CloudBees

CASE STUDY

Balancing Our CI/CD Approach to Move Our Digital Transformation Forward

Industry
Finance

Geography
North America

Product
CloudBees CI

"Thanks to CloudBees CI, we have a stable, maintainable and reliable CI/CD pipeline. Our deployments are faster and smoother and our feature engineering teams are all on the same page. Our developers enjoy a better quality of life and are free to define new processes."

Senior Software Engineer
Leading Provider of SaaS-based M&A Technology

Contributed by Senior Software Engineer, Leading Provider of SaaS-based M&A Technology

I work for a leading Software as a Service (SaaS) provider for the mergers and acquisitions (M&A) industry, and we specialize in M&A technology. The M&A field is wrought with processes that have to be fulfilled to the finest detail in order to make sure deals are complete. Our company has modernized these processes by providing a SaaS platform that helps companies manage the end-to-end due diligence process. This allows our customers to share and verify documents on the cloud to expedite the deal lifecycle and improve the quality of life for everyone involved in getting the deal done.

New Operational Approaches

I'm a software engineer, but I was hired to bridge the gap between our engineering teams and our application support staff. It was an experimental position. The idea was that I would sit down with feature engineers, learn about the intricacies of their applications and efforts and relay this information to application support so they could get ahead in shaping out the support they'd need to provide for new features being delivered to our customers.

The experiment went well, and its initial success led to the creation of a site reliability engineering (SRE) team, which I became a part of. The core objectives of the SRE team were to make sure our microservices were ready for production by adhering to readiness standards that were developed by the team. This helped us make sure that what we delivered to production was monitored for performance and anomalous behavior(s). This made application insights easily accessible, which in turn gave us the ability to get ahead of production support issues. We also helped provide tools that reduced the toil of repeat production support efforts. All of this has continued to coalesce into less time spent in production support and more time in developing and delivering valuable features to our customers.

"We could not have accomplished this with the open source version of Jenkins alone. CloudBees CI has given us the way forward."

– Senior Software Engineer

At about this time, we started talking about overhauling our entire CI/CD process. I had worked with our existing CI/CD platform in the past, so I volunteered for the job without any hesitation. It turned out to be bigger in scope and much more involved than I imagined.

“We were able to use CloudBees CI as a management layer that allowed us to enforce standards and practices as needed while giving teams the freedom to move around within their individual environments.”

– Senior Software Engineer

Balancing Scalability, and Maintainability

Our CI/CD process really needed an overhaul at that point. Initially, our feature engineering teams were doing all their builds on a single Jenkins server, but that wasn’t scalable, and it was quite fragile. The server was never up to date and dependencies were clashing because all the different plugins and build tools amassed on the build server over time.

To solve our scalability problem, we decided to give each team a separate Jenkins server based on the same base virtual machine (VM) image. We automated the process and spun up VMs that allowed them to isolate their operations. This introduced a maintainability problem where the servers we distributed started to drift apart. Everyone started adding plugins and using different tools and we couldn’t keep track of the changes they made to their CI/CD platforms. We now had a distributed maintainability problem on our hands. A single server wasn’t scalable, multiple servers weren’t maintainable and our plugin use was inconsistent. We needed a better balance.

Striking this balance between scalability and maintainability was key in driving our CI/CD into the modern DevOps world. We agreed on CloudBees CI as the tool that could fulfill this balance. CloudBees is the premier sponsor of the Jenkins project and they provided a service that allowed us to give every team their own server, on top of a centralized management plan. With CloudBees CI, we are now scalable and maintainable. Once the tool was in place, it was a matter of deciding on common processes and freedoms to give the teams to optimize our CI/CD approach. This would involve a mass migration from the existing approach, which had its own unique series of challenges.

Reshaping Our CI/CD Approach

We started by creating a common CI/CD framework that we could distribute to all of our engineering teams, along with a team server. We shaped this framework by defining the common steps of our software development lifecycle (SDLC) and identifying all the nuances of the technology stacks we were using, including Spring Boot, Node.js and React. Once completed, we distributed a copy of the CI/CD framework to every feature team. With this framework in the teams’ hands, they could then build on it to customize their CI and CD.

We got pushback immediately. Either the teams didn’t like how we chose to implement something or they had no interest in owning a copy of the framework and developing on it. But the biggest complication was our own continuous integration of this CI/CD framework. Now every time we had a new process in place, we’d have to work with every team to update their frameworks in order to make sure they got the latest changes. Fortunately, we realized this early on and it wasn’t too late to rethink our approach.

We decided instead of a CI/CD framework to bootstrap the teams, we’d go with a CI/CD common library that would act as an internal open source project and would be overseen by the DevOps team. This way we could implement and standardize new elements of our SDLC, feed our ongoing efforts of enforcing production readiness, while also giving the developers the ability to contribute other new features that other teams could benefit from. This was a much more welcomed approach. I was now convinced that we were on the right track, but we had a lot more work to do.

Providing CI/CD and DevOps as a Service

By the time I came on board, our company had already purchased CloudBees CI as our CI/CD platform. We were able to use CloudBees CI as a management layer that allowed us to enforce standards and practices as needed while giving teams the freedom to move around within their individual environments.

With CloudBees CI, we had the capability to implement a CI/CD approach that was scalable, maintainable and flexible. It provided the tools we needed to enforce standards and practices – and roll out a core set of plugins – while giving the developers the correct level of freedoms they need to be successful.

With CloudBees CI, we had the flexibility we needed to rethink our approach and pivot to a workflow that better reflected the needs of our developers.

Right now, we have 20 Jenkins team controllers all running in an Azure Kubernetes cluster. We always have the right, up-to-date plugins installed and our dependencies and security patches are always up to date. Everyone is on the same page, but at the same time, there's enough flexibility baked into our CI/CD approach to allow teams to innovate.

Our deploy times have roughly been cut in half, and our ability to deploy hotfixes and recover from incidents has gone down from a factor of (sometimes) hours to minutes.

We have created a core DevOps management team that acts as the fabric of our CI/CD platform, but any developer in the company can contribute to its improvement. CloudBees CI is the hub, and our coders are the spokes on a wheel moving us forward.

I now manage and lead DevOps and CI/CD. One of my responsibilities is affecting a shift in our mindset about DevOps. Instead of solely focusing on maintaining technology, we established our new mandate as being an internal service that treats our feature teams as customers. It is an approach that echoes my work as a member of the application support team, but now, the feature engineers sit down with me and not the other way around.

CloudBees CI runs on top of Kubernetes, which keeps our platform available and scales when needed. It allows us to spin up resources quickly in a repeatable fashion and maintains a desired state. It takes care of itself and manages our maintainability problems. This means that we can focus on larger projects and innovation and not have to focus on keeping our CI/CD platform fed and cared for.

Now, we have been able to turn our attention to real business problems that we solve using a DevOps approach rather than being the gatekeepers and the maintainers of legacy platforms. We want to continue to improve the lives of developers and make engineering the best it can be.

The Perfect Balance

Thanks to CloudBees CI, we have a stable, maintainable and reliable CI/CD pipeline. Our deployments are faster and smoother and our feature engineering teams are all on the same page. Our developers enjoy a better quality of life and are free to define new processes.

Ultimately, the developers care that the features they're developing are getting into production smoothly and reliably. Our CI/CD is the lifeblood making that happen. Our new CI/CD codebase is not only robust, but easily extensible. If we find a tool that isn't part of our existing environment, we can fold it into our CI/CD workflow with minimal fuss.

When it comes to DevOps, it's all about improving the quality of life for developers through operations that are automated, well maintained and well understood. Our feature teams have the flexibility and the autonomy they need to develop world-class features for our customers, all while being aligned with standardization that ensures these features are delivered reliably and production-ready. We could not have accomplished this with the open source version of Jenkins alone. CloudBees CI has given us the way forward.

"Our deploy times have roughly been cut in half, and our ability to deploy hotfixes and recover from incidents has gone down from a factor of (sometimes) hours to minutes."

– Senior Software Engineer

CloudBees CI is built on top of Jenkins, an independent community project. Read more about Jenkins at: www.cloudbees.com/jenkins/about

0122v00

CloudBees

CloudBees, Inc.
4 North Second Street | Suite 1270
San Jose, CA 95113
United States
www.cloudbees.com
info@cloudbees.com