

03.

# Hosting

Traditional DCs, cloud giants and  
frontend-focused hosting



**Yan Cui**

AWS Serverless Hero  
and [Independent Consultant](#)

# ” The future of frontend development calls for platforms such as Netlify and Vercel which provide easy-to-use features for frontend-focused teams.

The results of the State of Frontend 2020 survey appear especially interesting when it comes to the subject of hosting. When you look at the numbers, there's everything: from traditional DCs through cloud providers to the new kids on the block like Netlify and Vercel.

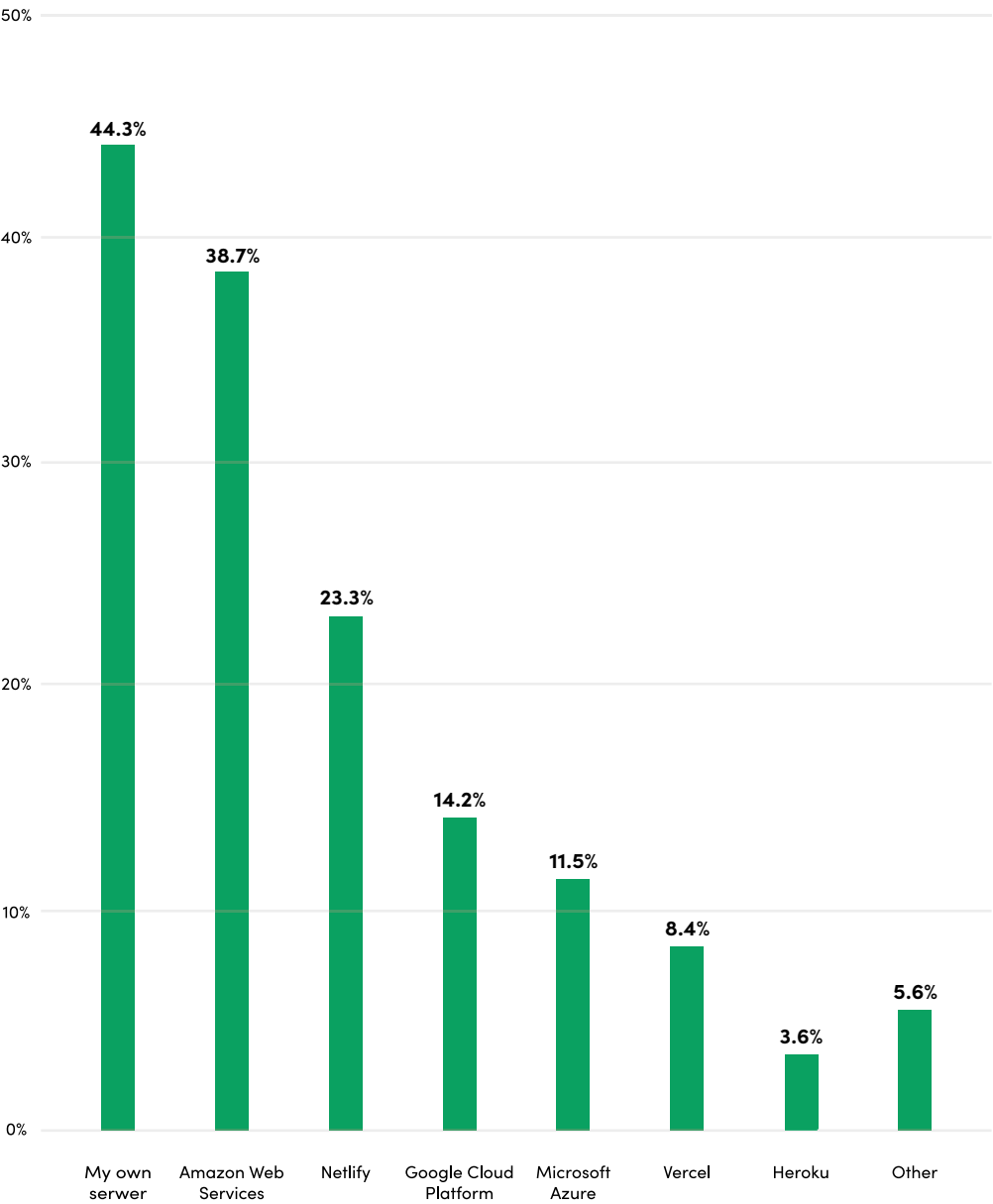
For me, by far the biggest surprise from these results is that 44.3% of respondents are still deploying their applications to their own web servers! Once again, it's a reminder that there is still a massive market for traditional DCs and that there's still much growth opportunities for the public clouds.

It's not really surprising that Amazon Web Services is the most popular deployment target among the cloud providers (38.7%). However, it may amaze (pun intended) some of you that the AWS's share is bigger than those of Google Cloud Platform (14.2%) and Microsoft Azure (11.5%) combined!

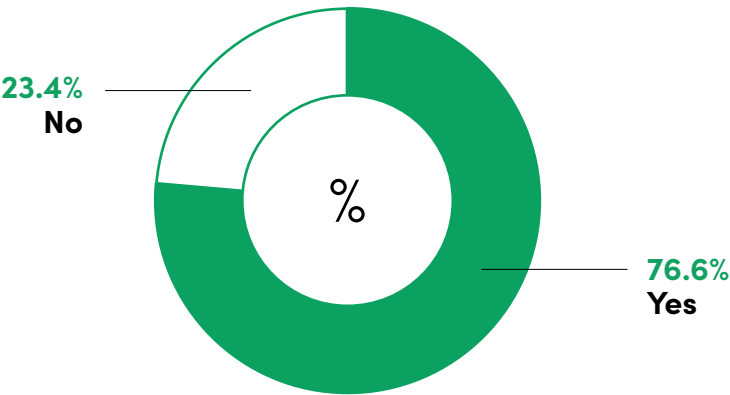
Also, the fact that Netlify has greater penetration (23.3%) than both GCP and Azure is a testament to how great a job they have done. It strengthens a growing school of thought that the future of frontend development calls for platforms such as Netlify and Vercel, which provide easy-to-use and yet powerful abstractions for backend infrastructures for frontend-focused teams.

AWS's continued push for Amplify would suggest they too see the potential here. And, on the other hand, one can't help but feel that GCP had a missed opportunity with their acquisition of Firebase all those years ago and their failure to develop it into the market leader it had the potential to become.

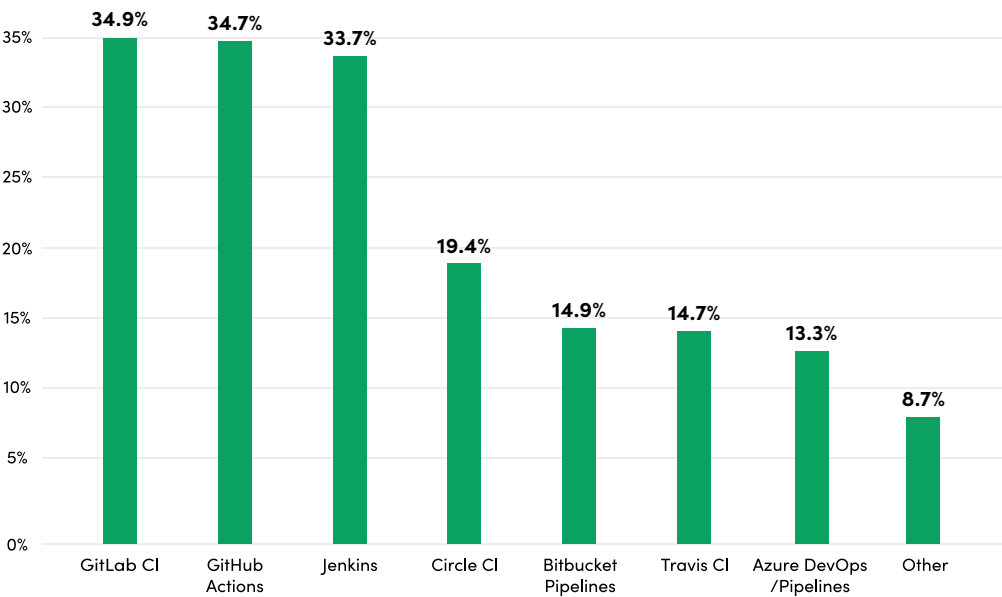
# Where do you usually deploy your applications to?



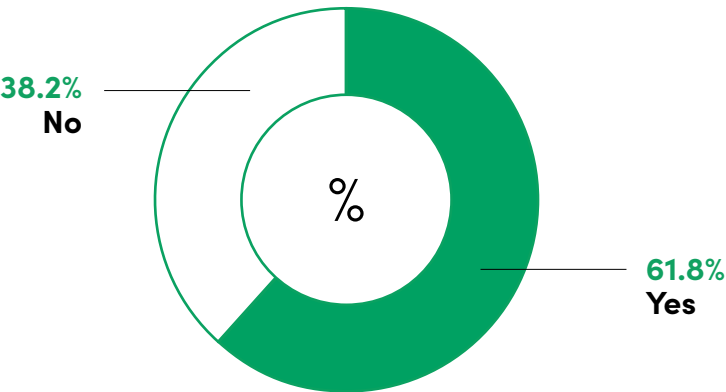
# Do you use Continuous Integration?



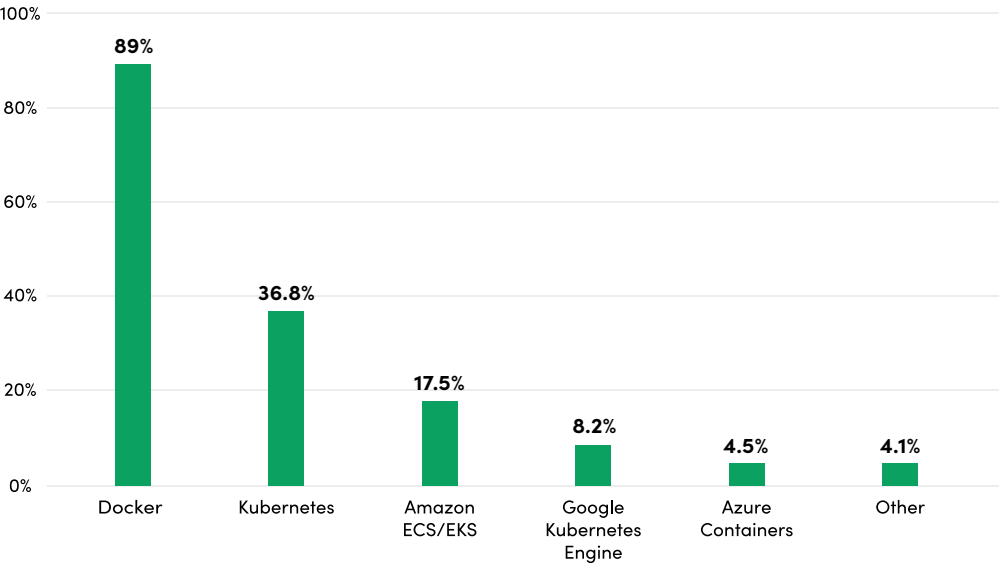
## Which CI solutions have you used during the last year?



# Do you make use of containerization?



## Which container management solutions have you used during the last year?



04.

# Jamstack

Ecstatic about static



**Tim Neutkens**

Head of Next.js at [Vercel](#)

**” Let me break the news:  
we expect an even larger  
share of developers  
building Jamstack  
websites in the upcoming  
months.**

It's great to see that almost one third of respondents have built a Jamstack (JavaScript, APIs, Markup) website lately. Also, it makes me personally happy that more than half of them have used Next.js – a React framework for Jamstack we've created at Vercel. And let me break the news: we expect an even larger share of frontend developers building Jamstack websites in the upcoming months.

To me, the appeal of Jamstack is that it lets us do less and accomplish more. With Jamstack, instead of rendering a page on every request (Server-Side Rendering), you pre-render a page before request time (Static Generation). This can be shared by all edges on a CDN (Content Delivery Network) for optimal performance, higher availability, lower costs and zero maintenance overhead.

Furthermore, Jamstack frameworks are evolving beyond static and adopting the flexibility of dynamic. For example, Next.js allows you to statically generate additional pages or regenerate

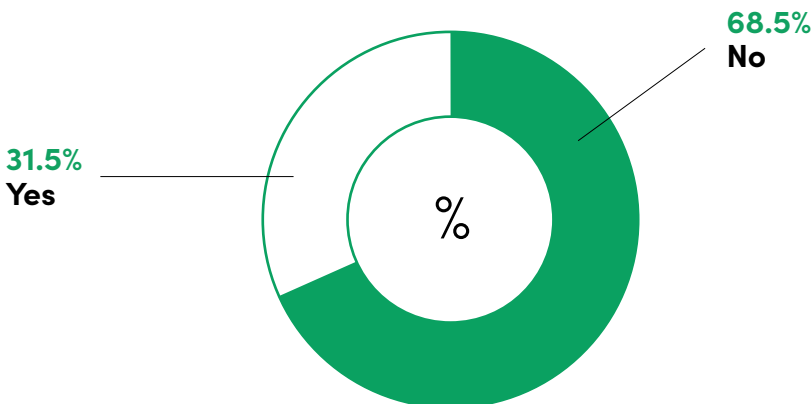


existing pages after the production build (Incremental Static Generation). Even if your app has millions of pages, the initial build will complete instantly as those pages can be generated incrementally.

Reusable APIs (the "A" in Jamstack) are on the rise too. The market now has many providers of headless CMS, headless e-commerce, headless identity, and more. No wonder that frameworks are evolving with these trends as well. I'm speaking from experience: Next.js has the preview mode feature which lets you conditionally bypass static generation when you're previewing a page on a headless CMS.

I'm very excited about the future of Jamstack. It's great to see that the respondents are using a wide variety of Jamstack solutions. It shows that developers are experimenting with different ideas – and that's pushing the Jamstack community forward for a more simple and performant web.

## Have you built JAMstack websites?



# Which static site generators have you used during the last year?

