



MORNINGSTAR®

CHALLENGE

Improve build processes while laying the groundwork for an organizational move to continuous delivery and a DevOps culture

SOLUTION

Use continuous integration with the CloudBees Jenkins Platform to increase automation, improve consistency, lower administrative overhead, and facilitate the adoption of continuous delivery practices and a DevOps culture

RESULTS

- >> Administration time and overhead reduced 80%
- >> Time to resolve typical support issues reduced by 70%
- >> Release cycles shortened

Morningstar: Bullish on Continuous Delivery and DevOps

Morningstar, Inc. is a leading provider of independent investment research. The company serves individual investors, financial advisors, asset managers, and retirement plan providers and sponsors and has operations in 27 countries. Morningstar provides data on more than 500,000 stocks, mutual funds, and other investment offerings, as well as real-time global market data on more than 17 million equities, indexes, futures, options, commodities and precious metals.

To enable rapid responses to client needs and market demands, Morningstar designs and builds its websites and software applications in-house. A globally distributed team of about 1,000 developers, QA specialists, project managers, system administrators and database administrators works together in a highly collaborative, agile environment. The teams build, test and deploy web and cloud-based software to support Morningstar clients and internal business processes.

Morningstar practices continuous integration (CI) with the CloudBees Jenkins Platform to improve consistency and increase automation—vital steps along the organization’s path to continuous delivery (CD) and a DevOps culture. “The CloudBees Jenkins Platform is a catalyst for our teams’ adoption of best practices,” says Fenghua Guan, head of technology operations at Morningstar. “Without enabling tools, it’s difficult to tell people to build as early and as often as possible, to perform unit tests and to analyze code quality. With process automation, it’s much easier to push those best practices through.”

Challenge: More Automation and Consistency, Less Administration and Frustration

Development teams at Morningstar started small and nimble. Individual teams tended to build and deploy tools using different methodologies. The company was able to deliver software solutions to clients quickly. However, as the firm’s client base and business grew rapidly, it became clear that common standards to improve automation, accuracy, scalability and consistency were needed. Morningstar established a team responsible for improving build and deployment processes and began using open source Jenkins.

As more and more teams relied on Jenkins, two new issues emerged. The first was a need for timely support. “We recognized that formal support for Jenkins is necessary to keep things up and running, because CI had become such a critical element for our processes,” Christopher Jones, DevOps manager at Morningstar says. The second issue was access control and ensuring that teams could not inadvertently change other teams’ build jobs. “It was difficult for us to implement permission controls without setting up a separate master for each team,” Guan says. “The administration and management overhead was huge.”

Jones had identified several opportunities for using automation to help lay the groundwork for establishing a DevOps culture. “For example, we had a lengthy process that required one team member to manually apply updates on 60 servers in the production environment. It took about six hours,” Jones explains. “We also had deployment processes that would take a long time to complete—maybe an hour, an hour and a half—and they were difficult to perform consistently because they were manual.”



Solution: Standardize on CI, Transition to CD and a DevOps Culture

Morningstar standardized on the CloudBees Jenkins Platform to expand its use of process automation and to support its adoption of CD and DevOps practices.

To consolidate 20 Jenkins masters to one and ensure only authorized personnel could access specific projects or folders, the Morningstar engineering team used the [Role-based Access Control plugin](#) to define and assign security roles, enabling them to implement an authorization strategy for team-limited access to Jenkins jobs.

This change, along with the widespread adoption of automation orchestrated with the CloudBees Jenkins Platform, has led to more consistency across diverse teams. “Whether our teams are using waterfall or agile development, the moving parts all tend to be the same,” Jones says. “Our build process includes compilation and unit testing. The packaging process then constructs all the appropriate artifacts, and the publication process gets those artifacts to where they need to go.”

On a recent greenfield project, the development team set up simple build and unit testing jobs in Jenkins. After integrating Selenium and SonarQube to automate tests and monitor code quality, they set up the Build Breaker plugin to mark builds as failed if the project’s coverage and code quality criteria are not met. Successful builds are deployed to the development server on each check-in.

The team created another Jenkins job for QA deployment. Based on roles defined using the Role-based Access Control plugin, the QA team accesses the artifacts ready for deployment, rather than interrupting developers to ask for the latest build.

“On the day we deployed to production, everyone on our team was confident because our application had been deployed approximately 100 times already to the development server, 25 times on QA and a dozen times to staging,” Avinash Kumar, lead developer for product services at Morningstar says. “We saved a lot of review time. Anyone who checked in code that caused integration test failures or merge conflicts had to fix it immediately.”

The mobile development team at Morningstar is seeing similar advantages from automation and governance supported by the CloudBees Jenkins Platform. “Jenkins serves as a single source of authority,” Brian Grant, the team’s technology lead says. “‘It worked on my machine’ is no longer an excuse. If it fails on Jenkins, you’re not done; you have to get back in there and fix it.”

Standardizing on Jenkins has increased team staffing flexibility, because consistent tooling and processes make it easier for developers to switch between teams. It has also fostered a community in which teams are ready and able to help their colleagues. “My team is not yet set up to qualify builds based on test coverage, but when we are ready I know I can walk across the hall, talk to Avinash and have it up and running in an hour,” Grant says.

Jones and his team worked with software delivery teams across Morningstar to help them automate processes and establish a DevOps culture of teamwork and shared goals. “We demonstrated how following standardized processes and practices can make the teams’ lives better by alleviating pain, anguish, suffering, late nights, early mornings—that sort of thing,” Jones notes. He adds, “For DevOps to be effective you need to break down walls between teams and have conversations about how to make the pipeline from concept to production faster, smoother and more reliable with fewer handoffs, less coordination and fewer bottlenecks.”

“Continuous delivery depends on having foundational elements like CI and best practices in place. Similarly, for DevOps to really be effective you need to break down walls between teams and have conversations about how to make the pipeline from concept to production faster, smoother, and more reliable with fewer handoffs, less coordination, and fewer bottlenecks.”

>> Christopher Jones,
DevOps Manager,
Morningstar

The team used automation to significantly reduce the time for the manual deployment process to 60 servers. “Once that process was automated, it took about 40 minutes to run. And because it’s automated, everything is infinitely more reliable,” Jones says. The 90-minute deployment processes were similarly accelerated via automation. “We worked with several teams to restructure and re-sequence processes using IBM UrbanCode Deploy. This shortened processes 20 to 30 minutes, which freed up the QA testers to focus on the much harder-to-automate activities. In addition to the speed increase, the teams recognized value in improved reproducibility and consistency.”

Jones and his team are gradually migrating other groups to automate processes using the CloudBees Jenkins Platform and a variety of other tools, as appropriate. The results speak for themselves—and are enabling a DevOps culture of teamwork and shared goals.

Results

- >> **Infrastructure administration time and overhead reduced by 80%.** “The Role-Based Access Control plugin enabled us to go from administering 20 Jenkins masters to just one—that’s huge,” Guan says. “The process is much more streamlined and efficient for our engineering team. We can easily set up a team structure with defined roles, and managing access to jobs is not a challenge anymore. We estimate that administration overhead was reduced by about 80%.”
- >> **Time to resolve typical support issues reduced by 70%.** “We see commercial support for Jenkins as a big advantage,” Guan says. “When we submit a request to the CloudBees support team we normally get a response or resolution within hours. We estimate that commercial support has reduced our typical trouble-shooting time by more than 70%. That’s impressive.”
- >> **Release cycles shortened.** “The teams that have adopted CI early are definitely seeing faster release cycles,” Guan says. “In fact, one of the additional values that Jenkins provides is maintaining those metrics for us so that we can perform data mining and analysis, going forward.”

Learn More About Morningstar:

www.global.morningstar.com

CloudBees, Inc.
2001 Gateway Place
Suite 670W
San Jose, CA 95110
United States
www.cloudbees.com
info@cloudbees.com



The CloudBees Jenkins Platform is built on top of open source Jenkins, an independent community project. Read more at: www.cloudbees.com/jenkins/about

© 2016 CloudBees, Inc. CloudBees is a registered trademark and CloudBees Jenkins Platform, CloudBees Jenkins Enterprise, CloudBees Jenkins Operations Center and DEV@cloud are trademarks of CloudBees, Inc. Other product or brand names may be trademarks or registered trademarks of their respective holders.

0116v00