Elastic improves Kibana CI/CD run time by 70% after switching to Buildkite Pipelines

Elastic decreased pipeline run time from 3 hours to 55 minutes while improving developer experience and decreasing costs by migrating to Buildkite.

elastic

Industry	Software
Tech	GCP, Terraform, TypeScript, Jenkins
Founded	2012
Customer since	2021

Improved productivity
Developers have faster feedback loops, can troubleshoot independently, and spend more time on high-value work.

Cut cost
Cloud infrastructure spend reduced by nearly three quarters.

Accelerated build times
CI/CD pipeline execution time decreased from 3 hours to 55 minutes.

Overview

Elastic is the Search AI Company, enabling everyone to find the answers they need in real-time using all their data, at scale. Elastic's solutions for search, observability, and security are built on the Elastic Search AI Platform, the development platform used by thousands of companies.

In 2021, Elastic reached the practical limits of the legacy continuous integration and continuous deployment (CI/CD) tool it was using. Buildkite offered a more flexible, scalable platform that helped Elastic to **decrease pipeline run times, reduce infrastructure costs,** and **improve developer productivity** to enable engineering teams to deliver critical updates and new features on time.

Kibana was the first Elastic product to migrate its CI/CD pipelines to Buildkite. Using Buildkite's highly scalable architecture and unique dynamic pipelines functionality, Elastic significantly reduced cloud infrastructure cost and decreased the Kibana CI/CD pipeline run time from 3 hours to 55 minutes.

Elastic also took advantage of Buildkite's developer-friendly UI to streamline CI error resolution, turning the CI system from the least-liked system in its developer experience surveys, to the most-liked.

“Designing the CI system we needed to meet our goals was so much easier once we moved to Buildkite... Buildkite just works the way you want it to—in so many scenarios.”

— **Brian Seeders,**
Staff Engineer,
Elastic Kibana

Opportunity

Massively scalable CI, helpful and intuitive UI

Elastic has enjoyed much success over the last decade, and is known for creating Search AI products that developers and enterprises can rely on. As the business grew, the Elastic platform team found that the automation and CI that served their developers well ten years ago needed updating to meet the needs of the increased throughput and requirements of a modern, sophisticated product engineering organization.

First generation CI/CD solutions, originally developed in an age when teams deployed on a cadence of at most once a week, helped teams run automated testing in a standard environment before deploying. Today, teams expect continuous deployment (deploying multiple times an hour) across hundreds of engineers and high volumes of code changes.

"Investing in our existing system to make it more reliable or faster didn't make sense for us," says Brian Seeders, principal engineer at Elastic. "Buildkite provided the capabilities we needed to easily meet the requirements of the business, and so the age-old choice of build-versus-buy was an easy one. This has allowed us to focus our efforts on innovation rather than maintenance."

It was not just the CI system that hit a scale limitation. As the Kibana team grew, so did the natural interdependencies between developer teams, which resulted in increased complexity when triaging and resolving problems. Without a developer-friendly self-service interface, Elastic's team members spent an increasing amount of time troubleshooting infrastructure logs across the organization to prevent developers from waiting on cross-team dependencies.

After thoroughly evaluating the market, Elastic found that Buildkite's massively scalable architecture and unique dynamic pipeline generation would allow it to create well-architected, easy-to-maintain pipelines, and also take advantage of cost savings in cloud infrastructure. Buildkite's built-in developer interface would also improve CI troubleshooting to deliver a more positive developer experience.

Elastic needed a flexible, scalable CI/CD solution that would enable them to reduce both pipeline run times and cost, while at the same time free developers to work independently.

Solution

Reducing costs and iteration cycles using Buildkite's dynamic pipelines and unlimited concurrency

"Designing the CI system we needed to meet our goals was so much easier once we moved to Buildkite," says Seeders. Elastic used Buildkite's unlimited concurrency to parallelize and scale out as much as they wanted. "This allowed us to break things up into short, small steps and use spot VMs to reduce costs," Seeders says. "Next, we worked on bandwidth optimization. At the end of the day, we were able to significantly reduce our cloud infrastructure costs."

To reduce pipeline run time, the Kibana team leveraged Buildkite's dynamic pipelines capability to create a system that automatically splits test suites across a growing number of Buildkite steps. "Previously, we had statically-defined groups of test suites that someone would have to manually rebalance when CI got too slow," says Seeders. "Now, our CI system dynamically breaks up our tests into steps based on their average historical runtime such that they all complete within our target build time. It's still running reliably, within our target, years later. This would not have been practical with our previous CI solution."

Features leveraged

Dynamic pipelines

Decrease CI/CD run times by customizing pipelines on the fly. Use the full power of your preferred programming language to generate pipeline step logic, all based on environmental scenarios encountered as the pipeline is running.

Learn more →

Pipeline annotations

Increase developer productivity by pushing feedback relevant to your team in the build UI, such as test result summaries, graphs of codebase analyses, and links to artifacts.

Learn more →

Scalable orchestration

Scale infinitely with Buildkite's best-in-class orchestration engine. Handle 100k+ connected agents from your favorite cloud provider, flexibly and efficiently.

Learn more →

Monorepo builds

Selectively build parts of your codebase depending on which components have changed. Use our official plugin, or use dynamic pipelines to implement your own build and test strategy.

Learn more →

Pipeline dashboard

Monitor, control, and visualize all your pipelines in one place with Buildkite's intuitive web app. Take action from metrics that show the health and performance of your pipelines.

Learn more →

To remove the need for developers to rely on other teams to troubleshoot CI errors, Elastic turned to Buildkite's pipeline dashboard. Out of the box, the Buildkite build page (developers' first port of call when troubleshooting) provides easy access to logs and other information that was either confusing to find or completely unavailable in Elastic's previous CI. Elastic also took advantage of Buildkite's pipeline annotations, a feature that enables teams to push customized information to the top of the build page. "Developers don't have to search or go to other teams for error information anymore," says Seeders. "It's just right there."

This improved developer experience meant that CI team members no longer spent as much time helping others track down logs and errors, freeing them to focus on projects that bring more efficiency and value to the company. Elastic reports that the migration itself was very smooth.

"We ended up not needing to re-implement a significant portion of our Jenkins pipeline files because the functionality that we used to accomplish via extra Groovy utilities came built-in with Buildkite pipelines," says Seeders. "Buildkite just works the way you want it to in so many scenarios."

Next, Elastic is working to further reduce run times by selectively building and testing only the parts of their codebase that changed, or were affected by a change, rather than building and testing the full monorepo on every pull request. Using Buildkite dynamic pipelines, an initial pipeline can triage further workflows at runtime, only triggering the necessary pipelines, jobs, or commands specific that build or change (rather than running through a series of preconfigured jobs that may not be applicable).

> **"Buildkite provided the capabilities we needed to easily meet the requirements of the business, and so the age-old choice of build-versus-buy was an easy one. This has allowed us to focus our efforts on innovation rather than maintenance."**
>
> — **Brian Seeders,**
> Staff Engineer,
> Elastic Kibana

Outcome

Supporting fast, cost-effective CI with Buildkite

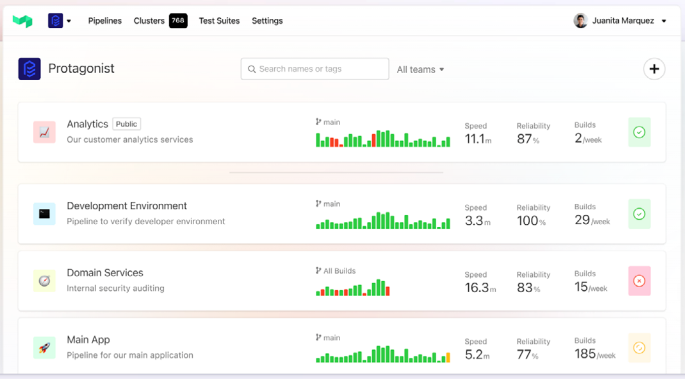
"The impact on turnaround time for developers has been huge," reports Seeders, "We reduced the wait time on the pipeline that runs pull requests for Kibana from 3 hours to 55 minutes. That's the difference between pushing a change and finding out if it's good tomorrow, and pushing a change and being able to continue your work today."

"From the engineers' perspective, the improvement was immediate," says Seeders. "We've only gotten positive feedback about how much easier and faster the new system is."

Sign up for free, and start liking CI/CD again

Every new signup gets a free 30-day trial of the Pro plan

Get started →   Contact us



Analytics Public	Our customer analytics services	main	Speed 11.1m	Reliability 87%	Builds 2/week
Development Environment	Pipeline to verify developer environment	main	Speed 3.3m	Reliability 100%	Builds 29/week
Domain Services	Internal security auditing	All Builds	Speed 16.3m	Reliability 83%	Builds 15/week
Main App	Pipeline for our main application	main	Speed 5.2m	Reliability 77%	Builds 185/week