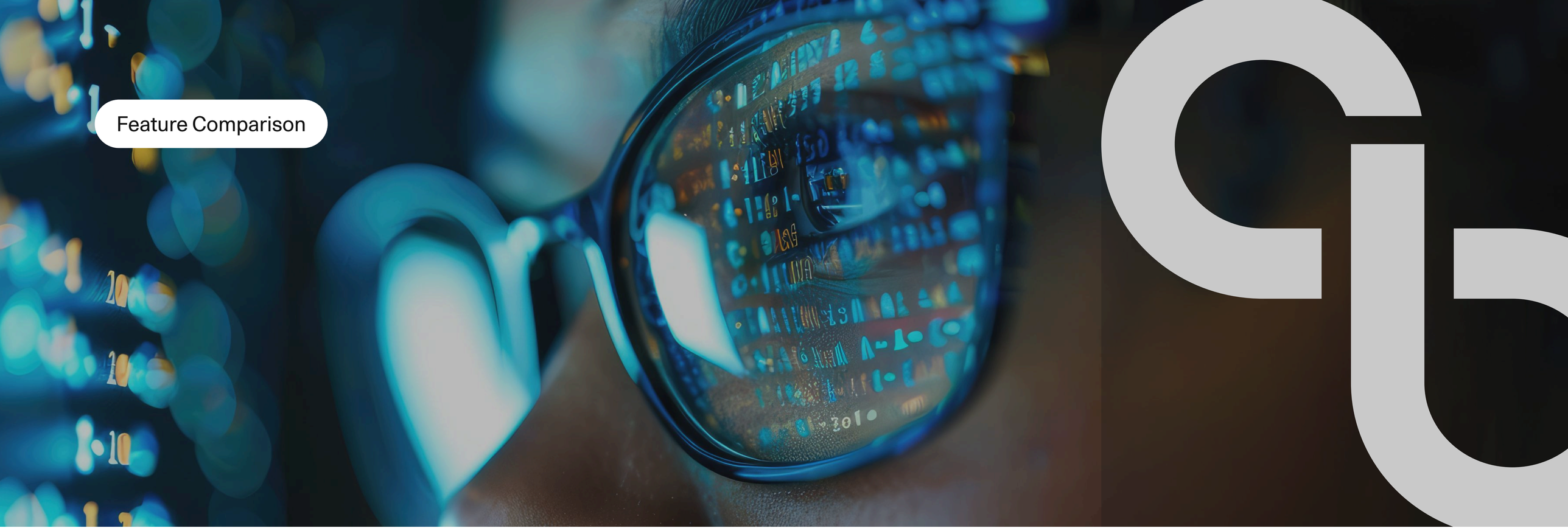


CloudBees CI: Enhancing Jenkins for mission-critical environments

Managing enterprise-wide software delivery has always been a balancing act between stability and speed. With the rise of agentic coding, far more code is flowing through Jenkins®, from developers and agents alike, and much of it reaches production via controllers configured and secured one at a time.

Securing each controller separately makes it hard to enforce the same security policies everywhere, and even harder to show an auditor who changed what. CloudBees CI brings that control back without asking teams to leave Jenkins.

Overview of Jenkins and CloudBees CI

Jenkins

Jenkins, one of the most widely used tools in the CI/CD space, automates the process of building, testing, and deploying software. With more than 2,000 plugins, Jenkins can be extended to a wide range of tools and technologies, making it adaptable to different workflows and environments.

CloudBees CI

CloudBees CI builds on Jenkins to provide large enterprises with central management, governance, and support across all controllers. It's designed for organizations, like Salesforce and Autodesk, that rely on Jenkins but need to govern it consistently at scale.

What ungoverned Jenkins costs

Configuring and securing each controller on its own leads to costs that accumulate undetected across teams, budgets, and audit cycles, surfacing only when something goes wrong.

- Security policies drift: Controllers that start with the same settings and permissions slowly diverge, until nobody can say for certain which rules apply where.
- Fixes reach teams slowly: A security patch has to be applied controller by controller, so a vulnerability can stay open on some teams long after others have closed it.

- Unvetted plugins pile up: Teams install their own plugins and versions, causing bugs and letting code nobody has reviewed into production.
- Admin time goes to upkeep: Platform teams spend their days maintaining controllers and plugins one by one, leaving little time to set and enforce policy.
- Infrastructure runs idle: Without a single view of every controller, unused controllers and agents keep running, and nobody can see what they cost.

Key enhancements CloudBees CI brings to Jenkins

1. Granular RBAC and centralized policy

Data from our State of Code Abundance 2026 report shows that while 93% of organizations have a formal review process for AI-generated code, only 56% always enforce it.

Policies only work when they're applied everywhere. Managing role-based access control (RBAC) in a single Jenkins instance is straightforward, but it becomes complex as the number of instances grows.

CloudBees CI adds granular permission settings and centralized policy management through the operations center. Administrators can set permissions from system-wide settings to specific jobs or pipelines, assign roles based on job functions, give automation and coding agents their own scoped service accounts, and apply them across every controller at once. Security policies stay consistent across all Jenkins instances without requiring admins to maintain them individually.

2. Centralized management with operations center

Managing multiple Jenkins controllers across teams means separate configurations, plugin updates, and user management, and that decentralized approach lets inconsistencies creep in. Operations center gives you a single place to deploy and manage controllers across clusters, monitor every Jenkins instance, and apply security settings consistently, so you always know what's running and how it's configured.

3. Approved plugins and configurations from day one

Configuration as code (CasC) bundles let you define a controller's setup, plugins, and RBAC once and roll it out to every team, so all controllers start from an approved baseline. Keep bundles in version control, and you have an attributable record of configuration changes: who changed what, when, and under what policy.

Plus, the CloudBees Assurance Program includes a curated set of plugins tested for stability, security, and compatibility, so teams aren't pulling unvetted code into the production path.

4. Keeping governance consistent as you scale

Traditional Jenkins setups often leave some agents idle while others are overwhelmed. Shared agents let idle capacity on one controller serve another, and automated agent provisioning scales resources to match demand. High availability and horizontal scalability, scoped to your environment, keep controllers available as load grows. All of it runs under the same policies set in the operations center.

Moving to CloudBees CI without starting over

Moving from open-source Jenkins to CloudBees CI preserves the jobs and pipelines your teams already rely on. Some plugins may need to be updated to the CloudBees Assurance Program versions, so it's worth testing before you cut over. Here are the essential steps for a smooth migration:

- Back up your Jenkins data, including job configurations and pipeline scripts.
- Run the migration in a test environment first.
- Make sure the instance runs a supported Java Development Kit (JDK).
- Install the plugins your configurations need, using CloudBees Assurance Program versions where possible.
- Install Jenkins Health Advisor by CloudBees to flag out-of-date components and known vulnerabilities early.
- Transfer your jobs and import them into CloudBees CI.
- Run sample jobs to confirm they work, then monitor resource use.
- Schedule a fixed window for the final data sync to minimize disruption.

See it on your own controllers

Contact us for a personalized demo or schedule a consultation to see how central policy, RBAC, and consistent configuration would work across the Jenkins controllers your teams already run.

Jenkins® is a registered trademark of LF Charities Inc.
Read more about Jenkins at: www.cloudbees.com/jenkins/about

© 2026 CloudBees, Inc., CloudBees® and the Infinity logo® are registered trademarks of CloudBees, Inc. in the United States and may be registered in other countries. Other products or brand names may be trademarks or registered trademarks of CloudBees, Inc. or their respective holders.