

Grundlagen über die File Hosting Services Amazon S3 und Google Cloud Storage

Pascal Krause

Hochschule Mannheim
Fakultät für Informatik
Paul-Wittsack-Straße 10
68163 Mannheim
922045@stud.hs-mannheim.de

Zusammenfassung Dieser Artikel beschäftigt sich mit den File Hosting Services „Simple Storage Service“ von Amazon und dem dazu in Konkurrenz stehendem „Cloud Storage“ von Google. Es werden sowohl die Grundlagen erklärt um mit diesen Services arbeiten zu können, als auch auf die Unterschiede der beiden Services eingegangen. Um einen schnellen Einstieg in diesen Infrastruktur-dienst zu ermöglichen, werden die Grundfunktionen anhand der REST-API erklärt.

In den letzten Jahren wurde es immer wichtiger, schnellen Zugriff auf seine Daten von überall aus dem Internet zu haben. Einer der Gründe für diesen Wandel ist, dass man heutzutage viele verschiedenen Endgeräte benutzt und es aus Synchronisationsgründen ungünstig ist die Daten lokal zu speichern. Ein weiterer Grund ist das Cloud Computing. Wenn sich ein Unternehmen Rechenleistung in der Cloud mietet, müssen die zu bearbeitenden Daten erst in die Cloud hinein kopiert werden um dort bearbeitet werden zu können. Würden die Daten bereits in der Cloud gehalten werden, könnte man sich diesen Zeitintensiven Schritt sparen.

1 Allgemeines zu diesem Dokument

Dieses Dokument ist ein Beitrag zum Cloud Computing Seminar im SS2012 an der Hochschule Mannheim. Es soll den Einstieg in S3 ähnliche File Hosting Services vereinfachen und deren Grundlagen vermitteln. Es richtet sich an Studenten im Hauptstudium, welche sich bereits mit den Grundlagen von Cloud Computing und Webservices auseinandergesetzt haben. Dieser Artikel gibt zuerst generelle Auskunft über S3 und S3 ähnlichen Dienste und geht danach auf deren Benutzung unter Zuhilfenahme der REST API ein. Im folgendem Artikel ist mit S3 die Gesamtheit der S3 kompatiblen File Hosting Services gemeint, sofern nicht extra etwas anderes erwähnt wird.

2 Grundlagen

In diesem Abschnitt wird auf die Grundlagen von Amazons „Simple Storage Service (S3)“ und Googles „Cloud Storage (CS)“ eingegangen. Es wird erklärt was ein File Hosting Services ist und Beispiele zur Integration dieser in bereits bestehende Webdienste gezeigt.

2.1 Was ist S3?

S3 steht für „Simple Storage Service“. Wie dem Namen schon zu entnehmen ist, handelt es sich bei S3 um einen sehr einfachen Speicherdienst für Daten, die im Internet abgelegt werden sollen. Die Daten werden in sogenannten Buckets abgelegt. Die Dateien müssen einen Eindeutigen Pfad innerhalb eines Buckets haben. Bei der Namensfindung für einen Bucket muss darauf geachtet werden, dass dieser keine Unterstriche und nur kleine Buchstaben enthält. Außerdem darf der Name eines Buckets nicht mit einem Strich enden und muss zwischen 3 und 63 Zeichen lang sein.[1][2] Nachdem ein Name für den Bucket gefunden wurde, muss noch die Access Control List (ACL) angegeben werden. Durch die ACL wird die Zugriffsberechtigung auf einen Bucket definiert. Die Wichtigsten Werte für die verschiedenen Berechtigungen sind *private*, *public-read* und *public-read-write*. [3][4] Falls für einen Bucket keine Berechtigung übergeben wird, wird standardmäßig *private* benutzt. Alle Dateien, welche in den erstellten Bucket hoch geladen werden, bekommen automatisch dessen Zugriffsberechtigung. Es kann jedoch für jede einzelne Datei eine spezielle Zugriffsberechtigung eingerichtet werden.[5] Danach kann mit folgendem Schema auf die Dateien zugegriffen werden.

<http://meinBucket.myS3Server.tld/PfadZurDatei> oder
<https://meinBucket.myS3Server.tld/PfadZurDatei>

2.2 Beispiele für die Nutzung von S3

Einige bekannte Unternehmen benutzen derzeit bereits S3. Das wohl bekannteste Beispiel für die Integration von S3 in einen Webservice ist der Dienst Dropbox¹. Doch auch kleinere Dienste wie z.B. SmugMug², Ubuntu One³ oder Foursquare⁴ benutzen S3 um ihre Daten im Internet ablegen zu können.

¹<http://www.dropbox.com/> (07.05.2012)

²<http://www.smugmug.com/> (07.05.2012)

³<https://one.ubuntu.com/> (07.05.2012)

⁴<https://de.foursquare.com/> (07.05.2012)

2.3 Angebote für S3

Um S3 benutzen zu können braucht man natürlich einen Server der diesen File Hosting Services unterstützt. Es ist naheliegend die Server der beiden Marktführer Amazon und Google zu verwenden, jedoch wird der Dienst von diesen nicht unentgeltlich angeboten. Die folgende Tabelle (Tabelle 1) soll helfen eine ungefähre Preisvorstellung, für das Speichern von Daten auf einem S3 Server innerhalb Europas, zu vermitteln.

Tabelle 1. S3 Preise Mai 2012

Leistung	Amazon ⁵	Google ⁶
1GB Storage	0,125\$	0,125\$
1GB Traffic	0,120\$	0,120\$
1.000 PUT Requests	0,010\$	0,010\$
10.000 GET Requests	0,010\$	0,010\$

Verfügt man jedoch über eigene Server-Ressourcen, oder möchte einfach nicht an andere Dienstleister gebunden sein, kann man auf open source Implementierungen eines S3-Servers zurückgreifen. Folgende open source Cloud-Infrastrukturen bieten S3-Unterstützung an.

- Nimbus⁷ mit Cumulus
- Eucalyptus⁸ mit Walrus-Addon
- Cloudstack⁹

2.4 Unterschiede zwischen Amazon S3 und Google Cloud Storage

Die Grundfunktionen von S3, Buckets zu erstellen sowie zu löschen und Dateien hochzuladen, herunterzuladen oder zu löschen werden natürlich von Amazons S3, aber auch von Googles CS unterstützt. Neben den Grundfunktionen gibt es jedoch auch speziellere Funktionen wie zum Beispiel das Fortsetzen des Hochladens, oder das Hochladen einer Datei in mehreren Teilen. Das Fortsetzen beim Hochladen wird nur von Google unterstützt, dafür ist es bei Amazon jedoch möglich eine große Datei in mehrere kleine Teile aufzuspalten. Des weiteren kann man bei Amazon auch Webseiten in einem Bucket ablegen oder festlegen, dass man über Veränderungen innerhalb eines Buckets Benachrichtigt werden möchte. Die Benachrichtigung erfolgt dann über den „Simple Notification Service“ (SNS) von Amazon.[6]

⁸<http://aws.amazon.com/de/s3/pricing/> (07.05.2012)

⁹<https://developers.google.com/storage/docs/pricingandterms> (07.05.2012)

⁷<http://www.nimbusproject.org/> (07.05.2012)

⁸http://open.eucalyptus.com/wiki/EucalyptusStorage_v1.4/ (07.05.2012)

⁹<http://www.cloudstack.org/> (07.05.2012)

3 Zugriff auf S3 über die REST API

In diesem Abschnitt werden die Grundfunktionen von S3 mit Hilfe der REST API[7][8][9] beschrieben. Es handelt sich nicht um eine vollständige S3-API Dokumentation sondern nur um kleine Teile, welcher dafür jedoch bei beiden Diensten Amazon S3 und Google CS gleichermaßen funktionieren. Die S3-Funktionen werden in drei verschiedenen Kategorien mit den Namen, Servicefunktionen, Bucketfunktionen und Dateifunktionen aufgeteilt.

Bei allen Anfragen über die REST API kommen folgende Parameter sehr häufig im *Request Header* vor. Diese Header-Informationen bilden dementsprechend die Basis für jede der hier aufgeführten REST Anfragen. Welche Parameter für welche Funktion genau benötigt werden, steht bei jeder Anfrage noch einmal explizit, in einer beispielhaften Anfrage, dabei. Bei Google CS muss außerdem immer die Version der API mitgegeben werden.

Tabelle 2. Basis Header

Request Header	Beschreibung	immer Erforderlich
Host	URI des S3-Servers	Ja
Date	Datum und Uhrzeit	Ja
Content-Length	Rumpfgöße in Byte	ja
Authorization	Autorisierungsschlüssel	ja
x-goog-api-version	Die Version der API	Ja
x-goog-project-id	Die Projekt-Id	nein
x-goog-acl	Google Zugriffsberechtigung	nein
x-amz-acl	Amazon Zugriffsberechtigung	nein

Der Autorisierungsschlüssel setzt sich bei Amazon aus einer AccessKeyId und einer Signatur zusammen, wobei hingegen Google zum Autorisieren OAuth2 verwendet. Zur Veranschaulichung folgen zwei Beispielschlüssel von Amazon und OAuth2.

Amazon:

AWS AKIAIOSFODNN7EXAMPLE:frJIUN8DYpKDtOLCwo//y1lqDzg=

OAuth2:

Bearer ya29.AHES6ZQhdquTfme0h17X9qU0QSdgLC2c7SN1J-94a1qj-hzIk0vkAA

3.1 Servicefunktionen

Servicefunktionen sind Funktionen die auf den Service an sich angewendet werden. Bisher gibt es allerdings nur eine Servicefunktionen. Diese Funktion dient dazu alle mit dem Autorisierungsschlüssel verbundenen Buckets abzufragen.

GET Service

Beispiel Anfrage zum Anzeigen aller Buckets (GET Service)

```
GET / HTTP/1.1
Host: commondatastorage.googleapis.com
Date: Fri, 01 Oct 2010 22:25:53 GMT
Authorization: Bearer ya29.AHES6ZQhd...
Content-Length: 0
—— Anbieter spezifische Parameter ——
x-goog-api-version: 2
x-goog-project-id: 24885729025
```

Beispiel Antwort mit allen Buckets (GET Service)

```
HTTP/1.1 200 OK
Content-Length: 648
Content-Type: application/xml
Date: Fri, 01 Oct 2010 22:25:53 GMT
<?xml version='1.0' encoding='utf-8'?>
<ListAllMyBucketsResult xmlns='http://...' >
  <Owner>
    <ID>84fac329bceSAMPLE777d5d22b8SAMPLE77d85ac2SA</ID>
    <DisplayName>foobar</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>my-first-bucket</Name>
      <CreationDate>
        2009-12-16T18:36:25.369Z
      </CreationDate>
    </Bucket>
    <Bucket>
      <Name>foobar-bucket</Name>
      <CreationDate>
        2009-12-16T18:42:37.408Z
      </CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

Der Container <Owner> enthält die *ID* und den *DisplayName* des Besitzers. Bei Google CS ist der *DisplayName* jedoch noch nicht implementiert, dort wird immer ein leerer String zurück gegeben. Der Container <Buckets> enthält alle mit dem Autorisierungsschlüssel in Verbindung gebrachten Buckets.

3.2 Bucketfunktionen

Die drei wichtigsten Bucketfunktionen sind die Funktionen zum Erstellen eines neuen Buckets, das Löschen eines bereits existierenden Buckets und das Anzeigen aller Dateien in einem Bucket. Es gibt noch weitere Bucketfunktionen¹⁰, jedoch wird auf diese nicht weiter eingegangen, da sie nur von Amazon S3 und nicht Google CS unterstützt werden.

PUT Bucket

Beispiel Anfrage zum Erstellen eines Buckets (PUT Bucket)

```
PUT / HTTP/1.1
Host: bucketName.commodatastorage.googleapis.com
Date: Fri, 01 Oct 2010 22:25:53 GMT
Authorization: Bearer ya29.AHES6ZQhd...
Content-Length: 0
—— Anbieter spezifische Parameter ——
x-goog-api-version: 2
x-goog-project-id: 24885729025
x-goog-acl: private
x-amz-acl: private
```

Bei der Anfrage für diese Bucketfunktion sind zwei entscheidende Parameter zu beachten. Zum einen muss bei dem Parameter *Host* der Name des Buckets mit angegeben werden, zum anderen muss die Berechtigung mit dem Parameter *x-**-acl* festgelegt werden. Wird keine Berechtigung übergeben, wird standardmäßig *private* benutzt.

Mögliche Antworten (PUT Bucket)

```
200 – Erstellung war erfolgreich.
400 – Bucketname beinhaltet verbotene Zeichen.
409 – Der angegebene Bucketname existiert bereits.
```

DELETE Bucket

Beispiel Anfrage zum Löschen eines Buckets (DELETE Bucket)

```
DELETE / HTTP/1.1
Host: bucketName.commodatastorage.googleapis.com
Date: Fri, 01 Oct 2010 22:25:53 GMT
Authorization: Bearer ya29.AHES6ZQhd...
Content-Length: 0
—— Anbieter spezifische Parameter ——
x-goog-api-version: 2
```

Mögliche Antworten (DELETE Bucket)

¹⁰<http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTBucketOps.html>
(07.05.2012)

204 – Entfernen war erfolgreich.
404 – Bucket existiert nicht.
409 – Bucket ist nicht leer

GET Bucket

Beispiel Anfrage zum holen des Bucketinhalts (GET Bucket)

```
GET / HTTP/1.1
Host: bucketName.comondatastorage.googleapis.com
Date: Fri, 01 Oct 2010 22:25:53 GMT
Authorization: Bearer ya29.AHES6ZQhd...
Content-Length: 0
—— Anbieter spezifische Parameter ——
x-goog-api-version: 2
```

Beispiel Antworten mit dem Bucketinhalt (GET Bucket)

```
HTTP/1.1 200 OK
Content-Length: 4061
Content-Type: application/xml
Date: Wed, 17 Feb 2010 23:31:57 GMT
Expires: Wed, 17 Feb 2010 23:31:57 GMT

<?xml version='1.0' encoding='utf-8'?>
<ListBucketResult xmlns='http://.....' >
  <Name>travel-maps</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>europe/france/cannes.pdf</Key>
    <LastModified>2010-02-17T22:11:12.487Z</LastModified>
    <ETag>"53fc311c15eda0a031809982ccf92aac"</ETag>
    <Size>5061631</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>84fac329bceSAMPLE777d5d22..... </ID>
      <DisplayName></DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>europe/france/paris.pdf</Key>
    <LastModified>2010-02-17T22:09:57.457Z</LastModified>
    <ETag>"53fc311c15eda0a031809982ccf92aac"</ETag>
    <Size>5061631</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
```

```
<ID>84fac329bceSAMPLE777d5d22 ..... </ID>
  <DisplayName></DisplayName>
  </Owner>
</Contents>
</ListBucketResult>
```

Wichtig sind die `<Contents>` Container, denn sie enthalten pro Container jeweils eine Datei. Der Name dieser Datei ist mit dem Tag `<Key>` versehen. In dem Tag `<Size>` ist die Größe der Datei in Bytes hinterlegt und anhand des Tags `<LastModified>` kann man das Datum der letzten Änderung einer Datei feststellen. Als Veränderung zählt jedoch nur ein erneutes Hochladen der Datei.

3.3 Dateifunktionen

Die Dateifunktionen sind den Bucketfunktionen sehr ähnlich. Auch hier sind die drei wichtigsten Funktionen wieder das Erstellen einer Datei, das Löschen einer Datei und das Herunterladen einer Datei.

PUT Object

Beispiel Anfrage zum Erstellen einer Datei (PUT Object)

```
PUT /picture.svg HTTP/1.1
Host: bucketName.commondatastorage.googleapis.com
Date: Fri, 01 Oct 2010 22:25:53 GMT
Authorization: Bearer ya29.AHES6ZQhd...
Content-Length: 13234
—— Anbieter spezifische Parameter ——
x-goog-api-version: 2
x-goog-acl: private
x-amz-acl: private
```

Beim Hochladen von Dateien, ist es äußerst wichtig die korrekte Länge der Datei in Byte zu übergeben. Dies geschieht mit dem Parameter *Content-Length*. Außerdem ist es ratsam die Berechtigung zu setzen. Wird keine Berechtigung gesetzt wird automatisch die Berechtigung des Buckets verwendet.

Mögliche Antworten (PUT Object)

```
200 – Das Hochladen der Datei war fehlerfrei.
400 – Fehler beim Hochladen der Datei.
404 – Bucket existiert nicht.
```

DELETE Object

Beispiel Anfrage zum Löschen einer Datei (DELETE Object)

```
DELETE /picture.svg HTTP/1.1
Host: bucketName.commondatastorage.googleapis.com
Date: Fri, 01 Oct 2010 22:25:53 GMT
```

```
Authorization: Bearer ya29.AHES6ZQhd...
Content-Length: 0
—— Anbieter spezifische Parameter ——
x-goog-api-version: 2
```

Mögliche Antworten (DELETE Object)

```
204 – Das Entfernen der Datei war erfolgreich.
404 – Die Datei existiert nicht.
```

GET Object

Beispiel Anfrage zum Löschen einer Datei (GET Object)

```
GET /picture.svg HTTP/1.1
Host: bucketName.commdatastorage.googleapis.com
Date: Fri, 01 Oct 2010 22:25:53 GMT
Authorization: Bearer ya29.AHES6ZQhd...
Content-Length: 0
—— Anbieter spezifische Parameter ——
x-goog-api-version: 2
```

Mögliche Antworten (GET Object)

```
200 – Das Herunterladen der Datei ist erfolgreich.
404 – Die Datei existiert nicht.
```

4 Schlusswort

Der File Hosting Service S3 stellt für einige Bereiche im Cloud Computing eine gute Lösung dar. Das System ist sehr leicht und schnell zu verstehen und somit auch weniger anfällig für Fehler. Bevor man jedoch den Schritt geht und entscheidet S3 in die Infrastruktur einzubinden, sollte man davor genau prüfen, ab welcher Größenordnung sich dieser Schritt überhaupt rentiert. Oftmals ist es für kleine Datenmengen günstiger diese selbst bereitzustellen. Auch muss man sich über die Sicherheit der Daten Gedanken machen. Die Rechenzentren von Google und Co, sind in der Regel zwar besser abgesichert als die eigenen, jedoch stehen diese zum Teil unter amerikanischem Recht, welches z.B sehr große Unterschiede zum deutschen Datenschutzrecht aufweist. Doch möchte man auf keinen Fall auf S3 verzichten, gibt es immer noch die Möglichkeit einen eigenen S3 Server aufzusetzen.

Literatur

1. Bucket and Object Naming Guidelines (07.05.2012)
<https://developers.google.com/storage/docs/bucketnaming>

2. Create a Bucket, Amazon Started Guide (07.05.2012)
<http://docs.amazonwebservices.com/AmazonS3/2006-03-01/gsg/CreatingABucket.html>
3. Google Parameters for the Access Control List (07.05.2012)
<https://developers.google.com/storage/docs/reference-headers?hl=de#xgoogacl>
4. Amazon Parameters for the Access Control List (07.05.2012)
http://docs.amazonwebservices.com/AWSSDKforPHP/latest/index.html#c=amazons3/ACL_AUTH_READ
5. Amazon REST API, PUT Object (07.05.2012)
<http://docs.amazonwebservices.com/AmazonS3/latest/API/RESTObjectPUT.html>
6. Amazon Developer Guide, Setting Up Notification of Bucket Events (07.05.2012)
<http://docs.amazonwebservices.com/AmazonS3/2006-03-01/dev/NotificationHowTo.html>
7. Google REST API (07.05.2012)
<https://developers.google.com/storage/docs/developer-guide?hl=de#overview>
8. Amazon REST API (07.05.2012)
<http://docs.amazonwebservices.com/AmazonS3/latest/API/APIRest.html>
9. Murty, James. Programming Amazon Web Services
1. Auflage O'Reilly. 2008 ISBN 978-0-596-51581-2