Amazon S3 - Daten in den Wolken

Barbara von Kalm

Fakultät für Informatik
Hochschule Mannheim
b.vonkalm@stud.hs-mannheim.de

27.11.2009

Agenda

- Motivation
- Grundlagen
  - Cloud-Computing
  - Amazon Web-Services
  - Speicher in einer Cloud
  - EBS
- Amazon S3 in der Praxis
- Schnittstellen
  - Zugriffsmöglichkeiten
  - Quick Reference
  - Bittorent
  - Beispielimplementierung
  - Hilfreiche Tools
- Kosten
- Fazit
- Literatur

Motivation

- Festplatte zu klein - Kein Problem: Speicher gibt es »unendlich« in den Wolken

Cloud-Computing

- Technologien »On demand« [REE09]
  - ... solange man sie braucht
  - ... zu dem Zeitpunkt, zu dem man sie braucht
  - ... soviel man sie braucht
- Technologien hinter einer Cloud:
  - Virtualisierung
  - Web-Services
- Zuordnung der verschiedenen Services zu:
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
  - Software as a Service (SaaS)

Amazon Web-Services

- Überbegriff, für die webbasierten Dienste, die Amazon seid 2006 anbietet [AWS09]
- Abdeckung von einer Vielzahl möglicher Technologien in einer Cloud
- Eine Auswahl:
  - Amazon Elastic Cloud Compute (EC2)
  - Amazon Simple Storage Service (S3)
  - Amazon Simple Queue Service (Amazon SQS)
  - Amazon Cloud Front
  - Amazon SimpleDB

Speicher in einer Cloud - 1

Anforderungen an Speicher in einer Cloud:

- Speicherung einer ≫unbegrenzten≪ Datenmenge
- Zugriff von überall
- Zugriffskontrolle
- Datensicherheit
- ständige Verfügbarkeit der Daten

Speicher in einer Cloud - 2

Arten von Speicher in einer Cloud:

- Persistenter Speicher
  - Elastic Block Store
- Flüchtiger Speicher

Speicher in einer Cloud - 3

verschiedene Angebote:

- Amazon S3
- Mossos CloudFS - Beta
- Memopal
- EMC^2

Elastic Block Storage

- Persistenter Speicher innerhalb der Amazon Cloud
- Vergleichbar mit einem Storage Area Network, aber kostengünstiger [REE09]

[1]

- Erzeugung von sogenannten Volumes in der Größe 1GB - 1TB
- Notwendig zur Instanziierung von EC2
- Der aktuelle Zustand einer EC2-Instanz kann als sogenannter EBS-Snapshot gespeichert werden.

[1] http://www.elektronik-kompendium.de/sites/net/0906071.htm

Was ist Amazon S3

- Amazon Simple Storage Service
- Persistenter Speicher in einer Cloud
- Speicherung von beliebigen digitalen Daten
- Speicherung (theoretisch) unendlich großer Mengen an Daten
- Daten sind von überall abrufbar
- Daten können gegen unerwünschte Zugriffe geschützt werden
- Standardisierte Schnittstellen zum Zugriff auf die Daten

Wozu kann man Amazon S3 einsetzen?

- Backups
- Kurzfristiges Speichern großer Datenmengen bei Peaks
- Datenaustausch zwischen Anwendungen
- Hilfreicher Datenspeicher in der Cloud mit Hilfe von Tools wie s3Fox

Entitäten im S3

- Objects
- Buckets
- Keys [API06]

Objects

- Die Dinge, die gespeichert werden sollen
- Über eine URL eindeutig abrufbar
- Speicherung von Meta-Daten in Key-Value-Pairs

Buckets

- Container, in denen die Objekte gespeichert werden
- Der Name muss amazonweit eindeutig sein
- Buckets können nicht verschachtelt werden
- Achtung: nicht vergleichbar mit Ordnern im Dateisystem
- Jeder Benutzer kann 100 Buckets verwalten
- Anzahl der Objekte in einem Bucket ist aber beliebig

Keys

- Jedes Objekt ist über einen Schlüssel eindeutig zuordenbar
- Kombination aus Bucketname und Objektname
- Beispiel: http://johnsmith.s3.amazonaws.com/photos/puppy.jpg
- Operationen zu einem Objekt mit Hilfe des Keys sind atomar zur Gewährleistung der Datenkonsistenz
- Hinweis: Man kann mit Hilfe der CNAME-Funktion dafür sorgen, dass man anhand der URL keinen Rückschluss auf Amazon ziehen kann [RUB08].

Operationen

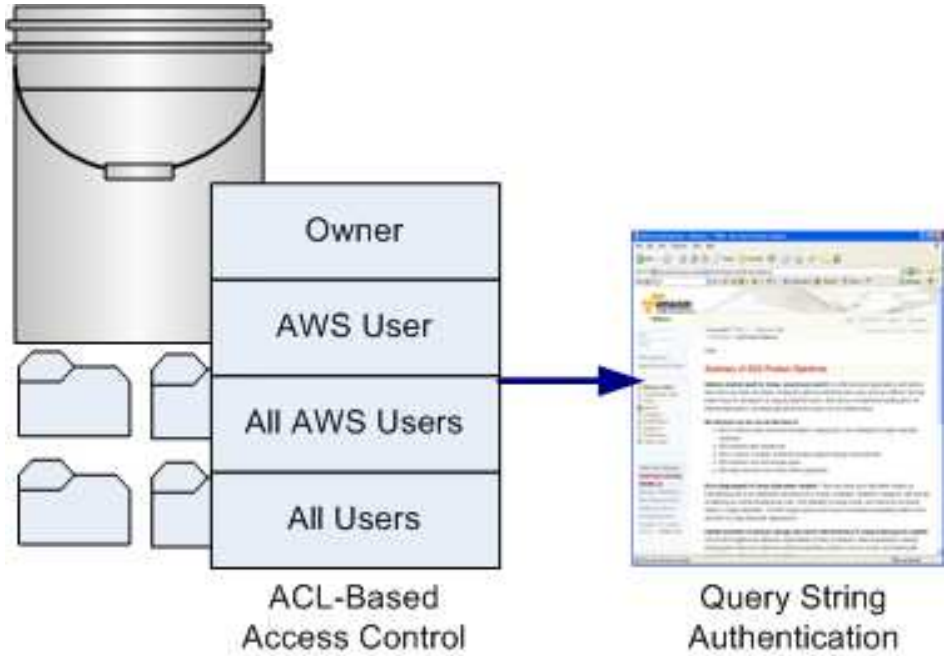
- Bucket erstellen und Daten darin speichern
- Daten-Objekte hoch laden und in einem Bucket ablegen
- Daten-Objekte aus einem Bucket downloaden
- Auflistung aller Objekte in einem Bucket

Sicherheit

- Wie sicher sind die abgelegten Daten:
  - Vertrauen in den Anbieter
  - Physikalische Sicherheit
  - Backups
- Amazon verspricht ein vertrauenswürdiger Partner zu sein
  - Regelmäßige SOX-Zertifizierungen
  - Industriespezifische Zertifizierungen
- Daten werden redundant an verschiedenen physikalischen Orten abgelegt
- Backups sind standardmäßig Bestandteil der Serviceverträge

Zugriffskontrolle

- Individuelle Gestaltung der Zugriffe auf einzelne Objekte



- Default: Nur private Nutzung
- Zugriff über SSL möglich
- Authentifizierung mit einer *Access Key ID* und *Secret Access Key*
- Hinweis: Geheime Daten sollten dennoch vor dem Ablegen verschlüsselt werden

Zugriffsmöglichkeiten

- Amazon S3 Application Programming Interface (API)
- REST Interface
- SOAP Interface
- Bittorent

- Es gibt sehr viele APIs für alle möglichen Programmiersprachen, die bereits fertige Schnittstellen anbieten.

Quick Reference

Service Operations

GET service
Returns a list of all buckets owned by the authenticated request sender.

```
GET / HTTP/1.1
Host: s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
```

Bucket Operations

PUT bucket
Creates a new bucket belonging to the account of the authenticated request sender. Optionally, you can specify a Europe location constraint.

```
PUT / HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
Content-Length: {0 | length}
[<CreateBucketConfiguration>
  <LocationConstraint>EU</LocationConstraint>
</CreateBucketConfiguration>]
```

GET bucket
Lists information about the objects in a bucket for a user that has read access to the bucket.

```
GET ?prefix=prefix&marker=marker&max-keys=max-
keys&delimiter=delimiter HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
```

GET bucket location
Lists the location constraint of the bucket for the bucket owner.

```
GET /?location HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
```

DELETE bucket
Deletes the specified bucket. All objects in the bucket must be deleted before the bucket itself can be deleted.

```
DELETE / HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
```

Object Operations

GET object
Gets an object for a user that has read access to the object.

```
GET /destinationObject HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
[Range:bytes=byte_range]
[x-amz-metadata-directive: metadata_directive]
[x-amz-if-match: etag]
[x-amz-if-none-match: etag]
[x-amz-if-unmodified-since: time_stamp]
[x-amz-if-modified-since: time_stamp]
```

PUT object
Adds an object to a bucket for a user that has write access to the bucket. A success response indicates the object was successfully stored; if the object already exists, it will be overwritten.

```
PUT /destinationObject HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
Content-Length: length
Content-MD5: md5_digest
Content-Type: type
Content-Disposition: object_information
Content-Encoding: encoding
Cache-Control: caching
Expires: expiration
<request metadata>
```

COPY object
Copies an object for a user that has write access to the bucket and read access to the object. All headers prefixed with x-amz- must be signed, including x-amz-copy-source.

```
PUT /destinationObject HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
x-amz-copy-source: /source_bucket/sourceObject
[x-amz-metadata-directive: metadata_directive]
[x-amz-copy-source-if-match: etag]
```

HEAD object
Retrieves information about an object for a user with read access without fetching the object.

```
HEAD /destinationObject HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
```

DELETE object
Deletes the specified object. Once deleted, there is no method to restore or undelete an object.

```
DELETE / HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
Date: date
Authorization: AWS AWSAccessKeyId:signature
```

Miscellaneous

Bucket Name Restrictions
Amazon S3 bucket names must:

- Only contain lowercase letters, numbers, periods (.), and dashes (-)
- Start with a number or letter
- Be between 3 and 63 characters long
- Not be in an IP address style (e.g., "192.168.5.4")
- Not end with a dash
- Not contain dashes next to periods (e.g., "my-.bucket.com" and "my.-bucket" are invalid)

Note: Although legacy bucket names can be up to 255 characters, include underscores, and end with a dash, they are not recommended.

REST Request Signature
Construct a signature by making an RFC2104 HMAC-SHA1 of the following and converting it to Base64.

Item	Example
HTTP-Verb + "\n" +	GET\n
[Content-MD5] + "\n" +	4gJE4saaMU4BqNR0kLY+lw==\n or \n
[Content-Type] + "\n" +	image/jpeg\n or \n
Date + "\n" +	Tue, 6 Mar 2007 19:42:41 +0000\n
[CanonicalizedAmzHeaders +/n]	x-amz-acl:public-read\n

Bittorrent

- Offenes Peer-to-Peer Protokoll zum Anbieten von Dateien
- Nutzung von Bittorrent, um öffentliche Dateien aus dem S3 anzubieten
- Vorteil: Es können Kosten gespart werden
- Jeder Nutzer lädt einen Teil der Datei runter und kann diesen gleichzeitig an andere verteilen
- So fungieren einzelne PCs auch als Server und die ganze Netzwerklast liegt nicht mehr bei Amazon
- Weniger Datentransfer auf den Server von Amazon, desto geringer die Kosten

Bittorrent - Was braucht man dafür

- Bittorent-Client
- *.torrent - Datei
- ?torrent an die REST-GET-Anfrage anhängen
- Kann nicht mit SOAP genutzt werden
- Achtung: Zugriffszeit auf ein Objekt kann je nach Größe des Objekts beim ersten Zugriff deutlich langsamer werden

Beispielimplementierung

- LIVEDEMO [JETS3]
- Verwendung der jets3t - API

Hilfreiche Tools

- Firefox Plugin: Amazon S3 Organizer [S3F09]
  - Buckets erstellen/löschen/umbenennen
  - Objekte hoch- und herunterladen
  - Zugriffsrechte vergeben
- Amazon S3 Authentication Tool for Curl
  - Command-line-Tool zur Interaktion mit HTTP-Services
  - Berechnung von Signaturen
- CloudBerry Explorer
  - Dateien in der Cloud managen
  - Bucketname-Validierung (gültige Symbole)
  - Generierung von URLs

Kosten

- Speicherkosten

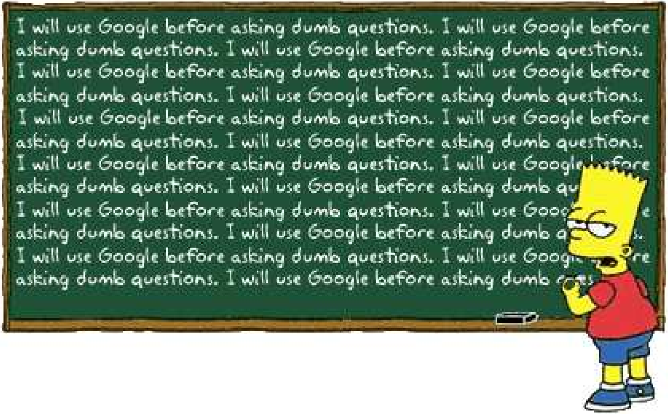
GB pro Monat	USA	Europa
ersten 50 TB	$0.150	$0.180
nächste 50 TB	$0.140	$0.170
nächste 400 TB	$0.130	$0.160
über 500 TB	$0.120	$0.150

- Datentransfer

GB pro Monat	USA / Europa
ersten 10 TB	$0.170
nächste 40 TB	$0.130
nächste 100 TB	$0.110
über 150 TB	$0.100

Fazit

- Tolle Möglichkeit schnell, viele Daten abzulegen
- Für Firmen hilfreich bei Projekten, wo plötzlich große Datenmengen verwaltet werden müssen
- Kann sehr teuer werden, wenn viele Daten heruntergeladen werden
- Einfach in Anwendungen zu integrieren mit Hilfe der gut beschriebenen APIs
- Man muss dem Anbieter vertrauen


I will use Google before asking dumb questions. I will use Google before
asking dumb questions. I will use Google before asking dumb questions.
I will use Google before asking dumb questions. I will use Google before
asking dumb questions. I will use Google before asking dumb questions.
I will use Google before asking dumb questions. I will use Google before
asking dumb questions. I will use Google before asking dumb questions.
I will use Google before asking dumb questions. I will use
asking dumb questions. I will use Google before asking dumb
I will use Google before asking dumb questions. I will use
asking dumb questions. I will use Google before asking dumb
I will use Google before asking dumb questions. I will use
asking dumb questions. I will use Google before asking dumb

Quellen

- [RUB08] Rubner, Stefan: *Eimerweise Online-Speicher*. c't 2008, Heft 23, Seite 186-187
- [REE09] Reese, George: *Cloud Application Architectures*. 1. Auflage. O'Reilly Media. April 2009
- [AWS09] Dokumentation von Amazon unter `http://aws.amazon.com/`
- [API06] Technical Documentation S3. *API Version 2006-03-01*.
- [ARM09] Armbrust, M.: *Above the Clouds: A Berkeley View of Cloud Computing*.
  `http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html`
- [S3F09] Amazon S3 Firefox Organizer, Version 0.4.8 - September 11, 2009
  `https://addons.mozilla.org/en-US/firefox/addon/3247`