



CASE STUDY

Embracing Developer Autonomy and Building a DevOps Culture with CloudBees



Industry

Oil and Energy

Geography

Spain

Product

CloudBees CI

“With CloudBees, we can deploy new code every two weeks, or sooner, should the need arise. And we can do it with less training every day as our developers are more mature with cloud native architectures, CI/CD, CloudBees and CEPSA DevOps framework.”

Dario Fernández Barrio

Former Lead DevOps
Cepsa

Contributed by Dario Fernández Barrio, Former Lead DevOps, Cepsa

DevOps is a journey, not a destination. You can talk all you want about containerized application, serverless patterns and CI/CD pipelines, but the tools and technology are as important as the philosophy itself. DevOps is best described as an approach to “Velocity”, a approach that speeds the delivery of applications and services in a faster, more streamlined, predictable way than traditional software development cycles and provides more autonomy to coders.

At Cepsa we had in place a solid DevOps pipeline for our on-prem containerized docker workloads. But we started our journey to serious cloud production workloads in 2018 and began to build a mature DevOps culture in 2019/2020. We’ve come a long way in this journey but have further to go.

Cepsa is a multinational oil and gas company headquartered in Madrid, Spain. We operate in several European countries, as well as Algeria, Canada, Colombia, Morocco and Brazil. As our DevOps lead, the “DevOps” team is responsible for developing our company-wide DevOps strategy and software delivery model. I oversee a six-person team of senior solutions software architects that design, develop, implement, install and deploy the platform and tools our developers use.

We are a Platform Engineering Team responsible of building the CI/CD & dev platform that allows our developers to quickly deploy apps using a high-visibility workflow, which integrates with cloud and on-prem systems and applications.

Our Developers Lacked Autonomy and a Predictable CI/CD Platform

When we started the journey to the cloud, we were in a common place a lot of organizations that come from a more classic approach find themselves in. Our developers lacked autonomy and the ticket based layered teams approach does not simply negate most of the benefits of leveraging cloud native services.

We didn’t have a predictable CI/CD pipeline or software delivery model for our cloud products. Even the most straightforward deployment took a week or two of manual work sometimes based on trial and error.



143x
improvement in
deployments per
month

Our software delivery workflow was so complex that our devs couldn't handle deployments on their own. They weren't comfortable with the process and lacked the maturity and authority to do the work themselves.

Sometimes deployments in QA environments require refactor of baseline architecture with the help from a team of "super experts" who had an unending queue of requests and were constantly falling behind. The delivery pace was slow and the manual approach to quality assurance (QA) multiplied the potential for human error, leading to bugs and security issues.

Building Enterprise CI/CD Pipelines with CloudBees

We needed to streamline our delivery workflow and reduce the time between deployments. I was familiar with CloudBees from my previous job and recognized it as a flexible, low-maintenance solution that allowed us to set up CI/CD pipelines that needed very little management. I presented it to my colleagues as the best tool to build a stable and standardized development platform.

We did our due diligence and ran proofs of concept with other industry-leading solutions, but in the end, CloudBees won out. It offers an enterprise-grade centrally managed and monitored Jenkins implementation that provides a single source of truth, whether you're talking plugin libraries or analytics.

Open-source Jenkins was an operational nightmare that didn't fit well with a large organization like ours, with dozens of teams working on different platforms. Moving to CloudBees eliminated the variables introduced by the open-source approach. Everyone shares a standard pipeline across the entire organization and the teams can create their own pipelines as they see fit. We no longer have teams using different versions of Jenkins or outdated plugin libraries because updates are automated and pushed out to all our people.

CloudBees defines everything – including Jenkins – as code. We can give each dev team a managed Jenkins controller just by adding two yaml files in a central cloud-based Kubernetes Docker container that incorporates the Jenkins plugins and source code they need. There's no need for a separate team to author and manage pipelines, which drastically reduces deployment times.

Embedding Senior Architects to Build Up a DevOps Culture

To aid with the transition to CloudBees, we implemented a two-pronged strategy. We had to choose the right tool and then change our culture to take full advantage of its features. Cepsa uses a number of QA tools so there was some hesitation.



The only way we could do this was to give our developers more autonomy and training was key to making them more independent and transitioning them to CloudBees.

We embedded senior solutions architects from the architectural DevOps team into production teams to guide our devs through the transition. These senior architects taught developers how to use DevOps tooling and Jenkins pipelines, generate QA jobs, IaC, cloud native basics and tools. They also boosted security, enhanced compliance and reduced operational risks by providing written instructions and daily support as they are part of the developer teams.

By embedding our senior solutions architects into production teams, we iterated faster. We had immediate and pertinent feedback that allowed us to refine our CloudBees workflow as we were rolling it out. At the same time, this approach eliminated the silos that emerge when separate teams work in silos.

Politics and Culture are as Important as Tools

Part of the success of a DevOps model is the technical implementation and the other is the political support to break down the traditional silos and the understandable reluctance of middle-management in organizations.

For this reason, I want to emphasize that none of this would have been possible without the support of our CDO and CIO and, above all, the effort and commitment of our developers to learn and improve their skills every day.

Standardizing our DevOps workflow across our cloud development allowed us to build a DevOps culture from the ground up. Our DevOps model is not yet fully mature, but it is evolving to meet our present and future needs.

Faster Sprints with More Visibility and Added Value

Since implementing CloudBees, we accelerated our development and deployment cycle. Our developers now offer more value to the business by deploying new features and functionalities faster than ever, with more visibility into the process. Before we embark on a sprint, we consult with relevant stakeholders within Cepsa to confirm what features and functionalities they need. We then code these changes and deploy them in our QA environment to ensure that they don't break previous versions of our apps.

We can confidently deploy new features now into our production environment because they have been thoroughly tested by our developers and vetted by the Cepsa stakeholders who will be using them.

With CloudBees, we can deploy new code every two weeks, or sooner, should the need arise. And we can do it with less training every day as our developers are more mature every day with cloud native architectures, CI/CD, CloudBees and CEPESA DevOps framework.

Autonomy, Ownership and an Evolving DevOps Strategy

Since we rolled out CloudBees in April 2020, we have created more than 50 managed controllers and have gone from an average of seven deployments a month to 1,000 deployments a month – a 143X improvement. These deployments are pushed into production by our developers, who now have complete autonomy over the CI/CD pipeline, and no longer spend endless days emailing super experts to trigger builds.

We have also built a custom analytics dashboard that leverages the CloudBees API. It tracks metrics like deployment time, deployment frequency, product change frequency and pipeline failures. It is an ad hoc solution for now, but we hope to formalize this dashboard to provide external metrics and deliver value to the rest of the business and product owners. When it is finalized, this dashboard will be the last word on product stability, quality, maturity and release frequency.

In the year since we deployed our CI/CD pipeline, CloudBees has proven to be a partner, not just another service provider. Our model has just begun, we have a lot to improve but I think we settled a good foundation by avoiding the ticket-layered-silo ways in our cloud strategy. So the road is far from complete, but CloudBees has given us a predictable CI/CD environment that increases developers' autonomy and allows them to deploy code to production faster. CloudBees also will help us move our on-premise resources to the public cloud. It is the glue that holds together our production environment and a foundation of Cepsa's evolving DevOps strategy.

