

# CloudBees Feature Flags

Enterprise feature flag management

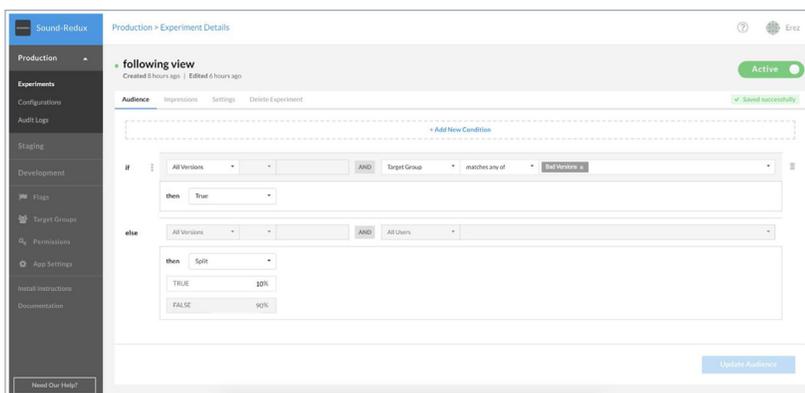
Feature flagging is a method to show or hide specific software features at runtime for a defined audience segment. Their usage was pioneered by leaders in software development, such as Facebook, Uber and Netflix. Due to their clear benefits in risk mitigation, developer productivity and increased feature quality (via shortened user feedback loops), the use of feature flags by developers is rapidly growing. In fact, a recent study conducted with CloudBees and Atlassian found that 95% of developers are already using feature flags to control their code, and 81% estimated that their feature flag usage would increase in the next year. However, for most organizations, scaling management of those flags across development teams and non-technical employees proves to be difficult over time.

One of the primary reasons for this is most development teams are using “homegrown” solutions to manage their feature flags. The ever-increasing number of flags and complexity of the use cases in which feature flags are used is simply too difficult to manage at scale for most organizations. The consequence is time-consuming,

manual management by various developers who aren’t necessarily following the same runbook, leaving lots of room for human error. And as feature flags continue to be further integrated across CI/CD toolchains, the complexity of management and dependencies grow even further.

CloudBees Feature Flags is an enterprise-ready feature flag management solution available for deployment via SaaS, with robust deployment rules, audience segmentation and a testing framework for controlling which audiences are exposed to what features. With CloudBees Feature Flags, you can rollout and rollback features, test and target features for specific customer segments for fast feedback and use gradual rollout and rollback capabilities to sidestep the danger of “big bang” or “big bust” deployments. CloudBees Feature Flags speeds the pace of development by enabling the delivery of software changes as soon as they are code complete or even for testing a minimum viable product (MVP) within a subset of internal or external users.

## Segment Audience and Set Percentages for Gradual Rollout



## Feature Flag Usage is On the Rise

- » 81% say feature flag usage will increase over the next year.
- » 37% report the cost of in-house development of a feature flag management system is higher than purchasing a solution.
- » 32% say they encountered more complexity in building a feature flag management system than anticipated.

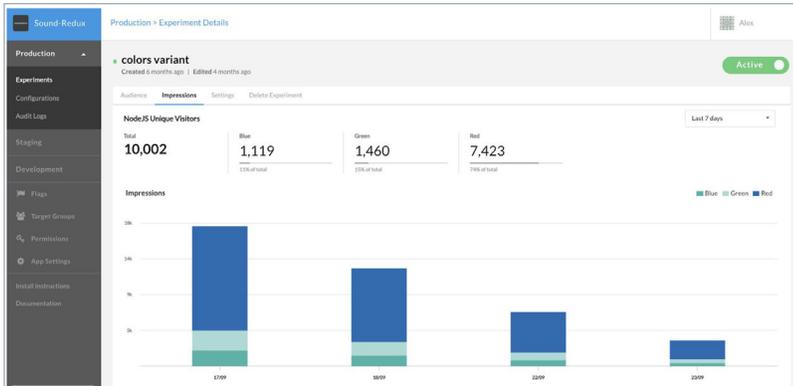
As the practice becomes pervasive, the need for enterprise-level feature flag management will become an undeniably necessary addition to the DevOps toolchain.

\* Source: [Study conducted by Rollout.io and Atlassian](#)

*“Working with CloudBees has been smoother than we could have hoped. Not only does the Testim team benefit from reduced development costs, but our customers gain access to new features much more quickly than ever before without compromising reliability. It’s a win-win.”*

Rhon Shoshani  
VP of R&D  
Testim.io

## Value Across CI/CD Pipelines



### Move Fast. Break Nothing.

CloudBees Feature Flags allow you to separate software into discrete, controllable features. By decoupling feature deployment from code release, you can tailor customer experiences in production, in real-time, with the flip of a switch. Suddenly, the business is no longer constrained by technical backlog.



**Faster Delivery** - Gain velocity through smarter CI that embraces trunk-based development. Avoid merge conflicts associated with long-lived feature branches. Merge incomplete code in an inactive state and simply turn the feature on when it's ready for prime time. Avoid delaying releases due to late-breaking features with a higher probability of being buggy.



**Safety** - Test new code safely in both pre-production and production environments, and perform canary deployments. This can be carried out with targeted user groups, such as a subset of the user base, beta users, internal employees and other user constituencies. If a bug is discovered – even days later – don't wait for a costly rollback of the whole application or a time-consuming patch and release process; just use the killswitch to turn off that feature and then repair it.



**Experimentation** - Perform A/B testing by enabling new features only for certain customers. Analyze your metrics to make a data-driven decision about what works best.



**Customization** - Turn specific features on or off for particular customers, based on preferences, location, licensing or other criteria.

Benefits	Use Cases
<p><b>Enterprise-ready scalability and governance</b></p>	<p>Advance beyond homegrown feature management with:</p> <ul style="list-style-type: none"> <li>» Centralized dashboard for flag management across features</li> <li>» Audit logs and localized flag rules</li> <li>» Robust Security with SAML / SSO Integration</li> <li>» Zero access to end user Personally Identifiable Information (PII)</li> </ul>
<p><b>Increased Developer Productivity</b></p>	<p>Workflows:</p> <ul style="list-style-type: none"> <li>» Add flags to code first and see them on the dashboard</li> <li>» Flags-as-code: Take a "GitOps" approach to log changes to feature flags in the SCM, in the same fashion as code and configuration changes in applications</li> <li>» Set only one flag for all development languages</li> </ul>
<p><b>Shortened feature feedback loops</b></p>	<p>Gradually release features to subsets of users based on predefined criteria (country, language, version, etc.) and use your current analytics tools and teams to ensure the features will be well-received before full release</p>

## Schedule a Demo Today

Contact us: [info@cloudbees.com](mailto:info@cloudbees.com)