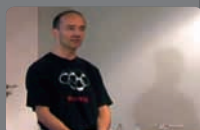


Java™ magazin

Java • Architekturen • Web • Agile

www.javamagazin.de

DVD-INHALT



JAHRES-ARCHIV 2010

Alle Ausgaben und
Quellcodes aus dem
Jahr 2010

JAX TV

- Wie Softwarearchitekten lernen (Dr. Gernot Starke)
- Spring in die Cloud (Christian Dupius)

BUCHAUSZÜGE

- JPA mit Hibernate
- IT Survival Guide
- Java Web Services mit Axis2

Alle DVD-Infos ab Seite 2

JBoss ESB

Prozessmodellierung im
Service Bus » 71

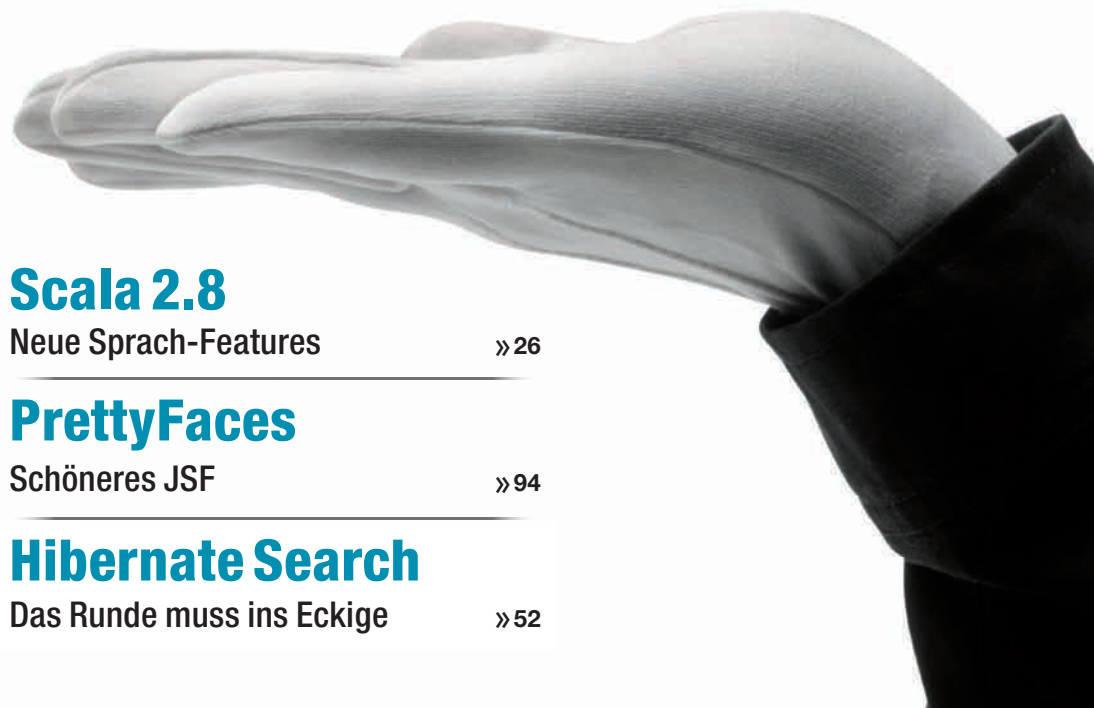
WIKI Plus

Atlassian
Confluence » 60

jax 2011
Alle Infos im Heft » 50

Continuous Integration

- ▶ **State-of-the-Art mit Hudson/Jenkins > 36**
- ▶ **Interviews mit Kohsuke Kawaguchi und Jason van Zyl > 44, 46**
- ▶ **Was steckt hinter CloudBees > 48**



Scala 2.8

Neue Sprach-Features » 26

PrettyFaces

Schöneres JSF » 94

Hibernate Search

Das Runde muss ins Eckige » 52



Datenträger enthält
Info- und
Lehrprogramme
gemäß § 14 JuSchG

Die CloudBees-Cloud-Plattform

Build- und Runtime-Umgebung in der Cloud

Immer mehr Applikationen laufen wegen der größeren Flexibilität und Kosteneffizienz in der Cloud. Anwendungen nicht nur in der Cloud deployen, sondern auch entwickeln – das ist das Versprechen von CloudBees. Dahinter stehen Sacha Labourey (CEO) und Marc Fleury (Investor), die schon aus dem JBoss-Projekt bekannt sind – und natürlich Kohsuke Kawaguchi, einer der Köpfe hinter der Continuous-Integration-Lösung Hudson/Jenkins. Grund genug, einen Blick zu wagen.

von Eberhard Wolff



Vorab: CloudBees ist erst seit dem 31. Januar für die Öffentlichkeit verfügbar. Es wird sicher noch Änderungen und Verbesserungen geben. Der Artikel beschreibt einen sehr frühen Stand (Februar) und kann nur einen ersten Eindruck geben, was CloudBees sein will und sein wird. Die gute Nachricht: Im Moment sind die Dienste kostenfrei nutzbar. Klassisch unterscheidet man bei Cloud-Systemen IaaS (Infrastructure as a Service), bei dem es virtuelle Rechner und Storage gibt, PaaS (Platform as a Service), auf denen man, ähnlich wie bei einem Application Server, Anwendungen deployen kann, und SaaS (Software as a Service), die eine Software für Endanwender zur Verfügung stellen. Bei allen Modellen werden nur genutzte Ressourcen in Rechnung gestellt. Während man beispielsweise bei einem Hostler einen Rechner für ein Jahr mietet und es einige Zeit dauert, bis er bereitsteht, wird bei IaaS ein Rechner innerhalb weniger Minuten zur Verfügung gestellt und man bezahlt ihn stundenweise. CloudBees passt nicht wirklich in diese Einteilung. Das liegt daran, dass CloudBees nicht nur eine Ablaufumgebung für Anwendungen anbietet, sondern auch die Entwicklung von Anwendungen unterstützt. CloudBees bietet zum einen DEV@Cloud an, ein SaaS mit Tools für Softwareentwickler, und RUN@Cloud, eine PaaS, auf der Anwendungen laufen können. Dieser Ansatz ist anders als der vieler anderer PaaS, die nur auf die Ablaufplattform fokussieren. Dahinter steht vermutlich auch eine Geschäftsstrategie: Man macht die Plattform so für Entwickler interessant,

die dann natürlich auch CloudBees als Ablaufumgebung nutzen werden.

DEV@Cloud

Die Entwicklerwerkzeuge sind unter dem Begriff DEV@Cloud zusammengefasst. Konkret ist dies eine Hudson/Jenkins-Installation, die man für eigene Projekte nutzen kann. Weitere geplante Services wie ein Repository verraten sich zwar schon durch Menüeinträge, stehen aber noch nicht zur Verfügung oder existieren nur als geschlossene Betas. Hudson/Jenkins ist ein Continuous-Integration-Tool, mit dem Software bei Änderungen in der Versionskontrolle automatisch kompiliert und getestet wird. So ist gewährleistet, dass sie eine gewisse Mindestqualität hat und jederzeit ein Build möglich ist. Ein solcher Dienst ist für Cloud-Umgebungen besonders geeignet. Er benötigt nach einem Check-in viele Ressourcen für das Kompilieren und Testen, aber nur für eine kurze Dauer. Nachts hingegen werden zum Beispiel keine Ressourcen benötigt. Ressourcen kann man in der Cloud ohne Weiteres kurzfristig allokalieren und wieder freigeben. Das Installieren einer Continuous-Integration-Umgebung ist einfach: Man benötigt nur den Zugriff auf die Versionskontrolle – alles Weitere ist bei einem Build heutzutage voll automatisiert und standardisiert.

Wegen der Standardisierung der automatisierten Builds kann also Continuous Integration sehr schnell und einfach in die Cloud verlegt werden. Weil es nur für kurze Zeit viele Ressourcen benötigt, ist das dynamische Ressourcenmodell der Cloud für diesen Einsatzkontext

sehr nützlich. Allerdings liegt der Quellcode zumindest kurzfristig auf einer fremden Infrastruktur. Das kann gegebenenfalls den Sicherheitsrichtlinien widersprechen.

Für die Cloud-Unterstützung gibt es für Hudson/Jenkins ein Plugin, das auch ohne CloudBees die Nutzung des Amazon-EC2-IaaS erlaubt. Man kann damit Rechner in der Amazon-Cloud nutzen, aber dann muss eine eigene Hudson/Jenkins-Installation vorhanden sein. Bei CloudBees ist Hudson/Jenkins auf der Infrastruktur installiert. Es lockt also ein Kostenvorteil, da kein Aufwand für das Aufsetzen von Hudson/Jenkins anfällt. Die Installation umfasst auch einige Plug-ins, zum Beispiel für den Zugriff auf git-Repositories. Wie man es von Hudson/Jenkins gewohnt ist, kann man sehr einfach ein Projekt in das Continuous Deployment integrieren. Außer dem Quellcode gibt es auch die Möglichkeit, über WebDAV Dateien hochzuladen, die im Build referenziert werden können. Die CloudBees-Infrastruktur läuft auf Amazon-EC2-Servern. Zurzeit ist man auf *m1.small*-Instanzen eingeschränkt und kann auch keine schnelleren Systeme wählen. Hier wird sicher das Angebot in Zukunft erweitert.

RUN@Cloud

IaaS-Infrastrukturen wie Amazon EC2 bieten nur virtuelle Rechner, auf denen die Infrastruktur wie ein Application Server oder Monitoring und schließlich die eigene Anwendung installiert werden müssen. Unter der Bezeichnung DEV@Cloud bietet CloudBees eine Laufzeitinfrastruktur an, die schon eine Infrastruktur mitbringt (PaaS). Dabei setzt CloudBees auf den Tomcat-Server in der Version 6.0.26, auf dem Standard-WAR-Files deployt werden können. Näheres dazu, wie man Java-Anwendungen konkret in der CloudBees-Cloud laufen lassen kann, findet sich im Kasten „CloudBees RUN@Cloud praktisch“. Die CloudBees-Umgebung ist allerdings im Moment noch wenig flexibel. Die Tomcat-Instanzen haben 256 MB Heap, und diese Einstellung ist nicht änderbar. Es gibt die Möglichkeit, eine Instanz oder mehrere Instanzen auszuwählen. Wenn man mehrere Instanzen wählt, sind aber das Load Balancing und die Anzahl der Instanzen nicht konfigurierbar. Dynamisches Skalieren ist ebenfalls nicht möglich. Das ist die Möglichkeit, neue Instanzen abhängig von der Last zu starten. Gerade in Cloud-Umgebungen ist das nützlich, da es punktgenau die Ressourcen nutzt, die gerade benötigt werden. So können Lastspitzen gut abgedeckt werden. Die Replikation von Sessions ist über eine Datenbank möglich [2].

Im Bezahlangebot von CloudBees sollen die erwähnten Einschränkungen nicht mehr vorhanden sein. Die stärkere Vereinheitlichung des CloudBees-Ansatzes hat auch Vorteile: Mehrere Kunden von CloudBees können auf einem virtuellen Rechner untergebracht werden, da man pro Tomcat-Instanz nur 256 MB benötigt. Dadurch kann die Umgebung sehr kosteneffizient sein. Ansätze wie Amazon Beanstalk nutzen einen Rechner exklusiv für einen Kunden, der dann den ganzen Rechner zur Verfügung hat, ihn aber auch ganz bezahlen muss.

Ein einfaches Monitoring ist enthalten. So erhält man Statistiken über die Aufrufe, den Speicherverbrauch, Sessions und Load. Server-Log, Access-Log und Error-Log des Servers können über die Weboberfläche oder mit dem Werkzeug des CloudBees SDK (siehe Kasten) ausgelesen werden.

Zu der RUN@Cloud-Plattform gehört außerdem die Möglichkeit, mit der Weboberfläche oder dem CloudBees SDK MySQL-Datenbanken anzulegen. In einer Anwendung muss man die Datenbank dann mit einem Eintrag in das *WEB-INF/cloudbees-web.xml* konfigurieren.

Anzeige

Das Anlegen einer Datenbank ist sehr einfach, aber es ist nicht möglich, die Datenbankkonfiguration zu optimieren. Zugriff auf die Datenbank mit einem MySQL-Frontend, z. B. für Reporting, ist ebenfalls möglich. Allerdings ist die Verbindung zur Datenbank über das Internet dabei nicht verschlüsselt. Da CloudBees auf der Amazon-EC2-Infrastruktur läuft, sollte es möglich sein, alternativ Amazon Relational Database Service (RDS) [3] zu nutzen. Dieser Service basiert ebenfalls auf MySQL, bietet aber eine wesentlich bessere Skalierung.

Fazit

CloudBees hat mit dem Hudson/Jenkins-Service einen interessanten Ansatz, der die Unterstützung der Softwareentwicklung in den Fokus stellt, während typischerweise Cloud nur als Ablaufumgebung verstanden wird. Dieser Service ist heute schon im Projektkontext ohne Weiteres nutzbar und sehr einfach aufzusetzen. Mit Amazon EC2 als Basis kann CloudBees in Zukunft eine sehr gute Skalierbarkeit bieten. Bei RUN@Cloud ist die Plattform sowohl in Bezug auf Tomcat als auch auf die Datenbank kaum auf unterschiedliche Last anpassbar und optimierbar. Die praktische Nutzbarkeit für anspruchsvolle Projekte ist also im Moment fraglich. Aber die Plattform steht der Öffentlichkeit noch nicht lange zur Verfügung, und daher ist der Stand eher als ein Betatest zu interpretieren. Die Einschränkungen sollen beim Bezahlangebot nicht mehr vorhanden sein. Supporttickets werden recht schnell und freundlich be-

arbeitet und CloudBees ist offen für jegliches Feedback. Wenn man also ambitioniertere Projekte hat, kann man sicher auch auf die Entwicklung der Plattform Einfluss nehmen und entsprechende Features anfragen. Auf Kunden zu hören und anhand des Feedbacks das Projekt weiterzuentwickeln, ist zu so einem frühen Zeitpunkt im Produktzyklus vielleicht das Wichtigste – man kann auf die zukünftige Entwicklung gespannt sein.

Danke an alle adesso-Kollegen für das Review des Artikels!



Eberhard Wolff ist Gründungsmitglied der Java-Champions, Autor zahlreicher Fachartikel und Fachbücher und regelmäßiger Sprecher auf internationalen Konferenzen. Seine Schwerpunkte sind Enterprise-Systeme, Java, Spring und Cloud-Technologien. Er arbeitet als Architecture & Technology Manager für die adesso AG.

Links & Literatur

- [1] <http://wiki.hudson-ci.org/display/HUDSON/Amazon+EC2+Plugin>
- [2] <https://cloudbees.zendesk.com/entries/417088-application-clustering>
- [3] <http://aws.amazon.com/rds/>
- [4] <https://cloudbees.zendesk.com/entries/414109-cloudbees-sdk>
- [5] <https://cloudbees.zendesk.com/entries/421064-maven-guide>
- [6] <http://blog.cloudbees.com/2011/02/continuous-deployment-devcloud-to.html>
- [7] <https://cloudbees.zendesk.com/entries/417131-rails>

CloudBees RUN@Cloud praktisch

Unter [4] steht eine Einführung in das CloudBees SDK bereit. Mit folgenden Schritten kann man recht schnell eine Anwendung in CloudBees RUN@Cloud-Plattform ablaufen lassen:

- Zunächst ist eine Registrierung unter <https://grandcentral.cloudbees.com/account/signup> notwendig
- Unter <https://grandcentral.cloudbees.com/user/keys> stehen die Informationen bereit, die man beim Deployment in den späteren Schritten benötigt
- Unter <http://cloudbees-downloads.s3.amazonaws.com/sdk/cloudbees-sdk-0.5.1-dist.zip> kann das CloudBees SDK heruntergeladen werden
- Das SDK muss entpackt werden
- Das Verzeichnis, in welches das SDK entpackt worden ist, muss in den Pfad aufgenommen werden und es muss eine Umgebungsvariable BEES_HOME gesetzt werden, die auf das Verzeichnis verweist
- `bees help` gibt die Onlinehilfe für das SDK aus
- Mit `bees create -t basic MyBasicApp` erzeugt man eine Anwendung im Unterverzeichnis MyBasicApp, die lediglich aus einem Servlet besteht und die `cloudbees-web.xml`-Datei mit den CloudBees-spezifischen Konfigurationen enthält

- Man muss die Anwendung nun mit `ant` bauen
- Mit `bees app:deploy build/webapp.war` deployt man sie anschließend in die CloudBees-Plattform
- Der URL, unter dem man die Anwendung erreicht, wird beim Deployment ausgegeben

Man kann auch andere Anwendungen deployen, aber ein WAR muss für die CloudBees-Plattform eine eigene Konfiguration unter `WEB-INF/cloudbees-web.xml` enthalten, in der zum Beispiel der Name der Anwendung festgelegt ist. Es gibt auch ein Maven-Plug-in, mit dem man Goals für das Deployment in CloudBees erstellen kann [5]. Damit ist es auch sehr einfach möglich, von DEV@Cloud ein Continuous Deployment auf RUN@Cloud einzurichten [6].

Im Wesentlichen besteht das SDK aus einigen Groovy Scripts, die zum Beispiel auf Ant Libraries zugreifen. Es enthält aber schon einige wesentliche Optimierungen. So werden bei einem erneuten Deployment nur die Änderungen gegenüber der vorherigen Version übertragen, was die Zeit für ein Deployment signifikant reduziert. Das SDK liegt aktuell in der Version 0.5 vor und ist damit noch nicht final. Beispielsweise kann man im Moment nur eine Beispielanwendung mit dem Template `basic` generieren lassen. Für das Deployment von Ruby-on-Rails-Anwendungen gibt es ein Ruby Gem [7].