

Serverless

Node.js Webanwendungen mit AWS Lambda entwickeln

Oliver Gutperl

Über mich

Oliver Gutperl

Kindheit: Softwareentwickler

Architekt, VP, CTO

heute: Kapitän @  Digital Sailors

Autor: „Nginx richtig konfigurieren“



Was macht eine Webanwendung?

Liefert dynamische Webseiten aus

Verarbeitet Formulare und nimmt POST Requests entgegen

Stellt eine API bereit (REST, JSON, ...)

Server

 Zu viele oder zu wenige

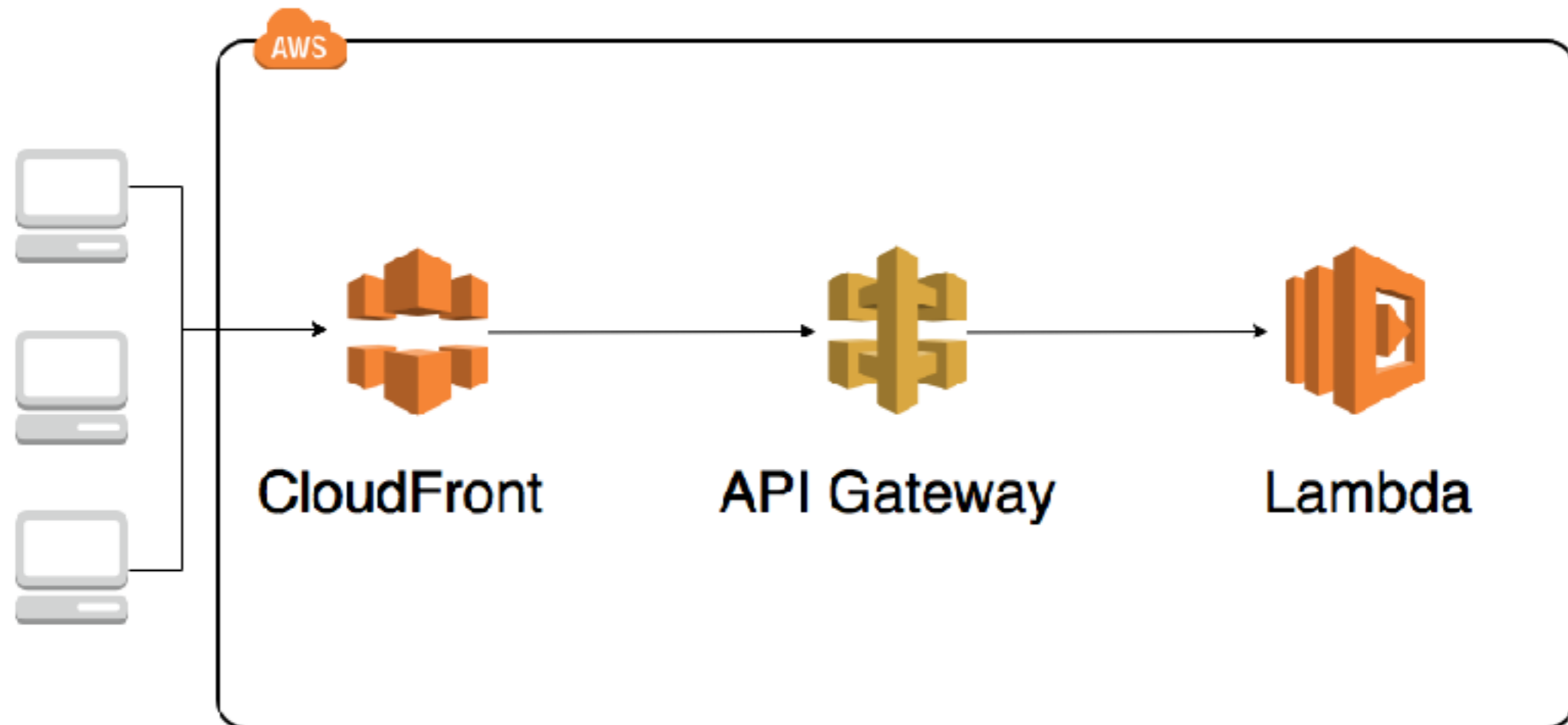
 Betriebssystemupdates

 SSH

 Deployment



Architektur



Was ist AWS Lambda?

„Lambda ist S3 für Programmlogik“

„Function as a Service“



Funktion?

~~FaaS - Function as a Service~~

REaaS - Runtime Environment as a Service

Größe: Eine „Funktion“ = Ein Microservice

Ein Microservice = Bounded Context/DDD, kann von einem Entwickler überblickt und verstanden werden

Andere Anbieter

Google Cloud Functions

Microsoft Azure Functions

IBM OpenWhisk (Open Source)



Umgebung

Node.js 6.10

Stateless

128 MB bis 1536 MB RAM

- mehr RAM 👉 schnellere CPU

512 MB flüchtiger(!) Speicher in /tmp

Event-basiertes Entwicklungsmodell

Minimale Lambda Funktion

```
exports.handler = function(event, context, callback) {  
  console.log(event);  
  callback(null, { output: 'Hello World' });  
};
```



API Gateway

Aufruf von Lambda Funktionen per HTTP

REST API

- Ressourcen und Methoden (GET /product/)

Lambda Proxy Integration

API Gateway

The screenshot displays the Amazon API Gateway console interface. The breadcrumb navigation at the top reads: Amazon API Gateway > APIs > Lambda Demo (ui1535tpi9) > Resources > /{proxy+} (1acgr8) > ANY. The left sidebar lists various API Gateway components, with 'Resources' selected under the 'Lambda Demo' API. The main content area is titled '/{proxy+} - ANY - Method Execution' and features a 'TEST' button with a lightning bolt icon. The diagram illustrates the flow of a request and response:

- Client:** Initiates the request.
- Method Request:** Shows 'Auth: NONE' and 'ARN: arn:aws:execute-api:eu-west-1:621073008195:ui1535tpi9/ /ANY/'.
- Integration Request:** Shows 'Type: LAMBDA_PROXY'.
- Integration Response:** Shows the message 'Proxy integrations cannot be configured to transform responses.'
- Method Response:** Shows 'HTTP Status: Proxy'.
- Lambda api-echo:** The target service for the proxy integration.

Event

```
{
  "resource": "/{proxy+}",
  "path": "/test",
  "httpMethod": "GET",
  "headers": {
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "Accept-Encoding": "gzip, deflate, br",
    "Accept-Language": "de,en-US;q=0.7,en;q=0.3",
    "Host": "j1thsk3qj3.execute-api.eu-west-1.amazonaws.com",
    [...]
  },
  "queryStringParameters": null,
  "pathParameters": {
    "proxy": "test"
  },
  "stageVariables": null,
  "requestContext": {
    "accountId": "123456789012",
    "resourceId": "splpe3",
    "stage": "prod",
    "requestId": "b967ad24-9515-11e6-a00a-3d71e07b9f7e",
    "identity": { [...] },
    "resourcePath": "/{proxy+}",
    "httpMethod": "GET",
  },
  "body": null
}
```

Event

```
exports.handler = function(event, context, callback) {  
  if (event.path === '/test') {  
    ...  
  }  
};
```

HTTP-Response

```
exports.handler = (event, context, callback) => {
  callback(null, {
    statusCode: 200,
    body: '<html><body>Hello World</body></html>',
    headers: {
      'Content-Type': 'text/html',
    }
  });
};
```

Weitere Services

require() funktioniert

- npm-packages ("node_modules") im zip-File mitliefern

aws-sdk ist direkt verfügbar

RDS - Relationale Datenbanken

SES - E-Mail Versand

SNS - SMS oder Push-Nachrichten

S3 - Speicher für Dateien

Polly - Sprachausgabe

Deployment

Grundkonfiguration per AWS Konsole

```
zip -r lambda-function.zip lambda-function.js node_modules  
&& aws lambda update-function-code --function-name lambda-  
function --zip-file fileb://lambda-function.zip --publish
```

Vom Laptop in Produktion

Versionierung innerhalb von AWS



Limits

maximale Ausführungszeit: 5 Minuten

Request/Response Body Payload: 6 MB

Gleichzeitige Ausführungen: 1000

Code: 250MB (50MB gezippt)

Tipps

Cold Start

- kann 3-5 Sekunden dauern
- kann durch Inaktivität ausgelöst werden

No-Op in Funktion zur Starthilfe

- z.B. regelmäßiger Cloudwatch Event
- Ping bei POST-Formular



Preise

Preise sind Architekturkriterium

Dauer [msec] x Speicher [128MB - 1536MB]
x 0,00001667 USD pro GB/Sekunde
+ 0,0000002 USD pro Ausführung
= Gesamtpreis

Beispiel:

300 msec x 128MB x 0,00001667 USD/GB/sek
+ 0,0000002 USD
= 0,000000626 USD

1 USD \approx 1,6 Mio. Ausführungen



Testing & Debugging

Logs in CloudWatch

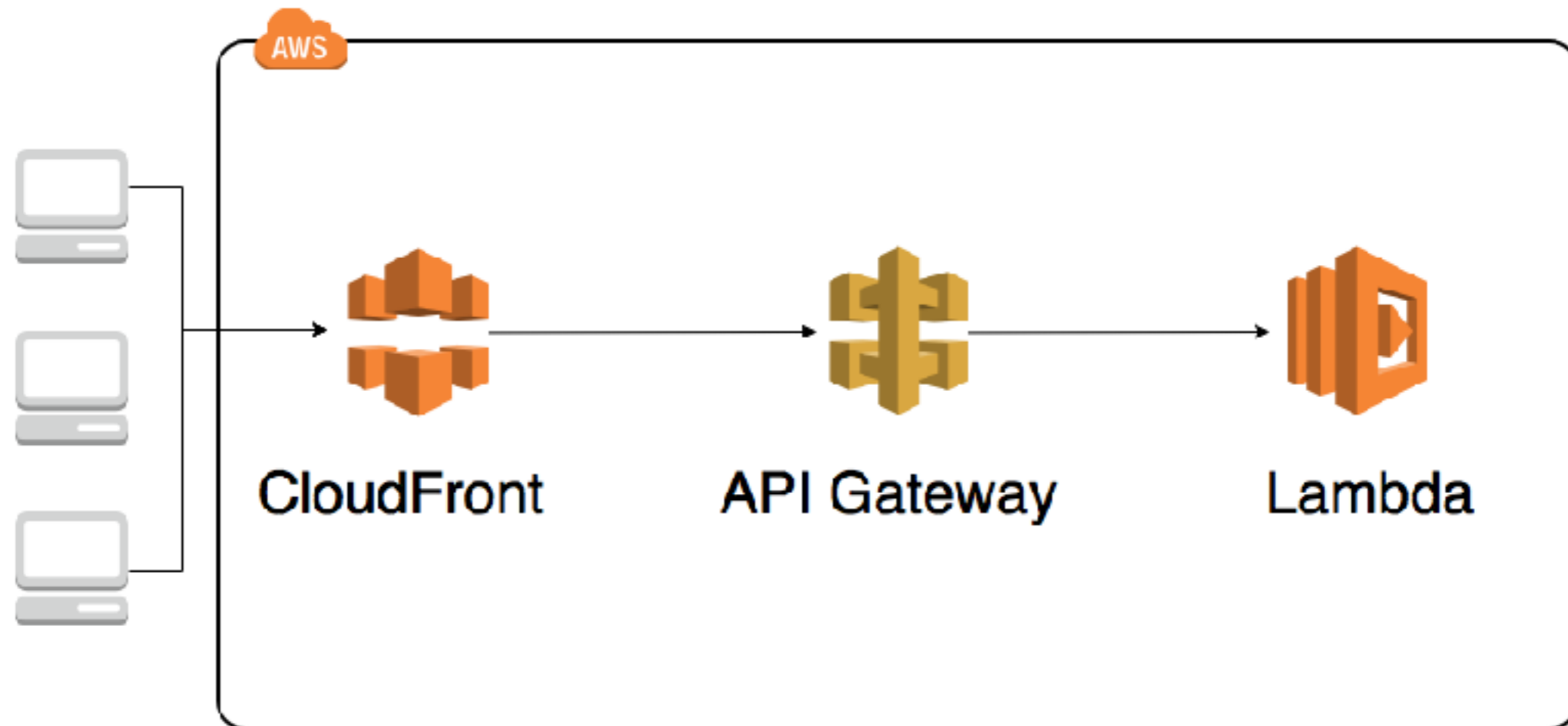
Tracing durch AWS Services mit X-Ray

Leicht in in Testframeworks integrierbar:

```
exports.handler({ input : 'Hello Lambda' },
null, function(err, value) {
  if (err) {
    assert.fail(err);
  } else {
    assert(value);
  }
});
```



Architektur



CloudFront

Routing

IPv6

HTTP/2

Domain

Caching

Shield/WAF

Content Delivery Network



Kosten

1 Mio. Requests, 20KB, 200 ms

- Lambda: <1 EUR
- API Gateway: <5 EUR
- CloudFront: <5 EUR


Gesamt <11 EUR

Grobe Abschätzung, Freikontingente nicht berücksichtigt

Zusammenfassung

- Serverless
- Automatische Skalierung
- Einfacher Entwicklungszyklus
- Ausprobieren!

 Fragen?

 oliver.gutperl@digital-sailors.de

 @Digitalkapitaen

 <https://www.digital-sailors.de>