cache or CDN. Both features cut resource consumption, lowering a site's overall carbon footprint.

## Carbon emissions per Jamstack page view



## Carbon emissions per Jamstack page view



Jamstack lowers the carbon footprint of a website considerably.

Sources: <https://almanac.httparchive.org/en/2020/cms>, <https://almanac.httparchive.org/en/2020/jamstack>

“  
Delivering a fast, secure,  
and scalable online  
store is enormously  
important to our clients.  
Jamstack has given us  
the perfect foundation  
to make this happen.”



**Enrico Franke**  
CEO  
Fruchte GbR

### Attract talented developers



- Decoupling and integrated Git allow developers to use the latest frameworks and work in agile, autonomous teams.

## Potential for development

### Integration of dynamic content



- Adding a contact form or a store as a microservice has many architectural advantages, but integration takes more time and effort than it would with a conventional CMS.

### Convenient content editing



- Headless CMS are a great way to edit and add content to Jamstack sites, even though they don't provide the same visual experience that WordPress and other conventional CMS offer.

### Design changes only via code



- Customizing the design of a Jamstack website requires editing its code. This can add to software developers' workloads.

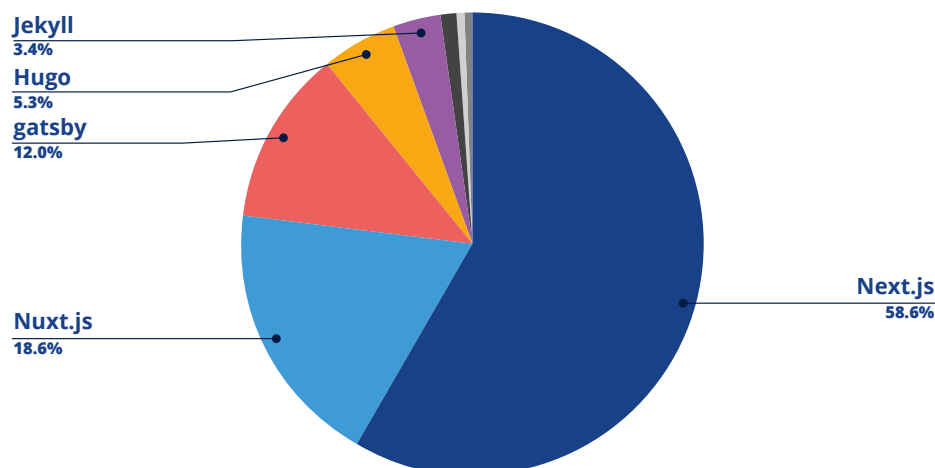
## Adapt the Jamstack toolchain to your use case

While the complexity of the Jamstack toolchain shouldn't be underestimated, its sheer versatility offers a number of options for meeting your project's needs.

Tool	Use case	Provider
JavaScript frameworks	Modern, component-based frameworks for the frontend of your Jamstack site	<ul style="list-style-type: none"> <li>Vue</li> <li>React</li> <li>Angular</li> </ul>
Static site generators	Static HTML file generation	<ul style="list-style-type: none"> <li>Jekyll</li> <li>Hugo</li> <li>Nuxt</li> <li>Gatsby</li> </ul>
Headless CMS	Simple content management via a graphical user interface	<ul style="list-style-type: none"> <li>Contentful</li> <li>Forestry</li> <li>Sanity</li> </ul>
Git	Code and asset repositories that facilitate versioning and collaboration	<ul style="list-style-type: none"> <li>Github</li> <li>GitLab</li> <li>Bitbucket</li> </ul>
eCommerce APIs	Convert Jamstack sites into online stores via connected microservices	<ul style="list-style-type: none"> <li>Stripe</li> <li>Snipcart</li> </ul>
Forms	Easy integration of database-free contact forms	<ul style="list-style-type: none"> <li>Typeform</li> <li>Formspree</li> </ul>

### Jamstack adoption share

Web Almanac 2020: Jamstack (Mobile)



### Most popular site generators

Source: <https://almanac.httparchive.org/en/2020/jamstack>

## Performance Jamstack hosting with Deploy Now

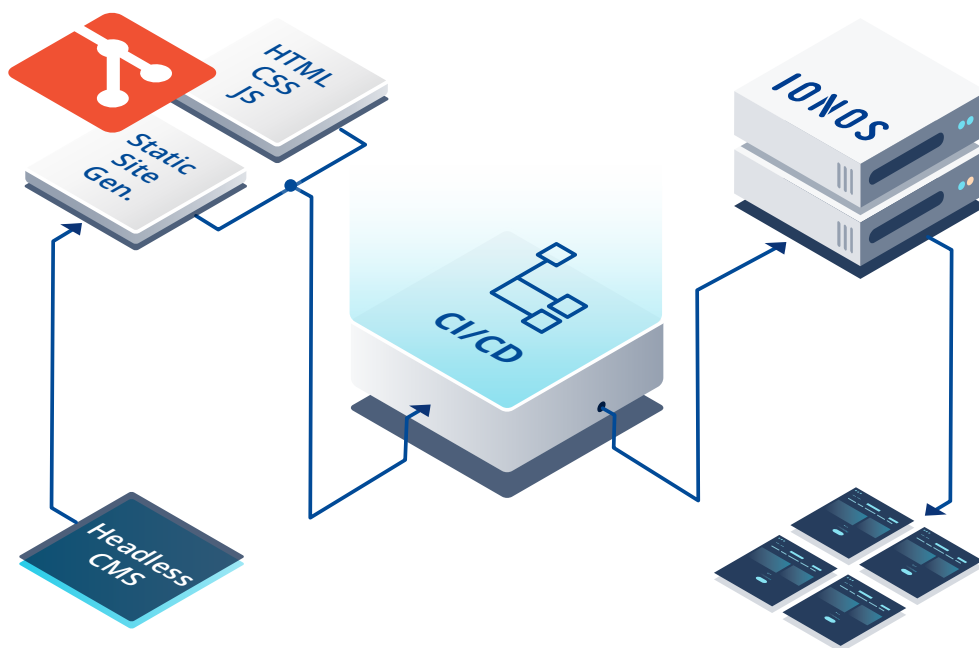
Using Deploy Now you can easily deploy your Jamstack website from your GitHub repository on IONOS infrastructure. Your first three projects are available at no cost to you. Each project allows you to deploy productive branches and link them to your domain, as well as create feature branches and use them as a staging environment. All your deployments go live with SSL certificates on georedundant, DDoS-protected infrastructure.



# Deploy Now

[Try it for free](#)

## Client projects with Deploy Now in just a few steps



### Step 1: Choose the right framework

Find the static site generator that best suits your project. Jamstack has put together [a list of supported generators](#) for popular scripting languages like JavaScript, Go, Ruby, and Python. Many JavaScript generators are also based on Angular, React and Vue.

## Step 2: Set up a Git repository

Ideally, the entire dev team can work locally on website code, synchronizing it via a shared Git repository. This way, developers can open their own feature branches without causing code conflicts. Learn more about setting up your workflow quickly at [GitHub](#).

## Step 3. Select your repository

Web projects don't usually start from ground zero. By cloning an existing Git repository, you get your web project started quickly. For instance, you can create your own starter repositories, use code templates from the web, or get started with one the [Deploy Now samples](#) from our documentation.

## Step 4: Connect to Deploy Now

As soon as you've chosen your repository, you can conveniently connect it to Deploy Now. From this point on, any code changes you make will be automatically updated by the static site generator and deployed to your web server as HTML and CSS files.

## Step 5. Develop with staging

It's a good idea to review changes to feature branches in staging deployments before you push them to your live website. This helps prevent errors and conflicts. With Deploy Now, you'll get a preview URL for each staging deployment, which you can easily share to collect feedback.

## Step 6: Merge and push to the production branch

Once you are happy with your code edits, you can conveniently deploy them to the master branch for your site. You can do this by either committing changes from feature branches in Git to your master branch, or by defining your feature branch as the new master branch in Deploy Now.

## Step 7: Connect your domain and go live

Last step. You easily replace your preview URL with your own domain, Deploy Now automatically adds an SSL certificate, and your site goes live!

## **Publisher**

1&1 IONOS Inc.  
701 Lee Road, Suite 300  
Chesterbrook, PA 19807

### **Contact IONOS**

Phone: +1-484-254-5555

Email: [info@ionos.com](mailto:info@ionos.com)

Online: <https://www.ionos.com>

### **Executive Board**

Hüseyin Dogan, Dr. Martin Endress, Claudia Frese, Henning Kettler,  
Arthur Mai, Britta Schmidt, Achim Weiss

### **Chairman of the Supervisory Board**

Markus Kadelke

## **Copyright**

This white paper has been created with great care. However, we cannot guarantee the correctness, completeness or relevance of its contents.

© Copyright 1&1 IONOS Inc. 2021

All rights reserved, including those relating to the reproduction, editing, distribution and exploitation of the contents of this document – or parts thereof – beyond the scope of copyright law. Any such actions may only be carried out with the written consent of IONOS. IONOS reserves the right to update and change the contents of this white paper.