PaaS

A Lecture for A4M39WA2

Ing. Tomáš Vondra

Cloud Computing

- A new form of IT outsourcing
  - Replaces server rental, webhosting, managed applications
  - Correspoding layers: IaaS, PaaS, SaaS
- Brings new features
  - Elasticity, automation, infinite scale
  - Pay per use, accounting
  - Self service, web services integration
- Three layers:

IaaS

- Computing infrastructure offered on the web
  - CPU power, memory, storage, network
- Enabling technology – Virtualization
  - Flexible allocation and sharing of servers
  - Separation of customers = multitenancy
  - -> granularity – one VM instance and hour
- Different usage model from raw virtualization
  - Multiple VMs from one read-only image
  - Agile infrastructure – 1 server, multiple purposes

PaaS

- Public solution stacks for web applications
  - OS, web server, language interpreters, provisions for automatic scaling, all shielded from the user
- Each system only has a few supported languages
  - Automatic deployment and scaling not trivial
- Offers development tools
  - Libraries for specific services
  - IDE plugins, deployment tools
- Two types – Instance PaaS, Framework PaaS

SaaS

- A new form of application delivery
- Global availability – web application, user data
- Multitenancy – users use the same installation
- Licensing – ideally pay per use
  - When monthly payment, it should be elastic
- Third layer of cloud doesn't imply use of underlying layers, but elastic infrastructure is advisable
  - <- No control over the number of users

PaaS properties

- Gives the programmer a solution stack
  - Web server, database engine, scripting language
- Simple deployment, no worries about servers, storage, network, scaling, updates, …
- Guarantees multitenancy for better security
- Users isolated by virtualization or OS means
- Accounting and billing of used resources
  - Different at every vendor
- Development tools

Comparison with IaaS

- IaaS better for migrating existing applications
  - More flexible, you install your environment
- PaaS has lower demands on administration
- PaaS will take care of scaling if applications use correct frameworks, also redundancy and CDN
- -> PaaS better for new applications
- BUT has dangers of vendor lock in if platform specific functions are used
  - IaaS instance can be copied to your server.

Comparison with Webhosting

- Webhosting essentially does the same – offers a platform for web sites / applications
- Minus scalability, multitenancy, accounting
- Plus personal contact – negotiation, support
- Different languages, cloud focuses on scalability
  - Hosting: PHP, ASP, some Perl and Python
  - Cloud: Java, Ruby, PHP (due to demand), Node.js
- Added value – e-mail and domain hosting
  - vs. development tools and web services in PaaS

PaaS types

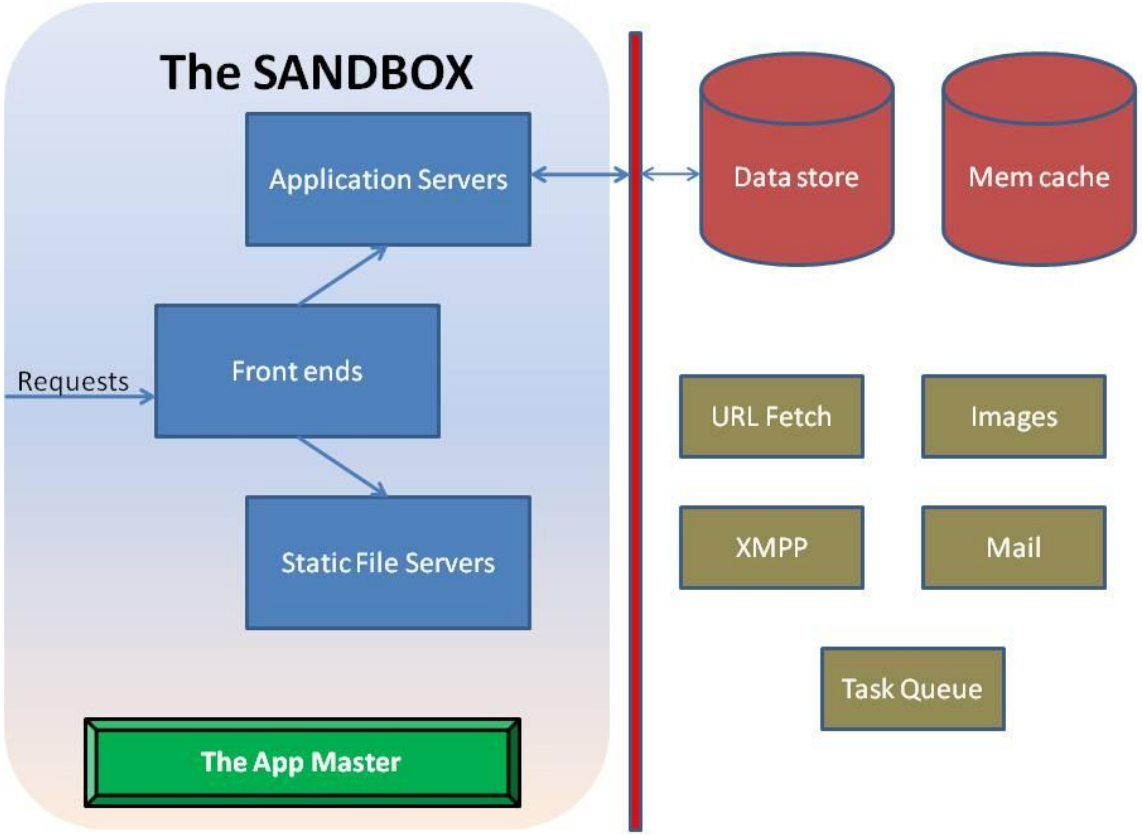
- Instance PaaS
  - Depends on IaaS layer for multitenancy
    - Better security and performance guarantees
  - Deploys applications to IaaS instances
- Framework PaaS
  - Uses OS capabilities for multitenancy
    - Better resource utilization and accounting granularity
  - Requires specific frameworks to be used
  - Can benefit from cloud infrastructure, but is not dependent on it
- Metadata PaaS
  - Client configures his service through metadata

Available systems (selected)

- Instance PaaS
  - Amazon Elastic Beanstalk
  - Microsoft Azure
- Framework PaaS
  - Google App Engine
  - VMware Cloud Foundry

Google App Engine

- Since Apr. 2008, commercial in Sep. 2011
- Languages (in order) – Python, Java, Go
- Typical Framework PaaS
  - Multitenancy by limiting system library functions
    - No filesystem writes and network sockets
    - Must use specific database and network services
      - High vendor lock-in potential (ex. emulation projects)
  - Quotas needed to support massive multitenancy
    - Daily quotas – billing, fair use
    - Per minute quotas – spike prevention



Architecture of Google App Engine

GAE - Quotas

- Basic quota enough for development
  - 6,5 h CPU time per day (granularity 15 min)
  - 1 GB datastore, 5 GB blobstore, 1 GB network traf.
- If billing enabled, minor service quotas raised
  - $0.08 CPU hour 600 MHz (of what?), larger inst.
  - $0.12 1 GB net traffic
  - $0.24 datastore, $0.13 blobstore, $0.1 100k writes
- If exceeded
  - Main quotas: user gets HTTP 403
  - Service quotas: OverQuotaError exception

GAE – Services

- Data storage based on Google File System
  - Over it BigTable noSQL database engine
    - Providing Datastore and Blobstore APIs (GQL lang.)
      - For Java JDO and JPA compatibility layers
        - Problem – not SQL: no indexes, no joins –> very limited
  - Datastore – Master-Slave or High Replication
- Google Accounts, HTTP fetch, e-mail send, XMPP, Memcache, image manip. and others
- Client libraries for all APIs in Eclipse plugin

GAE – Limitations

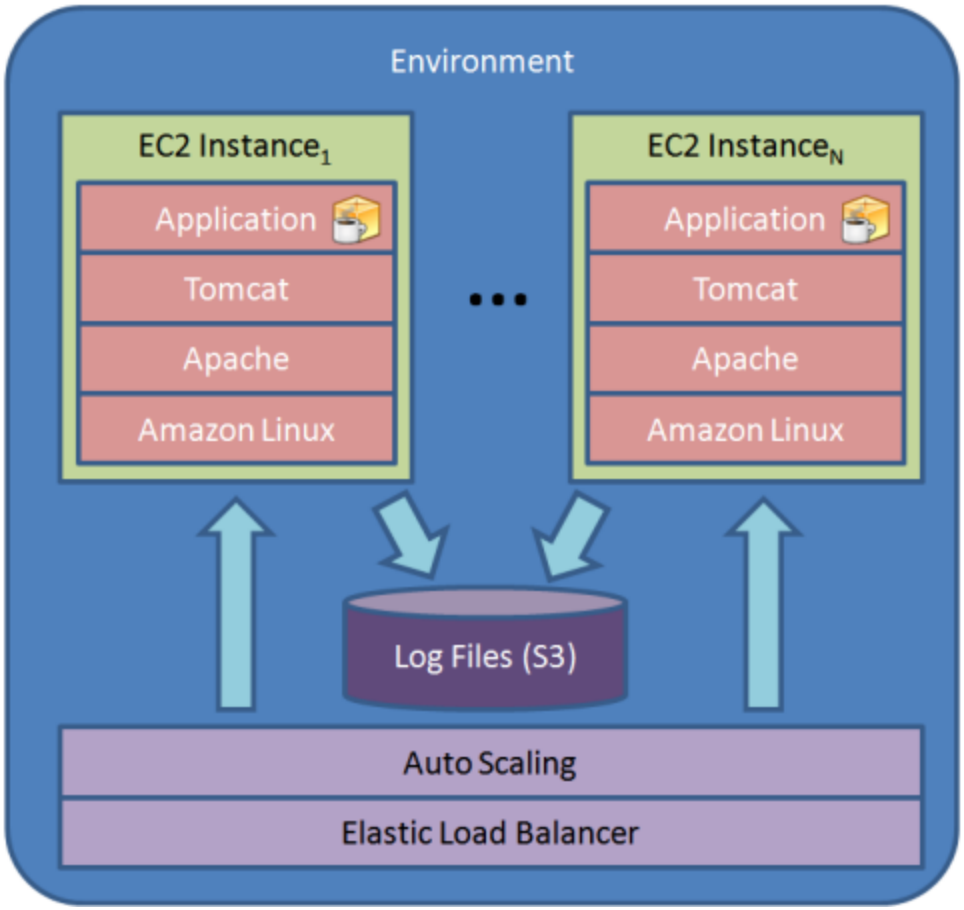
- No state information between HTTP requests
  - All sessions etc. in datastore
  - Good for instance unloading and scalability
- Event style programming, 30 s per request
  - Single-threaded instances by default!
  - Task queues (60s tasks) and backend instances
- Specific programming style
  - Practically impossible to deploy existing apps.
  - Vendor lock-in

Amazon Elastic Beanstalk

- Since beginning of 2011, still Beta
- Currently only Java, should be extensible
  - Java interpreters – jRuby, Quercus, ..
- Amazon – major IaaS provider -> Instance PaaS
- Other PaaSs on Amazon: Heroku, AppFog, ..
- Multitenancy by virtualization
  - Each user gets his own VM instances
  - Some services shared, some need more instances
  - Minimal limitations – can migrate to IaaS

Beanstalk – Pricing

- Free – only a management tool for other services
  - Prepares images with solution stacks, w/upgrades
- Deploys images on EC2 instances:
  - m1.small – $0.085 (1,7 GB RAM, 1 ECU)
  - t1.micro – $0.02 (613 MB, 10% of 2 ECU)
- Uses Elastic Load Balancer:
  - $0.025 / 1 hour, $0.008 / 1 GB
- Applications and logs are on S3:
  - $0.14 / 1 GB and month
- Uses Elastic Autoscaler, which needs CloudWatch
  - $3.5 per instance and month
- Data transfers outbound from cloud
  - $0.12 / 1 GB basic price, degressive for larger volumes


Environment
EC2 Instance1
Application
Tomcat
Apache
Amazon Linux
EC2 InstanceN
Application
Tomcat
Apache
Amazon Linux
Log Files (S3)
Auto Scaling
Elastic Load Balancer

Beanstalk – Services

- The service itself has no database, Amazon has:
- Database VM instances
  - per hour licensing for IBM DB2 and Informix, Oracle, MS SQL, Sybase, Postgres Plus, normal instance price for free DBs
- RDS – automatically managed MySQL and Oracle
  - does updates, backups, scaling, HA between av. zones
  - Paid per hour, $0.11, 2x in hot-standby, $0.1 / 1 GB
- Non-relational databases with variable row format & sharding
  - SimpleDB - $0.25 / 1 GB, $0.14 / machine hour (shared)
    - Max size 10 GB, automatic indexing, website use, 1 GB & 25 h free
  - DynamoDB - $1 / 1 GB, $0.01 for 50 kB/s (granularity 1h)
    - On SSD storage, guaranteed IOPS, for data mining
- Other services: MapReduce, ElastiCache, Route 53, e–mail sending, Simple Queue Service, Simple Notification Service

Beanstalk – Limitations

- Minimal.
  - Applications are deployed to standard application servers (currently Apache Tomcat)
  - Standard relational databases available
  - Can disable the service and continue as IaaS only
- Local storage is ephemeral, design for scalability
- High price ~ $40 for minimal conf. + database
- Eclipse plugin provided, API and CLI tools

Microsoft Azure

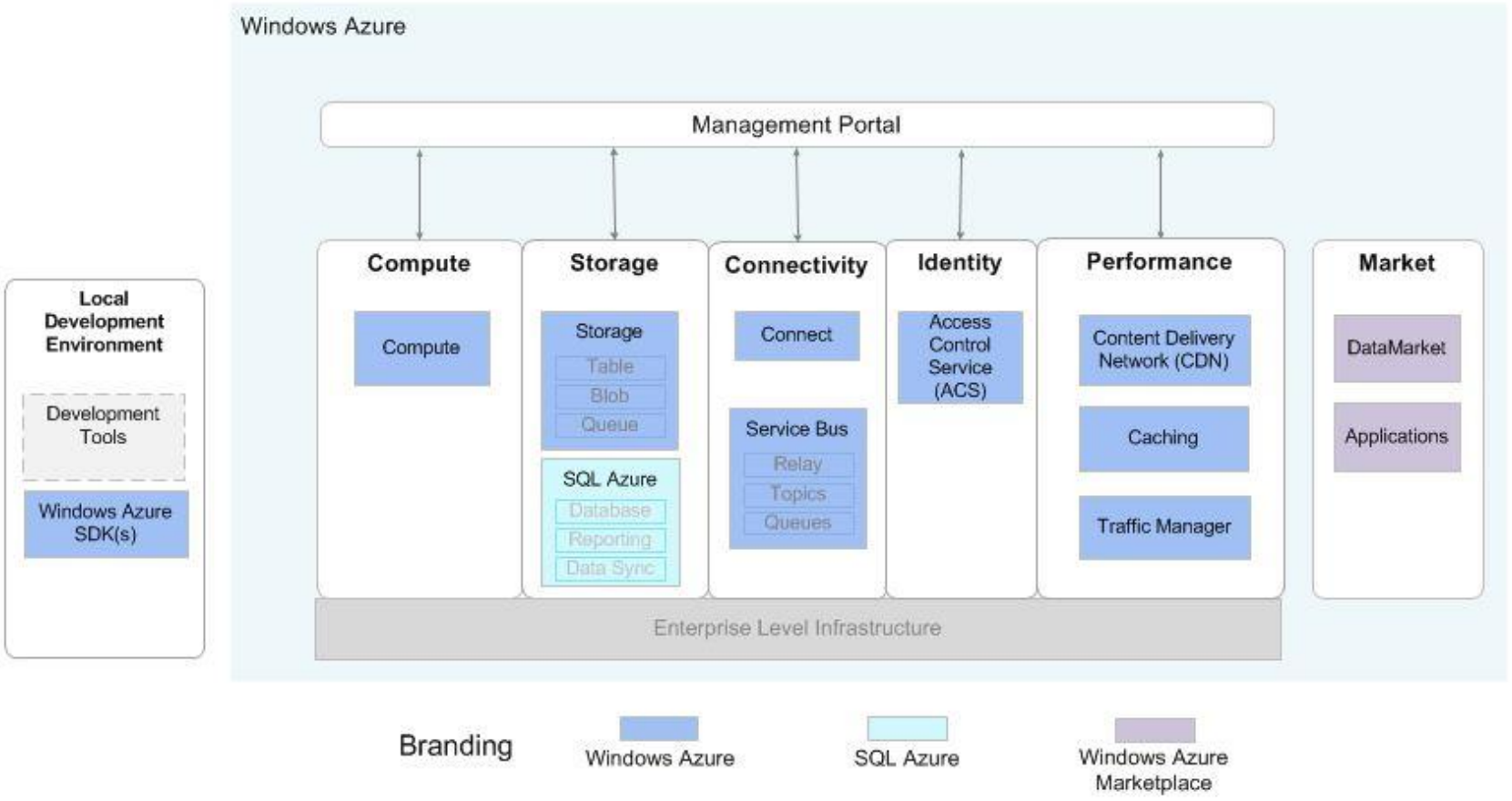
- Since Oct. 2008, commercial in Feb. 2010
- Languages: ASP.NET, PHP, Node.js, Java
- Internally Instance PaaS, VM Role only recently
  - Web Role – runs IIS server on W2k8 and Hyper-V
    - Application is a web server root directory
  - Woker Role – no IIS
    - Application is a package with EXE file
    - For services outside IIS, ie. Java, Node.js
  - VM Role – runs your VHD images of W2k8
    - Essentially IaaS

Azure - Pricing

- Instance PaaS – accounting by instance hours
- 5 instance types, prices exactly as Amazon + Win
  - Small: 1 CPU, 1,75 GB RAM, 230 GB disk, $0.12
  - ExtraSmall, 768GB RAM, 20 GB disk, $0.04
- Possible to buy discount packages
  - Like Amazon's Reserved Instances
- Interesting SLA – 99.95% only if you have 2 inst.
  - Automatic updates need reboot :-P

Azure – Services

- noSQL storage
  - Tables – variable row format, max. 255 attributes, manual sharding - “partition key”
  - Blobs – for storing large objects
    - Block blobs: 4 MB blocks, addressable individually or as a whole, max. size 200 GB, may use CDN
    - Page blobs: 512 B pages, sparse writes, snapshots
      - Azure Drives: page blobs containing VHD images
    - Queues: 8 kB XML messages for IPC
  - Counted together at $0.14 / 1 GB and $0.01 / 10k I/Os
- SQL Azure – modified MS SQL Server with clustering
  - Un-cloudlike flat fees, minimum $5 for max. 100 MB
- AppFabric – Access Control, SOA bus; Connect (VPN), Market (apps and datasets), Caching, MPI


Windows Azure
Management Portal
Compute
Storage
Connectivity
Identity
Performance
Market
Compute
Storage
Table
Blob
Queue
SQL Azure
Database
Reporting
Data Sync
Connect
Service Bus
Relay
Topics
Queues
Access Control Service (ACS)
Content Delivery Network (CDN)
Caching
Traffic Manager
DataMarket
Applications
Enterprise Level Infrastructure
Local Development Environment
Development Tools
Windows Azure SDK(s)
Branding
Windows Azure
SQL Azure
Windows Azure Marketplace

Azure – Limitations

- MS technologies.
  - Windows command line, T-SQL language, Caching is not memcached, MPI is not OpenMPI
  - Some may view it as an advantage
- A cloud classic – local storage is ephemeral
- No e-mail sending service
- Deployment requires writing an XML descriptor
  - Also needed for scaling – no built-in automation
- Prices higher than Amazon – need 2 instances
- Plugins for MS Visual Studio and Eclipse

Cloud Foundry

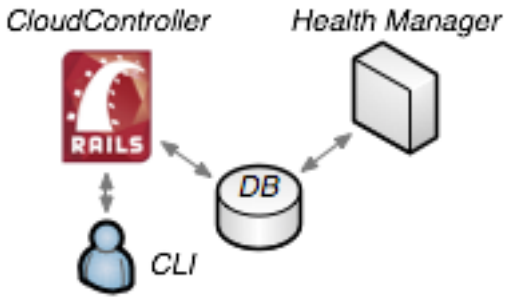
- Since beginning of 2011, still beta
- The only open source PaaS to date
  - There will be multiple providers, private clouds
- Supports Spring for Java, Ruby on Rails (and Sinatra), node.js, Grails and Scala on Lift
  - Typical Framework PaaS
    - Unlike Google, uses standard frameworks
  - More added by companies and community
    - PHP, Django on Python, Erlang
- Three versions: cloudfoundry.com, private, micro (in a VMware Player image)

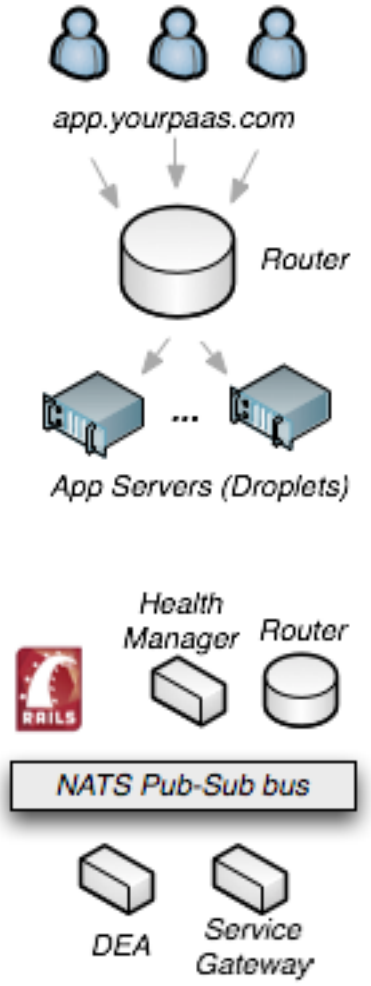
Cloudfoundry.com

- Public version is free in beta
- Limits:
  - 2 GB total memory per user
    - Apps can have from 64 MB to 2 GB per instance
  - Maximum 20 applications
  - 128 MB data in MySQL
  - 16 MB Redis, 240 MB MongoDB
- Which sums up the services, except RabbitMQ
  - More may be added later

CloudFoundry – Structure

- Droplet Execution Agent
- Router
- Cloud Controller
- Health Manager
- NATS message bus





Cloud Foundry – Limitations

- Not significant if programmer adheres to framework's design patterns
  - For Java – contains Tomcat application server
- Low Security – only standard UNIX user/group
  - A local exploit will compromise the DEA, exposing other users' apps and data
  - Nimbula IaaS – launches a whole CF for each user
  - ActiveState Stackato – a derivative using LXC
    - (Iron Foundry – a derivative running .NET)
- Scaling is manual, local storage ephemeral
- Eclipse plugin available, "vmc" CLI tool