

Deltacloud

Stefan Düsing

Hochschule Mannheim
Fakultät für Informatik
Paul-Wittsack-Straße 10
68163 Mannheim
stefanduesing@googlemail.com

Zusammenfassung Heutzutage gibt es viele verschiedene Anbieter von Clouddiensten im Internet. Von den Virtuellen Servern in Amazon's Elastic Compute Cloud (EC2) bis zu vSphere von VMWare. Fast alle Dienstanbieter bieten jeweils auch Schnittstellen an um mit ihren Diensten einfach zu kommunizieren. Jedoch hat diese hohe Anzahl an Anbietern hat dazu geführt, dass unterschiedlichen Schnittstellen von Anbieter zu Anbieter entstanden sind. Ein bestimmter Standard für all diese Dienste ist von Nöten. In dieser Lücke kommt Deltacloud zum Einsatz. Dabei bietet es eine Schnittstelle an, die über alle Schnittstellen der Clouddienst-Anbieter steht.

Deltacloud bietet eine Schnittstelle zwischen Cloud-Anbieter und Client, die es ermöglicht jeden Cloud-Anbieter mit den gleichen Befehlen anzusprechen. Diese API wird über Rest angesprochen. Dabei ist es auch möglich sich zusätzliche Treiber für noch nicht unterstützte Clouds zu schreiben.

Bei Cloud Diensten spielen unter anderem folgende Faktoren eine wichtige Rolle: Skalierbarkeit: Die Software, Plattform oder die Infrastruktur muss je nach Bedarf leicht skalierbar sein.

1 Deltacloud

Mit Diensten wie Amazon Web Services, OpanStack, CloudStack, VMware haben Entwickler keinen Mangel an angebotenen Infrastructure as a Service Diensten (IaaS). Unglücklicherweise müssen sie sich auch mit den einzelnen APIs auseinandersetzen. Dies bringt einen erhöhten Aufwand mit sich, da die APIs sehr unterschiedlich sein können. Deltacloud, ein top-level Apache project, wurde erstellt um einerseits den Entwickler die Auseinandersetzung mit den einzelnen komplexen APIs zu bewahren und andererseits ihn die Möglichkeit geben einfach mit mehreren Diensten zu kommunizieren.

Zur Zeit arbeitet Deltacloud mit 11 verschiedenen Rechen APIs (von EC2 bis hin zu vSphere) und fünf verschiedenen Speicher APIs (unter anderem S3, Eucalyptus Walrus und Google Storage).

Was tut eigentlich Deltacloud? Es ist ein Server, der eine Standard REST-API für die verschiedensten Funktionen anbietet.

Bei vielen Clouddiensten bucht man sich Rechenzeit in Form von virtuellem Maschinen (z.B. bei Amazon EC2). Die meisten Anbieter benennen diese Maschinen mit „Instanzen“. Auf dieser Seite bietet Deltacloud Funktionen wie das Erstellen einer neuen Instanz, Neustarten einer Instanz, Abrufen der Hardware-Profile, d.h. die angebotene Hardware-Ausstattung des Clouddienst-Anbieters (z.B. CPU Geschwindigkeit, Arbeitsspeichergröße und Festplattengröße). Auch bietet Deltacloud das Abrufen der angebotenen Images (Images enthalten Software (Betriebssystem plus weitere Software), die man auf seine Instanzen spielen kann) an.

Wiederum besteht bei einigen Clouddiensten der Dienst aus dem Zurverfügungstellen von Speicher. So bietet z.B. Amazon mit seinem Simple Storage Service (S3) genau diesen Service an. Der Benutzer legt sich sogenannte „Buckets“ an, welche seinen Speicher darstellen.

Hier bietet dann Deltacloud eine Standard-API für das Erstellen von Buckets, das Manipulieren von Buckets und „Speicher-Blobs“ und dem Lesen und Schreiben von Daten.

Zusammengefasst bietet Deltacloud die Basis-Operationen an, die der Benutzer für das Interagieren mit Clouddiensten benötigt. Dabei kann er diese dann von jeglichen Management-Tools oder Applikation ausführen.

Um seine Dienste manuell zu verwalten bietet Deltacloud auch eine grafische Oberfläche in Form einer Weboberfläche an, die über den Browser abrufbar ist. Näheres dazu in Abschnitt XYZ.

Abbildung 1 veranschaulicht die Arbeitsweise von Deltacloud. Der HTTP-Client kommuniziert mit dem Deltacloud-Server über die REST API. Der Deltacloud Server übersetzt die angenommene Anfrage in den dienst-spezifische API-Aufruf des IaaS-Anbieters der ihn unterstützt. Falls ein Dienst keine eigene API bereitstellt, so benachrichtigt Deltacloud den Entwickler, sodass dieser darauf agieren kann. [3]

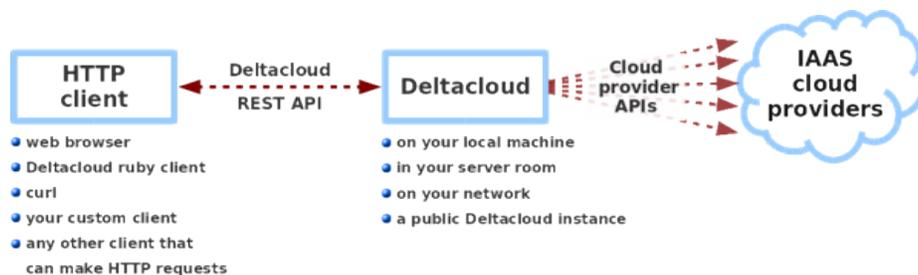


Abbildung 1. Das Design von Deltacloud

2 REST API

Die REST-API ist das Herzstück von Deltacloud. Denn durch sie ist der Entwickler gegenüber Änderungen an den einzelnen APIs der Dienste geschützt. Die einzelnen Cloud-APIs können sich ändern, aber diese hat keine Auswirkung auf die darüberstehende REST-API. Die Befehle funktionieren solange sie die eine REST-API befriedigen. [2]

2.1 Collections

Die folgenden Begriffe repräsentieren die Abstraktionen in der Apache Deltacloud API und werden hier eingeführt um sie dem Leser zu erklären. Jede repräsentierte Datensatz ist eine virtuelle Maschine oder ein Image im „back-end“ des Cloud-Anbieters. Es sollte beachtet werden, dass nicht alle Clouds die folgenden Collections unterstützen. Es funktionieren nur bestimmte Collections auf dem jeweiligen back-end. (z.B. Microsoft Azure driver unterstützt nur die collection „Buckets“)

Realms Eine eindeutige organisatorische Einheit, welche das back-end enthält, sie ist sowas wie ein Datacenter. Ein realm kann, muss aber nicht zwingend eine eindeutige geographische location der Ressourcen sein.

Instances Eine virtuelle Maschine, welche auf einer back-end cloud läuft. A realized virtual server, running in a given back-end cloud. Diese sind von den server Images instantiiert worden.

Images Dies sind Vorlagen (Virtual Machine Images), aus denen Instanzen erstellt werden. Jedes Image definiert die Root-Partition und den Speicher für die Instanz-Betriebssysteme.

Instance_States Diese stellen die Instanz-zyklus dar. Eine Instanz hat dauerhaft einen der Status: pending, running, stopped, shutting_down, finished.

Keys Stellen die Anmeldeautorisierung dar um auf die laufende Instanz zuzugreifen. Diese können vom Typ Schlüssel (z. B. ein RSA -Schlüssel) oder des Typs Kennwort (dh mit Benutzername und Kennwort -Attribute) sein.

Storage_Volume Dies ist ein virtuelles Speichergerät, das zu einer Instanz gehört und mit dieser die Instanz gestartet werden kann.

Storage_Snapshot Dies sind Kopien, snapshots eines Storage_Volume zu einem bestimmten Zeitpunkt.

Bucket Ein Container für den Datentyp Blob. Die organisatorische Einheit eines generischen Schlüssels ==j Wert basierten Datenspeicher (wie Rackspace CloudFiles oder Amazon S3). Einzelne Datenelemente (Blobs) sind eine Unterkategorie des buckets.

Blob Eine generische Binärdaten Element, das mit einem bestimmten bucket existiert (Ein Object in n Amazon S3 und Rackspace CloudFiles).

Address Stellt IP-Adressen dar. Abhängig von den Back-End-Cloud-Anbieter können diese öffentlich oder private sein. Im letzteren Fall, können diese nur innerhalb des back-end Netzwerkes benutzt werden.

Load Balancer Kümmt sich um die Verteilung von eingehenden Netzwerkverkehr einer IP-Adresse an einer Gruppe von Instanzen.

Firewalls Diese sind Regelwerke, die die Zugänglichkeit einer laufenden Instanz über das öffentliche Internet definieren. Die Einschränkung wird über die Freigabe nur bestimmter Ports an den Instanzen erreicht.

2.2 Client-Anfragen

Clients machen ihre Anfragen über HTTP mit Hilfe von REST unter Verwendung der üblichen Bedeutungen der Standard HTTP Verben (GET, POST, PUT und DELETE). Dabei folgt Deltacloud den REST Style Guides der Fedora Project Cloud API.

Die URL ist so aufgebaut, dass sie eine Ansammlung der Ressourcen beinhaltet, d.h. eine genaue Beschreibung der Anfrage.

Eine Anfrage würde dementsprechend folgendermaßen aussehen:

`http://localhost:3001/api/Realms/ [2]`

2.3 Authentication

Die Deltacloud API-Server ist stateless und hält keine Informationen über den aktuellen Client. Insbesondere behält er nicht die Anmeldedaten der cloud mit der der Client grade interagiert. Stattdessen verwendet er die HTTP Basic-Authentifizierung, und Clients müssen den Benutzername/Passwort für die Backend-Cloud bei jeder Anfrage mitsenden.

Die Informationen die gesendet werden müssen variieren dabei von Cloud zu Cloud. So verlangen machen Cloud-Dienste nur einen Benutzernamen und ein Passwort. Wiederum andere verlangen einen speziellen Schlüssel. [2]

2.4 Server-Antwort

Der Server ist in der Lage auf Client-Anfragen in verschiedenen Formaten zu antworten. Das Format der Antwort wird vom HTTP Inhalt bestimmt. Das Hauptformat ist XML. Es besteht aber auch die Möglichkeit JSON und HTML als Ausgabeformat zu wählen. So kann der Client explizit in seiner Anfrage bestimmen in welchem Format ihm geantwortet werden soll. Dies wird durch den „format=“ Paramater festgelegt:

```
http://deltacloudserver.foo/api?format=xml
```

oder

```
http://deltacloudserver.foo/api?format=json
```

Es kann auch z.T. der Detailgrad der Response gewählt werden. So würde z.B. GET /api/realms nur eine Liste der Objekte(in diesem Fall Realms) zurückliefern. Möchte man genauere Angaben zu einem Realm haben, muss man GET /api/realms/:id mit der ID-Angabe des jeweiligen Realms angeben. [2]

2.5 API Konventionen

Jedes XML-Element, das ein Objekt darstellt, wie z.B. eine Instanz hat ein href und ein id-Attribut. Das href-Attribut gibt die URL an, an der objekt-spezifische Aktionen durchgeführt werden können. So kann man z.B. wenn man die URL mit einem GET-Befehl verknüpft die Details zu einem Objekt abrufen. Das id-Attribut biete einen Identifier des Objekts an, welcher innerhalb der Sammlung eindeutig ist. Z.B. eindeutige ID für eine Instanz innerhalb einer Sammlung von Instanzen. [2]

Im generellen haben Objekte einen von menschen lesbaren Namen. Dieser wird in den <name/> angegeben.

2.6 API Stabilität und Evolution

Deltacloud garantiert, dass zukünftige Änderungen an der REST-API in solcher Form durchgeführt werden, dass trotzdem noch alte Clienten, die gegen die ältere Version der API entwickelt wurden, auch weiterhin mit der neuen API funktionieren. [2]

2.7 Beispielanfrage und Antwort

Um zu zeigen, wie eine Request und eine Response aufgebaut sind wird hier ein Beispiel gezeigt. Die Anfrage umfasst das Abfragen der verfügbaren Stand-

orte(Realms) für eine Instanz. Es wird verlangt, dass die Antwort in XML-Format gesendet werden soll. Danach werden die Authentifizierungsdaten und der User-Agent angegeben und zuletzt der Host. [2]

GET /api/realms

```
GET /api/realms?format=xml HTTP/1.1
Authorization: Basic AU1J3UB2121Afd1DdyQWxLaTYTmJMNF4zTXBoRGdhMDh2RUw5ZDAN9zVXVa=
User-Agent: curl/7.20.1 (i386-redhat-linux-gnu)
Host: localhost:3001
Accept: */*
```

Antwort

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 639
<?xml version='1.0' encoding='utf-8' ?>
<realms>
<realm href='http://localhost:3001/api/realms/us-east-1a'
id='us-east-1a'>
<name>us-east-1a</name>
<state>available</state>
</realm>
<realm href='http://localhost:3001/api/realms/us-east-1b'
id='us-east-1b'>
<name>us-east-1b</name>
<state>available</state>
</realm>
<realm href='http://localhost:3001/api/realms/us-east-1c'
id='us-east-1c'>
<name>us-east-1c</name>
<state>available</state>
</realm>
<realm href='http://localhost:3001/api/realms/us-east-1d'
id='us-east-1d'>
<name>us-east-1d</name>
<state>available</state>
</realm>
</realms>
```

3 Installation

Um Deltacloud einzurichten müssen folgende Schritte durchgeführt werden.

3.1 Dependencies

Deltacloud hat folgende Abhängigkeiten

- ruby (min 1.8.7) and ruby-devel
- gem (RubyGems)
- gcc-c++
- libxml2 and libxml2-devel
- libxslt and libxslt-devel
- rake

3.2 Installation

Nachdem die Dependencies installiert worden sind, muss man nur noch:

```
sudo gem install deltacloud-core  
eingeben.
```

4 Starten des Servers

Nachdem alles installiert worden ist muss nur noch

```
deltacloud -i driver -r ip -p port  
eingegeben werden.
```

-i gibt den zu ladenden Treiber an. z.B. EC2

-r ist nur notwendig, wenn man den Server auf einem externen Host laufen lässt. Dort wird die öffentliche IP eingetragen. Wenn -r nicht angegeben wurde wird standardmäßig localhost genommen

-p gibt den Port an. Wenn -p nicht angegeben ist wird standardmäßig der Port 3001 benutzt.

5 HTML-Benutzeroberfläche

Deltacloud bietet auch eine HTML-Benutzeroberfläche an über die die Möglichkeit des Verwaltens der Cloud-Dienste über den Webbrowser besteht. Dies stellt eine interessante Alternative zur direkten Nutzung der REST-API. Alle Funktionen die über die REST-API angesprochen werden können, sind auch in der HTML-Seite eingebaut.

6 Fazit

Mit seinen Funktionen bietet Deltacloud eine interessante Möglichkeit der Ansprache von Cloud-Diensten an. Der Entwickler muss sich nicht anständig

ändernde und unterschiedlichste APIs der einzelnen Cloud-Dienst-Anbieter anpassen. Er muss sich nicht durch die einzelnen APIs, die an Zahl nicht geringfügig sind, arbeiten. Deltaclouds Motto „Deltacloud. Many clouds. One API. No Problem.“ bringt das auf den Punkt. Einziges Manko ist, dass es noch einige Komplikationen beim Nutzen des EC2-Dienstes von Amazon gibt. Die aufgetretenen Probleme waren:

- Das Starten einer Instanz funktioniert nicht. Stattdessen wird eine Exception geworfen
- Beim Stoppen einer Instanz wird diese gelöscht, anstatt dass sie nur gestoppt wird.

Literatur

1. Deltacloud Homepage
<http://deltacloud.apache.org/>
2. Deltacloud Documentation - REST API and Developer Guide
<http://deltacloud.apache.org/api.html>
3. A look at DeltaCloud
<http://www.readwriteweb.com/cloud/2012/02/a-look-at-deltacloud-the-multi.php>