



Serverless Development in der AWS-Cloud

Marcus Treffert



Über mich

- Vor beinahe 20 Jahren: Ausbildung zum Fachinformatiker/ Anwendungsentwicklung
- Backend Developer bei BHS Corrugated Maschinen- und Anlagenbau GmbH
- Nebenberuflicher Koizüchter bei teichjuwelen GbR

Disclaimer



Serverless?



“Erstellen und Ausführen von Anwendungen
ohne sich über Server Gedanken machen zu
müssen”

- AWS



Wie funktioniert das?

Verwendung vorgefertigter, verwalteter Compute-, Storage- und Integrationslösungen.

- Betriebsaufwand fällt nahezu weg
 - Höhere Entwicklungsgeschwindigkeit
 - Fokus auf das Kundenproblem
- Skalierbar und hochverfügbar
- Pay-for-value Abrechnungsmodell

Populäre Serverless-Dienste



AWS Lambda



Amazon S3



Amazon API
Gateway



AWS AppSync



AWS Fargate



Amazon
DynamoDB



Amazon SQS



Amazon SNS



Amazon
Aurora



AWS Step
Functions



Storage und Datenbanken

S3 (Simple Storage Service)

- Object Storage
- Buckets und Keys
- Unbegrenzte Anzahl an Files
- 1 Byte - 5 TB Files pro File
- 11x9 Durability, 99,99 Availability
- Verschiedene Speicherklassen möglich
- Preise:
 - 0,023 USD / GB für Standardzugriff
 - 0,00099 USD / GB für Deep Archive

DynamoDB

- NoSQL-Datenbank
- Managed, Hochverfügbar
- Strongly consistent / Eventual consistent
- Abrechnungsmodelle On Demand / Provisioned Capacity
- Id besteht aus PartitionKey und SortKey
- Selektivität ist eingeschränkt, auch mit Indizes \Rightarrow Anwendungsfälle müssen vorher bekannt sein



Integration-Services

SQS (Simple Queue Service)

- Pull-basierter Queue Service
- Standard-Queues
 - Kein Transaction Limit für Standard
 - Best Effort-Order
 - At Least once-Delivery
- Fifo-Queues
 - 300 Transaktionen / s
 - Exactly once-Delivery
- Nachrichten bis 256 KB

SNS (Simple Notification Service)

- Push based / Publish Subscribe
- Flexible Subscriber:
 - Email
 - Http
 - SMS
 - SQS
 - Apple
 - Google Firebase
 - Lambdas



API-Gateway

- REST- und Websocket-APIs
- Zahlung ausschließlich für Aufrufe und Datenübertragung
- Standardmäßig bis zu 10000 Anforderungen pro Sekunde
- Staging
- Authorization-Verfahren (Cognito, IAM, Api-Key)
- Metriken



Lambda

- Ausführung von Code ohne Server bereitzustellen
- Function as a Service
- Event-Based
- Möglich in vielen Programmiersprachen (Java, Python, Node.js, Go)
- Inzwischen sind auch Custom-Runtimes möglich
- Skaliert automatisch horizontal (bis zu einem gewissen Grad)
 - Lifecycle (Init, Invoke, Shutdown)
 - Cold-Start führt zu Latenz
- Zustandslos (beinahe)
- Einschränkungen:
 - Laufzeit maximal 15 Minuten
 - Maximal 10 GB RAM



Cloudformation

- Infrastructure as Code - Service der AWS-Cloud
- Gewünschte Infrastruktur wird deklarativ in JSON definiert
- Cloudformation provisioniert dann den beschriebenen Stack
- Stack kann aktualisiert und wieder abgerissen werden
- ⇒ Ideal für CD
- Cloudformation selbst ist kostenlos. Die Infrastruktur verursacht aber Kosten
- Sehr kleinteilige Definition. Bsp.: Lambda, zugehörige Role und Log-Groups müssen separat definiert und verknüpft werden.
- Code muss bereits in der Cloud sein um verwendet werden zu können
- Abstraktionsschichten vereinfachen die Entwicklung ⇒ Amplify, Sam, Cloud Development Kit CDK, Serverless-Framework



Serverless-Framework

- <https://www.serverless.com/>
- Beschreibt Infrastruktur in einem YAML-File
- Vereinfachte Definition der Basisumgebung, Funktionen und Events
- Automatische Erstellung der Abhängigkeiten (z.B. Log-Groups, S3-Buckets, APIs)
- Ressourcen-Block ermöglicht klassische Ressourcen-Definition via Cloudformation-Templates



Szenario: Verwaltung einer Artikelbestandsliste

- Bestandsdaten werden per Upload als CSV bereitgestellt.
- Die Daten können sehr groß werden, deshalb wird das speichern in der Datenbank mittels Queue entkoppelt
- Die gespeicherten Bestände sind via REST-Schnittstelle abrufbar



Prerequisites

- Serverless ist installiert (z.B. via npm)
- User und Deployment-Role in AWS
- Eintrag in [HOME]/.aws/credentials

```
##### dcn-serverless-demo #####

[dcn-serverless-demo]
aws_region=eu-central-1
aws_access_key_id=XXX
aws_secret_access_key=XXX

[dcn-serverless-demo_deployment-role]
region=eu-central-1
role_arn=arn:aws:iam::XXX:role/dcn-serverless-demo_deployment
source_profile=dcn-serverless-demo
```



Deploy/Delete

- `$ serverless deploy --aws-profile=dcn-serverless-demo_deployment-role`
- `$ serverless remove --aws-profile=dcn-serverless-demo_deployment-role`

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with subtle diagonal lines.

Vielen Dank!

Marcus Treffert