

# AWS SERVERLESS MIT CDK

ein interaktiver Workshop mit Max Tharr

MAYFLOWER

1. Introduction
2. Was meint Serverless?
3. AWS Serverless Services
4. Cloud Development Kit (CDK)
5. Setup for this workshop
6. Contribute your own Lambdas!

# SERVERLESS



A: Die Infrastruktur, die für eine Anwendung benötigt wird, wird vollständig vom Cloud-Anbieter verwaltet.

B: Wenn deine Software Geld kostet, wenn sie niemand verwendet, dann ist es *nicht* Serverless

# AWS SERVERLESS SERVICES\*

\*Eine Auswahl

# LAMBDA



- Serverless Computing Service
- AWS stellt eine Umgebung und ein SDK bereit, so dass man 'nur' Code schreiben muss

# SQS (SIMPLE QUEUE SERVICE)



- Asynchroner messaging service
- Gut zum decouplen von workloads
- Gut integriert in andere AWS services

# API GATEWAY



- API Endpunkt
- Rest API oder generische HTTP(S) API
- Kümmert sich um SSL und validieren von Requests, leitet an andere Services weiter

# DYNAMODB

- "NoSQL" Datenbank (Nicht wie MongoDB)
- Ein Key-Value-Store, der auch mit sehr vielen Einträgen sehr schnelle Antworten liefert
- Prinzip: Filtern ist schneller als ein Join



# CLOUD WATCH

Logs und Metriken



Tracing, integriert in CloudWatch



**CLOUD DEVELOPMENT KIT**



- Infrastructure-as-Code Ansatz von AWS
- Basiert auf CloudFormation
- Modulares Erstellen von Stacks in für Entwickler vertrauten Programmiersprachen
- Funktioniert u.a. mit Python, TypeScript, Java, C#



- Infrastructure-as-Code Ansatz von AWS
- Basiert auf CloudFormation
- Modulares Erstellen von Stacks in für Entwickler vertrauten Programmiersprachen
- Funktioniert u.a. mit Python, TypeScript, Java
- **Funktioniert richtig gut nur mit TypeScript**

## Beispiel:

```
const entryQueue = new cdk.aws_sqs.Queue(this, 'entry-queue')

const entryFunction = new cdk.aws_lambda_nodejs.NodejsFunction(
  this, 'entrypoint', {
    runtime: cdk.aws_lambda.Runtime.NODEJS_18_X,
    tracing: Tracing.ACTIVE,
    environment: {
      QUEUE_URL: entryQueue.queueUrl
    }
  }
)
entryQueue.grantSendMessages(entryFunction);
```

# WORKSHOP



- Ich habe einen CDK Stack vorbereitet, der das Kinderspiel "Stille Post" mit Hilfe von Lambdas nachspielt.
- Jedes Lambda bekommt einen String als Input, soll ihn transformieren und dann an eine SQS Queue weitergeben. Dabei wird die ursprüngliche Nachricht immer weiter verfälscht, so wie bei Stille Post

**WIE GEHT DAS KONKRET?** 🤔



Ich habe zwei Beispiele im Ordner `workers` angelegt.  
Einmal **JavaScript** , einmal **Python** 

### WAS IHR TUN MÜSST:

Das Projekt auf github forken, den Ordner `py-example` oder `js-example` kopieren, dabei muss euer neuer Ordnername auch den prefix `py-` oder `js-` behalten, damit mein CDK-Code erkennt, welche Runtime benutzt wird.

Danach im Code die Lambda nach euren Wünschen verändern!

## ONE MORE THING 🍏:

Falls ihr ChatGPT verwenden wollt: Ich habe in der Environment einen API key hinterlegt und ein Beispiel mitgeliefert 😊

<https://github.com/mayflower/telephone-game>

M  YFLOWER

