



CHALLENGE

Eliminate infrastructure maintenance and devote more time to software development

SOLUTION

Implement DEV@cloud and RUN@cloud to automate Jenkins maintenance and fully leverage continuous integration

RESULTS

- >> 100% focus on software development with DEV@cloud
- >> 33% increase in developer productivity
- >> Expert support from CloudBees for Jenkins CI issues

SERVICES

- >> DEV@cloud
- >> RUN@cloud

jclouds Boosts Productivity and Focuses on Delivering Customer Value with the CloudBees Platform

jclouds provides portable compute and storage abstractions for most major cloud service providers while giving developers the freedom to use their cloud-specific features, simplifying the use of the cloud without sacrificing its power or flexibility. jclouds is open source under the Apache license and packaged as a library that you can embed in applications. “jclouds lets you focus on using the cloud, rather than troubleshooting implementations,” explains Adrian Cole founder, jclouds, and chief evangelist, Cloudsoft.

Using jclouds, developers can unlock the code they write - that would otherwise be tied into a specific cloud provider—and migrate it elsewhere. For example, a developer can use jclouds to manage workloads inside the firewall in VMware’s vCloud with the same code used for Amazon’s EC2. jclouds covers over 30 cloud providers and cloud software stacks, including Amazon, GoGrid, Ninefold, vCloud, OpenStack and Azure. jclouds’ ecosystem is equally strong, with companies such as Adobe, CloudBees, enStratus and Red Hat using jclouds to support their platforms.

Challenge

The jclouds open source community uses Jenkins CI as their platform for continuous integration (CI). While Jenkins provided jclouds developers with a useful, usable service in terms of software quality control, the corresponding provisioning and maintenance that was required to keep the on-premise servers up and running was time-consuming and tedious. It was also becoming costly to maintain the environment and developers were distracted from their main objective—writing code to satisfy the needs of the community.

Considering many volunteer hours are contributed to jclouds development, there is a premium placed on optimizing developer productivity. Any opportunity to streamline back-end IT operations and eliminate time spent on infrastructure maintenance activities means a commensurate gain in hours of actual development time for the project. As Cole explains, “We want to run lean on infrastructure support, so we can focus our resources on developing more and better offerings, and servicing our customers.” Cole’s goal was to continue to leverage the advantages of the Jenkins environment but to automate and outsource

Jenkins provisioning and maintenance to a third party. Doing so would enable the development team to focus solely on application development and also make their pro bono work more satisfying.

Solution

The jclouds team researched Platform as a Service (PaaS) offerings, feeling a PaaS solution would meet the objectives of eliminating infrastructure maintenance. One of the solutions the team explored was DEV@cloud,™ from CloudBees. During an evaluation of the DEV@cloud platform, Cole and his team were able to easily shift their Jenkins CI work to DEV@cloud and immediately saw productivity payoff in the automation of Jenkins CI infrastructure maintenance.

By utilizing DEV@cloud, Cole estimates that he and his team are saving up to four hours per week—time that had previously been spent manually maintaining and updating the Jenkins

“Since we implemented the CloudBees platform, we are able to truly leverage continuous integration in our development process...Overall, for jclouds, DEV@cloud and RUN@cloud have been game-changers.”

>> Andrew Phillips, jclouds contributor

server environment. “That marked a significant increase in productivity,” says Cole. “With the CloudBees platform, Jenkins CI is now extremely easy for us to use. We don’t have to concern ourselves with provisioning and maintaining the Jenkins servers, which has freed up valuable development time. I’m not interested in running a server farm, I’m interested in running builds . and with DEV@cloud, we are doing that.”

Andrew Phillips, veteran jclouds contributor, elaborates on his experience with Jenkins CI before and after working with the CloudBees platform. “We were using Jenkins CI, but managing it manually was a painful process,” says Phillips. “After porting our builds over to DEV@cloud, those pain points were vanquished almost instantly. Support from CloudBees has been great—technical resources are instantly available and helpful. Turn-around time has been spectacular for fixes. Furthermore, the CloudBees tech support personnel make excellent suggestions to help us resolve various development-related issues. Today, we’re building all of our release branches and all of our integration builds on DEV@cloud.”

Phillips estimates he has personally saved about 2-3 hours per week, out of the 6-8 hours he works on jclouds development. “That’s about 25% - 33% more productivity I am realizing,” says Phillips. “I am 33% more productive—and my work is now 33% more interesting!” The jclouds developers also found the Maven and Eclipse integration offered within DEV@cloud useful.” The availability of third party plugins—instantly, from within the CloudBees environment—allows the team to easily extend the platform to get the functionality we need.”

Phillips summarizes, “Since we implemented the CloudBees platform, we are able to truly leverage continuous integration in our development process. Every change we make gets deployed right away, automatically, using RUN@cloud.”™ Build time is much more predictable and our builds break a lot less. General operational failures are down to nothing and server side build failures happen rarely, if ever.

The whole environment is easier to navigate now - and all without an admin resource on the back-end. Additionally, we need only to make tiny alterations to the automated deployment process to scale up to the likes of Force.com and Elastic Beanstalk. Overall, for jclouds, DEV@cloud and RUN@cloud have been game-changers.”

Results

- >> **100% focus on software development.** With DEV@cloud, the jclouds team is saving about four hours of time per week; time that was previously spent on infrastructure maintenance. Chores such as software updates, server reboots, server sizing and patches are automatically performed for them, from within the CloudBees service. The development team now focuses 100% of their time writing and enhancing jclouds.
- >> **33% increase in developer productivity.** Phillips comments on the efficiency gains realized from implementing DEV@cloud, “My first ‘aha moment’ occurred after I set up DEV@cloud. I made a commit and DEV@cloud automatically built the latest release branch, OK’d everything and then went on to build our various demos. This used to be an incredibly complex process for us. Now, with DEV@cloud, I just get an email that effectively says ‘everything is complete.’ It’s now an incredibly efficient process that has taken tedious tasks off of our plate and lowered frustration amongst the developers because they no longer have to deal with infrastructure issues.”
- >> **Expert support from CloudBees for Jenkins CI issues.** “CloudBees does what needs to be done to resolve our issues,” says Cole. “Our ambition is not to be Jenkins experts. We are a cloud project and we want to focus on our application. If we think about developer resources available to us, I would classify CloudBees as another productivity tool for the team.”

Learn More About
jclouds Here:

www.jclouds.org

CloudBees, Inc.

400 TradeCenter, Suite 4950
Woburn, MA 01801 USA
+1.781.404.5100

www.cloudbees.com
info@cloudbees.com