CloudBees.

CASE STUDY

Up Close and Personal with CloudBees CD's Enterprise-Grade Orchestration Management Solution

CREDO mobile

Industry
Telecommunications

Geography
North America

Product
CloudBees CD

"CloudBees CD has given CREDO the freedom to increase our velocity and adjust our trajectory as needed and that is worth every penny."

Dan Pritzl
Former Systems Architect
CREDO Mobile

Contributed by Dan Pritzl, Former Systems Architect, CREDO Mobile

CREDO Mobile is a wireless service provider that believes in making a difference. Founded in 1985, we are committed to progressive social change and have donated more than $92 million to humanitarian relief efforts and social justice and human rights campaigns. Based in San Francisco, we're a small company and we like to keep our business personal, which applies equally to our customers and our teams.

I'm CREDO Mobile's systems architect and work with a team of 12 engineers. I've been with the company for four years and I've stayed this long because no day is the same. When I accepted the job, our hiring manager promised me, "Dude, you'll never get bored." And he was right.

An Enterprise-Grade Solution for a Small Company

Although we're a small company, we're not afraid to invest in the same tools used by businesses far bigger than ours. When it was time to simplify our application orchestration process, we went with an enterprise-grade CD platform designed for massive teams that require both scale and speed. It might have seemed like overkill at the time, but it was exactly what we needed.

My work centers around infrastructure automation and providing systems that speed up the delivery process for our software engineers developing our web platform. Our issue has never been scale, but we have a ton of legacy complexity. We are using newer web technologies, having recently migrated our microservices from container-based solutions to an AWS Lambda-based serverless model. At the same time, though, we have to maintain a fleet of Linux-based servers for backend services that aren't compatible with a serverless environment, as well as a legacy .NET ecosystem running on older-school VM-based Windows servers.

I wear a lot of hats and do many things, but my one responsibility is to enable the rest of my peers to do their jobs better. By simplifying their workflows, I can free them to focus on providing the best quality solutions to CREDO and our customers. Our data management folks can focus on SQL, our app engineers can focus on Go, Python, or .NET, QA can focus on testing/feature delivery and so on.

Our Legacy Issues Impacted Our Velocity

As a result of these legacy issues, our attempts at CD pipelining were anything but streamlined, which negatively impacted our velocity. To keep up with the complex demands of the mobile industry, we need coordinated deployments of our storefront, microservices and .NET components. Satisfying dependencies across all these fronts is complex and we were doing triple duty by running three different delivery systems. Our .NET and SQL Server deployments are governed by Octopus Deploy and TeamCity while our storefront stack is managed by a mix of Jenkins® pipelines and manual processes. The serverless stuff is the most CI/CD complete, has been built 100% on Jenkins, but is still a discrete workflow requiring cognitive effort for coordination with the aforementioned stacks. Engineers were burning out keeping up with it all.

To do everything in the correct order and validate every step and component of a deployment required a level of orchestration we weren't comfortable maintaining in Jenkins. We also had to deal with the complications of our aged-out TeamCity and Octopus Deploy tools, so the quality and frequency of our releases suffered. We didn't have adequate testing capabilities and couldn't catch all the edge cases and production issues that might adversely impact our customers.

"They gave me exactly the insight I needed, when I needed it, so once I completed the setup, CloudBees CD was bulletproof."

– Dan Pritzl,
Former Systems Architect

Going with a Trusted Name

We were already using CloudBees for paid Jenkins support. When I brought up these issues with our customer success rep, he suggested we try CloudBees CD orchestration platform. My team was very comfortable with Jenkins, but we don't have the resources required to code and maintain the rigorous quality-control mechanism we needed in Jenkins alone.

We had already started to look at potential solutions, but CloudBees CD stood out. It offered orchestration and delivery in an elegant package that gave my engineers the choice of defining workflows in a point-and-click GUI or via configuration code. We all know it's impossible to please everyone, but on small, skill-diverse teams, we find it's worth the effort of accommodating, as much as is practical, the preferences of all your team members. After all, happy engineers are productive engineers.

Six months ago, I installed a demo of CloudBees CD in our production environment and took it for a test drive. I figured out a lot of the installation and setup independently, so the CloudBees support team didn't have to walk me through much of the process. The times I did ask for their help, they were fantastic. They gave me exactly the insight I needed, when I needed it, so once I completed the setup, CloudBees CD was bulletproof.

Flattening the Learning Curve with CloudBees CD

As with any tool, there is a learning curve. CloudBees' support documentation is quite thorough, so admins will likely be able to self-serve in most cases, but users may be a different story. My team needed some training when we moved from the demo to the implementation phase of CloudBees, but it has been mostly painless and simple. Thanks to the robust role-based permissions governance of CloudBees CS, I can simply hand a URL to a given teammate, they log in and only get to see the things that are pertinent to their work. Sure, they had to memorize some new workflows, but the UI is simple, uses industry standard iconography, and most folks didn't have trouble finding their way around. For the most part, I targeted engineers with personalized training on the elements of CloudBees CD platform and UI components they needed to do their work. They had an easy time understanding the basic concepts because CloudBees CD's visual presentation sets users up for success right out of the gate.

Despite its ease of use once everything is wired up, I found the steepest part of the CloudBees CD learning curve was understanding the platform's nomenclature. Terms like "pipeline," "application," and "release" have CloudBees-specific meanings which are just different enough from some other automation systems to be a tad confusing, but once the core product concepts make sense, it's easy to point folks in the right direction. And, as I mentioned, CloudBees CD gives my engineers a choice. Those who aren't comfortable with or simply don't like writing configuration code to define their pipeline can click and drag to create configuration files, export them in YAML, and commit them to version control without writing a single line of code.

CloudBees CD is a robust platform that makes it easy for those who have a better grounding in its use to simplify the work of those just learning it. I use CloudBees CD's permissions modeling to give my engineers access to the appropriate tools based on their function within the team. As a bonus, this approach prevents engineers from stepping on each other's toes by only allocating resources to a project precisely the moment they're needed.

The service catalog feature allows me to predefine sets of workflows and objects that make it easy to onboard new team members. When a new hire comes in, I can ask them to log in and access everything they need to set up their development environment and pipelines. I tell them what to click and CloudBees CD builds everything they need to get started in about five minutes.

"CloudBees CD has simplified my life. I no longer have to spend hours or even days troubleshooting build issues."

– Dan Pritzl,
Former Systems Architect

Analytics Keep us Honest

CloudBees CD has simplified my life. I no longer have to spend hours or even days troubleshooting build issues. That process had a domino effect on other engineers. When we encountered a problem, we often had to divert several highly-paid engineers to interpret error messages, parse logs, or otherwise figure it out for other engineers who aren't as experienced with troubleshooting "DevOps problems." CloudBees CD does an admirable job of surfacing the exact reasons for failure quickly, provides options for dealing with it, and effectively turned an expensive context switch into a self-service script.

CloudBees CD's analytics tools give us qualitative data about the effectiveness of our processes and systems. Half my work is building and orchestrating an application, but the other part is reporting and analytics. CloudBees CD provides detailed feedback on what breaks, how it breaks and why it breaks. We can spot recurring problems and focus our energies there.

Because the process is automated, there are no egos involved. Nobody wants to admit that they made a mistake, but we're all human. CloudBees CD analytics help us keep each other honest without wagging fingers or rebuilding the proverbial walls between teams that we have worked so hard to demolish. And we can celebrate successes together, too—our analyses have shown that our engineers and our code fail far less than we thought we did.

We also use these analytics tools to provide feedback to our product and executive stakeholders so they can see where projects are struggling and take data-driven, informed action.

Creating an Organizational Structure with Templates

One of CloudBees CD's best features is templating. From environment definitions to workflows to individual applications, everything in CloudBees CD is an object that can be reused. Template-based approaches allow us to create a basic framework and organizational structure with the flexibility for users to fill in the details as needed. This standardization and governance means that we no longer run into random edge case errors that we didn't see during the deployment, QA or staging phases.

Templating also gives us standardized interfaces between our different polyglot systems, like Octopus Deploy and TeamCity supplying its output a downstream Jenkins pipeline execution. Instead of defining (often obtuse) approval gates or handoffs between systems in Groovy or ad hoc shell commands during a deployment, we set up a standardized template object that routes output from one system to another system's input. There's no guesswork or assuming how data is flowing because it travels across a predefined pipeline.

Although these templates are very rigid, they are not set in stone. We define the change control process as code that anyone can alter. We've made templates changeable because everyone has a stake in the deliverability and success of a product. Making the templates accessible and alterable in a collaborative, peer-reviewed process helps us evolve and embrace change in a balanced, sustainable way.

A Worthwhile Investment

CloudBees CD is an enterprise-grade pipeline management tool that is equally suited to smaller companies like CREDO. At first glance, the price and learning curve of the platform can seem steep when compared to solutions scaled to smaller businesses. But it offers great value, especially when considering the alternative: hiring another qualified engineer to handle everything that CloudBees CD automates.

CloudBees CD is a flexible platform that works the way you do. You can set it and forget it out of the box, or if you're programmatically inclined, you can bend it to your will. You can also rewrite CloudBees CD plugins to your heart's content without violating your end-user license or terms of use. CloudBees CD has given CREDO the freedom to increase our velocity and adjust our trajectory as needed and that is worth every penny.

"CloudBees CD does an admirable job of surfacing the exact reasons for failure quickly, provides options for dealing with it, and effectively turned an expensive context switch into a self-service script."

– Dan Pritzl,
Former Systems Architect

CloudBees.

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