

Plattform as a Service: eine Marktübersicht

Wer seine Anwendungen direkt in der Cloud betreiben will, hat die Wahl zwischen mehreren Plattform-as-a-Service-Angeboten. Der Markt ist jung und über weite Strecken noch in der Betaphase. Hier eine Bewertung der wichtigsten Offerten.

Von **Eberhard Wolff***

Der Nachteil einer IaaS-Cloud (Infrastructure as a Service) wie Amazon EC2 ist, dass nur „nackte“ Rechner und Storage zur Verfügung stehen. 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 Plattform as a Service (PaaS) 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 Programme. 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

Google App Engine

Vorteile

- Automatische Skalierung der Anwendung;
- lange am Markt und bewährt.

Nachteile

- Wesentliche Einschränkungen gegenüber dem üblichen Java-Programmiermodell;
- zurzeit keine relationale Datenbank verfügbar;
- durch die Einschränkungen ergeben sich Abhängigkeiten in der Anwendung (Lock in);
- keine Eingriffs- oder Tuning-Möglichkeiten auf Ebene des Web-Servers oder Betriebssystemes.



Übersicht über die Anwendungen in der Google App Engine.

Service, über den per Kreditkarte mit einem Kunden abgerechnet werden kann. Durch die Nutzung solcher Dienste ist man allerdings sehr von der jeweiligen 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.

1 Google App Engine

Google hat bereits 2008 damit begonnen, eine PaaS anzubieten, und war damit ein Pionier im Java-PaaS-Markt. Neben Java unterstützt die Google App Engine (GAE) auch Python-Anwendungen. Es gibt ein GAE-Angebot für den Betrieb von Anwen-

Anlegen einer neuen Anwendung mit der Google App Engine.

dungen, die im Internet jedem zur Verfügung stehen sollen. Die Kosten hängen davon ab, wie viel Bandbreite, Storage, CPU-Zeit etc. genutzt wird. Zum Testen gibt es ein freies Kontingent. Es ist auch möglich, ein Budget für eine Anwendung festzulegen, um so die Kontrolle über die Kosten zu haben. Für Anwendungen, die nur innerhalb einer Firma verwendet werden sollen, gibt es die

Google App Engine for Business. Dabei erfolgt die Abrechnung pro Anwendung und Benutzer, außerdem gibt es andere SLAs.

Google App Engine treibt den PaaS-Ansatz recht weit: Benutzer haben keinen Zugriff auf das zugrunde liegende Betriebssystem. Anwendungen können auf einem oder vielen Servern laufen, ohne dass der Entwickler es bemerkt. Lastverteilung und Skalierung der Anwendung sind transparent. Um dies zu ermöglichen, hat GAE ein sehr eingeschränktes Programmiermodell. Die auf GAE nutzbaren Java-Klassen sind in einer White List beschrieben. So ist es beispielsweise nicht möglich, neue Threads zu starten. Der Zugriff auf das File-System ist nicht erlaubt, und ein Request darf nicht mehr als 30 Sekunden Bearbeitungszeit erfordern. Durch diese Einschränkungen ist die Nutzung einiger Java-Frameworks gar nicht möglich oder nur sehr aufwendig zu realisieren. Google bietet eine Liste von Frameworks an, die unter GAE laufen.

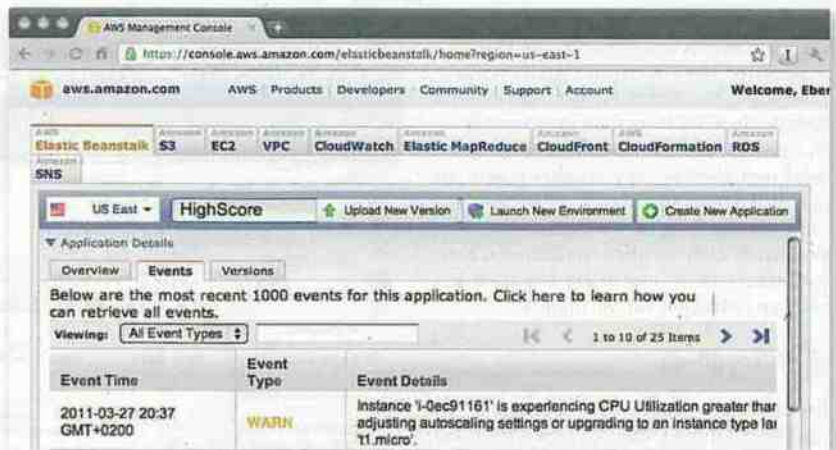
Bislang nur NoSQL

Als Datenbank steht ein NoSQL-System zur Verfügung. Eine relationale Datenbank ist angekündigt, soll dann aber nur im Rahmen von Google App Engine for Business zur Verfügung stehen. Daher muss die Persistenz-Architektur einer Anwendung, die auf GAE laufen soll, entsprechend angepasst werden. Eine Portierung auf eine andere Infrastruktur wird dadurch erschwert.

GAE bietet zahlreiche zusätzliche Services wie einen Cache, XMPP-basierendes Instant Messaging, Bildverarbeitung oder die Speicherung von Blobs. Außerdem kann von GAE aus natürlich auf andere Google-Services zugegriffen werden, die sich über Web-Services-APIs ansprechen lassen. Es gibt auch eine Integration in die Google-User-Verwaltung. Für Entwickler stehen darüber hinaus Plug-ins für Eclipse zur Verfügung, so dass ein Deployment von dieser Entwicklungsumgebung direkt auf GAE möglich ist. Mit Hilfe des GAE SDK (Software Development Kit) sind Anwendungen für Testzwecke auch lokal installierbar. Zudem kann GAE mit Hilfe des SDK auch von der Kommandozeile aus gesteuert werden. Ein einfaches Monitoring ist ebenfalls für Anwendungen verfügbar.

2 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).



Übersicht über die Events einer Anwendung in Amazon Beanstalk.

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 lassen sich unterschiedliche Indikatoren für die Last nutzen, etwa 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 In-



EVENT:
Best in Cloud
Info Seite 49

stanz 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 später eine Wiederherstellung möglich ist. Es gibt außerdem ein Eclipse-Plug-in, das den Zugriff auf einen Beanstalk-Server so einfach macht wie sonst auf einen lokalen Tomcat-Server. Ein Kommandozeilen-Werkzeug steht ebenfalls zur Verfügung. Das System kann an eigene Anforderungen angepasst werden: Die Parameter für den Tomcat sind modifizierbar, ein Login auf die laufende Maschine ist möglich. Sogar die Images für die Rechner sind veränderbar, so dass praktisch jede erdenkliche Anpassung möglich ist.

Schneller Einstieg

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 man ein klassisches Enterprise-Java-Programmiermodell verwenden kann. Alternativ sind auch NoSQL-Lösungen wie Amazon SimpleDB verfügbar. Außerdem gibt es zahlreiche andere Dienste, die teilweise 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 dagegen kostenpflichtig. Allerdings bietet Amazon Neukunden für ein Jahr ein großzügiges Kontingent an kostenloser

Amazon Beanstalk

Vorteile

- Bewährte Cloud-Infrastruktur als Basis;
- übliches Enterprise-Java-Programmiermodell (Tomcat Web-Server und relationale Datenbank);
- daher geringe Lock-in-Gefahr auf Codeebene;
- zahlreiche weitere Services (RDS, SimpleDB, Payment etc.).

Nachteile

- Derzeit noch in der Betaphase und deshalb regional eingeschränkt verfügbar;
- grobgranulare Skalierung durch zusätzliche Rechner.

► Nutzung an, so dass man die Infrastruktur ausführlich testen kann, bevor sie für produktive Anwendungen genutzt wird.

Amazon Beanstalk ist derzeit noch im Betastadium. Da Beanstalk aber nur Dienste aus dem Amazon-Cloud-Angebot kombiniert und diese einfach nutzbar macht, ist das Risiko beim Einsatz dieser Betaversion eher gering. Wegen des Betastatus steht Beanstalk nicht in allen Amazon-Rechenzentren, sondern nur in der US-amerikanischen Ostregion zur Verfügung.

3 Cloudbees

Bei Cloudbees handelt es sich um ein Start-up-Unternehmen, das sich auf den Java-Cloud-Markt spezialisiert hat. Das Angebot von Cloudbees 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 Sourcecode aus der Versionskontrolle, kompiliert ihn und führt automatisierte Tests aus. Diese Aufgabe ist gut für Cloud-Systeme geeignet: Sie benötigt viele Ressourcen, wenn Entwickler den Code modifizieren, dann allerdings nur für 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.

Einstieg über Entwickler

Die Ergebnisse eines Kompilierungsvorgangs können in einem von Cloudbees angebotenen Maven Repository abgelegt werden, so dass man sie in anderen Projekten weiterverwenden kann. DEV@Cloud bietet einen kostenlosen Service mit einer entsprechend wenig leistungsfähigen Infrastruktur sowie verschiedenen kostenpflichtigen

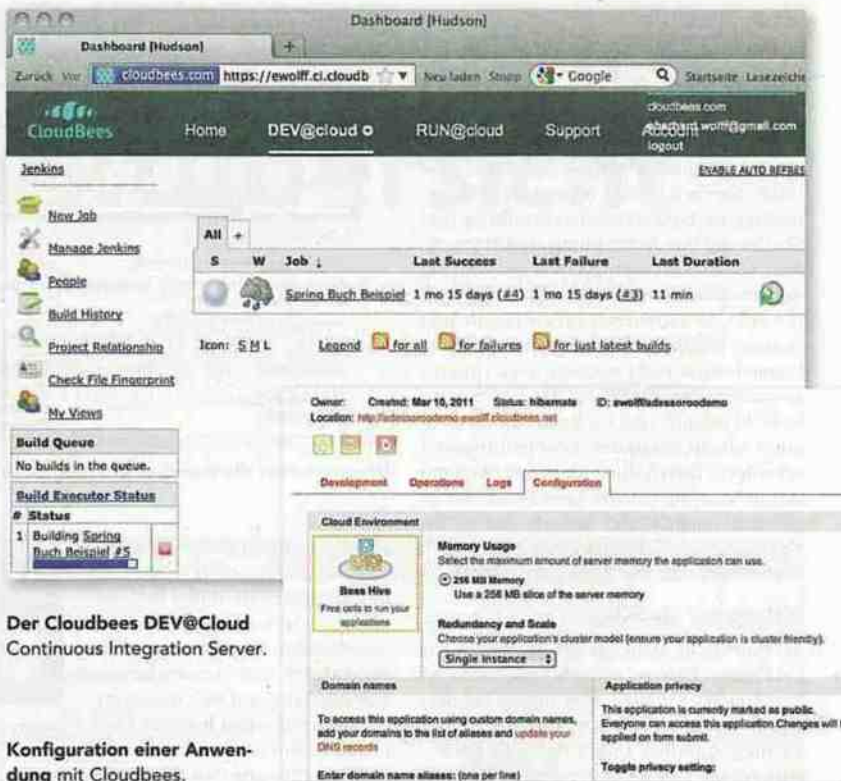
Cloudbees

Vorteile

- Attraktiv für Entwickler bei Continuous Integration und wegen des Repository-Servers;
- RUN@Cloud kostenlos;
- EC2-Infrastruktur, daher leicht mit Amazon-Angeboten kombinierbar.

Nachteile

- RUN@Cloud zurzeit noch in der Beta-phase;
- kein echtes Skalierungskonzept;
- keine SLAs;
- Datenbanklösung eher rudimentär;
- Tomcat-Installation kaum anpassbar.



Der Cloudbees DEV@Cloud Continuous Integration Server.

Konfiguration einer Anwendung mit Cloudbees.

Angewandten. Die Strategie von Cloudbees ist, die Plattform für Entwickler möglichst attraktiv zu gestalten, um dadurch Marktanteile zu gewinnen und Kunden zusätzliche Dienste wie eine Ablaufumgebung anzubieten.

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 Skalierungsoptionen – 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.

Kosteneffiziente Architektur

Ein Vorteil des Cloudbees-Ansatzes ist, dass sich mehrere Kunden einen virtuellen Rechner teilen können: Mehrere Tomcat-Instanzen lassen sich auf einem Rechner unterbringen, was die Lösung letztlich kosteneffizienter macht 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 gibt keine Backup-Funktionen. Auf einer solchen Datenbankinstallation 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 sicher einige der Probleme gelöst sein.

4 VMware Cloud Foundry

VMware ist als einer der wichtigsten Anbieter von Lösungen für Virtualisierung und Private Cloud bekannt. Über Cloud Foundry steigt VMware mit einer PaaS in den Public-Cloud-Bereich ein. Das stellt eine Abkehr vom bisherigen Geschäftsansatz dar, der auf einem 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 sich auf einem Desktop-PC oder Laptop betreiben lässt, um Anwendungen für die 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.

Unterstützte Tools-Umgebungen

Neben Java bietet Cloud Foundry auch Support für die Programmiersprache Ruby mit den Frameworks Rails und Sinatra sowie für Server-seitige 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.

Flexible Einsatzszenarien

Der Vorteil des Cloud-Foundry-Ansatzes ist, dass die Anwendung ohne Änderungen auf verschiedenen Cloud-Foundry-Umgebungen laufen kann, also beispielsweise einer Ent-

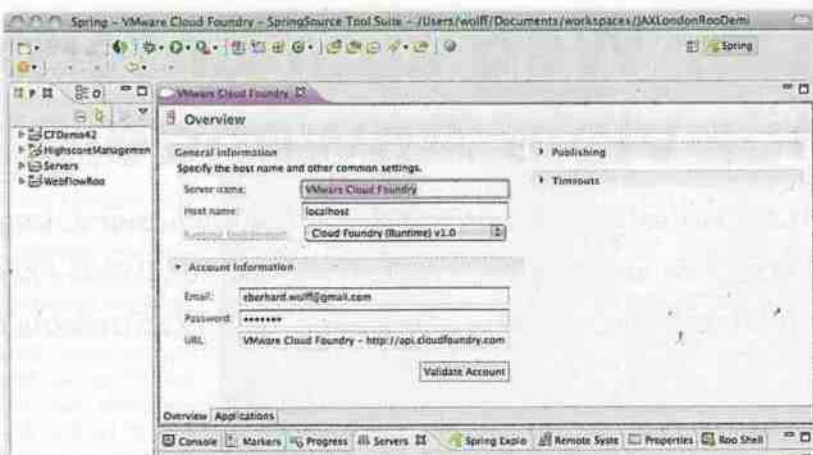
VMware Cloud Foundry

Vorteile

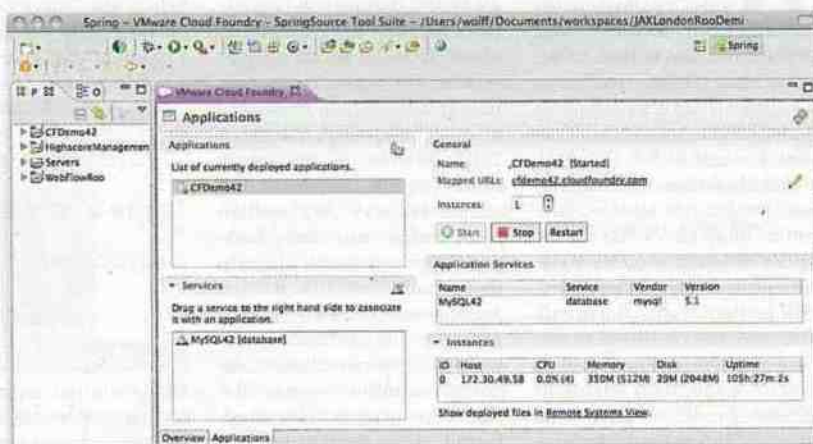
- Offene Plattform;
- Open-Source-Lizenz;
- Services-Konzept;
- Flexibilität (Nutzung auf Amazon EC2 und lokal möglich, Private Cloud denkbar).

Nachteile

- ➖ Zurzeit noch in der Betaphase;
- ➖ benötigt Framework-Integration;
- ➖ Public-Cloud-Infrastruktur im Moment nur in den USA.



Integration von Cloud Foundry in die Eclipse-Entwicklungsumgebung.



Binden einer Anwendung an einen Service mit Cloud Foundry.

wicklungs-, 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-Plug-in, so dass die Entwicklung mit der Plattform gut unterstützt wird.

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-Techniken, 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 lässt sich die trotz der jüngsten IT-Ausfälle am Markt etablierte Amazon-Infrastruktur nun wesentlich einfacher für Java-Applikationen nutzen. Cloudbees hingegen verfolgt mit dem auf Entwickler abgestimmten Services einen ganz anderen Ansatz, kann aber als Ablaufumgebung kaum überzeugen. 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/ue)

*Eberhard Wolff ist Manager für Architecture & Technology bei der Adesso AG in Dortmund.