Titel: Design und Implementierung einer Serverless Infrastructure Anwendung beim Cloud Anbieter Amazon Web Services

Bachelorarbeit
zur Erlangung des Grades *Bachelor of Science*

an der
Hochschule Niederrhein
Fachbereich Elektrotechnik und Informatik
Studiengang *Informatik*

vorgelegt von
Oktavius Wiesner
Matrikelnummer: 1082104

Datum: 7. Juli 2020

Prüfer: Prof. Dr. Peter Davids
Zweitprüfer: Maik Glatki

Zusammenfassung

Themen:

- Was ist Serverless

- Prinzipien von Serverless Architektur

- Architekturmodelle Vergleich (Monolithisch, Microservice, Serverless)

- Function as a Service

- Serverlessdienste bei AWS (Lambda, API . . .)

- Security Aspekte bei Serverless

- Kosten / Vergleich zu Server Infrastruktur

- Aufbau Projekt bei AWS

Abstract

Die vorliegende Bachelorarbeit beschäftigt sich im Detail mit der Serverless Architektur und Function as a Service. Zur Verständlichkeit werden die verschiedenen Architekturmodelle verglichen und bewertet. Die Bachelorarbeit beschränkt sich auf den Cloud Provider Amazon Web Services und der entsprechenden Dienste. Dabei werden AWS Dienste wie Lambda, Cognito, AppSync, DynamoDB und Amplify genauer untersucht und bewertet. Auf Basis der untersuchten Dienste wird ein Prototyp bei AWS entwickelt und implementiert. Das Ziel ist eine moderne Web Applikation für die Mitarbeiter der Mediengruppe RTL, die komplett auf Serverless Architektur basiert und die in der Bachelorarbeit erwähnten Vorteile vollständig ausnutzen kann.

Eidesstattliche Erklärung

Name:	Oktavius Wiesner
Matrikelnr.:	1082104
Titel:	Titel: Design und Implementierung einer Serverless Infrastructure Anwendung beim Cloud Anbieter Amazon Web Services English title

Ich versichere durch meine Unterschrift, dass die vorliegende Arbeit ausschließlich von mir verfasst wurde. Es wurden keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt.

Die Arbeit besteht aus _____ Seiten.

_____ _____

Ort, Datum Unterschrift

Hinweis

Bei allen Ausführungen im Folgenden, die auf Personen bezogen sind, meint die gewählte Formulierung beide Geschlechter, auch wenn aus Gründen der sprachlichen Vereinfachung und der besseren Lesbarkeit die männliche Form gewählt wurde.

Danksagung

Hier kommt eine Danksagung...

Inhaltsverzeichnis

1 Einleitung

1.1 Aufgabenstellung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1.2 Motivation

Die Abteilung Datacenter and Clouds innerhalb von CBC beschäftigt sich mittlerweile seit einigen Jahren mit Cloud Infrastruktur. Zu Beginn wurden größenteils dynamisch Linux Server mit einer SQL Datenbank und einem Loadbalancer realisert(Infrastructure as a Service). Mit der Zeit wurden auch Container basierte Varianten sowie Platform as a Service Lösungen umgesetzt. Mittlerweile gibt es den Wunsch sich auch mit Function as a Service zu beschäftigen um vorallem eine schnelle Bereitstellung von Diensten zu geringen Kosten zu ermöglichen.

Da sehr viele Firmen innerhalb der Mediengruppe RTL mit AWS und anderen Cloud Providern intensiv arbeiten ist zum Beispiel die Kostenzuweisung unübersichtlich geworden. Deshalb besteht innerhalb der Abteilung Datacenter and Clouds der Wunsch nach einer modernen Web Applikation welche bestimmte Informationen der jeweiligen Cloud Provider zentral sammelt und zur Verfügung stellt. Im Rahmen der Bachelorarbeit soll dafür ein Prototyp entstehen der effizient und einfach in Zukunft um weitere Anforderungen erweitert werden kann.

1.3 CBC Cologne Broadcasting GmbH

Die Bachelorarbeit wird innerhalb der Räumlichkeiten der Firma CBC Cologne Broadcasting GmbH in Köln Deutz realisiert. CBC ist ein Unternehmen der Mediengruppe RTL Deutschland. Dazu gehören unter anderem die Fernsehsender RTL Television, RTL Nitro, N-TV und Vox. Mit ca. 550 festen Mitarbeitern ist CBC für die Produktion, Programmverbreitung, Sendeabwicklung sowie die IT Infrastruktur verantwortlich. Die Abteilung Datacenter and Clouds beschäftigt sich dabei um jegliche Infrastruktur, sowohl OnPremises als auch bei den Cloud Providern Amazon Web Services, Microsoft Azure sowie Google Cloud Plattform. Beispiele für Projekte die in der Cloud umgesetzt wurden sind die Streaming Plattform TVNow sowie die Internetpräsenz des Nachrichtensenders N-TV.

2 Theorie

2.1 Definition

Was genau ist Serverless?

Serverless bedeutet... blaa blaaa blaa...

3 Fazit

3.1 Zusammenfassung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

3.2 Ausblick

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Literaturverzeichnis

Anhang

A Anhang Teil 1

Listing 1: Formularmanipulation

```
$( document ).ready(function() {


   if ( $('#TimeUnits').length )
   {
      $('#TimeUnits').autocomplete({ disabled: true });
   }


   var ITPROLABCurrentAction = $.getUrlVar('Action');

   if ( ITPROLABCurrentAction == 'CustomerTicketMessage' )
   {
      var ITPROLABQueue   = $( "#Dest option:selected" ).text();
      alert( ITPROLABQueue );

      if (ITPROLABQueue != 'ITPROLAB' ) {
         $('#ITPROLABServiceID').fadeOut();
      }
   }



$("<style>")
   .prop("type", "text/css")
   .html("\
   [title=\"10-blue\"] {\
      background-color: blue;\
      color: blue;\
      font-size: 1px;\
      opacity: 0.7;\
   }")
   .appendTo("head");
```

B Anhang Teil 2

Listing 2: Formularmanipulation

```
#!/usr/bin/perl
##
```

```
# generate PDF from Latex Template LK.tex --> lk.tex --> lk.pdf
# wird von machform formular aufgerufen.
##

use CGI qw/:standard/;
use DBI;

my $debug = 0;



my $template;
my $german = "LK/LK-D.tex";
my $english = "LK/LK-E.tex";
my $arabic = "LK/LK-A.tex";
my $lk    = "LK/lk.tex";



my $message = "<pre>\n";
my $dbm = DBI->connect (
   ↪ "dbi:mysql:$database:$server",$userid,$passwd ) || die "Could
   ↪ not connect to $database";


$stm = $dbm->prepare ( qq{ select id,element_1, element_2,
   ↪ element_5, element_6, element_7, element_8,
   ↪ element_10_1,element_12 from $table ORDER by id DESC LIMIT 1
   ↪ });
$stm->execute;
$stm->bind_columns(\$id,\$e1,\$e2,\$e5,\$e6,\$e7,\$e8,\$e10,\$e12);
if ($stm->fetch) {

   if ($e5 =~ /-/) {
      $e5 = substr ($e5,8,2) . "." . substr ($e5,5,2) . "." . substr
         ↪ ($e5,0,4);
   }
   $message .= " ID = $e1\n";
   $message .= " Name = $e2\n";
   $message .= " Datum = $e5\n";
   $message .= " Adresse= $e6\n";
   $message .= " E = $e7\n";
   $message .= " K = $e8\n";
   $message .= " data = $e10\n";
   $message .= " lang = $e12\n";

}
else {
   $message .= " nicht gefunden\n";
}
```

```
printf "$message\n" if ($debug);

undef $stm;
undef $dbm;
```

C Anhang Lambda

Listing 3: Lambda-Code

```
   /* Amplify Params - DO NOT EDIT
   ENV
   REGION
Amplify Params - DO NOT EDIT */

var AWS = require('aws-sdk')
AWS.config.update({region: 'eu-central-1'});

var docClient = new AWS.DynamoDB.DocumentClient

// Create S3 service object
const s3 = new AWS.S3({apiVersion: '2006-03-01'});


const s3listfunction = new Promise((resolve, reject) => {
  let s3buckets_data = s3.listBuckets().promise();
  if(s3buckets_data ) {
    resolve("Works" + s3buckets_data);
  }
  else {
    reject("NOPE")
  }
  //return s3.listBuckets().promise()
});

const s3listfunction_1 = () => {

  return s3.listBuckets().promise()
};

var params = {
    TableName: 'Comments-ifdtxan4k5fglip5ynnopk6bky-dev',
 //  Key: {'id': '4de8a026-4599-465a-ac76-9259b4adf1fc'}
   };


   async function getAllDynamoDBItems(){
      console.log("Starting Function")
```

```
    try {
        var result = await docClient.scan(params).promise()
        console.log(JSON.stringify(result))
        console.log("Success")
    } catch (error) {
        console.log("FAILED")
        console.error(error);
    }
}



exports.handler = async (event) => {

    //getAllDynamoDBItems()
    //console.log( "S3 Ausgabe: " + s3.listBuckets().promise() )

    //const msg = await getAllDynamoDBItems()
    //const msg1 = await s3listfunction_1()
    //console.log(msg1)

    s3listfunction_1.then((res) => console.log(res), (err) =>
        ↪ alert(err));

    // TODO implement
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
    //return s3.listBuckets().promise();
    return response
};




function getCoins(callback) {
    console.log("Function starts")
  docClient.scan(params, function(err, data) {
    if (err) {
      callback(err)
      console.log("IFFFF")
      console.log(err)
    } else {
        console.log("ELLSEE")
        console.log(data.Items)
      callback(null, data.Items)
    }
```

```
  });
}
```