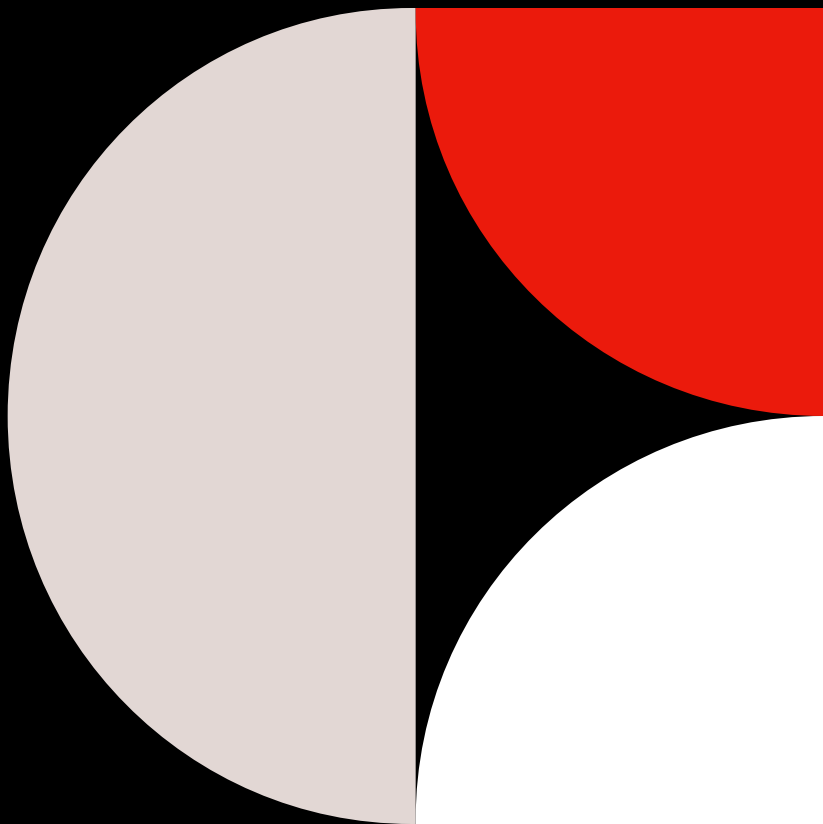
EBOOK

The Modern Application Development Playbook

How to solve the biggest challenges in app development by shifting to the edge



fastly

Introduction

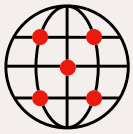
The 3 biggest challenges in modern application development

Organizations consistently encounter 3 big challenges as they work to build applications with great user experiences. It makes sense to aspire to iterate quickly and incorporate capabilities like real-time and personalized experiences, while also keeping things fast and secure, but here's where you run into problems.

The DevOps Struggle is Real

Speed of innovation

How to empower App Developers to innovate faster without creating risk by loosening controls, without over-complicating internal processes, and without getting bogged down by tech debt.



Performance at global scale

How to ensure that the websites and applications that the developers build will be highly performant, reliable and secure – and that they'll stay that way over time as things age and the user base grows! And how can DevOps scale themselves to provide this to every application being built or maintained across an org building for a growing customer base?

Optimization & Cost reduction

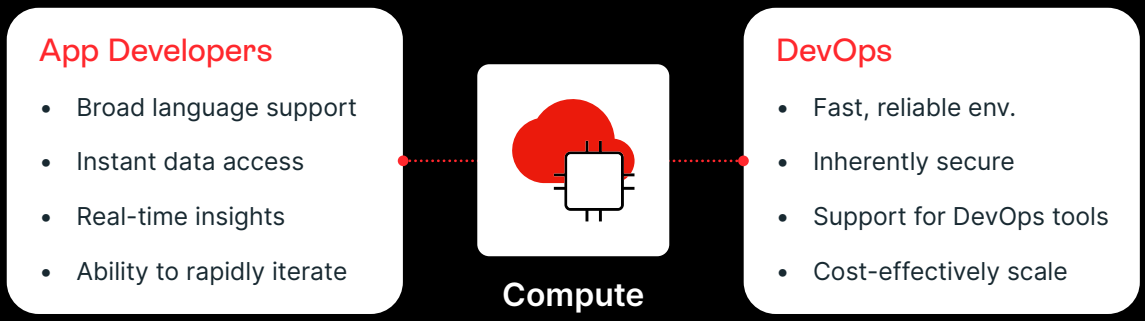
How to control operations and development costs associated with developing, delivering and protecting apps without putting restrictions on application developers or app features and capabilities.

DevOps needs an infrastructure solution that

- Removes friction for app developers without adding risk
- Works with the tools and workflows that developers already use
- Supports modern development and delivery practices like Continuous Integration and Continuous Delivery
- Scales their team's efforts to support a growing organization, expanding portfolio of applications, and growing user base
- Creates safe self-service global deployment options for other teams like SecOps to build internal applications without risk
- Provides a highly performant development environment

Comparing the different needs of DevOps vs application developers

While addressing the 3 big challenges above, both Central Platform teams and App Developers have different needs. In the past there hasn't been a good solution that serves all of them, but serverless edge computing has finally arrived with the right combination of capabilities to make everyone happy. Here's a closer look at how their needs differ, and how other approaches to this problem fall short.



App developers need

To move quickly and add features like real time communications and personalization. But they need to do it without risking taking the platform down, opening up critical vulnerabilities, or overwhelming the ops and security teams with too many applications and too much variance to maintain.

Platform teams need

To provision application developers while ensuring reliability and security. Often this comes at the cost of flexibility and speed. Even though these teams want to enable their app developer colleagues, they often are stuck in the position of limiting the platforms, tools, speed, and flexibility of the rest of the org in order to ensure reliability, security, and prevent ballooning maintenance costs or tech debt. They also need to be careful to keep compute and storage costs down, and not to rack up massive surprise bills through errors or misconfigurations.

Most teams settle for limited solutions

Most of the central platform and devops teams of today typically use CDNs to accelerate and protect their websites and applications, but these solutions create restrictions and limitations for the application developers who are trying to innovate and enhance on their existing work or build new applications for the organization. Teams often turn to virtual machines, containers, or serverless (not at the edge) solutions to try to bridge the gap, but each of these has big gaps when you compare them to everything that can be enabled with a serverless solution at the edge. Here's a chart that shows how they stack up.

How does Fastly's serverless Compute stack up against other solutions for App Developers and DevOps teams?

What's the need? (and who needs it)	Legacy CDNs	Virtual machines	Containers	Serverless (but not at edge)	Fastly Compute (serverless at the edge)
App developers					
Improves latency for users regardless of what device they are on, what type of content they are consuming or where they are	✗	✗	✗	✗	✓
Secure-by-design sandboxing isolates every request, and minimizes attack surface with no extra effort from developers	✗	✗	✗	✗	✓
Allows them to rapidly scale their services globally without the overhead of ops complexity	✗	✗	✗	✓	✓
Pulls real-time data from devices and apps so they can deliver faster more accurate user experiences	✗	✓	✓	✓	✓
Allows them to add more personalization to enhance these end user experiences	✗	✓	✓	✓	✓
Is API-friendly so they can easily link different software functions into one product (or easily integrate with 1st and 3rd party SaaS and PaaS providers)	✗	✗	✗	✓	✓

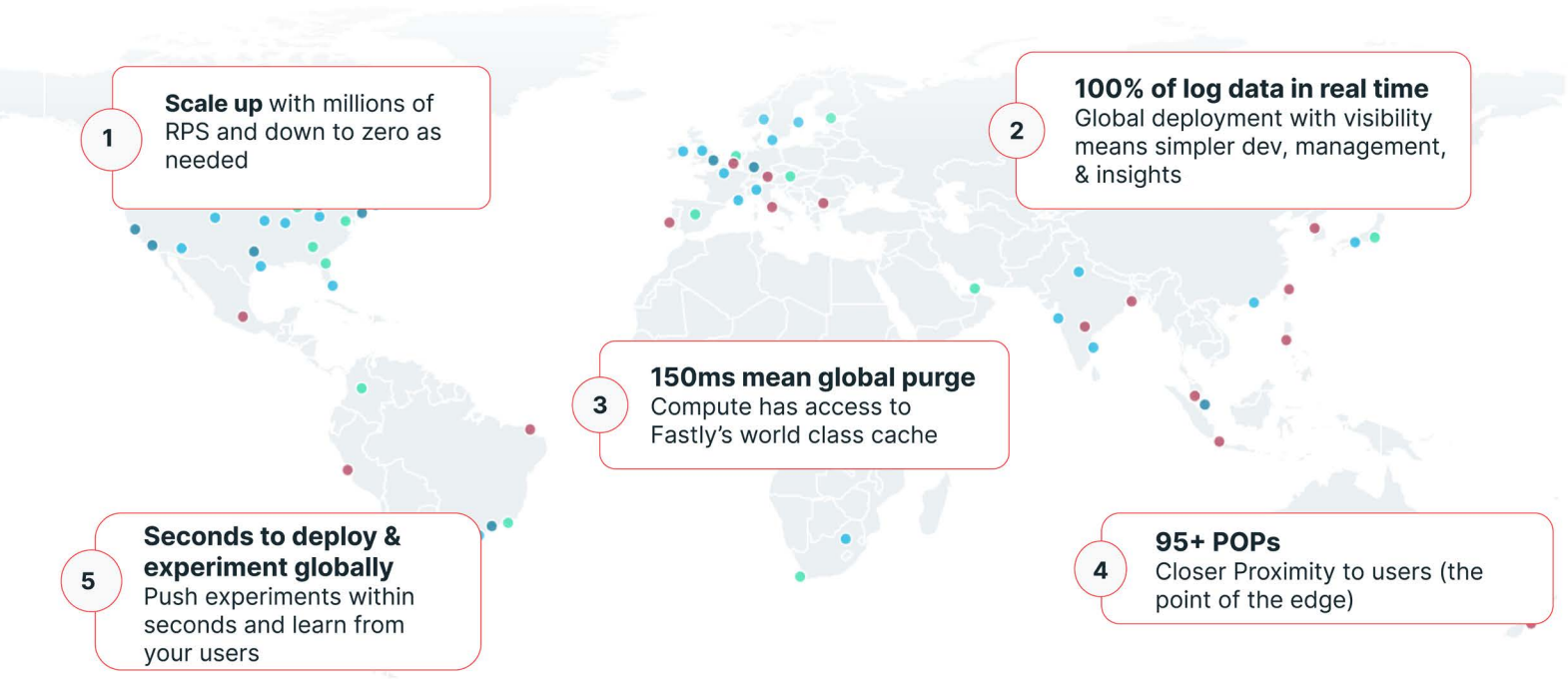
How does Fastly's serverless Compute stack up against other solutions for App Developers and DevOps teams?

What's the need? (and who needs it)	Legacy CDNs	Virtual machines	Containers	Serverless (but not at edge)	Fastly Compute (serverless at the edge)
DevOps teams					
Accelerates website/app/API performance despite rich/media experiences and complex app stacks	✗	✗	✗	✓	✓
Secure-by-default architecture improves DevOps processes by simplifying global deployment and observability for multiple dev teams.	✗	✗	✗	✗	✓
Rapid prototyping of global workstreams	✗	✓	✓	✗	✓
Allows them to leverage existing investments in core cloud infrastructure and common DevOps tools like GitHub, Terraform, DataDog, Splunk, Amazon S3 etc.	✗	✓	✓	✓	✓
Supports increasingly common API-heavy workflows by acting as a central tool to fetch data from multiple backends and services and dynamically stitch them together into one cohesive experience	✗	✗	✗	✓	✓

Serverless at the edge—a better approach

Fastly's serverless platform at the edge allows teams to move more of their development to the edge to take advantage of all of these benefits. It's the solution that is needed for truly modern app development. Our Compute platform sits between our customers' origin servers, and the end users who they are interacting with. We're right in the path between the requests and responses for every customer interaction, which means we're already in the right position to make things work better.

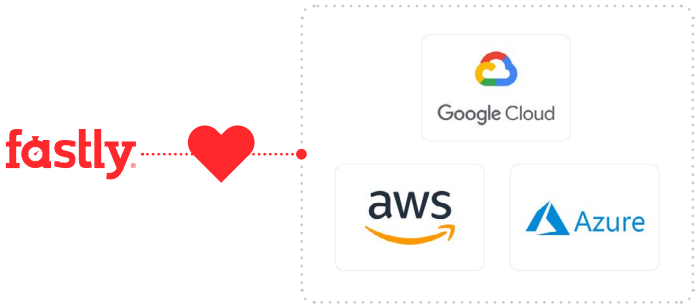
1. Accelerate and optimize responses with solutions like load balancing and image optimization
2. Protect those responses with solutions like our Next-Gen WAF and DDoS mitigation
3. Provide access to full configurability of the network to enable modern app workflows and processes like CI/CD for rapid innovation
4. Provide data store capabilities at the edge like **KV Store** and **Config Store** to enable real-time decisioning, personalization, A/B testing and more capabilities, all while continuing to cut out latency, trips to origin, and egress charges
5. Access to realtime data and messaging capabilities via Webhooks, plus even more real time capabilities than the competition with **Fanout**, our one-to-many realtime messaging solution for everything from chat and messaging during live events to ensuring real-time data updates are federated globally and instantly
6. Low code access to edge capabilities through Glitch.com, which abstracts away the edge while giving developers and easy place to start building and collaborating on projects



Fastly loves your core cloud

Fastly complements your core cloud investments – we love working with them, and they work better for you with us. Your core cloud is ideal for certain workloads such as:

- Apps that aren't latency-sensitive or dependent on large scale data analysis
- Data warehousing
- General purpose compute power



But when you have applications, or pieces of applications that require more time-sensitive and low latency tolerance, or rapid personalization of data, then those are the elements that should live on the edge. We're not replacing your core cloud, but we'll help you do more and better things without having to rip and replace.

More performant and secure with WebAssembly

When we tackle problems at Fastly, we always look for the best long term solution that makes future innovation faster, easier, and more secure. We avoid quick fixes that end up creating problems and complications over the long term, so we knew that building the best serverless computing platform on the edge meant we had to pick the right technology for the job. This new development environment was going to sit on top of our network and advanced caching platform – best-in-breed, fully software-defined, built on modern architecture and hardware. We needed the compute environment to be fast, scalable and secure – to be every bit as excellent and able to support future innovation as the rest of our platform. We chose the right building blocks carefully for maximum performance and security.

We looked at existing technologies like reusable containers, but nothing could provide all of the elements we knew we needed, so we decided to leverage WebAssembly (Wasm) instead. Here are the choices we made along the way, and why we made them.

"We see WebAssembly as having the potential to do what Kubernetes did for containers, but for serverless approaches to development."

— Devin Dickerson, Forrester Principal Analyst

Watch this on-demand webinar to hear more

Choice 1: We invested in WebAssembly because it's superior to existing technologies

Rather than relying on existing technologies for serverless compute, like reusable containers, we decided to leverage WebAssembly (Wasm) which translates several popular languages into an assembly language that runs incredibly quickly – it comes close to letting you run even higher level languages like javascript with the speed of systems level languages, offering big performance benefits without learning new languages. Wasm also allows developers to run complex code in microseconds across multiple environments. Our choice of WebAssembly was also in keeping with our long-term goal of being an open-source champion.

Choice 2: We built our own compiler to be faster

The common method chosen by competitors is to compile and run WebAssembly using the Chromium V8 engine, but this is too large and slow. (It's also less secure, but we'll get to that next.) To achieve the level of **performance, scalability, and security** that Fastly would feel comfortable putting our name on, we opted to build our own compiler and runtime in Rust, intensely optimizing for performance, security and compactness.

Because our Wasmtime compiler is WAY smaller and pared down to just what is important, it can provide **startup times under 50 microseconds**, and provide separate sandboxes for each request. Fastly is uniquely able to compile WebAssembly to fast, efficient native assembly code, and enforce safety and security at superior levels, and it's only going to get better. Because we didn't inherit all of the size and technical debt of the V8 engine, we are positioned to continue making even more improvements into the future.

"Starting with Fastly was probably the fastest infrastructure onboarding we've ever done. In less than three weeks, we went from 'Hey, we should talk with someone at Fastly' to having Compute fully in production."

— Jake Loveless, Edgemesh

Read this customer use case to learn more

"WebAssembly programs are sandboxed and isolated from one another and from the host, so they can't read or write external regions of memory, transfer control to arbitrary code in the process, or freely access the network and filesystem."

Read the full article on bytecodealliance.com

Choice 3: We built our own compiler because it's safer than the Chromium V8 engine

Fast and efficient is great, but it isn't enough. If we're building a platform for the future of development on the edge, then it also needs to be "secure by design." We built our compiler in Rust because it's a safer language with more guarantees than C++. Fastly's huge reduction in size for our compiler helps with performance as we already mentioned, but it also means a giant reduction in attack surface. We have less code to worry about securing, built in a significantly more secure-able language, and we use it to execute your code with way more granular sandboxing.

Fastly's Wasmtime compiler is able to provide per-request sandboxing, which means that a unique sandbox is created and destroyed for each and every request that comes through the platform. No matter how small. This is an impossible level of security for containers or virtual machines to execute, and even our competitors using the V8 engine and WebAssembly can't match it because their performance is not up to the task.

Per-request sandboxing effectively limits the risk of buggy code, or config mistakes from other users, and reduces the overall risk for your entire organization. Even for exploits like log4j, if an attacker is able to successfully exploit your code, they only gain access to what was available within that unique sandbox for that single request – an extremely limited exposure even for the worst exploits.

Fastly's Compute Benefits for Application Developers

✓ **Reduced complexity**	Serverless edge computing offers a powerful environment to build and rapidly deploy at scale without the complexity of virtual environments
✓ **Less DevOps dependency**	Because it's so easy and safe to build and deploy globally from the serverless edge, DevOps teams usually feel more comfortable with application devs building on the edge.
✓ **Speed and latency**	Moving more logic to the edge, closer to the end user, delivers performance benefits by reducing the amount of work that needs to go all the way back to origin servers. Combine that with our fast code execution startup times and you can enjoy significant improvements in application latency compared to other solutions.
✓ **Security**	Compute is "secure by design" with per-request sandboxing that limits risk for applications built and run on Fastly's serverless edge.
✓ **Fits into your existing workflows**	Compute has Terraform support and is API-friendly, with pre-built integrations into the tools you already use like AWS logging and backends, Okta, DataDog, New Relic, and more
✓ **Bring your own language**	Program in familiar languages like JavaScript, Go, Rust and WebAssembly.
✓ **Reduced cost**	Edge operations can scale up and down effortlessly, including all the way to zero cost when not being utilized. More savings are realized by reducing egress charges and reducing the distance between the operations and the end user. You only pay for the resources needed to fulfill the request instead of paying for hardware or a compute environment in case it's needed.
✓ **Predictable pricing**	Fastly's flat-rate packages provide predictable pricing for Compute (and other services) without ever charging for overages or surprise spikes in cost. Pick the package that's right for you and rest easy.

How our Compute Offering Solves for Central Platform Team's Problems

Fastly's Compute also solves for many of the challenges Central Platform teams face when trying to support modern app development efforts

- They can facilitate rapid prototyping for App Developers by providing a powerful serverless compute environment where developers can quickly spin up new code to execute on a function and then rapidly decommission it
- This development environment sits on Fastly's edge cloud platform so requests can be served from any POP on our massive globally distributed network meaning they are always routed to the most optimal location for faster performance.
- Our edge platform is also connected to all the major cloud providers by interconnects which also minimizes latency and improves reliability. If one cloud provider goes down, our built-in load balancing capabilities enable us to automatically switch to another one, so these revenue-generating websites and apps are not impacted.
- We support API-heavy workflows by acting as a central tool to fetch data from multiple backends and services and dynamically stitch them together into one cohesive experience
- The environment is isolated so it doesn't negatively impact the Central Platform team's efforts to fine-tune performance and SEO for existing apps and services

Additional resources

How our Compute Offering Solves for SecOps Team's Problems

Fastly's Compute solves for many SecOps problems by allowing customers to deploy it everywhere, consolidated down to a single WAF that protects all your organization's applications and APIs across any kind of architecture including complicated hybrid environments. It also enables security teams to extend its capabilities with simple additional logic at the edge to extend its capabilities. All you need is a little bit of Javascript or another supported language similar to consolidate all of your application security needs with Fastly's Next-Gen WAF combined with Compute.

Do more with the NGWAF by adding some logic at the edge:

- **Password validation**
- **Malware verification**
- **Edge rate limiting**
- **Visibility into origin server metrics for new rule creation**

Explore Common Use Cases

Fastly's compute has unleashed many creative use cases where App Developers have been able to improve performance and reduce costs by building on our powerful platform:

Edge Authentication:

Digital publishers can authenticate subscribers, and retailers can authenticate buyers at the edge for a faster login experience.

Explore

Geolocation:

Create a better user experience by keeping it local; this feature, when given an IP address, will provide information such as latitude, longitude, continent, country, city, and region.

Explore

Personalization:

Delivering personalized, high-performance content unique to each user is the ideal online experience—and the edge is the place for providing it.

Explore

Rate Limiting:

Handle hundreds of terabits of traffic per second, whether you're preventing bad bot traffic or managing a queue for a product release, our dedicated Edge Rate Limiting product is a great solution at Fastly.

Explore

Redirects at Scale:

An edge lookup enables the shifting of all static redirects to the edge, thereby reducing traffic to origin for lower infrastructure costs and lower latency.

Explore

A/B Testing:

Developers can rapidly push out new tests and adjust them as needed based on near-instant insights, all of which improves CX and helps drive higher conversion rates.

Explore

Waiting Room:

Ecommerce and travel & hospitality companies can help reduce operational costs and provide a better customer experience when managing traffic bursts by assigning waiting room tokens at the edge.

Explore

API Gateways/GraphQL:

An API gateway on the edge can improve scalability and enables increased cacheability for content that was not previously cacheable.

Explore

SEO:

Fastly has a number of features to maintain close control when managing all the aspects of SEO and maintaining a high-quality site experience that never compromises on performance.

Explore

Content Stitching:

By deploying logic at the edge to stitch together general and user-specific data into one combined request, more content can be cached to reduce trips to origin and lower infrastructure costs—perfect for personalizing experiences for customers of ecommerce and travel & hospitality companies.

Explore