

Cloud Computing

Prof. Dr. Stefan Tai
stefan.tai@kit.edu



KIT – The cooperation of Forschungszentrum Karlsruhe GmbH and Universität Karlsruhe (TH)

About the Speaker: Stefan Tai



- Professor,
Karlsruhe Institute of Technology (KIT), Germany
 - Institute for Applied Informatics (AIFB)
 - Karlsruhe Service Research Institute (KSRI)
- Director,
FZI Research Center for Information Technology in
Karlsruhe, Germany

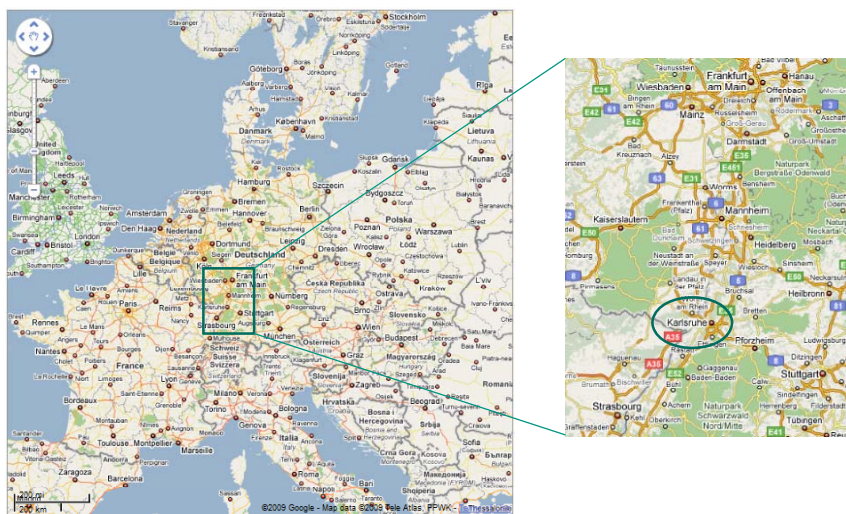
From 1999-2007:

- Research Staff Member, IBM T.J. Watson Research Center, New York, USA

Prior to 1999:

- Eurocontrol Experimental Center, Paris, France
- Fraunhofer ISST, Berlin, Germany
- TU Berlin, Germany

Karlsruhe, Germany



eOrganization

Overview
People
Research
Teaching

News archive

Shortcuts

Teaching Winter 2009

These offers

Upcoming events

Welcome to eOrganization - a new research group dedicated to explore challenging research problems in the field of service computing.

Our multi-disciplinary group directs its research and teaching activities to the Internet in its combined role as a platform for technical, economic, organizational and social networks.

Driven by the critical importance of the Web in supporting our increasingly service-led economy, we are committed to devise new concepts and technology to support the design, development, operation, utilization, and impact analysis of software service infrastructures, platforms and applications.

The eOrganization research group was one of the founders of the Karlsruhe Service Research Institute and provides a clear demonstration of KIT's vision of a new culture of sustainable innovation.

eOrganization focus research areas are:

- Cloud Computing & Cloud Service Engineering
- Situational Applications & Collaborative Services
- Service Value Networks & Service Communities

The research group is part of:

- Karlsruhe Service Research Institute (KSRI), KIT
- Institute for Applied Informatics (AIFB), KIT
- Information and Process Engineering Department (IPE), FZI

You are currently not logged in - Login

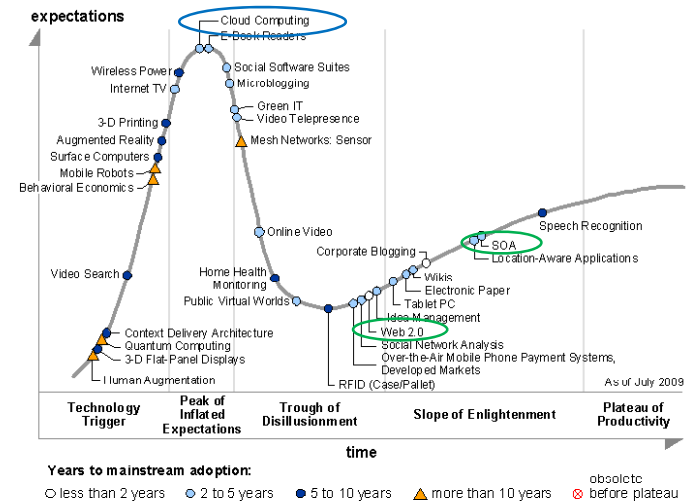


www.eOrganization.de

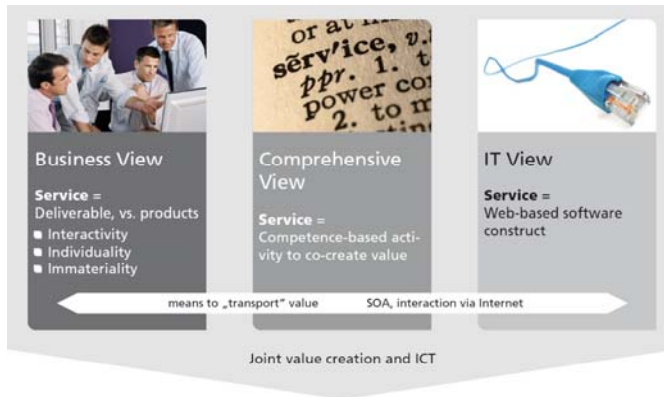
What is Cloud Computing?



Gartner's Hype Cycle of Emerging Technologies, July 2009



First of all: What are Services?



Cloud Computing: Infrastructure, Platforms, and Software as Services

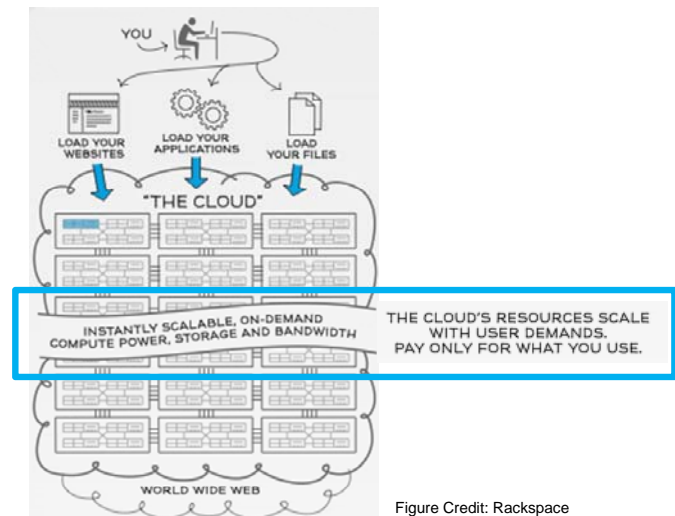
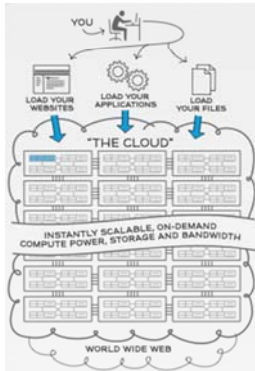


Figure Credit: Rackspace

...set in multiple contexts



↔ Enterprise Computing

↔ Web Computing

↔ Cloud Economics

Cloud Computing is about:



Enterprise-grade systems management

- Under-utilized server resources waste computing power and energy
- Over-utilized servers cause interruption or degradation of service levels

Cloud Computing is about:



Internet-scale service computing

- Provide and consume sophisticated infrastructure, platforms and business applications as modular (Web) services
- Disrupt traditional industries and offer rich, highly dynamic experiences



Enterprise-grade systems management

- Under-utilized server resources waste computing power and energy
- Over-utilized servers cause interruption or degradation of service levels

Cloud Computing is about:



Understanding business opportunities

- Faster time-to-market, and cost-effective innovation processes
- Dynamic (trans-)formation of open service and business networks
- Leveraging the participation Web and mass programming



Internet-scale service computing

- Provide and consume sophisticated infrastructure, platforms and business applications as modular (Web) services
- Disrupt traditional industries and offer rich, highly dynamic experiences



Enterprise-grade systems management

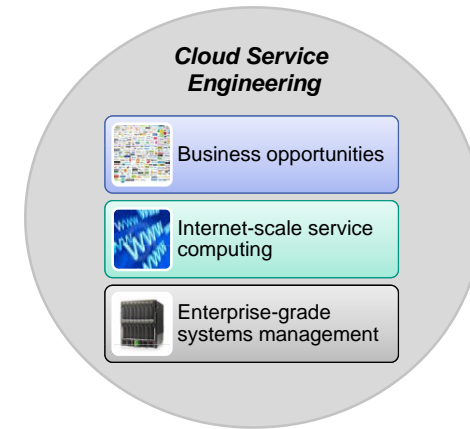
- Under-utilized server resources waste computing power and energy
- Over-utilized servers cause interruption or degradation of service levels

Our Definition

“Building on compute and storage virtualization, **cloud computing** provides scalable, network-centric, abstracted IT infrastructure, platforms, and applications as on-demand services that are billed by consumption.”

“**Cloud service engineering** leverages cloud computing in the context of the Internet in its combined role as a platform for technical, economic, organizational and social networks.”

To keep in mind: Three Dimensions



Clouds vs. Grids

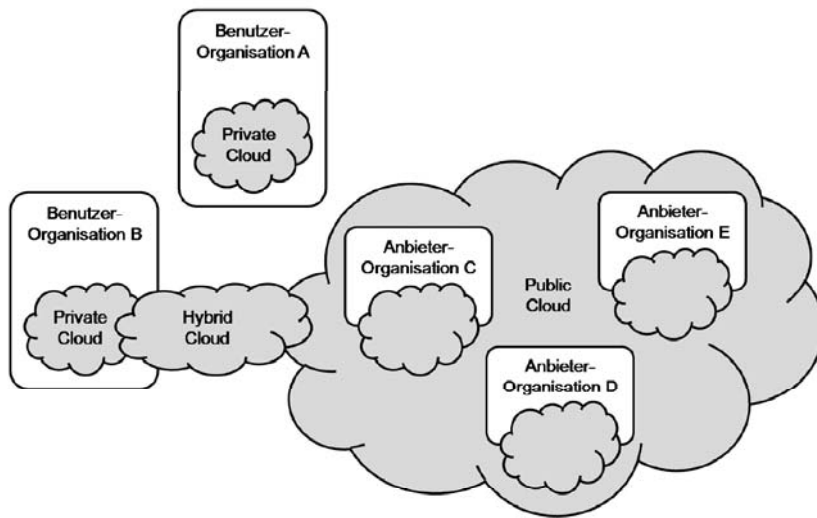
	Cloud Computing	Grid Computing	
Functional	Objective	Provide desired computing platform via network enabled services	Resource sharing Job execution
	Infrastructure	One or few data centers, heterogeneous/homogeneous resource under central control, Industry and Business	Geographically distributed, heterogeneous resource, no central control, VO Research and academic organization
	Middleware	Proprietary, several reference implementations exist (e.g. Amazon)	Well developed, maintained and documented
	Application	Suited for generic applications	Special application domains like High Energy Physics
	User interface	Easy to use/deploy, no complex user interface required	Difficult use and deployment Need new user interface, e.g., commands, APIs, SDKs, services ...
Non-Functional	Business Model	Commercial: Pay-as-you-go	Publicly funded: Use for free
	Operational Model	Industrialization of IT Fully automated Services	Mostly Manufacture Handcrafted Services
	QoS	Possible	Little support
	On-demand provisioning	Yes	No

(Thanks to M. Kunze, KIT)

Cloud Architecture and Ecosystem

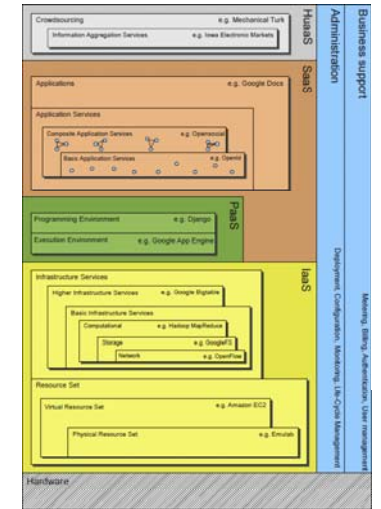


Organizational Cloud Architecture: Public-/Hybrid-/Private-Cloud

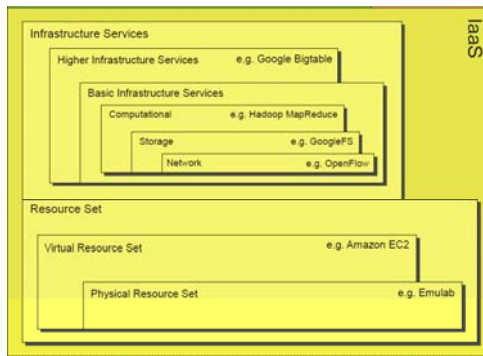


Technical Cloud Architecture: Cloud Computing Stack

- Generic Approach
- Layered architecture
- **Everything as a Service** concept
 - Standard layers
 - Infrastructure as a Service
 - Platform as a Service
 - Software as a Service
 - Extra Layers
 - Human as a Service
 - Administration/Business Support



Infrastructure as a Service



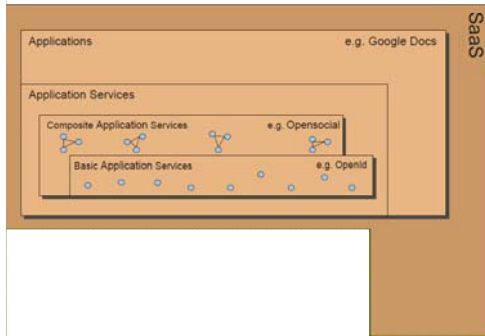
- **Infrastructure Services**
 - Storage
 - Computational
 - Network
 - Database
 - e.g. Google Bigtable, GoogleFS, Hadoop MapReduce, HadoopFS
- **Resource Set**
 - Machine Images
 - e.g. EC2, Eucalyptus

Platform as a Service



- **Programming Environment**
 - Programming Language, Libraries
 - e.g. Django, Java
- **Execution Environment**
 - Runtime Environment
 - e.g. Google App Engine, Java Virtual Machine

Software as a Service



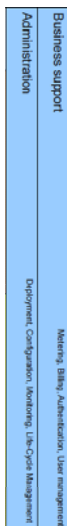
- Applications
 - User Interface
 - Frontend Application
 - e.g. Google Docs, Yahoo Email
- Application Services
 - Webservices Interface
 - Basic or Composite
 - e.g. Opensocial, Google Maps

Human as a Service



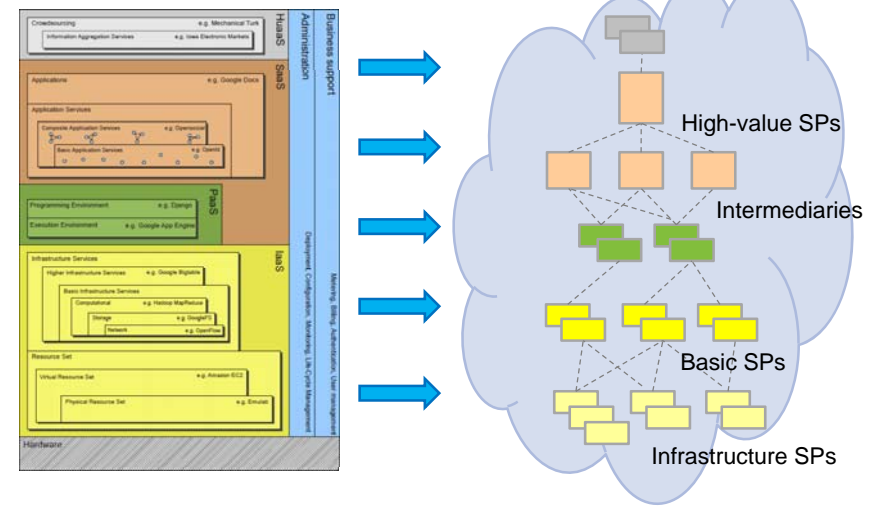
- Crowdsourcing
 - Enabling Collective Intelligence
 - e.g. Mechanical Turk
- Information Markets
 - Prediction of events
 - e.g. Iowa Electronic Markets

Administration/Business Support



- Available on all layers
- Administration
 - Deployment
 - Configuration
 - Monitoring
 - Life cycle management
- Business support
 - Metering
 - Billing
 - Authentication
 - User management

Cloud Architecture → Cloud Players



Players



- Cloud **infrastructure service providers** – raw cloud resources
IaaS (infrastructure-as-a-service)
- Cloud **platform providers** – resources + frameworks; PaaS (platform-as-a-service)
- Cloud **intermediaries** – help broker some aspect of raw resources and frameworks, e.g.,
 - server managers, application assemblers, application hosting
- Cloud **application providers** (SaaS)
- Cloud **consumers** – users of the above

[Thanks to M. Maximilien, IBM]



Players: Providers



- Programmatic access via Web Services and/or Web APIs
- “Pure” virtualized resources
 - CPU, memory, storage, and bandwidth
 - Data store



Joyent



Versus

- Virtualized resources plus application framework
- (e.g., RoR, Python, .NET)
 - Imposes an application and data architecture
 - Constrains how application is built



Google App Engine



[Thanks to M. Maximilien, IBM]



Players: Cloud Intermediaries



- Resells (aspects of) raw cloud resources, with **added value** propositions
 - Packaging resources as bundles
 - Facilitating cloud resource management, e.g., setup, updates, backup, load balancing, etc.
 - Providing tools and dashboards
- Enabler of the cloud ecosystem



[Thanks to M. Maximilien, IBM]



Players: Application Providers



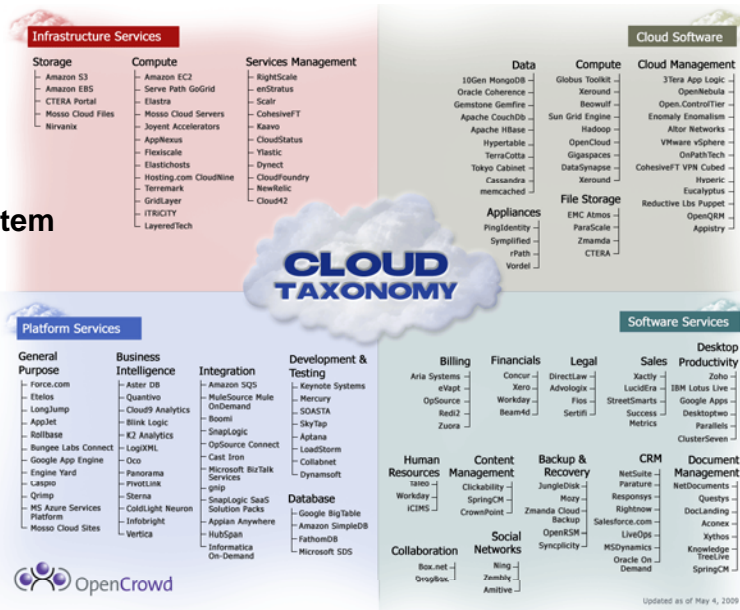
- Software as a Service (SaaS):
Applications provided and consumed over the Web
- Infrastructure usage (mostly) hidden



[Thanks to M. Maximilien, IBM]



The Cloud Ecosystem



Cloud Computing by example: AWS



Cloud computing by example: AWS



Amazon Web Services (AWS) Cloud Offerings:

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Queuing Service (Amazon SQS)
- Amazon SimpleDB

- Amazon Elastic MapReduce
- Amazon CloudFront
- Amazon DevPay
- AWS Import/Export



Amazon Simple Queue Service (SQS)



[About AWS](#) |
 [Products](#) |
 [Solutions](#) |
 [Resources](#) |
 [Support](#) |
 [Your Account](#)

- Infrastructure Services
 - Amazon Elastic Compute Cloud (Amazon EC2)
 - Amazon SimpleDB
 - Amazon Simple Storage Service (Amazon S3)
 - Amazon CloudFront
 - Amazon Simple Queue Service (Amazon SQS)
 - AWS Premium Support

- Payments & Billing
- On-Demand Workforce
- Alexa Web Services
- Amazon Fulfillment & Associates

- Related Services
 - AWS Premium Support
Gain direct access to our technical support team.

Home > Products > Amazon Simple Queue Service (Amazon SQS)

Amazon Simple Queue Service (Amazon SQS)

Amazon Simple Queue Service (Amazon SQS) offers a reliable, highly scalable, hosted queue for storing messages as they travel between computers. By using Amazon SQS, developers can simply move data between distributed components of their applications that perform different tasks, without losing messages or requiring each component to be always available. Amazon SQS makes it easy to build an automated workflow, working in close conjunction with the Amazon Elastic Compute Cloud (Amazon EC2) and the other AWS infrastructure web services.

Amazon SQS works by exposing Amazon's web-scale messaging infrastructure as a web service. Any computer on the Internet can add or read messages without any installed software or special firewall configurations. Components of applications using Amazon SQS can run independently, and do not need to be on the same network, developed with the same technologies, or running at the same time.

[Sign Up For Amazon SQS](#)

Amazon Simple Queue Service (SQS)



- “Message queuing in the Cloud”
 - Basic message queuing model, except: queues are hosted by Amazon, and queues are accessed using Web service protocols
- Simple API
- Platform agnostic
- Basic support for access control and message locking

- Reliability
 - Runs within Amazon's high-availability data centers
 - Messages stored redundantly across multiple servers and locations
- Scalable to millions of messages a day

SQS Functionality



- Developers can create an unlimited number of Amazon SQS queues, each of which can send and receive an unlimited number of messages.
- New messages can be added to a queue at any time. The message body can contain up to 8 KB of text in any format.
- A computer can check a queue at any time for messages waiting to be read.
- A message is “locked” while a computer is processing it, keeping other computers from trying to process it simultaneously. If processing fails, the lock will expire and the message will again be available.
- Messages can be retained in queues for up to 4 days.
- Developers can access Amazon SQS through standards-based SOAP and Query interfaces designed to work with any Internet-development toolkit.

Source: aws.amazon.com

SQS API



- CreateQueue: Create queues for use with your AWS account.
- ListQueues: List your existing queues.
- DeleteQueue: Delete one of your queues.
- SendMessage: Add any data entries to a specified queue.
- ReceiveMessage: Return one or more messages from a specified queue.
- DeleteMessage: Remove a previously received message from a specified queue.
- SetQueueAttributes: Control queue settings like the amount of time that messages are locked after being read so they cannot be read again.
- GetQueueAttributes: See information about a queue like the number of messages in it.

Source: aws.amazon.com

Sample SOAP request of sending a message



```
POST /MyQueue HTTP/1.1
Host: queue.amazonaws.com
<other HTTP headers here...>

<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header
    xmlns:aws="http://security.amazonaws.com/doc/2007-01-01/">
    <aws:AWSAccessKeyId>ID9FVRAYCP1VJS767E02EXAMPLE</aws:AWSAccessKeyId>
    <aws:Timestamp>2008-02-10T23:59:59Z</aws:Timestamp>
    <aws:Signature>SZf1CHmQ/nrZbsrC13hCZS061ywsEXAMPLE</aws:Signature>
  </soapenv:Header>
  <soapenv:Body>
    <SendMessage xmlns="http://queue.amazonaws.com/doc/2008-01-01">
      <MessageBody>This is my message</MessageBody>
    </SendMessage>
  </soapenv:Body>
</soapenv:Envelope>
```

Source: aws.amazon.com

SQS Pricing



Pricing

Pay only for what you use. There is no minimum fee. Estimate your monthly bill using AWS Simple Monthly Calculator.

Please note that Amazon SQS introduced a new WSDL and pricing plan on February 6, 2008.

[Pricing for the current WSDL](#) (version 2008-01-01)

Requests

- \$0.01 per 10,000 Amazon SQS Requests (\$0.000001 per Request)

Amazon SQS requests are CreateQueue, ListQueues, DeleteQueue, SendMessage, ReceiveMessage, DeleteMessage, SetQueueAttributes and GetQueueAttributes

Data Transfer

- \$0.100 per GB – all data transfer in
- \$0.170 per GB – first 10 TB / month data transfer out
- \$0.130 per GB – next 40 TB / month data transfer out
- \$0.110 per GB – next 100 TB / month data transfer out
- \$0.100 per GB – data transfer out / month over 150 TB

Data transfer "in" and "out" refers to transfer into and out of Amazon SQS. Data transferred between Amazon SQS and Amazon EC2 in the U.S. is free of charge (i.e., \$0.00 per GB). Data transferred between Amazon SQS and Amazon EC2 in the E.U. will be charged at Internet Data Transfer rates on both sides of the transfer.

Source: aws.amazon.com

Amazon Simple Storage Service (S3)



Contact Us | Create an AWS Account

About AWS | Products | Solutions | Resources | Support | Your Account

Infrastructure Services

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon SimpleDB
- Amazon Simple Storage Service (Amazon S3)
- Amazon CloudFront
- Amazon Simple Queue Service (Amazon SQS)
- AWS Premium Support

Payments & Billing

On-Demand Workforce

Alexa Web Services

Amazon Fulfillment & Associates

Not a Developer?

- ↳ Looking for a consumer or business storage solution?
- ↳ Search for consumer storage solutions built on

Home > Products > Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3)

Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers.

[Sign Up For Amazon S3](#)

Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

This page contains the following categories of information. Click to jump down:

↳ [Amazon S3 Functionality](#)

↳ [Pricing](#)

↳ [Resources](#)

↳ [Amazon S3 Design Requirements](#)

↳ [Amazon S3 Design Principles](#)

↳ [Intended Usage and Restrictions](#)

Amazon S3



- Write, read, and delete objects containing from 1 byte to 5 gigabytes of data each. The number of objects you can store is unlimited.
- Each object is stored in a bucket and retrieved via a unique, developer-assigned key.
- A bucket can be located in the United States or in Europe. All objects within the bucket will be stored in the bucket's location, but the objects can be accessed from anywhere.
- Authentication mechanisms are provided to ensure that data is kept secure from unauthorized access. Objects can be made private or public, and rights can be granted to specific users.
- Uses standards-based REST and SOAP interfaces designed to work with any Internet-development toolkit.
- Built to be flexible so that protocol or functional layers can easily be added. Default download protocol is HTTP. A BitTorrent™ protocol interface is provided to lower costs for high-scale distribution. Additional interfaces will be added in the future.
- Reliability backed with the [Amazon S3 Service Level Agreement](#).

Source: aws.amazon.com

S3 Design Principles



- **Decentralization:** Use fully decentralized techniques to remove scaling bottlenecks and single points of failure.
- **Asynchrony:** The system makes progress under all circumstances.
- **Autonomy:** The system is designed such that individual components can make decisions based on local information.
- **Local responsibility:** Each individual component is responsible for achieving its consistency; this is never the burden of its peers.
- **Controlled concurrency:** Operations are designed such that no or limited concurrency control is required.
- **Failure tolerant:** The system considers the failure of components to be a normal mode of operation, and continues operation with no or minimal interruption.

S3 Design Principles (cont.)



- **Controlled parallelism:** Abstractions used in the system are of such granularity that parallelism can be used to improve performance and robustness of recovery or the introduction of new nodes.
- **Decompose** into small well-understood building blocks: Do not try to provide a single service that does everything for everyone, but instead build small components that can be used as building blocks for other services.
- **Symmetry:** Nodes in the system are identical in terms of functionality, and require no or minimal node-specific configuration to function.
- **Simplicity:** The system should be made as simple as possible (but no simpler).

Sample Code: Developer Community



Home > ... > Amazon Simple Storage Service > Sample Code & Libraries

Category: Sample Code & Libraries

Want to share your code? Submit your code.

Documents in Category Go to Page: [1 2 3 4 5] Next]

Sort by: [Select] Filter by Document Type: [Select]

- 1) **Manager for Amazon CloudFront**
A Windows application for uploading files to Amazon S3 and distributing them through Amazon CloudFront.
Last Modified: Jan 2, 2009 2:16 PM
- 2) **hS3: Haskell S3 Library**
hS3 is a Haskell library for accessing Amazon S3.
Last Modified: Jan 2, 2009 12:00 PM
- 3) **LitS3 - A Small and Fast C# Library for Amazon S3**
LitS3 is a library written in C# that provides comprehensive and straightforward access to Amazon S3 for .NET developers.
Last Modified: Oct 15, 2008 9:24 AM
- 4) **Code Sample for "Using S3, EC2, SQS, Lucene and Ruby for Web Spidering"**
This code sample goes along with the "Using S3, EC2, SQS, Lucene and Ruby for Web Spidering" article. The article and accompanying code show how Ruby and Rails developers can use freely available tools and simple techniques to spider sites and then,
Last Modified: Sep 21, 2008 1:51 PM
- 5) **Programming Amazon Web Services - Example Code**
Example code from the book Programming Amazon Web Services.
Last Modified: Sep 21, 2008 1:24 PM
- 6) **Command-line Access to S3**
A Perl script to access S3 from the command line. Has commands to work with S3 "ourelly" or

Sample S3 REST Usage

- Use standard HTTP requests to create, fetch, and delete buckets and objects
- A typical REST operation consists of a sending a single HTTP request to Amazon S3, followed by waiting for an HTTP response. Like any HTTP request, a request to Amazon S3 contains a request method, a URI, request headers, and sometimes a query string and request body. The response contains a status code, response headers, and sometimes a response body.
- Following is an example that shows how to get an object named "Nelson" from the "quotes" bucket.

```
GET /Nelson HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: AWS
15B4D3461F177624206A:xQE0diMbLRrepd3YB+FIEXAMPLE=
```

```
HTTP/1.1 200 OK
x-amz-id-2:
qBmkRcEwBBhH6XAqsKU/eg24V3jfkWKN9dJip1L/FpbYr9FDy7wWfufdQOEMcY
x-amz-request-id: F2A8CCCA26B4B26D
Date: Wed, 01 Mar 2006 12:00:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "828ef3fafa96f00ad9f27c383fc9ac7f"
Content-Type: text/plain
Content-Length: 5
Connection: close
Server: AmazonS3
```

ha-ha

Source: aws.amazon.com

S3 Pricing

United States

Storage

- \$0.150 per GB - first 50 TB / month of storage used
- \$0.140 per GB - next 50 TB / month of storage used
- \$0.130 per GB - next 400 TB / month of storage used
- \$0.120 per GB - storage used / month over 500 TB

Data Transfer

- \$0.100 per GB - all data transfer in
- \$0.170 per GB - first 10 TB / month data transfer out
- \$0.130 per GB - next 40 TB / month data transfer out
- \$0.110 per GB - next 100 TB / month data transfer out
- \$0.100 per GB - data transfer out / month over 150 TB

Requests

- \$0.01 per 1,000 PUT, COPY, POST, or LIST requests
- \$0.01 per 10,000 GET and all other requests*

* No charge for delete requests

Europe

Storage

- \$0.180 per GB - first 50 TB / month of storage used
- \$0.170 per GB - next 50 TB / month of storage used
- \$0.160 per GB - next 400 TB / month of storage used
- \$0.150 per GB - storage used / month over 500 TB

Data Transfer

- \$0.100 per GB - all data transfer in
- \$0.170 per GB - first 10 TB / month data transfer out
- \$0.130 per GB - next 40 TB / month data transfer out
- \$0.110 per GB - next 100 TB / month data transfer out
- \$0.100 per GB - data transfer out / month over 150 TB

Requests

- \$0.012 per 1,000 PUT, COPY, POST, or LIST requests
- \$0.012 per 10,000 GET and all other requests*

* No charge for delete requests

Source: aws.amazon.com

Data transfer "in" and "out" refers to transfer into and out of an Amazon S3 location (i.e., US or EU).
Data transferred within an Amazon S3 location via a COPY request is free of charge.
Data transferred between locations is charged at regular rates.
Data transferred between Amazon EC2 and Amazon S3 within the same region is free of charge (i.e., \$0.00 per GB). Data transferred between Amazon EC2 and Amazon S3 across regions (i.e. between US and EU), will be charged at Internet Data Transfer rates on both sides of the transfer.
Storage and bandwidth size includes all file overhead.

Amazon EC2



Contact Us | Create an AWS Account

About AWS | Products | Solutions | Resources | Support | Your Account

Infrastructure Services

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon SimpleDB
- Amazon Simple Storage Service (Amazon S3)
- Amazon CloudFront
- Amazon Simple Queue Service (Amazon SQS)
- AWS Premium Support

Payments & Billing

On-Demand Workforce

Alexa Web Services

Amazon Fulfillment & Associates

Running Databases on EC2

Find tips and technical resources for running databases on Amazon EC2.

Home > Products > Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon provides web-s

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 configures control to obtain, quickly required by all EC2 pr and is

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

This p

Amazon EC2



- EC2 presents a **true virtual computing environment**, allowing you to use web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage your network's access permissions, and run your image using as many or few systems as you desire.
- Create an Amazon Machine Image (AMI) containing your applications, libraries, data and associated configuration settings. Or use pre-configured, templated images to get up and running immediately.
- Upload the AMI into Amazon S3. Amazon EC2 provides tools that make storing the AMI simple. Amazon S3 provides a safe, reliable and fast repository to store your images.
- Use Amazon EC2 web service to configure security and network access.
- Choose which instance type(s) and operating system you want, then start, terminate, and monitor as many instances of your AMI as needed, using the web service APIs or the variety of management tools provided.
- Determine whether you want to run in multiple locations, utilize static IP endpoints, or attach persistent block storage to your instances.
- Pay only for the resources that you actually consume, like instance-hours or data transfer.

EC2 Instance Types



Instance Types

Standard Instances

Instances of this family are well suited for most applications.

- Small Instance (Default) 1.7 GB of memory, 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit), 160 GB of instance storage, 32-bit platform
- Large Instance 7.5 GB of memory, 4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each), 850 GB of instance storage, 64-bit platform
- Extra Large Instance 15 GB of memory, 8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each), 1690 GB of instance storage, 64-bit platform

High-CPU Instances

Instances of this family have proportionally more CPU resources than memory (RAM) and are well suited for compute-intensive applications.

- High-CPU Medium Instance 1.7 GB of memory, 5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each), 350 GB of instance storage, 32-bit platform
- High-CPU Extra Large Instance 7 GB of memory, 20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each), 1690 GB of instance storage, 64-bit platform

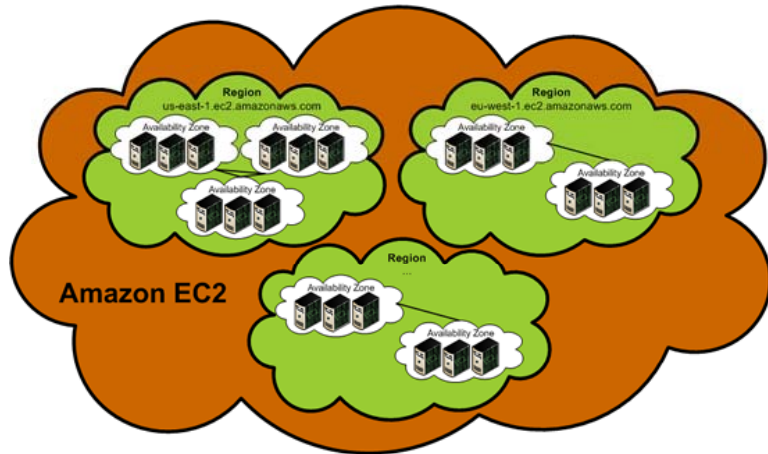
EC2 Compute Unit (ECU) - One EC2 Compute Unit (ECU) provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

EC2 Core Concepts



- Amazon Machine Image (AMI):** an encrypted file stored in Amazon S3, containing all the information necessary to boot instances of a customer's software
 - An AMI is like a bootable root disk, which can be pre-defined or user-built.
 - Public AMIs: Pre-configured, template AMIs
 - Private AMIs: User-built AMI containing private applications, libraries, data and associated configuration settings
- Instance:** The running system based on an AMI
 - All instances based on the same AMI begin executing identically. An instance can be launched in very few minutes. Any information on them is lost when the instances are terminated or if they fail.

EC2 Regions & Availability Zones



EC2 Operating Systems

Amazon Machine Images (AMIs) are preconfigured with an ever-growing list of operating systems. We work with our partners and community to provide you with the most choice possible. You are also empowered to use our bundling tools to upload your own operating systems. The operating systems currently available to use with your Amazon EC2 instances include:

Operating Systems

Red Hat Enterprise Linux	Windows Server 2003	Oracle Enterprise Linux
OpenSolaris	openSUSE Linux	Ubuntu Linux
Fedora	Gentoo Linux	Debian

EC2 Software Appliances

Amazon EC2 enables our partners and customers to build and customize Amazon Machine Images (AMIs) with software based on your needs. We have hundreds of free and paid AMIs available for you to use. A small sampling of the software available for use today within Amazon EC2 includes:

Databases	Batch Processing	Web Hosting
IBM DB2	Hadoop	Apache HTTP
IBM Informix Dynamic Server	Condor	IIS/Asp.Net
MySQL Enterprise	Open MPI	IBM Lotus Web Content Management
Oracle 11g		IBM WebSphere Portal Server
Microsoft SQL Server Standard 2005		
Application Development Environments	Video Encoding & Streaming	
Java Application Server	Wowza Media Server Pro	
JBoss Enterprise Application Platform	Windows Media Server	
IBM sMash	Windows Media Server	
Ruby on Rails		
Oracle WebLogic Server		

EC2 Pricing: On-demand Instance

United States	Europe	
Standard On-Demand Instances	Linux/UNIX Usage	Windows Usage
Small (Default)	\$0.11 per hour	\$0.135 per hour
Large	\$0.44 per hour	\$0.54 per hour
Extra Large	\$0.88 per hour	\$1.08 per hour
High CPU On-Demand Instances	Linux/UNIX Usage	Windows Usage
Medium	\$0.22 per hour	\$0.32 per hour
Extra Large	\$0.88 per hour	\$1.28 per hour

EC2 Pricing: Reserved Instance



United States Europe

Linux/UNIX		One-time Fee		
Standard Reserved Instances	1 yr Term	3 yr Term	Usage	
Small (Default)	\$325	\$500	\$0.04 per hour	
Large	\$1300	\$2000	\$0.16 per hour	
Extra Large	\$2600	\$4000	\$0.32 per hour	
High CPU Reserved Instances	1 yr Term	3 yr Term	Usage	
Medium	\$650	\$1000	\$0.08 per hour	
Extra Large	\$2600	\$4000	\$0.32 per hour	

EC2 Data Transfer



Internet Data Transfer

The pricing below is based on data transferred "in" and "out" of Amazon EC2.

Data Transfer In

All Data Transfer	\$0.10 per GB
-------------------	---------------

Data Transfer Out

First 10 TB per Month	\$0.17 per GB
Next 40 TB per Month	\$0.13 per GB
Next 100TB per Month	\$0.11 per GB
Over 150 TB per Month	\$0.10 per GB

- No costs within availability zones
- \$0.01/GB between regions
- \$0.01/GB for public and elastic IP data transfer

Elastic Block Store (EBS)



United States Europe

Amazon EBS Volumes

- \$0.11 per GB-month of provisioned storage
- \$0.11 per 1 million I/O requests

Amazon EBS Snapshots to Amazon S3 (priced the same as Amazon S3)

- \$0.18 per GB-month of data stored
- \$0.012 per 1,000 PUT requests (when saving a snapshot)
- \$0.012 per 10,000 GET requests (when loading a snapshot)

- For permanent data storage
- Uses S3

Elastic IP Addresses



Dynamic Mapping of IPs to virtual machines

No cost for Elastic IP addresses while in use

- \$0.01 per non-attached Elastic IP address per complete hour
- \$0.00 per Elastic IP address remap – first 100 remaps / month
- \$0.10 per Elastic IP address remap – additional remap / month over 100

Amazon Management Console (1): https://console.aws.amazon.com/



Instance	AMI ID	Security Groups	Type	Status	Public DNS	Key Pair Name
i-a32d2a7	ami-e402790	default_cloud	m1.small	running	ec2-79-125-10-186.eu-west-1.cloud	
i-972d2ae3	ami-8cae29c	vorlesung	m1.small	running	ec2-79-125-62-58.eu-west-1.c	vorlesung
i-4729e33	ami-2e0275a	default	m1.large	running	ec2-79-125-62-52.eu-west-1.c	vorlesung

AMI ID: ami-e402790
Zone: eu-west-1b
Security Groups: default_cloud
Type: m1.small
Status: running
Owner: 521141848536
Reservation: r-4b5b5c3f
Platform: -
Kernel ID: aki-7e0d230e
Elastic IP: -
Public DNS: ec2-79-125-10-186.eu-west-1.compute.amazonaws.com
Private DNS: ip-10-224-63-37.eu-west-1.compute.internal

Amazon Management Console (2): AMIs



AMI ID	Manifest	Visibility	Platform
ami-08ca27c	ec2-public-windows-image-eu-DE-SqSvExp20032-x86_64-Win1.02.manifest.xml	Public	Windows
ami-08ca27e	ec2-public-windows-image-eu-DE-SqSvExp20032-x86_64-WinAuth-v1.02.manifest.xml	Public	Windows
ami-10163a54	aws-toolkit-for-ec2-ami-aws-eu-haproxy-v1.0.2.manifest.xml	Public	Other Linux
ami-20163a5c	aws-console-quickstart-ami-eu-west-1-nubyclickstart.manifest.xml	Public	Other Linux
ami-20163a5c	aws-toolkit-for-ec2-ami-aws-eu-omcat-v1.0.0.manifest.xml	Public	Other Linux
ami-2a0275a	ec2-public-images-eu-fedora-9-i386-base-v1.08.manifest.xml	Public	Fedora
ami-2a0275a	ec2-public-images-eu-fedora-9-x86_64-base-v1.08.manifest.xml	Public	Fedora
ami-38163a4c	aws-console-quickstart-ami-eu-west-1-ghpquickstart.manifest.xml	Public	Other Linux
ami-5206252b	ec2-public-images-eu-fedora-9-i386-base-v1.07.manifest.xml	Public	Fedora
ami-5602529	ec2-public-images-eu-demo-pad-AMI-v1.07.manifest.xml	Public	Other Linux
ami-64ca210	ec2-public-windows-image-eu-EN-SqSvExp20032-i386-WinAuth-v1.02.manifest.xml	Public	Windows
ami-70ca204	ec2-public-windows-image-eu-EN-SqSvExp20032-i386-WinAuth-v1.02.manifest.xml	Public	Windows

Amazon Management Console (3): EBS



Volume ID	Capacity	Snapshot	Created	Zone	Status	Attachment Information
vol-69e10461	10 GB	--	2009-05-09 10:13 UTC+0200	eu-west-1b	in-use	i-972d2ae3 /dev/sdf (attached)
vol-c6e104af	10 GB	--	2009-05-09 09:50 UTC+0200	eu-west-1b	in-use	i-a32d2a7 /dev/sdf (attached)

0 Elastic Block Store Volumes selected
 Select an EBS volume above to view information about it here

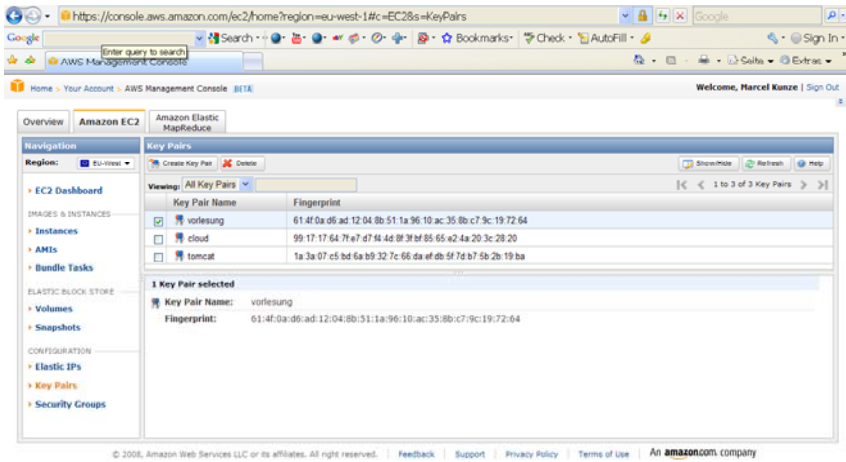
Amazon Management Console (4): Elastic IPs



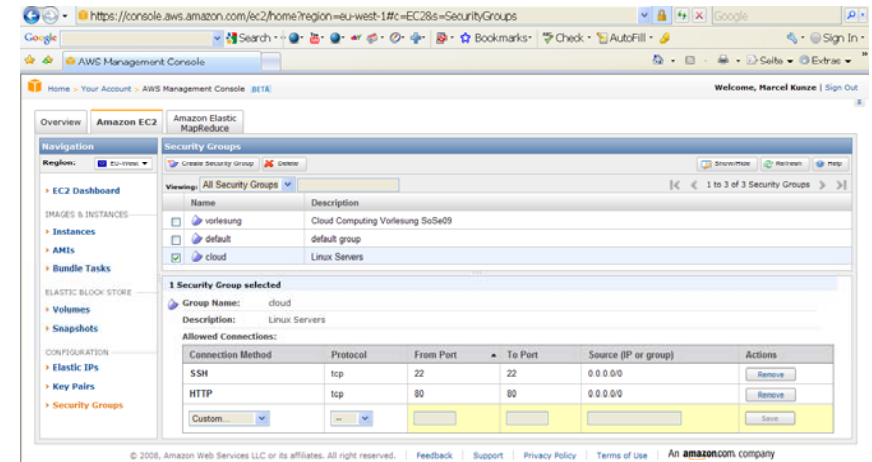
Address	Instance ID
79.125.10.186	i-a32d2a7

1 Elastic IP selected
 Address: 79.125.10.186
 Instance ID: i-a32d2a7

Amazon Management Console (5): Keys



Amazon Management Console (6): Security Groups/Firewall



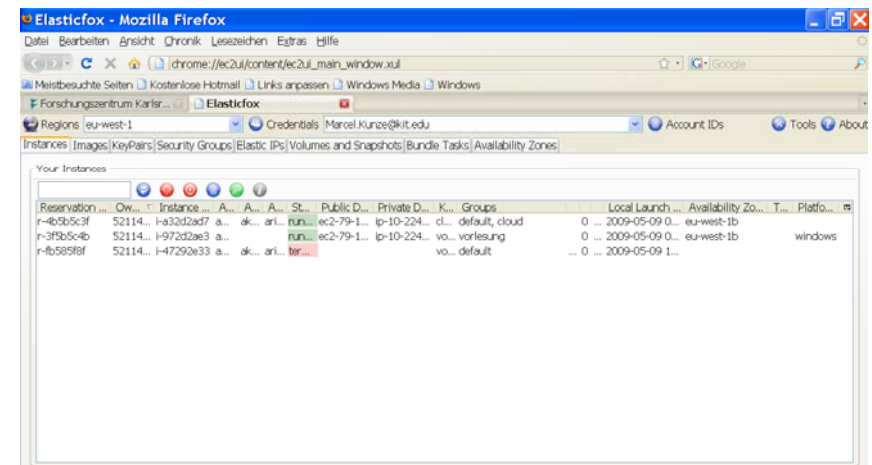
Amazon Command Line Tools



AMI Tools	Image Tools	Key Pair Tools
ec2-bundle-image Creates a bundled AMI from an operating system image created in a loopback file. ec2-bundle-image -k private_key -s user_id -c user_id -i image_path -r [size tag set] [-d destination] [-p aws_profile] [-s source] [-k kernel_kernel_id] [-r ramdisk_ramdisk_id] [-b block-device-mapping_block_device_mapping]	ec2-register Registers the AMI specified in the manifest file and generates a new AMI ID. ec2-register manifest	ec2-add-keypair Creates a new 2048 bit RSA key pair with the specified name. ec2-add-keypair key
ec2-bundle-vol Creates a bundled AMI by compressing, encrypting and signing a snapshot of the local machine's root file system. ec2-bundle-vol -k private_key -s user_id -c ec2_cert -r [size tag set] [-d destination] [-p aws_profile] [-s source] [-k kernel_kernel_id] [-r ramdisk_ramdisk_id] [-b block-device-mapping_block_device_mapping] [-s ...-batch]	ec2-deregister Deregisters the specified AMI. Once deregistered, the AMI cannot be used to launch new instances. ec2-deregister ami_id	ec2-describe-keypairs Describes the current state of each specified key. If no keys are specified, all of your keys are returned. ec2-describe-keypairs [key_id ...]
ec2-migrate-ami Copies a bundled AMI from one region to another. ec2-migrate-ami -k private_key -c ec2_cert -s access_key_id -d destination_bucket -s source_bucket -m manifest_manifest_path [-u username] [-p aws_profile] [-s user_id] [-r ramdisk_ramdisk_id] [-b block-device-mapping_block_device_mapping] [-r region] [-m region_name]	ec2-describe-images Returns information about AMIs, AKIs, and ARIs. If no parameters are specified, information about all images is returned. ec2-describe-images [ami_id ...] [-a all] [-o user ...] [-s user_id]	ec2-delete-keypair Deletes the specified key pair by removing the public key from Amazon EC2. ec2-delete-keypair key_pair
ec2-run-instances Launches one or more instances of the specified AMI. ec2-run-instances ami_id [-i instance_profile] [-g group] [-p group ...] [-i keypair] [-d addressing_type] [-t instance_type] [-m instance_market_type] [-a availability_zone] [-k kernel_kernel_id] [-r ramdisk_ramdisk_id] [-b block-device-mapping_block_device_mapping] [-m monitoring]	ec2-modify-image-attribute Modifies an attribute for the specified AMI. ec2-modify-image-attribute ami_id [-i image_id] [-r reset_image_attribute]	ec2-fingerprint-key Computes and displays the fingerprint for a private key produced by Amazon EC2. This operation is performed entirely on the client-side. Network access is not required. ec2-fingerprint-key keyfile
ec2-upload-bundle Uploads a bundle to Amazon S3 storage. ec2-upload-bundle -b s3_bucket -s manifest -a access_key_id -s secret_key -r aws_profile [-c ec2_cert] [-d directory] [-p part] [-i uri] [-r retry] [-s manifest]	ec2-modify-instance-attribute Modifies an attribute for the specified instance. ec2-modify-instance-attribute instance_id [-i instance_id] [-r reset_image_attribute]	ec2-allocate-address Acquires an elastic IP address for use with your account. ec2-allocate-address
ec2-download-bundle Downloads the specified bundle from S3 storage. ec2-download-bundle -b s3_bucket -s manifest -a access_key_id -s secret_key -r aws_profile [-p aws_profile] [-d directory] [-i uri]	ec2-terminate-instances Terminates the specified instances. ec2-terminate-instances instance_id	ec2-associate-address Associates an elastic IP address with an instance. If no addresses are specified, all addresses are returned. ec2-associate-address [public_ip ...]
ec2-unbundle Recreates the AMI from the bundled AMI parts. ec2-unbundle -m manifest [-d destination_directory] [-s source_directory]	ec2-describe-instances Describes the current state of the specified instance(s). If you do not specify instances, all of your instances are included in the output. ec2-describe-instances [instance_id availability_zone ...]	ec2-release-address Releases an elastic IP address associated with your instance. ec2-release-address ip_address
ec2-delete-bundle Deletes the specified bundle from Amazon S3 storage. ec2-delete-bundle -b s3_bucket -s access_key_id -s secret_key [-m manifest_path] [-p aws_profile] [-i uri] [-r retry] [-y] [-c clear]	ec2-region-and-availability-zone-tools Returns a list of regions and availability zones currently available to the account. ec2-describe-availability-zones	ec2-associate-address Associates an elastic IP address with an instance. If the IP address is currently assigned to another instance, the IP address is assigned from the old instance to the new instance. ec2-associate-address -i instance_id ip_address
	ec2-disassociate-address Disassociates the specified elastic IP address from the instance to which it is assigned. ec2-disassociate-address ip_address	

- http://awsdocs.s3.amazonaws.com/EC2/latest/ec2-qrc.pdf
- http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/

AWS Management using ElasticFox (Firefox plugin)



Go online!



Change	Description	Release Date
Windows Bundling Tools Updates	Slight changes were made to the Amazon EC2 Windows bundling tools. For more information on Windows bundling tools, see Bundling a Windows AMI .	23 January 2009
Signature Version 2	Added information about signature version 2, which is a more secure method for signing Query requests. For more information, see Query API Authentication . Also updated the guide to indicate that HTTPS is required for SOAP requests. For more information, see Request Authentication .	15 December 2008
Regions	Amazon EC2 now supports the ability to launch instances in multiple geographically distinct regions. These currently include the United States and the new region for Europe. For more information, see Regions and Availability Zones .	10 December 2008
Windows Operating System	Amazon EC2 now offers the Microsoft Windows Server 2003 R2 operating system on all Amazon EC2 instance types. For more information, see Amazon EC2 Running Windows .	22 October 2008
Elastic IP Improvements	Amazon EC2 is upgraded with a series of networking improvements that result in faster remapping of elastic IP addresses. Additionally, when an address is disassociated, the connection is immediately broken so you can reconnect to other instances. These changes are transparent and do not affect the APIs or command line tools.	22 October 2008
Amazon EBS	Amazon EBS enables a single Amazon EC2 instance to attach to a highly available.	20 August

EC2 SOAP API



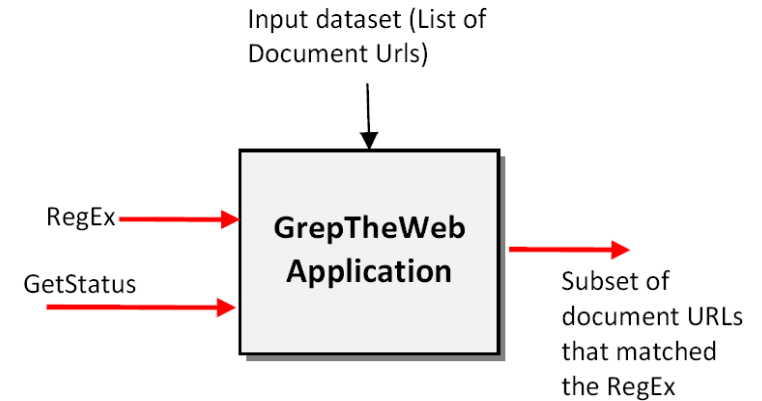
- AllocateAddress
- AssociateAddress
- AttachVolume
- AuthorizeSecurityGroup
- BundleInstance
- CancelBundleTask
- ConfirmProductInstance
- CreateKeyPair
- CreateSecurityGroup
- CreateSnapshot
- CreateVolume
- DeleteKeyPair
- DeleteSecurityGroup
- DeleteSnapshot
- DeleteVolume
- DeregisterImage
- DescribeAddresses
- DescribeAvailabilityZones
- DescribeBundleTasks
- DescribeImageAttributes
- DescribeImages

AWS Example: Grep the Web



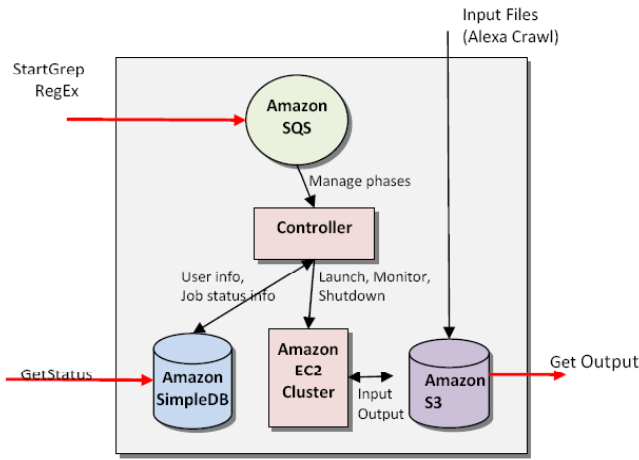
Source:
„Cloud Architectures“ by Jinesh Varia, Amazon, published online at <http://jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf>

GrepThe Web – Zoom Level 1



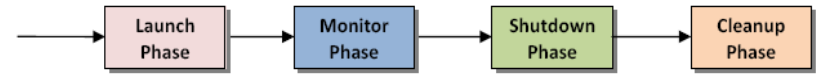
Source: amazon.com

GrepThe Web – Zoom Level 2



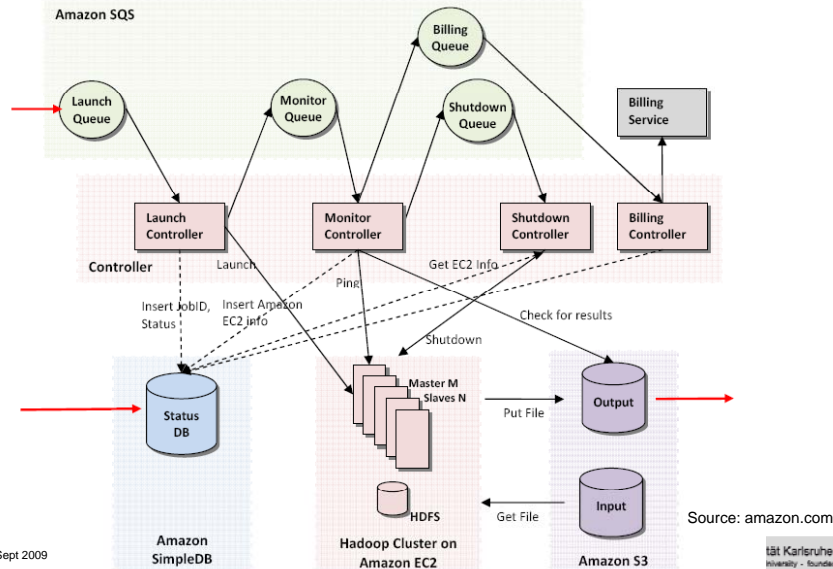
Source: amazon.com

Phases of GrepThe Web Architecture



Source: amazon.com

GrepThe Web – Zoom Level 3

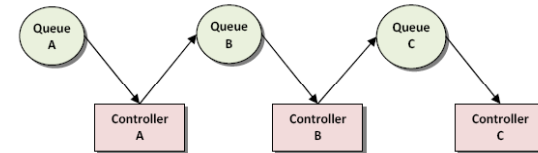


Source: amazon.com

Using Queues for Loose coupling



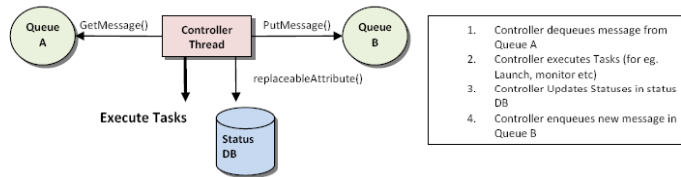
Tight coupling (procedural programming)



Loose coupling (independent phases using queues)

Source: amazon.com

Simple Controller Architecture



1. Controller dequeues message from Queue A
2. Controller executes Tasks (for eg. Launch, monitor etc)
3. Controller Updates Statuses in status DB
4. Controller enqueues new message in Queue B

Public Abstract BaseController (SQSMessageQueue fromQueue, SQSMessageQueue toQueue, SDBDomain domain)

Source: amazon.com

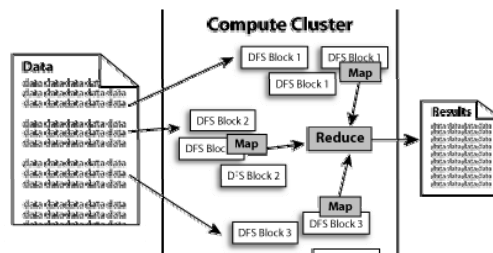
MapReduce



MapReduce Programming Model

Two functions:

- **map** (in_key, in_value) -> (out_key, intermediate_value) list
- **reduce** (out_key, intermediate_value list) -> out_value list



Google's MapReduce

Programming model

Input & Output: each a set of key/value pairs

Programmer specifies two functions:

```
map (in_key, in_value) -> list(out_key, intermediate_value)
```

- Processes input key/value pair
- Produces set of intermediate pairs

Example: Count word occurrences

- Combines all intermediate values for a particular key
- Produces a set of merged output values (usually just one)

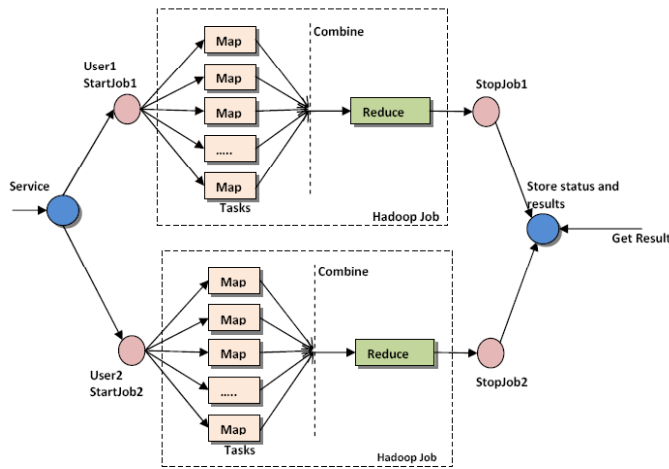
Inspired by similar primitives in LISP and other languages

```
map(String input_key, String input_value):
// input_key: document name
// input_value: document contents
for each word w in input_value:
  EmitIntermediate(w, "1");
```

```
reduce(String output_key, Iterator intermediate_values):
// output_key: a word
// output_values: a list of counts
int result = 0;
for each v in intermediate_values:
  result += ParseInt(v);
Emit(AsString(result));
```

<http://labs.google.com/papers/mapreduce.html>

MapReduce



Source: amazon.com



Hadoop

- OpenSource Apache Software Foundation Project (Yahoo!)

- <http://wiki.apache.org/hadoop/ProjectDescription>

- MapReduce programming model

- Distributed file system (HDFS)

- Parallel database

- <http://en.wikipedia.org/wiki/Hadoop>

- <http://code.google.com/edu/parallel/mapreduce-tutorial.html>

PaaS Example: Google App Engine



Web Applications on Google Infrastructure



Google App Engine: code.google.com/appengine/



code.google.com/appengine/docs/ whatisgoogleappengine.html



- “Google App Engine lets you run your web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve your users.”
- “You can serve your app using a free domain name on the appspot.com domain, or use Google Apps to serve it from your own domain. You can share your application with the world, or limit access to members of your organization.”
- “App Engine costs nothing to get started. Sign up for a free account, and you can develop and publish your application for the world to see, at no charge and with no obligation. A free account can use up to 500MB of persistent storage and enough CPU and bandwidth for about 5 million page views a month.”

App Engine Components



1. Scalable Serving Infrastructure
2. Python and Java Runtime
3. Software Development Kit
4. Datastore
5. Web based Admin Console



App Engine Developer Account



- Dozens of examples in App Gallery
 - Tools, Communication, Games, News, Finance, Sports, Lifestyle, Technology, Enterprise
 - <http://appgallery.appspot.com/>
- Register as a developer
 - <http://code.google.com/appengine>
- Free to get started
 - 500 MB storage
 - 2 GB bandwidth / day
 - ~ 5 million page views / month
- Pay-per-use if you need more



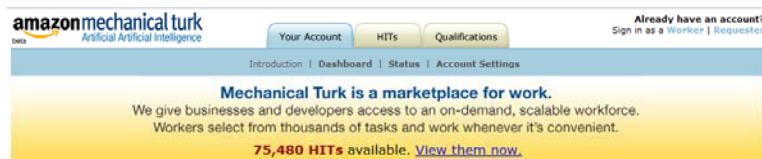
Free Quota and Pricing

Resource	Free Quota	Additional
CPU	Equivalent to 5M pageviews / month for a typical app	10-12¢ / core-hour
Storage		15-18¢ / GB-month
Bandwidth, Outgoing		11-13¢ / GB transferred
Bandwidth, Incoming		9-11¢ / GB transferred

HuaaS Example: On-demand Workforce



Amazon Mturk: www.mturk.com



amazonmechanicalturk
Artificial Intelligence

Your Account HITs Qualifications

Introduction | Dashboard | Status | Account Settings

Already have an account?
Sign in as a Worker | Requester

Mechanical Turk is a marketplace for work.
We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.
75,480 HITs available. [View them now.](#)

Make Money by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. [Find HITs now.](#)

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work



Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. [Get started.](#)

As a Mechanical Turk requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results



MTurk Basic Idea

- Amazon Mechanical Turk (MTurk) is a **crowdsourcing marketplace** that enables computer programs to co-ordinate the use of human intelligence to perform tasks which computers are unable to do.
- Requesters, the human beings that write these programs, are able to pose tasks known as **HITs (Human Intelligence Tasks)**, such as choosing the best among several photographs of a storefront, writing product descriptions, or identifying performers on music CDs.
- Workers (called Providers in Mechanical Turk's Terms of Service) can then browse among existing tasks and complete them for a monetary payment set by the Requester. To place HITs, the requesting programs use an open API, or the somewhat limited Mturk Requester site.

„Human(Intelligence)-as-a-Service“



amazon mechanical turk
beta Artificial Intelligence

Your Account | **HITS** | Qualifications | 75,545 HITS available now

Sign In

Search for **HITS** containing that pay at least \$ **0.00** for which you are qualified **GO**

All HITS
1-10 of 600 Results

Sort by: HITS Available (most first) | Show all details | Hide all details | 1 2 3 4 5 > Next >> Last

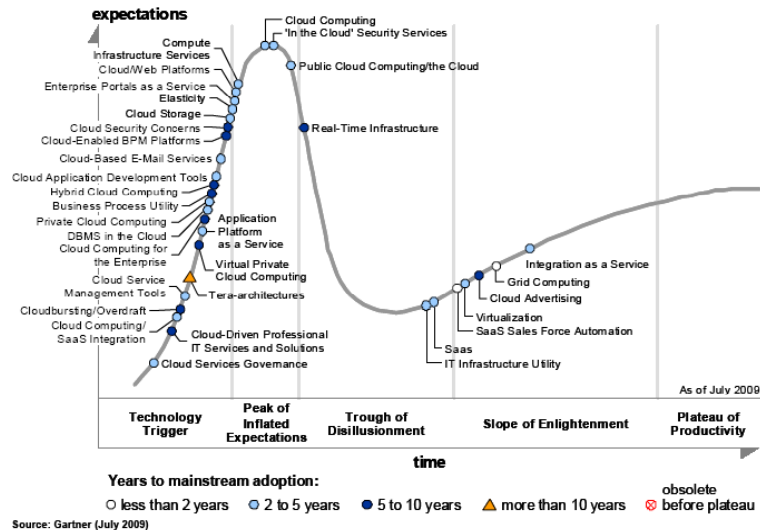
Find a Price for a Specific Wine View a HIT in this group			
Requester: Alan - HM, Inc.	HIT Expiration Date: Feb 5, 2009 (1 day 23 hours)	Reward: \$0.03	
	Time Allotted: 60 minutes	HITS Available: 39883	
IT Part Descriptions View a HIT in this group			
Requester: Corev Donovan	HIT Expiration Date: Mar 5, 2009 (4 weeks 1 day)	Reward: \$0.05	
	Time Allotted: 60 minutes	HITS Available: 7181	
Write a short review on music, books, movies, restaurants, destinations, or anything you choose! View a HIT in this group			
Requester: Vic Dickson	HIT Expiration Date: Feb 14, 2009 (1 week 4 days)	Reward: \$0.30	
	Time Allotted: 60 minutes	HITS Available: 4441	
Find the Product Image on the Manufacturer's Website View a HIT in this group			
Requester: VitaminLife.com	HIT Expiration Date: Feb 9, 2009 (6 days 3 hours)	Reward: \$0.02	
	Time Allotted: 60 minutes	HITS Available: 3592	
Get paid to rate porn (Warning: You WILL see software and hardcore pornography!) View a HIT in this group			
Requester: Faspek.com	HIT Expiration Date: Feb 6, 2009 (2 days 17 hours)	Reward: \$0.01	
	Time Allotted: 60 minutes	HITS Available: 1855	

Fertig www.mturk.com

Open Research Challenges and Opportunities, Select Ongoing Research Activities



Gartner's Hype Cycle for Cloud Computing, July 2009



Cloud Computing Obstacles and Opportunities



Table 1: Quick Preview of Top 10 Obstacles to and Opportunities for Growth of Cloud Computing.

Obstacle	Opportunity
1 Availability of Service	Use Multiple Cloud Providers; Use Elasticity to Prevent DDOS
2 Data Lock-In	Standardize APIs; Compatible SW to enable Surge Computing
3 Data Confidentiality and Auditability	Deploy Encryption, VLANs, Firewalls; Geographical Data Storage
4 Data Transfer Bottlenecks	FedExing Disks; Data Backup/Archival; Higher BW Switches
5 Performance Unpredictability	Improved VM Support; Flash Memory; Gang Schedule VMs
6 Scalable Storage	Invent Scalable Store
7 Bugs in Large Distributed Systems	Invent Debugger that relies on Distributed VMs
8 Scaling Quickly	Invent Auto-Scaler that relies on ML; Snapshots for Conservation
9 Reputation Fate Sharing	Offer reputation-guarding services like those for email
10 Software Licensing	Pay-for-use licenses; Bulk use sales

Above the Clouds: A Berkeley View of Cloud Computing. Armbrust et al., Technical Report No. UCB/EECS-2009-28. Electrical Engineering and Computer Sciences, University of California at Berkeley, USA, 2009.

Cloud Research in Karlsruhe



- Research, education, and innovation



- Inter-disciplinary approach
- Strong industry partnerships, projects, and strategic alliances



- European node in the OpenCIRRUS™ global, open Cloud Computing research testbed



OpenCIRRUS™ Cloud Computing Research Testbed



- An open, internet-scale global testbed for cloud computing research

- Data center management & cloud services
- Systems level research
- Application level research

- Structure: a loose federation

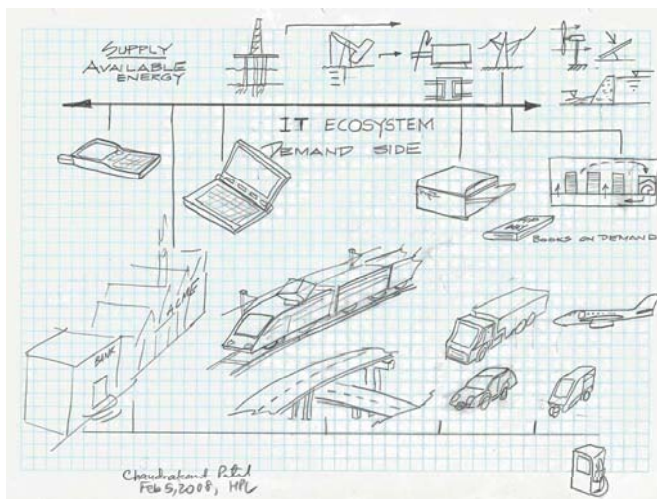
- Sponsors: HP Labs, Intel Research, Yahoo!
- Partners: UIUC, Singapore IDA, KIT, NSF
- Members: System and application development

- Great opportunity for cloud R&D

- <http://opencirrus.org>



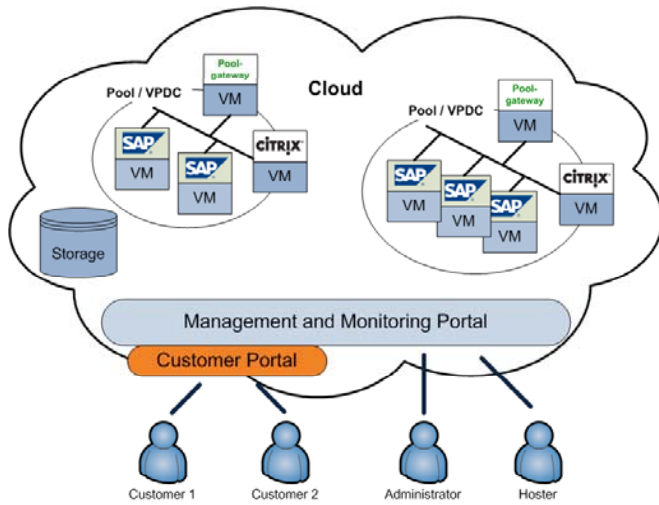
„Everything-as-a-Service (XaaS)“



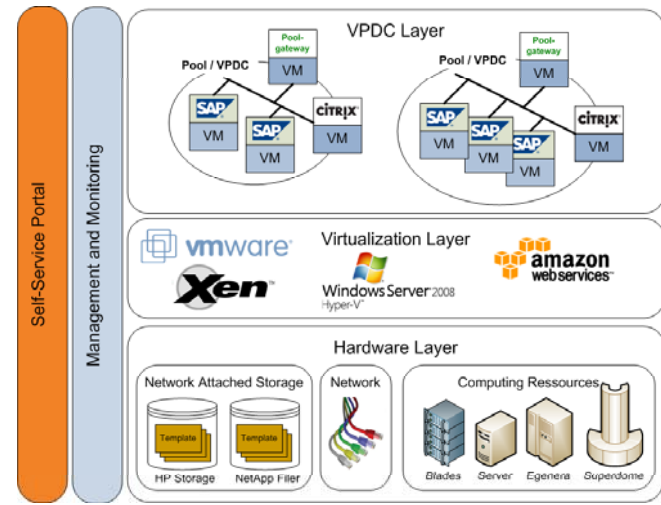
Example 1: Landscape as a service (LaaS), or, Virtual Private Data Centers (VPDC)



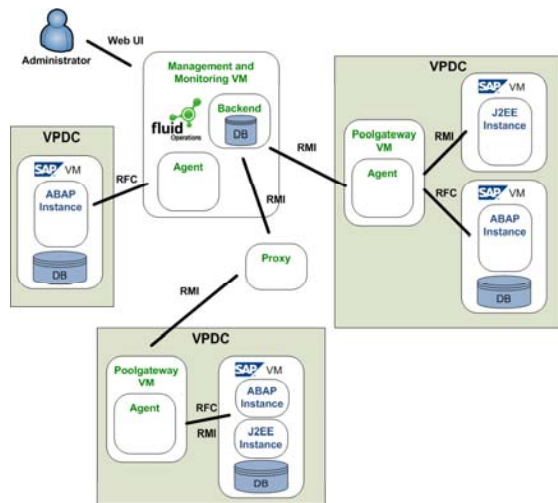
Migrating, resp. Provisioning, an entire (SAP-) Landscape as a Service



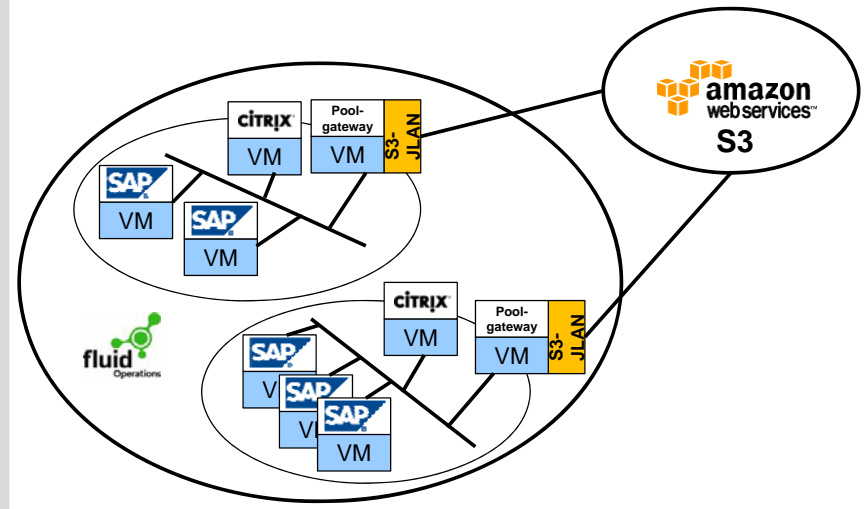
LaaS-Architecture



LaaS Management and Monitoring



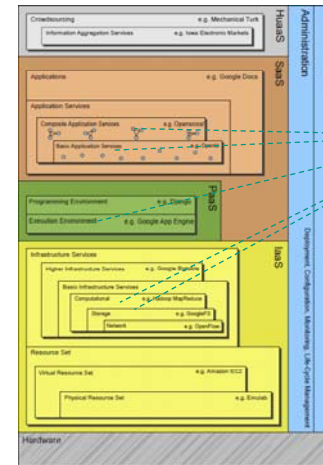
Extending the VPDC to External Resources



Example 2: Composing Cloud Services



Composing XaaS

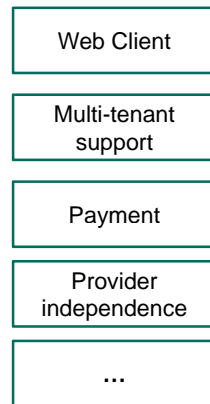


How can we mashup an open, authenticated, pay-per-use Hadoop service?

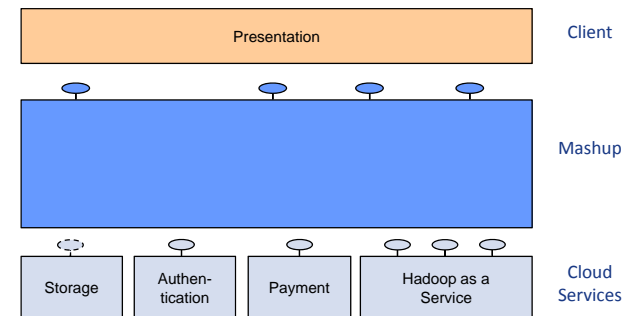
Hadoop reality

1. Reserve servers
2. Install Hadoop on each server
3. Configure one server as namenode, another one as jobtracker (all others are tasktrackers and will execute map-reduce)
4. Log into namenode (SSH)
5. Load data onto namenode
6. Load data onto HDFS
7. Log into jobtracker (SSH)
8. Start program
9. Observe SSH shell output for progress
10. Load date from HDFS onto namenode
11. Get results (SSH)

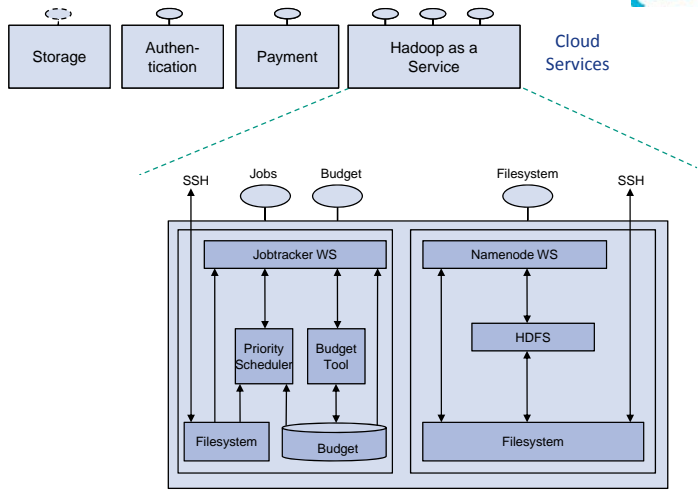
What's missing:



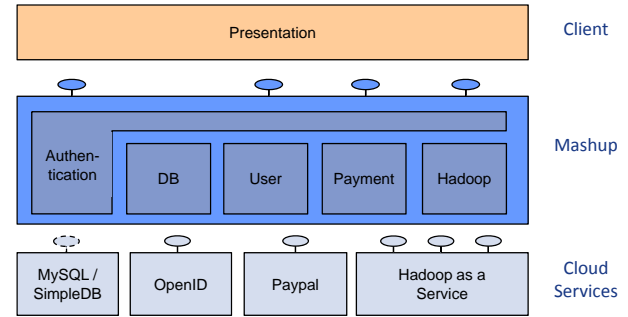
Basic Architecture



Cloud Services



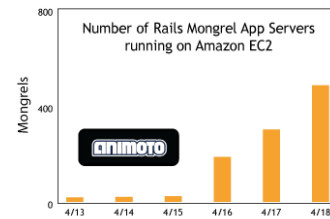
Cloud Service-specific Modules



Example 3: Understanding common Cloud Use Cases & Cost/Value Estimation



Animoto's Facebook Scale-up



...and Scale-down



Cloud Computing TCO (single consumer viewpoint, IaaS focus)



Collect real-world use cases and identify typical scenarios



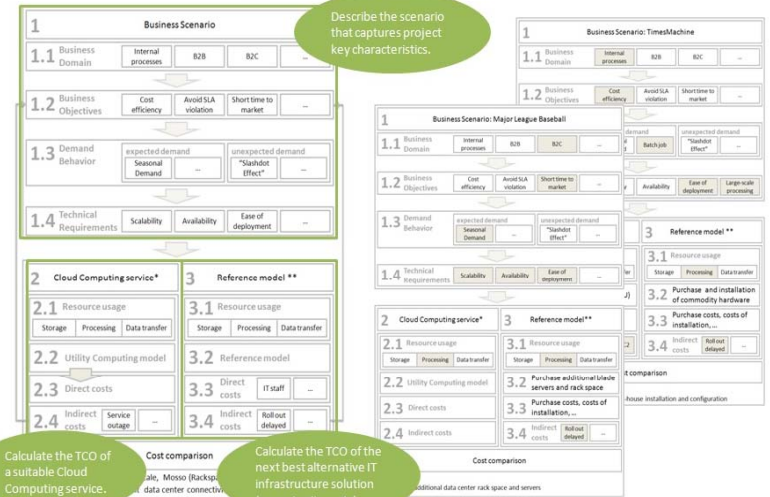
Examine key aspects from business and IT perspective

- business objectives**
- foster innovation
 - rapid prototyping
 - leverage Web as platform
- demand behavior**
- seasonal
 - temporary spikes
 - unpredictable
- IT requirements**
- scalability
 - reliable and stable platform
 - high availability

Understand and value benefits from cloud computing

- Estimate costs**
- variable costs
 - fixed costs
 - time to market
- Estimate value**
- Business value
 - Economic value
- Derive strategies**
- Decision processes
 - Recommendations
 - Business transformation

Estimating the Value of Cloud Computing



Summary



Summary and Discussion



- Cloud Computing has the potential to **fundamentally change** the way we design the technical architecture and the business architecture of modern enterprises
- Cloud Computing is a **disruptive technology**, leading to **creative disruption**

Creative Disruption [Schumpeter]



The opening up of new markets and the organizational development [...] illustrate the process of **industrial mutation** that incessantly revolutionizes the economic structure from within, incessantly destroying the old one, incessantly creating a new one...

[The process] must be seen in its role in the perennial gale of **creative destruction**; it cannot be understood on the hypothesis that there is a perennial lull.

- Business strategy can never assume an end-state or equilibrium
- The integrity and identity of any business is to some degree dependent on the external pressures exerted on it by the competitive environment; strategic success may be the greatest threat to future strategic success

Conclusion



More information: cloudwiki.fzi.de



New Book on Cloud Computing (in German) Springer-Verlag, September 2009



www.tinyurl.com/cloudbuch

Thank You!



FZI

Stefan Tai, stefan.tai@kit.edu
www.eOrganization.de

Acknowledgments



- Marcel Kunze, KIT
- Jens Nimis, FZI
- Alexander Lenk, FZI
- Markus Klems, FZI