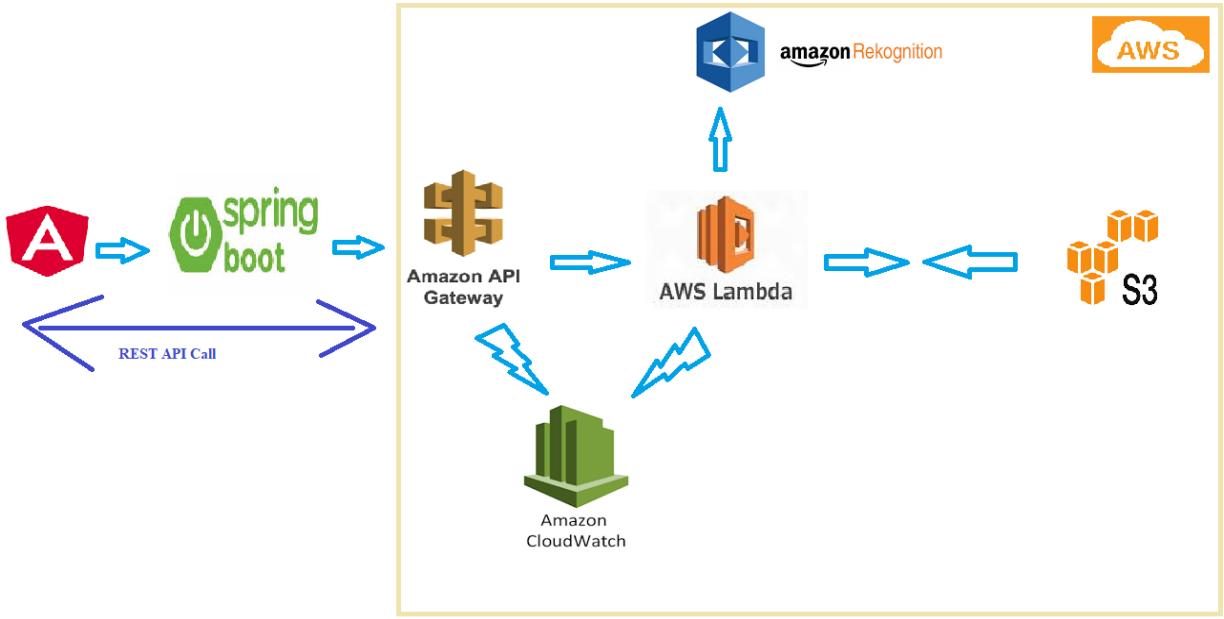
Image Comparison using AWS Rekognition

Objective: Compare each image in a s3 bucket to every other image in the bucket using AWS Rekognition and provide the matching results

Technical Approach



- Languages used: Angular for frontend and Java for backend.
- REST API call is made from Angular front end on [B] click.
- Spring Boot Cloud is used to connect to Lambda via AWS API Gateway. Spring Cloud has the Lambda handler.

```
AWSAPIGLambdaHandler extends
SpringBootRequestHandler<APIGatewayProxyRequestEvent,
APIGatewayProxyResponseEvent>
```

- Override apply method

```
@Override
public APIGatewayProxyResponseEvent apply(APIGatewayProxyRequestEvent input)
```

- Lambda is responsible for establishing s3 client and interacting with S3 to load all images HashMap array list. Each image is stored in ByteBuffer.
- Lambda establishes AWS Rekognition build object and passes two images from HashMap array list at a time for comparison.

- Receive compared result back from compareImages function and store the results in List array.
- Once all the images are compared with one another, the result is set to `APIGatewayProxyResponseEvent` response body.
- Results from backend is displayed on the Angular front end UI.

Source code location:

https://github.com/kiraniitd/Angular-Spring-AWS-APIG-Lambda-S3-Image-Comparision-Rekognition

front-end (Angular)

back-end (Spring Boot Cloud, AWS APIG, Lambda, S3)

To run front end

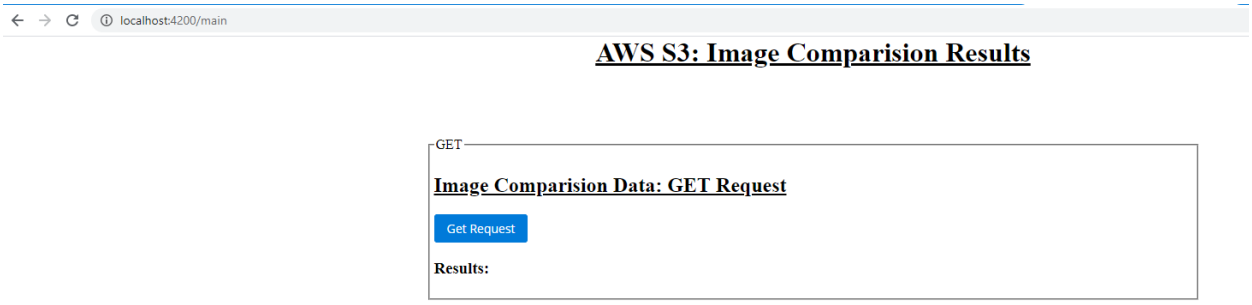
- npm install
- ng serve –open

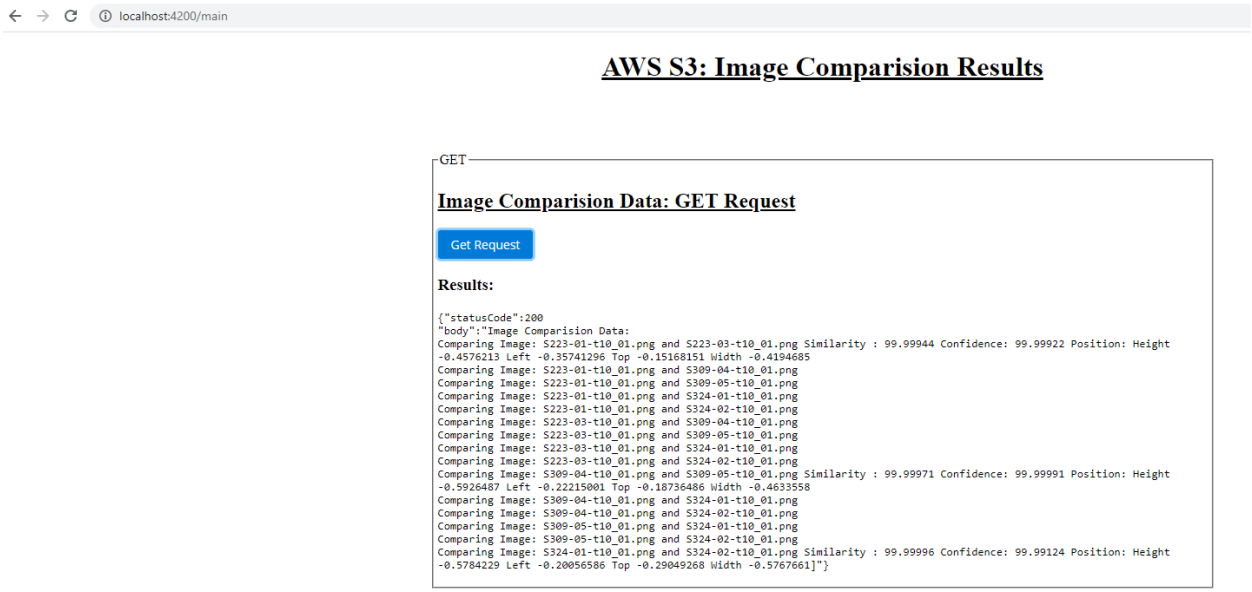
To run back end

- mvn clean install
- upload the jar file to lambda function
- Create AWS API Gateway > create resource > Assign lambda function > Deploy API to stages

To run the app

- http://localhost:4200
- Click on 'Get Request' [B] on the UI





Referred materials

- https://angular.io/
- https://spring.io/
- https://docs.aws.amazon.com/sdk-for-java/index.html
- https://aws.amazon.com/api-gateway/
- https://aws.amazon.com/lambda/
- https://aws.amazon.com/s3/
- https://aws.amazon.com/rekognition/

Hours Spent

Tasks	Angular	Spring Boot	AWS APIG	Lambda	S3	AWS Rekognition	Total Hours
Research	2	2	2	1	1	3	11
Development	3	3	3	4	2	4	19
Testing	1	1	1	1	1	2	7
Integration, Deployment and Documentation	1				1	1	3
Total Hours by Area	**7**	**6**	**6**	**6**	**5**	**10**	**40**