References


Reading


Cloud Computing
Examples

Google Apps
iTunes Store
Bittorent
Skype
Web mail
Facebook
Google Maps
"computation may someday be organized as a public utility"

John McCarthy
1960
Wikipedia Definition

IT-related capabilities are provided “as a service”

Services accesses anywhere via network access

IEEE

It is a paradigm in which information is permanently stored in servers on the Internet and cached temporarily on clients that include desktops, entertainment centers, table computers, notebooks, wall computers, handhelds, etc.
Key Characteristics

- Capital expenditure minimized for users
- Device and location independence
- Performance
- Reliability by way of multiple redundant sites
- Scalability
- Security
- Sustainability through improved resource utilization
- Multitenancy
Multitenancy

Single instance of software runs on a software-as-a-service (SaaS) vendor's servers

Google Apps

Serving multiple client organizations (tenants)
## Beware of Hype


<table>
<thead>
<tr>
<th>Date</th>
<th>Service</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008-08-26</td>
<td>FlexiScale</td>
<td>2 day outage</td>
</tr>
<tr>
<td>2008-08-12</td>
<td>Gmail</td>
<td>2 hour outage</td>
</tr>
<tr>
<td>2008-07-20</td>
<td>Amazon S3</td>
<td>8 hour outage</td>
</tr>
<tr>
<td>2008-07-10</td>
<td>MobileMe</td>
<td>outage up to week+</td>
</tr>
</tbody>
</table>
Other Definitions

http://cloudcomputing.sys-con.com/node/612375/print

21 IT professionals give their definition of cloud computing

Cloud Computing Journal
http://cloudcomputing.sys-con.com/
Michael Sheehan

Application
- Web email
- Software as a Service (SaaS)
- SalesForce

Platform
- Enables cloud applications
- Google App Engine, Heroku, Mosso, Engine Yard, Joyent, force.com

Infrastructure
- Enables cloud applications & platforms
- Amazon’s EC2, GoGrid, RightScale, Linode

http://cloudcomputing.sys-con.com/node/609938
Components (wikipedia)

<table>
<thead>
<tr>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clients</td>
</tr>
<tr>
<td>Services</td>
</tr>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Platform</td>
</tr>
<tr>
<td>Storage</td>
</tr>
<tr>
<td>Infrastructure</td>
</tr>
</tbody>
</table>

These terms are poorly defined in the Wikipedia article.
Infrastructure

Delivery of computer infrastructure as a service

Full virtualization (GoGrid, Skytap)
Grid computing (Sun Grid)
Management (RightScale)
Paravirtualization (Amazon Elastic Compute Cloud)
**Storage**

Delivery of data storage as a service
Often billed on a utility computing basis

Database
- Amazon SimpleDB
- Google App Engine's BigTable datastore

Network attached storage
- MobileMe iDisk component,
- Nirvanix CloudNAS

Synchronization
- Live Mesh Live Desktop component
- MobileMe push functions

Web service
- Amazon Simple Storage Service
- Nirvanix SDN
Platform

Facilitates deployment of applications

Web application frameworks
  Python Django (Google App Engine)
  Ruby on Rails (Heroku)

Web hosting
  Mosso

Proprietary
  Force.com
Application

Peer-to-peer/volunteer computing
  Bittorrent
  SETI@home
  Skype

Web application
  Facebook

Software as a service
  Google Apps
  Salesforce

Software plus services
  Microsoft Online Services
Service

Software system[s] designed to support interoperable machine-to-machine interaction over a network

Identity
  OAuth, OpenID
Integration
  Amazon Simple Queue Service
Mapping
  Google Maps, Yahoo! Maps
Payments
  Amazon Flexible Payments Service, Google Checkout, PayPal
Search
  Alexa, Google Custom Search, Yahoo! BOSS
Others
  Amazon Mechanical Turk
Client

Computer hardware and/or computer software which
Relies on The Cloud for application delivery,
Or which is specifically designed for delivery of cloud services

Mobile
Android, iPhone

Thick client/Web browser
Amazon Cloud Computing
Amazon Cloud Parts

Simple Storage Service (S3)
  Data storage
  Accessible from the Web

Elastic Compute Cloud (C2)
  Scalable, pay as you go compute capacity

SimpleDB
  Allows queries on structured data

Simple Queue Service
  Support messages between Web services
Getting Started

http://aws.amazon.com/

Create an account at http://aws.amazon.com/

Documentation
Select "Documentation" under Resources at http://aws.amazon.com/
Simple Storage System - S3

Store data on Amazon's servers

Objects
    Stored in buckets
    Accessed by key

Low level access
    SOAP
    REST

High level access
    Java
    C#
    Perl
    PHP
    Ruby
S3 Costs

Storage
$0.15 per GB-Month of storage used

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, POST, or LIST requests
$0.01 per 10,000 GET and all other requests
Objects

Objects contain
   Object data
   Metadata

Size
   1 byte to 5 gigabytes per object

Object data
   Just bytes
   No meaning associated with bytes

Metadata
   Name-value pairs to describe the object
   Some http headers used
      Content-Type
Buckets

Namespace for objects

No limitation on number of object per bucket

Only 100 buckets per account

Each bucket has a name
   Up to 255 bytes long
   Cannot be same as existing bucket name by any S3 user
Key

Unique identifier for an object within a bucket

Object Url

http://buckerName.s3.amazonaws.com/Key

http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl

Bucket = doc
Key = 2006-03-01/AmazonS3.wsdl
Access Control Lists

Each Bucket has an access control list
  Determines who has read/write access

Each Object can have an access control list
  Determines who has read/write access
Authentication

AWS Access Key ID
Key given to developer

AWS Secret Access Key
Used to generate signature for request

Signature
HMAC-SHA1 hash of request information
Uses Secret Access key to generate hash
Creating a Bucket Using Java

AWSAuthConnection conn =
    new AWSAuthConnection(
        "[aws-access-key-id]", "[aws-secret-access-key-id]";

Response response = conn.createBucket("[bucket-name]", null);
if (response.connection.getResponseCode() == 200) {
    // bucket was created
} else {
    // something bad happened
Writing an Object

AWSAuthConnection conn =
    new AWSAuthConnection("[aws-access-key-id]", "[aws-secret-access-key-id]");

S3Object simpleObject = new S3Object("this is a test".getBytes(), null);
Response response = conn.put("[bucket-name]", "[key-name]", simpleObject, null);
Reading an Object

AWSAuthConnection conn =
    new AWSAuthConnection("[aws-access-key-id]", "[aws-secret-access-key-id]");

GetResponse response = conn.get("[bucket-name]", "[key-name]", null);
String value = response.object.data;
Map metadata = response.object.metadata;
List values = (List)metadata.get("title");
String title = (String)values.get(0);