CASE STUDY

How Spotify aligned CDN services on Fastly for a lightning-fast streaming experience
A business with eyes on the future

Spotify has built its business on flawless content delivery. Their streaming platform serves up more than 50 million tracks (plus an array of images and other assets) to more than 230 million monthly active users around the world — making them one of the world’s leading streaming services. With content that feels instant and immersive, Spotify helps their customers have the best experiences possible with their favorite artists.

Behind the scenes, Spotify’s technology has evolved over time to achieve its user experience goals. After a decade of growth, it got to a place where it was using a number of disparate CDN solutions — which added complexity to its platform architecture, as well as inefficiency within the R&D organization. To solve for this, Spotify created a new team to focus on standardizing their CDNs across diverse engineering teams, as well as provide automated tools, governance, and support. By leveraging Fastly’s edge cloud platform, the CDN team was able to build a self-service solution that allowed R&D teams to easily implement a reliable, high-performance CDN that made their content fly.

The challenges of team autonomy

Spotify’s engineering culture champions team autonomy. The company’s R&D organization includes over 2,000 employees grouped into teams or “squads.” Each squad operates as a self-contained unit with all the skill sets needed to accomplish a particular mission. In this autonomous model, every squad has the authority to make its own decisions, including which technologies, tools, and processes work best for the project or the squad. This helps squads move fast within their mission, while the trade off is a risk of technology fragmentation.

Too many CDNs, too little oversight

One of the more painful areas of fragmentation was around content delivery. Spotify’s CDN solution for business-critical content, such as audio streaming, was performing well and had been honed in to achieve low latency and high bandwidth. However, CDN operations for everything else had become an inefficient jumble. Some content was being streamed directly from buckets like AWS S3 or Google Cloud Storage. Developers simply made the bucket readable, dropped the URI into their code, and moved on. Other squads took services that had been created for a particular use case, such as image resizing or watermarking, and used them for something entirely different. Our new solution also resolved issues for a few squads who inherited CDN endpoints without full knowledge about their configurations.

https://newsroom.spotify.com/company-info/ Date as of 10/15/2019
Enter the CDN ops team
To help unify the organization's approach to content delivery, Spotify created a separate team to manage CDN operations and help internal squads improve the content experience in their services. The team’s mission was to find the best way to deliver any type of Spotify content to end users — and to be successful, they needed to take ownership of all traffic between clients and CDNs. This seven-person team would handle everything from business negotiations to racking hardware. They also aimed to create a standardized approach to CDN across the company, and bring Spotify’s diverse squads into alignment.

Building a company-wide CDN solution on Fastly
As a first step toward standardization, the new CDN team created a simple workflow that allowed Spotify squads to get a new CDN service up and running quickly on Fastly. Spotify had been using Fastly’s edge cloud platform for a number of years to successfully deliver audio streaming, and the company wanted to achieve the same level of performance for other types of content. With this approach, the CDN team faced the organizational challenge of “autonomous alignment” head on.

Convincing squads to migrate
Spotify’s new workflow appealed to squads that were looking to implement a brand-new service. But for those who were already running an obscure CDN domain, it seemed like more work to migrate to a new system than to keep the old one running. To convince their squads, the CDN team not only guaranteed that migrating to Fastly would achieve the same performance benefits as audio streaming, but also offered to take care of the messier business details, such as ensuring a zero rating and a guaranteed set of IPs. Even better, the team would wear the pager, taking care of monitoring the whole CDN request path all the way to the origin, 24/7.

A plan for low-maintenance operations
Apart from troubleshooting, the CDN team also wanted a system that was easy for them to maintain. Low maintenance often equated to heavy automation. The team decided to leverage Fastly APIs to build an automated system that could accelerate operations and give them the freedom to focus on projects that added more value to the business.
Bringing together APIs and VCL
Spotify's audio streaming service relies heavily on Fastly's customizable edge computing language, Varnish Configuration Language (VCL), which allows developers to perform intelligent caching, push application logic to the network edge, and tailor a user's experience based on location, language, and device type. Spotify wanted to use VCL to build custom CDN functionality, such as handling errors and redirects, as well as token identification and more. The team already had some solid practices in place, like tracking VCL changes in Git, however much of that was still manual.

They decided to solve for this by combining Fastly APIs with VCL and leveraging the strengths of both. They began using APIs to set up simple things that were unlikely to change during the lifetime of a service, such as creating the new service, configuring host names, and adding origins or logging endpoints. Everything else is done in VCL.

A self-service tool to request CDN review
Taking the combination of Fastly APIs and VCL a step further, Spotify developed SquadCDN, an internal pre-deployment review service. Any Spotify squad can access the tool, type in a short YAML string, and add a few details like domain, origin, bucket, and config flags. The tool then submits a pull request for the CDN team to review and approve. With this in place, the team achieved their objective — a super simple workflow with a degree of automation that made life easier, both for their internal clients and themselves.

Lessons learned along the way
As the CDN team pioneered new territory for Spotify R&D, they encountered a number of tips and best practices that helped them achieve their goals — many of which apply universally to internal team management and business use cases.

- **Dog fooding is key.** When the CDN team created their MVP, they moved a few minor services over to Fastly. But the team quickly realized that they needed to move their critical delivery services over first — which meant anything that was public facing, such as audio, video, cover art, and artist images. In the process, they were able to review and clean up any spaghetti VCL, as well as fix any flaws in their automated pipeline.

- **Be mindful of API call limits.** Although the limit is high, too many API calls at once can force a deployment to fail. For example, recreating a new version of every single service across the fleet on every deploy may cause the deployment to hit the API limit and stop halfway through. Although Fastly’s UI enables developers to revert back, the manual process can be time consuming for dozens of services. Carefully planning API calls will help ensure everything runs smoothly.
• **Verify all the things.** When a squad submits a new service, the CDN team verifies a few details that are crucial for the service to work. For example, they check if Fastly can read from the origin, whether it be AWS S3, Google Cloud, or a custom service. They give the squad a test file that will help them catch problems with configurations or bucket permissions.

• **Don’t allow writing to the bucket.** It’s important that squads don’t create a CDN endpoint that enables someone to send a put request and overwrite a file because credentials allowed writing to that bucket. (Luckily, none of Spotify’s services has needed to write to the origin.)

• **Do a smoke test.** Once they’re done with deploys, the CDN team does an end-to-end test over the internet and curls for a file that they know should be there. Custom VCL ensures that the file doesn’t get cached so that the smoke test path allows them to reach the origin and read the right bytes back. If there’s something wrong with the set up after deployment, they can just automatically apply a rollback to the previous version and start debugging.

• **Enforce good practices.** If squads want uninterrupted delivery via HTTP, they need to request it. This gives the CDN team the opportunity to discuss the squad’s needs and determine the right use case for it. They also ask squads to tag services that handle personal data, so they can better maintain GDPR compliance. In addition, they promote sane defaults with respect to caching and purging.

• **Make it easy for others.** No one will use a system that requires a lot of effort. Spotify found that it’s better for them to do some work on behalf of others so that everyone benefits in the long run. One tactic they used proactively identified some outlying CDN endpoints and offered to modify the code, so that the squads who owned them could more easily move them over to Fastly. This enabled internal teams across the org to move with greater speed and agility, and saved time on needless back and forths.

• **Keep your secrets secret.** Security is a top concern at Spotify, and protecting user data is a key component of building trust. It was vital that sensitive data, such as passwords, get blanked out in the logs. The CDN team used Fastly’s Edge Dictionaries to maintain key store values that cannot be read by humans and only referenced in VCL. Even when viewing VCL in the admin tool, developers only see variables instead of private data.

**Successful CDN alignment**

A year and a half after forming the CDN team, Spotify’s CDN landscape has cleaned up significantly. Over 60+ squads have begun using their new configuration system, which represents more than 20% of the R&D organization. Over 80+ services are now pushing content through Fastly, and doing so more efficiently thanks to the team’s templatized tools and simplified workflows. Most importantly, engineers are happy because they no longer have to worry about CDN details and can focus on their core mission.