CASE STUDY

Transformation and Automation: Building a DevOps Culture at Vail Systems

Industry
Computer Software

Geography
North America

Product
CloudBees CI

"CloudBees has accelerated how we push out dependencies. Instead of waiting days for teams to update their code, the process now takes minutes or hours."

Daniel Lenar
DevOps Architect
Vail Systems

Contributed by Daniel Lenar, DevOps Architect, Vail Systems

DevOps engineers need to balance meaningful work and toil to stay happy. They prefer challenging tasks like R&D, platform improvements and coding new functions to managerial chores like managing infrastructure, scheduling builds and emailing other teams when developers commit code. Sometimes the boring, repetitive tasks can consume far too much of engineers' time. When they're mired in low-value work, productivity and workplace morale suffer, slowing down release cycles and resulting in substandard code. That's not a recipe for success. It's a cycle of distress.

Vail Systems was headed in that direction. We moved from a traditional workflow to a CI/CD pipeline based on Jenkins a couple of years ago. Adopting this agile methodology transformed the way we roll out new products and update existing ones, but two years into our DevOps journey, we hit a wall. We needed an enterprise-proven solution that could grow, scale and accelerate with us.

Thirty Years of Enhancing Voice Communications

Founded in 1991, Vail Systems has spent the last 30 years providing technology that enhances voice communication. Our founder and CEO, Jim Whiteley, believed that "computers are the only invention that extends the human mind" and looked for ways to rethink how we talk to one another.

We offer cloud-based telephony software that removes obstacles that can lead to bad caller experiences. We streamline conversations using in-app calling, automated authentication, enhanced call treatment and voice clarity, context-driven user experiences, speech analytics, customer dashboards and real-time quality feedback.

I joined Vail Systems as a software engineer in 2013. Back then, we placed little importance on automation and our legacy workflow was mostly manual. I spent my time writing C++ code, but I assumed a leadership role in the transformation team when we initially switched to Jenkins.

Baby Steps: Building a DevOps Culture with Open Source

About five years ago, Jim watched a video about a company called Mesosphere that offered a DevOps solution to help speed up a major software release. He liked what he saw and asked us to implement the approach here at Vail. It was

my job to get our teams on the DevOps bandwagon. The battle turned out to be cultural, not technological.

Developers may code cutting-edge apps, but they can be conservative when it comes to changing the way they work. I had buy-in at the top, but I needed to win over our coders. So, I tackled the problem team by team. I arranged small-scale demos and trials, educated our engineers about best practices and current industry standards and onboarded one group at a time.

We automated some of the more routine tasks and reduced human error right away, but we encountered a problem when it came to maintenance. We have dozens of teams at Vail developing different applications, and over time, it became common practice to upload their code to a single Jenkins instance. It turned into a maintenance nightmare. If team X wanted to add their code to the central build, they had to wait for teams Y and Z to complete jobs already in the pipeline. The teams would then have to coordinate with each other to find out when they were finished with the jobs.

Not only did this slow down the development process, but people were hesitant to make any big changes during business hours. We started asking ops folks to do some things outside of normal business hours, which encroached on their personal time. This, alongside manual tasks, contributed to a slump in morale. We knew there had to be a better, more scalable approach that would give teams the flexibility they need but with a standardized approach.

CloudBees: Enterprise Scale and Happy Developers

Six months ago, we switched to CloudBees to eliminate this bottleneck, standardize across our teams and improve our developers' lives.

CloudBees has enabled us to rapidly scale our CI/CD environment, improve manageability, scalability, security and support, and includes CloudBees Jenkins Operations Center (CJOC), which provides a centralized single-pane-of-glass management console.

CloudBees was the logical choice. Because we were already familiar with Jenkins, making the move required a slight shift instead of a massive change. We were sticking with what we knew but now getting extra capabilities that streamlined our CI/CD pipeline, standardize across teams and get enterprise-level support from experts instead of Google. All we had to do was migrate our existing configurations to this new environment. There was barely a hiccup.

Reigning in Chaos with Managed Controllers and Jenkinsfile Templates

We started using CloudBees' Managed controllers right away. It was the killer feature we needed to reign in some of the chaos. Every team now works with a separate Jenkins instance and no longer has to compete with other teams for resources. This approach simplifies how we administer the upgrade cycle and reduces the possibility of a buggy job crashing an entire app. Every team gets the tools they need, and no single team can halt our CI/CD pipeline should an issue emerge.

The other immediate benefit was the ability to create Jenkinsfile templates in CloudBees. We can set up a catalog of templates containing all the plugins our developers need to create a CI/CD pipeline. Instead of manually setting up every pipeline, they can grab an off-the-shelf template and start coding right away.

This "rinse and repeat" approach allows our engineers to focus on programming and developing applications. It's like getting into a car and driving away instead of having to build the vehicle from scratch. This shifts their attention away from the minutiae and moves their focus to the fun stuff – writing cool code.

Managing Application Dependencies

Our coders also use CloudBees' event messaging feature to manage application dependencies. When a team releases dependency A, CloudBees automatically alerts the team responsible for dependency B to update their code. When dependency B is built, CloudBees then signals team C, and so on down the line. This automation reduces toil and makes life easier for multiple teams.

In the past, we sent out email notifications whenever a new build was ready. We often found ourselves in a situation where team A and team C were both in the loop, but nobody on team B had read the email. Projects often stalled because somebody didn't get the memo.

CloudBees has accelerated how we push out dependencies. Instead of waiting days for teams to update their code, the process now takes minutes or hours. Improved dependency management has also led to increased cross-team collaboration. Event messaging means everybody knows what they're supposed to do next. Instead of spending time scheduling and coordinating, our engineering teams work with one another to build new features and write better code.

Striking the Right Balance

While we started our DevOps journey with Mesosphere, we now run CloudBees on top of Kubernetes and use the Kubernetes plugin to run dynamic agents. This allows us to run multiple parallel builds, distribute the builds across our Kubernetes cluster and be efficient with our resources.

Throughout this process, I've come to realize that you can't expect your programmers to learn everything about DevOps and CI/CD pipelines. You don't need everyone on your team to write Dockerfiles or containerize applications. Everyone is happiest and most productive when they are limited to working with the tools they need to succeed.

I want my developers to continue their increased focus on coding. There's no need to overload them by asking them to learn CloudBees from A to Z. Instead, the best way forward is to strike the right balance in everything you do. We've barely scratched the surface with CloudBees. My DevOps engineers are still exploring its many features and looking for ways to move everything to standardized templates. But we've already accelerated our development timelines and the automation we've adopted has reduced toil.

Adopting an enterprise CI/CD solution has transformed the work environment at Vail Systems. Our people love what they're doing and are building better software faster. We are more competitive and better equipped to create prod-ucts that remove obstacles to communication and extend the human mind.

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