JAVA IN THE CLOUD
PAAS PLATFORM IN COMPARISON

Eberhard Wolff
Architecture and Technology Manager
adesso AG, Germany
Agenda

- A Few Words About Cloud
- Java and IaaS
- PaaS – Platform as a Service
- Google App Engine
- Amazon Beanstalk
- VMware CloudFoundry
- Cloud Bees
A Few Words About Cloud
<table>
<thead>
<tr>
<th>Infrastructure as a Service</th>
<th>Platform as a Service</th>
<th>Software as a Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; Virtual Servers</td>
<td>&gt; Virtual App Server</td>
<td>&gt; Software or Service</td>
</tr>
<tr>
<td>&gt; Similar to Virtualization</td>
<td>&gt; Handles Scale-Out</td>
<td>that you use</td>
</tr>
<tr>
<td>&gt; Manage Everything Yourself</td>
<td>&gt; Mostly Managed by Provider</td>
<td>&gt; Components that you add/integrate into your app</td>
</tr>
</tbody>
</table>
Cloud Deployment Model

► Public
  > Available to general public

► Private
  > Available to only one organization
Why Cloud: Economics

► Public Cloud:
  > Pay only what you need (Pay-as-you-go) (Public Cloud)
  > No CapEx
  > Handle load peaks cheaply
  > Transparent cost model

► Private Cloud:
  > Better Resource Utilization
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances.

### Compute: Amazon EC2 On-Demand Instances:

<table>
<thead>
<tr>
<th>Instances</th>
<th>Description</th>
<th>Operating System</th>
<th>Instance Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Load Peaks</td>
<td>Linux/OpenSolaris</td>
<td>Small</td>
<td>0</td>
</tr>
</tbody>
</table>

### Compute: Amazon EC2 Reserved Instances:

<table>
<thead>
<tr>
<th>Instances</th>
<th>Description</th>
<th>OS</th>
<th>Type</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Grundlast</td>
<td>Linux</td>
<td>Small</td>
<td>3 yr term</td>
</tr>
</tbody>
</table>

### Storage: Amazon EBS Volumes:

<table>
<thead>
<tr>
<th>Volumes</th>
<th>Description</th>
<th>Provisioned Storage</th>
<th>Average IOPS in volume</th>
<th>Snapshot Storage*</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>10 GB-month</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

### Elastic IP:

- Number of Elastic IPs: 0
- Elastic IP Non-attached Time: 0

NEW! - Amazon RDS for Oracle, Custom Metrics in Amazon CloudWatch and Dedicated Instances in Amazon VPC

FREE USAGE TIER: New Customers get free usage tier for first 12 months

Language: English

Services Estimate of your Monthly Bill ($ 60.90)
Why Cloud: Business Agility

► From development to production with just one click
► Much faster
► Much simpler

► Application scales automatically
  > Higher load means more resources are used automatically

► Create a test environment with just one click
  > Production-like sizing
  > Cheaply (only paid during the test)

Adrian Colyer (CTO VMware):
Customers want Business Agility – even if it means higher prices

Werner Vogels (CTO Amazon):
Amazon Cloud is the answer to Amazon developers spending 70% of their time with scalability and technology
Why Cloud? Platform of the Next Generation

- Based on cheap commodity technologies
- No costly high available hardware
- Individual server may fail
- Network may fail

- But:
- Data and application can be held redundantly in multiple data centers
- Automatic distribution
- Starting new computers trivial

- Cheap systems with high availability and high data durability
- Just like Google, Amazon, Facebook...
- Needs different architectures
What this is all about...

WAR
So, let me get started

- Get an account at an IaaS provider
- …or virtualize your data center and create a self service portal
- Install your (Java EE) environment
- Install your (Java) application
- Done

- Wow, that was easy!
That is not enough

- How do you deal with peaks? Need more app server instances
- The server instances must be shut down after the peak
- ...otherwise you would pay for them
- Traditional middleware does not allow for that
- Elastic scaling

- Individual nodes fail – deal with it!

- RBMS prefer scale up (larger server)
- In the cloud it is easier to scale out (more server)
- That is why Amazon and Google use NoSQL / key-value stores
What you will eventually come up with

- A tool to take an Application
- ...and create a VM with all needed infrastructure etc
- Dynamically i.e. scale up and down

- Need tools to
  - Install software
  - Manage infrastructure
  - Configure infrastructure
  - Set up user etc
- Puppet, Chef etc.

- Like a factory for VMs
- Works on Private Cloud, Public Cloud or your local machine
Advantages of IaaS Approach

► Very flexible
► Works for any IaaS and any software to be installed
► Works for complex environments with many infrastructure pieces
  > Install a database server, some Tomcats, a load balancer and a cache server
  > Install your own and totally proprietary special solution
  > Fine tune all the parameters
► Can deploy different parts of the application to special nodes

► But often developers just want a platform to run applications on
► No fine tuning
► Standardized and uniform services
► Also: Developers need other non-Java-EE services
Not just automated…
PaaS
Platform as a service (PaaS) is the delivery of a computing platform and solution stack as a service.
PaaS: Advantages and Disadvantages

Advantages

• Easier to use than IaaS: You would need to install a server anyway
• Automatic scaling
  – Resources automatically added
• Can offer additional service
  • Tuned for Cloud
  • Technical e.g. data store, messaging, GUI elements
  • …but IaaS does the same (Amazon)

Disadvantages

• Less flexible
  • Pre-defined programming model
  • Defines environment
• Programming model might be different
  • Hard to learn & port existing code
Google App Engine
Google App Engine

- Pioneer: Very early in the market
- Supports Java, Python, Go (beta)
- For Java: Very restrictive environment
  - Java classes white list
  - Limited sandbox
  - Focus on NoSQL while typical Java applications use RDBMS
  - Limited Preview: Cloud SQL (MySQL)
  - Limit on start up time of application etc
  - Limit on response time (30 seconds)
  - No control or access to operating system
  - Can’t change configuration of the web server
- Benefits?
- Specialized frameworks have been created (Gaelyk for Groovy)
- Recently changed pricing
Amazon Elastic Beanstalk
Amazon Web Services

- Collection of Cloud Offerings (mostly IaaS)
- Elastic Compute Cloud (EC2)
- Elastic Map Reduce
- Auto Scaling
- SimpleDB: Big Table like NoSQL database
- Simple Queue Service (SQS)
- Simple Notification Service (SNS)
- Simple Email Service (SES)
- Virtual Private Cloud (VPC)
- Simple Storage Service (S3)
- Elastic Block Storage (EBS)
- ElastiCache

AWS is a marketplace: 3rd party offerings https://mongohq.com/ for MongoDB and https://cloudant.com/ for CouchDB
Amazon Elastic Beanstalk

- Based on the Amazon EC2 infrastructure
- …and Auto Scaling
- …S3 to host the WARs
- Adds Linux, OpenJDK, Tomcat 6 / 7

- Currently in beta
- …and only in US-East

- Eclipse Plug In available
- Supports versioning of applications
- Supports elastic scaling depending on load indicators
- Simple Monitoring built in
- Detailed control over the environment (Tomcat parameters, used VM image, log in to machine etc.)
Amazon Elastic Beanstalk

- Access to Tomcat logs etc.
- Access to the OS
- Fine tuning of Tomcat parameters possible
- Easy, yet powerful

- Very easy to get started
- Demo application based on Spring
- Uses also S3 (storage) and Simple Notification Service (SNS)

- Add Relational Database Service (RDS) for enterprise scale MySQL or Oracle
- …and all the other Amazon Web Services (AWS)
- …Virtual Private Cloud (VPC) to access your backend
- …Elasticache for performance
Amazon Elastic Beanstalk

- Can be much like your average Enterprise Java environment
  - = Tomcat + RDBMS
- Cloud features like elastic scaling available
- Can easily add other AWS elements
- Runs on a proven environment

- But: 1 server = 1 virtual machine
- GAE can run multiple applications on one machine
- Less efficient (?)
VMware
Cloud Foundry
VMware Cloud Foundry

- Open Source
  - At https://github.com/cloudfoundry/
    under Apache2 license

- No commercial offering yet
- Hosted at cloudfoundry.com, currently beta
Can run Java, Ruby and Node.js
- Spring, Grails, Scala / Lift, Rails, Sinatra & Node.js supported
- Erlang, PHP, Python, Play created by community
- Support for other languages currently in development by the community

- Spring / Grails / Lift use Tomcat internally
- 1 server runs multiple applications

- Command line tool available
- Eclipse Plug In available
- Only possible to add new instances, no elastic scaling depending on load indicators
- Well… you can build it 😊
VMware Cloud Foundry Services

- Relational Database Service (MySQL, PostgreSQL)
- NoSQL Key-Value Store (Redis)
- NoSQL Document Store (mongoDB)
- Messaging Service (RabbitMQ)

- Services are shared across applications
  - i.e. one server for multiple clients

- Framework support (e.g. Spring) allows easy access to services
  - Behind the scenes: Environment variable for server, user, password
  - i.e. can also use without framework support

- More to come in the future
VMware Cloud Foundry: Other Platforms

- Announced: Private Cloud Offering by VMware
- Beta: Activestate Private PaaS Offering

- Rightscale VM images for EC2

- Ubuntu 11.10 has a repository for Cloud Foundry
- Easy to create larger installations on EC2 using juju
- Ubuntu’s juju coordinates and installs a set of VMs

- VMware Cloud Foundry Microcloud: Try applications on your laptop
CloudBees DEV@Cloud and RUN@Cloud
CloudBees: DEV@Cloud

- Continuous Integration (Jenkins)
  - Good application for the Cloud: Peaks and high load only during working hours
  - Standardized and universally applicable service
  - Some Essentials Plug Ins in free version
  - More in Base / Pro / Enterprise pay version
  - Also more parallel build in pay version
  - …and faster build machines

- Maven repository
  - Snapshot / Release
  - Builds can be automatically deployed

- Might add other services in the future

- Partner: SonarSource (Sonar in the Cloud), Sauce Labs On Demand (Selenium)
CloudBees: RUN@Cloud

- Tomcat / Java EE 6 Web Profile
- Runs on Amazon EC2
- Multiple applications run on one machine

- Easily deploy a WAR
  - either by web interface
  - or command line utility (bees SDK)

- Simple monitoring (web / command line)
- Access to logs

- MySQL database
  - Very simple (i.e. just one server, but backup included)
  - Could use Amazon RDS and partner offering for MonogDB / CouchDB instead
Other Players

► Oracle
   > Just launched its Public Cloud based on Java EE

► Heroku
   > Pioneer for PaaS, in particular Ruby
   > Now support for Scala, Java etc
   > Acquired by Salesforce.com
   > git push into the Cloud

► RedHat
   > OpenShift Express: Public Cloud with Perl, Python, Ruby, and Java EE support
   > OpenShift Felx: Public Cloud for Java EE based on EC2
   > OpenShift Power (announced): Support for arbitrary Linux applications
Cloud

- Cloud is interesting because
  - Economics
  - Business Agility
  - Platform of the Future

- Google App Engine: The pioneer

- Amazon Beanstalk: The Tomcat you are used to – based on Amazon Web Services

- Cloud Foundry: Open Source platform with a lot of innovation

- CloudBees: Developer Focus

- Other players: Red Hat, Oracle and Heroku
Questions?

@ewolff

twitter