

COMPUTERWOCHE Fit für die IT 2011? Wissen testen, iPad2 gewinnen.

Home | **Management** | **Hardware** | **Software** | **Netzwerke** | **Mittelstand** | **Security**

Compliance & Recht | IT-Strategie | IT-Services | **Cloud Computing** | IT Macher | CIO des

Cloud Computing

Marktübersicht Cloud Computing

Softwareplattformen aus der Cloud

05.05.2011

Drucken | Empfehlen | PDF | Merken

 Share | 16 | Gefällt mir

von **Eberhard Wolff**, Architecture & Technology Manager bei der adesso AG

Platform as a Service (PaaS) erlaubt es, Anwendungen direkt in der Cloud zu betreiben. Was leisten die Dienste von Amazon, Google und Co. in der Praxis?

In einer IaaS-Cloud ([Infrastructure as a Service](#)) wie Amazon EC2 stehen nur "nackte" Rechner und [Storage](#) zur Verfügung. Soll eine selbst geschriebene Anwendung darauf zum Laufen gebracht werden, muss eine Infrastruktur wie zum Beispiel ein [Application Server](#) installiert werden.



Die Alternative dazu ist die Nutzung einer PaaS (Platform as a Service) im Rahmen des [Cloud Computing](#). Bei einer solchen Plattform ist die Installation zusätzlicher Infrastruktur nicht notwendig. Man kann Anwendungen direkt in der Cloud installieren. Außerdem bietet sie Lösungen für den Betrieb und das Monitoring der Anwendung. Das Skalieren der Anwendung bei Lastspitzen erfolgt automatisch.

PaaS bieten oft proprietäre Services mit fertigen Funktionalitäten an, die bei der Entwicklung von Anwendungen hilfreich sind. Beispielsweise enthält die Amazon-[Cloud](#)-Plattform einen Service, über den per Kreditkarte mit einem Kunden abgerechnet werden kann. Durch die Nutzung solcher [Services](#) ist man allerdings sehr von der PaaS abhängig, da Anwendungen nicht so einfach auf andere Plattformen migriert werden können, die solche Services nicht anbieten.

Bei der Entwicklung von Enterprise-Anwendungen stellt die Java-Plattform die wichtigste Ablaufumgebung dar. Bisher war das Angebot an PaaS-Plattformen für Java begrenzt, aber in den letzten Monaten sind viele neue Angebote aufgetaucht. Daher lohnt sich ein Blick auf den Markt in diesem Bereich.

Softwareplattformen aus der Cloud

Bild 1 von 13



Softwareplattformen aus der Cloud



Platform as a Service (PaaS) erlaubt es, Anwendungen direkt in der Cloud zu betreiben. Was bieten die Dienste von Amazon, Google und Co. in der Praxis?

Foto: Ye Liew, Fotolia.de



Anlegen einer neuen Anwendung mit Google App Engine.
Foto: Eberhard Wolff

GAE bietet zahlreiche zusätzliche **Services** wie einen Cache, XMPP-basiertes Instant Messaging, Bildverarbeitung oder die Speicherung von Blobs. Außerdem kann von GAE aus natürlich auf andere Google Services zugegriffen werden, die über **Web Services-APIs** angesprochen werden können. Es gibt auch eine Integration in die Google-User-Verwaltung. Für Entwickler stehen darüber hinaus Plugins für Eclipse zur Verfügung, so dass ein Deployment von dieser

Entwicklungsumgebung direkt auf GAE möglich ist. Mit Hilfe des GAE SDKs (Software Development Kit) können Anwendungen auch testweise lokal installiert werden und GAE kann mit dem SDK auch von der Kommandozeile aus gesteuert werden. Ein einfaches Monitoring ist ebenfalls für Anwendungen verfügbar.

Google App Engine auf einen Blick

Vorteile:

- ▣ Automatische Skalierung der Anwendung
- ▣ Lange am Markt, bewährt

Nachteile:

- ▣ Wesentliche Einschränkungen gegenüber dem üblichen Java-Programmiermodell
- ▣ Zur Zeit keine relationale Datenbank verfügbar
- ▣ Durch die Einschränkungen ergeben sich Abhängigkeiten in der Anwendung (Lock In)
- ▣ Keine Eingriffsmöglichkeiten oder Tuningmöglichkeiten auf Ebene des Web Servers oder Betriebssystems

INHALT DIESES ARTIKELS

- [Softwareplattformen aus der Cloud](#)
- [Google App Engine](#)
- [Amazon Beanstalk](#)
- [Cloud Bees](#)
- [VMware Cloud Foundry](#)
- [Fazit](#)

Cloud Computing

Marktübersicht Cloud Computing

Softwareplattformen aus der Cloud

05.05.2011

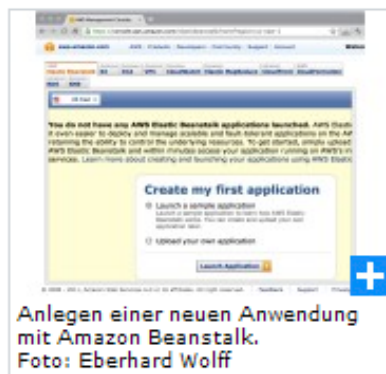
Drucken | Empfehlen | PDF | Merken



von **Eberhard Wolff**, **Architecture & Technology Manager bei der adesso AG**

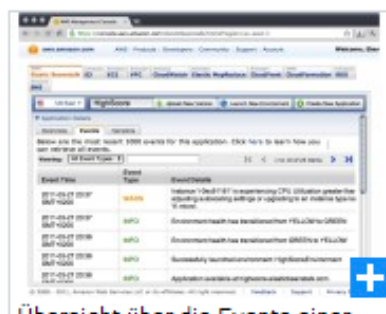
Amazon Beanstalk

Amazon bietet schon seit 2006 verschiedene [Cloud Services](#) an. Dazu zählen virtuelle Rechner (Elastic Compute Cloud, EC2), [Storage](#) (Simple Storage Service, S3), aber auch komplexere Dienste wie schlüsselfertige MySQL-kompatible Datenbanken (Relational Database Service, RDS).



Mit Beanstalk ("Bohnenranke") hat Amazon ein Angebot bereitgestellt, das die Installation und den Betrieb von Java-Anwendungen auf der Amazon-Plattform deutlich vereinfacht. Die Anwendungen werden dabei auf EC2 installiert. Auf dem virtuellen Rechner läuft ein [Linux](#) mit der OpenJDK-Java-Implementierung und dem Java-[Web-Server](#) Tomcat. Um mit Lastspitzen umzugehen, wird ein Load Balancer sowie der Amazon Auto Scaling Service verwendet.

Er startet bei hoher Last automatisch zusätzliche EC2-Instanzen. Dabei können unterschiedliche Indikatoren für die Last genutzt werden, wie die Latenzzeit eines Request oder die [CPU-Last](#) auf den einzelnen Rechnern. Die Skalierung durch das Starten zusätzlicher Rechner ist recht grobgranular: Wenn die Lastgrenze eines Rechners erreicht ist, muss gleich eine ganze neue Instanz gestartet werden, während sich beispielsweise bei [Google App Engine](#) mehrere Kunden einen Rechner teilen können.



Die verschiedenen Versionen einer Anwendung werden auf S3 gespeichert, so dass sie später wiederhergestellt werden können. Es gibt außerdem ein Eclipse Plugin, das den Zugriff auf einen Beanstalk-Server so einfach macht wie sonst den Zugriff auf einen lokalen Tomcat. Ein Kommandozeilen-Werkzeug steht ebenfalls zur Verfügung.

Anwendung bei Amazon Beanstalk.
Foto: Eberhard Wolff

Das System kann an eigene Anforderungen angepasst werden: Die Parameter für den Tomcat können modifiziert werden, ein Login auf die laufende Maschine ist möglich. Sogar die Images für die Rechner können modifiziert werden, so dass praktisch jede erdenkliche Anpassung möglich ist.



Monitoring einer Anwendung mit Amazon Beanstalk.
Foto: Eberhard Wolff

Der Einstieg in die Plattform ist sehr einfach: Beanstalk bietet beim ersten Login die Installation einer Beispielanwendung an und innerhalb weniger Minuten steht ein laufendes System zur Verfügung. Für das Ablegen der Daten steht der Relational Database Service bereit, so dass ein klassisches Enterprise-Java-Programmiermodell verwendet werden kann. Alternativ sind auch NoSQL-Lösungen wie Amazon SimpleDB verfügbar und es gibt zahlreiche andere Dienste, die

teilweise auch von anderen Firmen auf der EC2-Infrastruktur angeboten werden.

Amazon Beanstalk selbst ist kostenlos. Die verwendeten EC2-Instanzen und die anderen Amazon-Dienste wie S3 sind bezahlpflichtig. Allerdings bietet Amazon Neukunden für ein Jahr ein großzügiges Kontingent an kostenloser Nutzung an, so dass die Infrastruktur ausführlich getestet werden kann, bevor sie für produktive Anwendungen genutzt wird.

Amazon Beanstalk ist im Moment im Beta-Stadium. Da allerdings Beanstalk lediglich Dienste aus dem Amazon-Cloud-Angebot kombiniert und einfacher nutzbar macht, ist das Risiko beim Einsatz dieser Beta-Version eher gering. Wegen des Beta-Status steht Beanstalk nicht in allen Amazon-Rechenzentren, sondern nur in der US-East-Region zur Verfügung.

Amazon Beanstalk auf einen Blick

Vorteile:

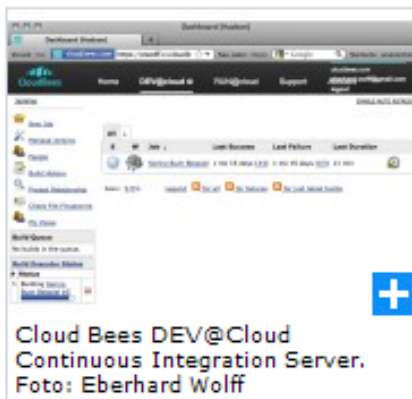
- ▣ Bewährte Cloud-Infrastruktur als Basis
- ▣ Übliches Enterprise-Java-Programmiermodell (Tomcat Web Server/ relationale Datenbank)
- ▣ Daher geringe Lock-In-Gefahr auf Code-Ebene
- ▣ Zahlreiche weitere Services (RDS, SimpleDB, Payment ...)

Nachteile:

- ▣ Zur Zeit noch in Beta und nur in US-East verfügbar
- ▣ Grobgranulare Skalierung durch zusätzliche Rechner

Cloud Bees

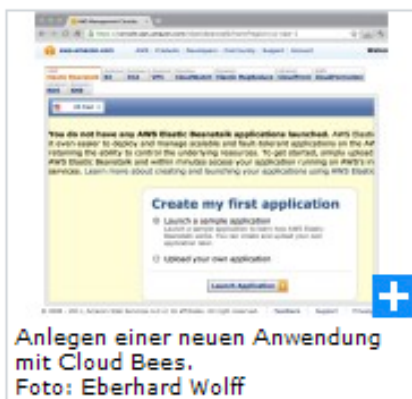
Bei Cloud Bees handelt es sich um ein Startup-**Unternehmen**, das sich auf den Java-**Cloud**-Markt spezialisiert hat. Das Angebot von Cloud Bees umfasst einige Dienste, die für Entwickler attraktiv sind. Sie werden unter dem Namen DEV@Cloud zusammengefasst. Dazu zählt ein Continuous Integration **Server**: Er holt sich bei jeder Änderung den Source Code aus der Versionskontrolle, kompiliert ihn und führt automatisierte Tests aus.



Diese Aufgabe ist gut für Cloud-Systeme geeignet: Sie benötigt zwar viele Ressourcen, aber nur, wenn Entwickler tatsächlich den Code modifizieren und dann auch nur für eine kurze Zeit, nämlich während der Code kompiliert wird. Die Aufgabe ist recht einfach auf einem beliebigen Rechner und damit auch in der Cloud ausführbar, da die Umgebungen für Continuous Integration standardisiert sind. Die Ergebnisse eines Kompilierungsvorgangs können in einem von Cloud

Bees angebotenen Maven Repository abgelegt werden, so dass sie von anderen Projekten weiterverwendet werden können.

DEV@Cloud bietet einen kostenlosen **Service** mit einer entsprechend wenig leistungsfähigen Infrastruktur und verschiedenen kostenpflichtigen Angeboten. Die Strategie von CloudBees ist, die Plattform für Entwickler möglichst attraktiv zu machen, um dadurch Marktanteile zu gewinnen und Kunden zusätzliche Dienste wie eine Ablaufumgebung anzubieten.

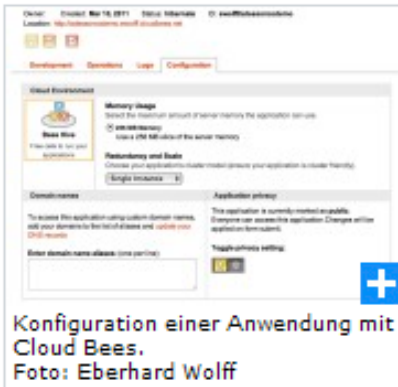


Anlegen einer neuen Anwendung mit Cloud Bees.
Foto: Eberhard Wolff

Als Ablaufumgebung bietet CloudBees RUN@Cloud an. Dahinter verbirgt sich die Möglichkeit, Anwendungen auf einem Tomcat-Server ablaufen zu lassen. RUN@Cloud ist im Moment noch ein kostenloses Angebot. Die Möglichkeiten sind daher sehr begrenzt: Der Tomcat kann praktisch nicht an eigene Bedürfnisse angepasst werden. Es gibt auch kaum Skalierungsmöglichkeiten - es kann nur zwischen einer oder mehreren Instanzen gewählt werden. Ein einfaches Monitoring ist

vorhanden. CloudBees bietet auch ein Kommandozeilen-Werkzeug an, um mit der Plattform zu interagieren.

Ein Vorteil des CloudBees-Ansatzes ist, dass sich mehrere Kunden einen virtuellen Rechner teilen können: Mehrere Tomcat-Instanzen können auf einem Rechner laufen und dadurch kann die Lösung letztendlich kosteneffizienter sein als beispielsweise Amazon Beanstalk, bei dem der Kunde einen virtuellen Rechner exklusiv nutzt.



Als **Datenbank** stellt CloudBees MySQL-Instanzen bereit. Aber auch hier ist der Dienst eher einfach. So ist der Zugriff auf eine solche Datenbank aus dem gesamten **Internet** möglich und es besteht keine Möglichkeit, Backups anzulegen. Auf einer solchen Datenbank-Installation wird wohl kaum jemand unternehmenskritische Daten ablegen. Übrigens läuft die CloudBees-Infrastruktur auf Amazon EC2, so dass die Kombination mit Diensten wie RDS problemlos möglich ist.

Man sollte bei der Bewertung von CloudBees berücksichtigen, dass RUN@Cloud kostenlos ist. Wenn der kostenpflichtige Dienst zur Verfügung steht, werden sicherlich einige der Probleme gelöst sein.

CloudBees auf einen Blick

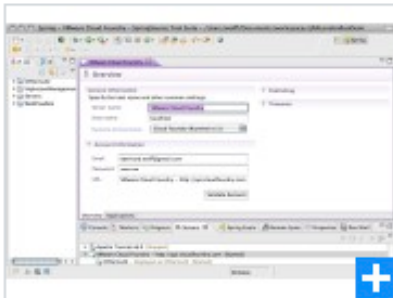
Vorteile:

- ▣ Attraktiv für Entwickler wegen Continuous Integration und Repository Server
- ▣ RUN@Cloud kostenlos
- ▣ EC2-Infrastruktur, daher leicht mit Amazon-Angeboten kombinierbar

Nachteile:

- ▣ RUN@Cloud zur Zeit noch in Beta-Phase
- ▣ Kein echtes Skalierungskonzept
- ▣ Keine SLAs
- ▣ Datenbank-Lösung eher rudimentär
- ▣ Tomcat-Installation kaum anpassbar

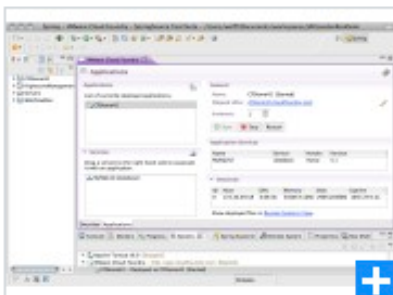
VMware Cloud Foundry



Integration von Cloud Foundry in die Eclipse-Entwicklungsumgebung.
Foto: Eberhard Wolff

VMware ist als einer der wichtigsten Anbieter von Lösungen für Virtualisierung und Private Cloud bekannt. Mit Cloud Foundry steigt VMware mit einer PaaS (Platform as a Service) in den Public-Cloud-Bereich ein. Das stellt eine Abkehr vom bisherigen Geschäftsansatz dar, der auf dem Verkauf von Lizenzen basiert. VMware baut für dieses Angebot eigene Rechenzentren auf, wobei der Betrieb im Moment in den USA stattfindet. Das Angebot ist zurzeit noch kostenlos.

Die Plattform ist vollständig Open Source, so dass sie auch auf anderen Infrastrukturen laufen kann. Die Firma Rightscale hat bereits gezeigt, dass Cloud Foundry auch auf Amazon EC2 betrieben werden kann. Geplant ist außerdem eine "Micro Cloud", die auf einem Desktop-PC oder Laptop betrieben werden kann, um Anwendungen auf der Cloud-Foundry-Infrastruktur zu testen und die Entwicklung für diese Plattform zu vereinfachen. Auf Basis des Cloud Foundry Open Source Projekts ist es auch denkbar, die Infrastruktur als Private Cloud auf eigenen Servern innerhalb einer Firma laufen zu lassen.

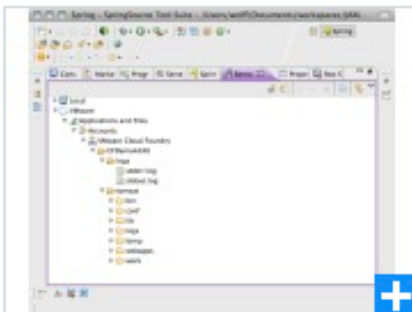


Binden einer Anwendung an einen Service mit Cloud Foundry.
Foto: Eberhard Wolff

Neben Java bietet Cloud Foundry auch Support für die Ruby-Programmiersprache mit den Frameworks Rails und Sinatra sowie für serverseitige JavaScript-Anwendungen, die mit dem Framework Node.js geschrieben sind. Cloud Foundry stellt den Anwendungen Dienste zur Verfügung. Dazu zählen im Moment die relationale Datenbank MySQL und die NoSQL-Datenbanken MongoDB und Redis. Die Anwendungen müssen nicht den konkreten Server kennen, auf dem der

Dienst installiert ist, sondern Cloud Foundry konfiguriert die Anwendungen entsprechend.

Dazu muss allerdings Unterstützung in dem benutzten Framework vorhanden sein. In der Java-Welt sind Spring, Spring Roo und Grails auf der Liste. Da aber die Integration in die Frameworks wie der Rest des Projekts als Open Source zur Verfügung steht und eine Integration auch recht einfach zu bewerkstelligen ist, werden in Zukunft sicher noch weitere Frameworks hinzukommen.



Zugriff auf die Dateien auf dem Cloud Foundry Server.
Foto: Eberhard Wolff

Der Vorteil des Cloud-Foundry-Ansatzes ist, dass die Anwendung ohne Änderungen auf verschiedenen Cloud-Foundry-Umgebungen laufen kann, also beispielsweise einer Entwicklungs-, Test- und Staging-Umgebung. Ohne Cloud Foundry müsste die Anwendung jeweils neu konfiguriert werden, damit sie den richtigen **Datenbank-Server** nutzt.

Für Cloud Foundry gibt es ein Kommandozeilen-Werkzeug und auch ein Eclipse Plugin, so dass die

Entwicklung mit der Plattform gut unterstützt wird.

VMware Cloud Foundry auf einen Blick

Vorteile:

- ▣ Offene Plattform
- ▣ Open-Source-Lizenz
- ▣ Services-Konzept
- ▣ Flexibilität: Nutzung auf Amazon EC2 und lokal möglich, Private Cloud denkbar

Nachteile:

- ▣ Zurzeit noch in Beta-Phase
- ▣ Benötigt Framework-Integration
- ▣ Public-Cloud-Infrastruktur im Moment nur in den USA

Fazit

Der Java-PaaS-Markt ist in Bewegung. Mittlerweile haben sich aber die großen Cloud-Computing-Player positioniert. Dabei sind die Konzepte sehr unterschiedlich: Google war als erster Anbieter am Markt, aber die Plattform des Suchmaschinenkonzerns bietet nur eine sehr eingeschränkte Unterstützung für typische Java-Technologien, so dass heute wohl kaum jemand darauf zurückgreifen wird, wenn nicht proprietäre Dienste wie beispielsweise die Nutzung der Google User-Verwaltung eine Rolle spielen.

Amazon Beanstalk ergänzt die sehr populäre Public-Cloud-Plattform durch eine Unterstützung für Java-Anwendungen. Damit kann die, trotz der jüngsten IT-Ausfälle am Markt etablierte Amazon-Infrastruktur nun wesentlich einfacher für Java-Applikationen genutzt werden.

Cloud Bees hingegen verfolgt mit dem auf Entwickler abgestimmten Services einen ganz anderen Ansatz, kann aber als Ablaufumgebung kaum überzeugen. Und schließlich gibt es mit Cloud Foundry von VMware eine Lösung, die ein interessantes Konzept verfolgt und dabei flexibel und offen ist. Es bleibt also spannend im Cloud-Computing-Marktsegment PaaS. (wh)