Getting started with *serverless (Lambda)*

Auditorium: Tech-Community

Host: Max

Agenda

Was haben wir vor

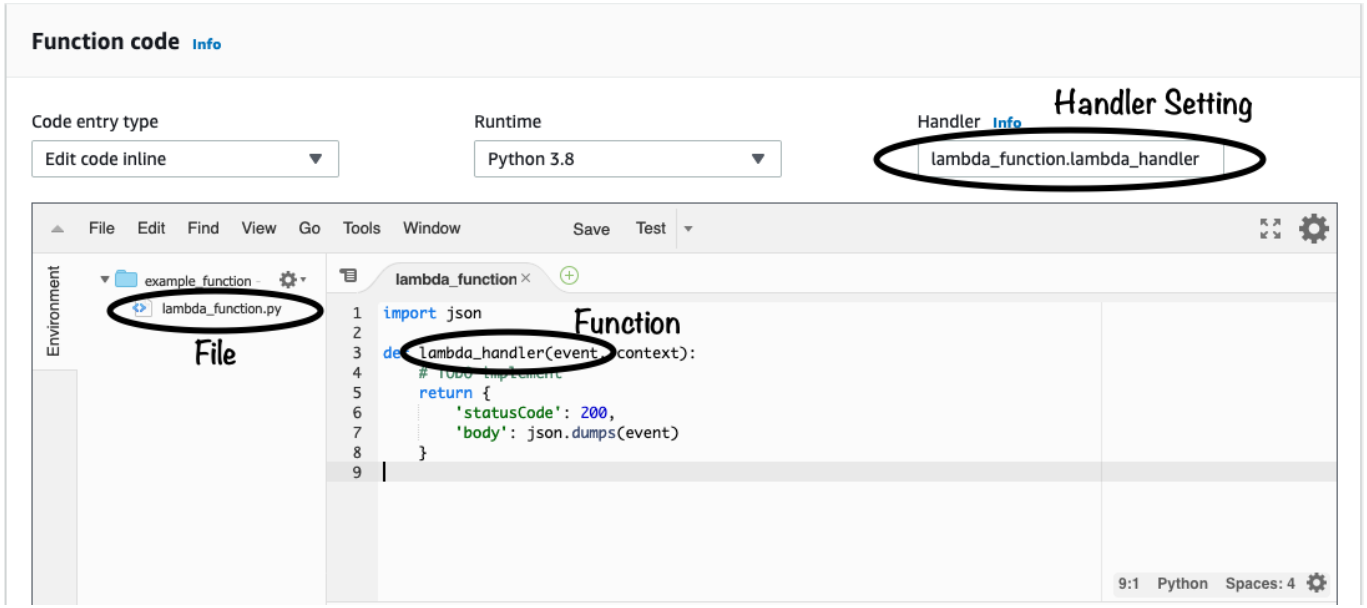
- Einfache Funktion mit AWS SDK **boto3** für Python
- Externe Module in Lambda Funktionen benutzen mit **Layers**
- Lambda-Funktionen lokal ausführen **ALS OB** man in einem AWS Environment wäre
- Serverlose **CRUD**-Anwendung mit **Lambda**, **API Gateway** & **DynamoDB**

Lambda

- Lambda ist das **AWS-eigene Produkt**, serverlose Funktionen bereitzustellen
- Lamba führt Code aus, ohne den **Aufwand zur Bereitstellung und Verwaltung von Infrastruktur**
- **Easy Scaling**
- **Integration** in viele verschiedene AWS Services
- **Ereignisgesteuert (events)**

Was ist das?

Lambda




```
def lambda_handler(event, context):
    return some_value
```

Standard Aufbau einer Lambda-Funktion. Es werden immer die Argumente und ein Rückgabewert benötigt.

Was ist das?

Lambda event

- Daten, die an die Funktion gesendet werden (kann von Allem kommen, was die Funktion auslöst, z.B. HTTP Request vom API-Gateway


```
{
    "resource": "/",
    "path": "/",
    "httpMethod": "GET",
    "requestContext": {
        "resourcePath": "/",
        "httpMethod": "GET",
        "path": "/Prod/",
        ...
    },
    "headers": {
        "accept": "text/html",
        "accept-encoding": "gzip, deflate, br",
        "Host": "xxx.us-east-2.amazonaws.com",
        "User-Agent": "Mozilla/5.0",
        ...
    },
    "multiValueHeaders": {
        "accept": [
            "text/html"
        ],
        "accept-encoding": [
            "gzip, deflate, br"
        ],
        ...
    },
    "queryStringParameters": {
        "postcode": 12345
        },
    "multiValueQueryStringParameters": null,
    "pathParameters": null,
    "stageVariables": null,
    "body": null,
    "isBase64Encoded": false
}
```

Was ist das?

Lambda context

- Stellt hauptsächlich Informationen über die derzeitig ausgeführte Umgebung bereit

Context methods

- **get_remaining_time_in_millis** – Returns the number of milliseconds left before the execution times out.

Context properties

- **function_name** – The name of the Lambda function.
- **function_version** – The version of the function.
- **invoked_function_arn** – The Amazon Resource Name (ARN) that's used to invoke the function. Indicates if the invoker specified a version number or alias.
- **memory_limit_in_mb** – The amount of memory that's allocated for the function.
- **aws_request_id** – The identifier of the invocation request.
- **log_group_name** – The log group for the function.
- **log_stream_name** – The log stream for the function instance.

Was ist das?

DynamoDB & API Gateway

- Serverlose NoSQL-Datenbank mit Key-Value Prinzip
- API Management zwischen Client und Backend, verknüpft Services

Let‘s build

Hands on !

Nützliche Links

Thema	Beschreibung	Ref.
Lambda	Produktübersicht	AWS Lambda Data Processing - Datenverarbeitungsdienste (amazon.com)
Lambda API	Dokumentation	API reference - AWS Lambda (amazon.com)
DynamoDB	Produktübersicht	AWS | Amazon DynamoDB – NoSQL Online Datenbank Service
API Gateway	Produktübersicht	Amazon API Gateway & API-Entwicklung – Amazon Web Services (AWS)
AWS SDK boto3 für Python	Dokumentation	Boto3 documentation — Boto3 Docs 1.26.16 documentation (amazonaws.com)
LambCI	Container Images for serverless	lambci/docker-lambda: Docker images and test runners that replicate the live AWS Lambda environment (github.com)
SAM	Lambda local testing	Testing Lambda container images locally - AWS Lambda (amazon.com)