Bericht

**Public Cloud Services HS23**

Dozent: Sebastian Graf

Studenten: Fabian Heuberger  
Etienne Frei  
Yannick Hohler

CI Pipeline on AWS

Ein Bild, das Rad, orange enthält.

Automatisch generierte Beschreibung

# Inhaltsverzeichnis

[Use Case 3](#_Toc154850147)

[Architektur 4](#_Toc154850148)

[Umsetzung 5](#_Toc154850149)

[Erkenntnisse 6](#_Toc154850150)

[Fazit 6](#_Toc154850151)

# Use Case

# Architektur

Gewählte Architektur inkl. evt. Alternativen

Verargumentieren der Entscheidung anhand der gelernten Prinzipien

Ein Bild, das Text, Diagramm, Screenshot, Plan enthält.

Automatisch generierte Beschreibung

Abbildung 1 Gesamte Architektur

# Umsetzung

Vorgehensweise für Umsetzung

Beschreibung der Implementation inklusive Configs

In diesem Kapitel wird der Vorgang beschrieben um die Pipeline in zu konfigurieren. Das Kapitel ist bezüglich den einzelnen Services unterteilt.

Rollen und

### IaC

Weil das Projekt nur auf AWS umgesetzt wird, wurde beschlossen AWS CloudFormation zu benutzen.

* Es bietet eine tiefere Integration mit anderen AWS-Services und -Funktionen
* Die Syntax ist deklarativ, was bedeutet, dass Sie Ihre gewünschte Infrastruktur konfigurieren
* CloudFormation bietet eine grafische Benutzeroberfläche namens AWS CloudFormation Designer

#### Stack erstellen

Jeder Stack basiert auf einer Vorlage. Eine Vorlage ist eine JSON- oder YAML-Datei, die Konfigurationsinformationen über die AWS-Ressourcen enthält.

### AWS S3

Von Hand erstellt

Name: [codebuild-artifacts-pcls](https://s3.console.aws.amazon.com/s3/buckets/codebuild-artifacts-pcls?region=eu-central-2&bucketType=general) (Für Artefakte von CodeBuild)  
Name: [angular-deployment](https://s3.console.aws.amazon.com/s3/buckets/angular-deployment?region=eu-central-2&bucketType=general) (Als Pipeline Source?)

### AWS CodeBuild

[] Erstellen Sie ein AWS CodeBuild-Projekt, das Ihren Build-Prozess definiert. Stellen Sie sicher, dass das Projekt so konfiguriert ist, dass es auf Änderungen im Quellcode reagiert.

* codebuild.yaml

[]Webhook zwischen Github und CodeBuild erstellen

**Create a personal access token in GitHub: (Owner or Admin of Repo)**

-Go to your GitHub account settings.

-Navigate to "Developer settings" and select "Personal access tokens".

-Click on the "Generate new token" button.

-Give the token a descriptive name and select the necessary scopes or permissions (e.g., repo, admin:repo\_hook).

-Click on the "Generate token" button.

-Copy the generated access token.

-Configure the access token in AWS CodeBuild:

-Open the AWS Management Console and navigate to the CodeBuild service.

-Find the CodeBuild project that you are working with.

-Open the project settings.

-Under the "Source" section, locate the "GitHub" configuration.

-Paste the access token you generated from GitHub into the "Access Token" field.

-Save the project settings.

### Amazon SQS

### Amazon Simple Queue Service (SQS) und Amazon CloudFront sind zwei separate AWS-Services, die nicht direkt miteinander konfiguriert werden können. Deshalb wird dieser Service manuell konfiguriert über die AWS Konsole.

### AWS Lambda

Kann mit Cloudfront konfiguriert werden: <https://docs.aws.amazon.com/de_de/AWSCloudFormation/latest/UserGuide/aws-properties-lambda-function-code.html>

Im zipFile ist der Code vorhanden. Dieses ZipFile kann auf einem s3 abgegelegt sein oder direkt im yaml-file festgelegt werden:   
ZipFile: |

const lambdaCode = () => {

// Hier ist der Code der Lambda-Funktion

}

### AWS CodePipeline

### Amazon EventBridge

Um Änderungen in Pipeline-Stufen zu überwachen, benötigen Sie eine AWS EventBridge-Ressource vom Typ AWS::Events::Rule.  
<https://docs.aws.amazon.com/de_de/AWSCloudFormation/latest/UserGuide/aws-resource-events-rule.html>

# Erkenntnisse

# Fazit

# Anhang