Text Extract Platform

CS 686- Cloud Computing Final Project

By- Akanksha Gupta

Introduction

Motivation

- Always read the nutrition information behind packed food products.
- Idea was to upload image and translate it from one language to another.
- Limited AWS LabRole , access restricted for AWS Translate.

Functionality (www.akanksha-cor.me)

- Login/Signup and verification
- Upload images in multiple languages and extract text
- View uploaded images and extracted text for a user.
- Delete any uploaded image.

Demo

Demo Video Link

Journey

- Architecture diagrams
- Discussions of multiple AWS services used

Hosting domain, CloudFront Distribution & SSL encryption

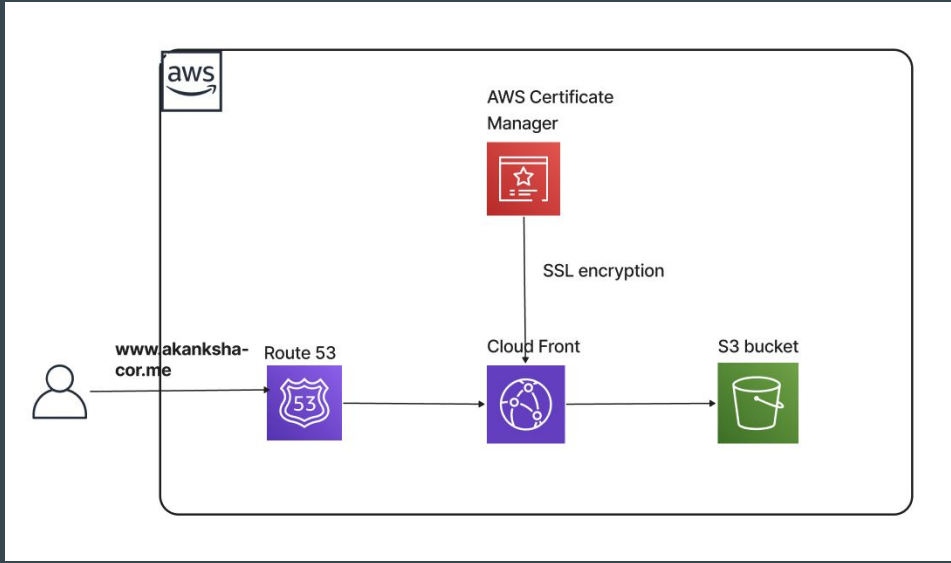
DNS record in Route 53 directs the request to the CloudFront distribution.

CloudFront handles requests over HTTPS.

CloudFront retrieves the HTML/JS/CSS files from S3 bucket.

Content delivered over HTTPS

- Data integrity
- confidentiality



AWS Cognito User Pool

Create an AWS Cognito User Pool

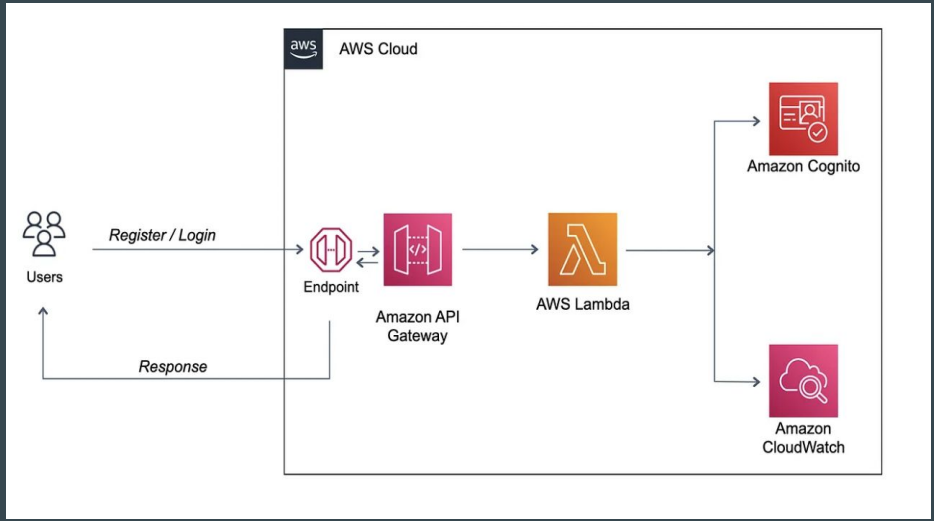
Used email, full name, password as required parameters

Email verification of account on Signup

Create a Lambda function to use AWS SDK for JavaScript

Instance of *CognitoIdentityServiceProvider* created

[User Guide]



MySQL DB instance

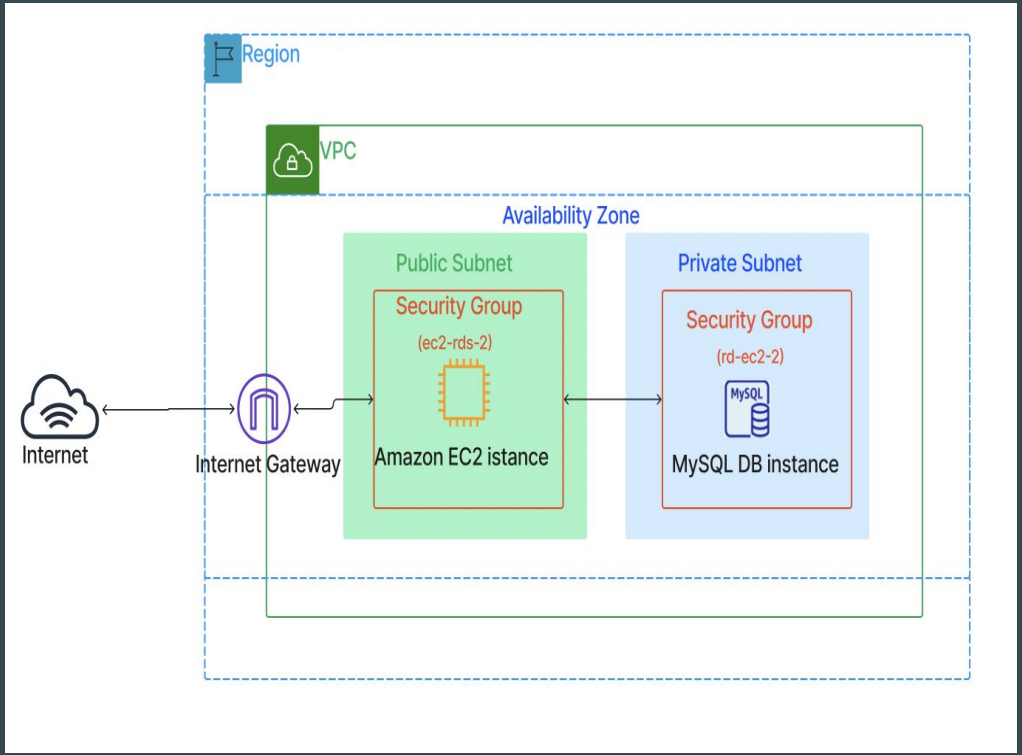
Created a MySQL RDS instance

Created an EC2 instance to connect to the database

Created table 'user_data' to store user data

- Username
- S3 FilePath
- Extracted Text
- TimeStamp

Used *pymysql* library to connect from lambda functions



User Guide

Lambda functions for CRUD operations

Used AWS Lambda for serverless backend tasks

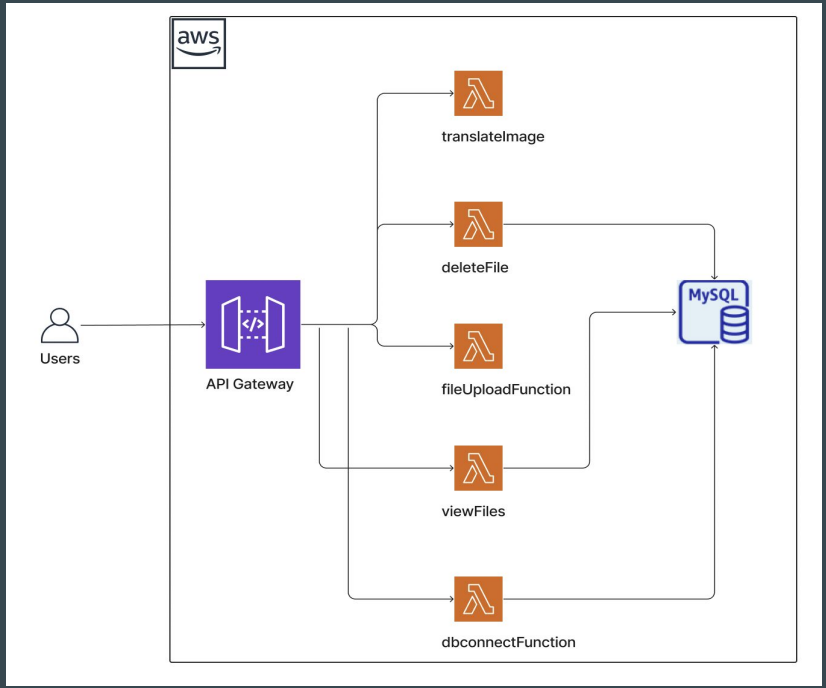
Connected Lambda functions to RDS MySQL DB

S3 bucket: *multilingual-content-platform-bucket*

- Store uploaded images

Extracted text from images using **AWS Textract**

Set up API Gateway for frontend-backend connectivity

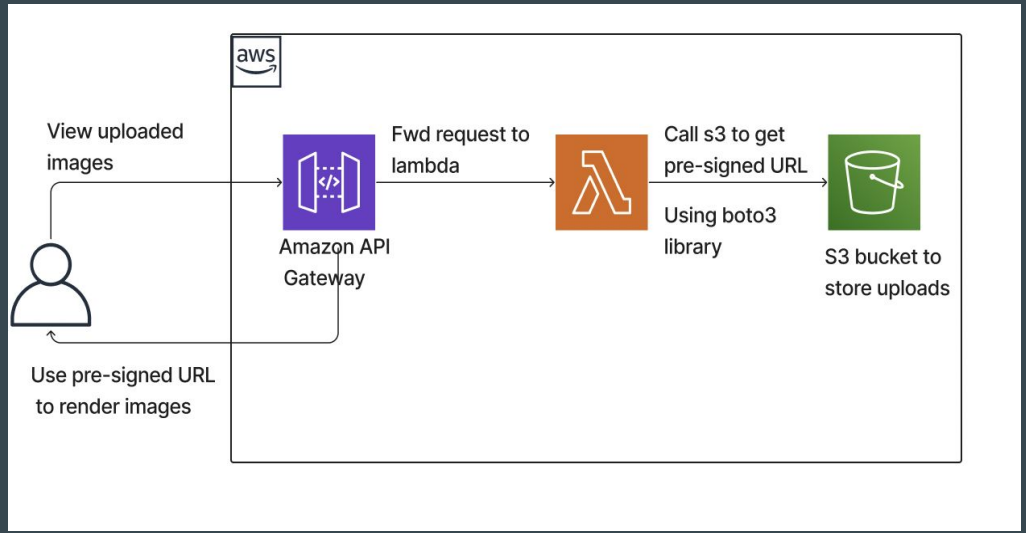


Pre-signed URLs to render uploaded images

Secure image access via S3 pre-signed URLs

Direct image rendering in the user interface with pre-signed URLs

Boto3 python library facilitates AWS service interactions

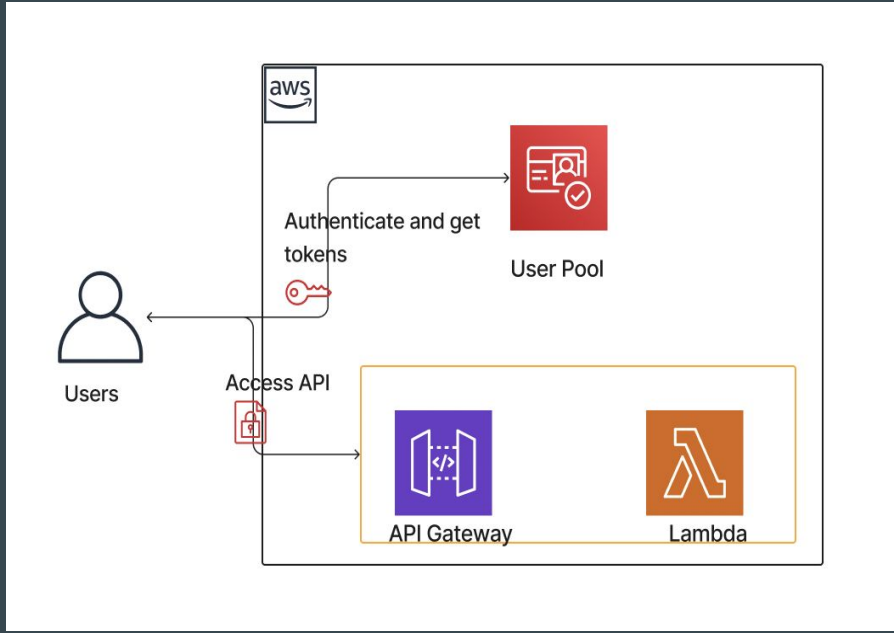


CORS and Authorization for APIs

CORS *access-Control-Allow-Origin* headers set to domain name.

Setting the JWT in the ‘Authorization’ headers of the APIs

Only users logged into website can access the APIs



User Guide

Experience

- Challenges
- Limitation of the web app
- Conclusion

Challenges

- Understanding CORS and making requests from browser.
- Sending images as payload to the request
  - Tried sending image as Multipart/form content-type caused Boundary Exceptions
  - Used Proxy requests, images in binary format
  - Tried generating pre-signed URLs and uploading, couldn't integrate end-to-end
  - Tried defining Mapping Templates for Integration Request
  - Edited API Gateway to add Binary Media Types
- Integration of initiateAuth(params) API using AWS SDK
  - Figuring correct authFlow
  - Actually needed to be enabled in AWS Cognito
- Generating RDS Instance
  - First tried postgreSQL, not able to connect through Lambdas
- Generating Pre-signed URL
  - Expiry Time mismatch error, stackoverflow
- *AccessDenied* exception for AWS Translate API

Limitations

Slow, can handle limited number of users.

AWS Textract efficiency ~70-80%.

Handling quality of images uploaded

No automatic refresh after deletion of images.

Can’t translate images from one language to another.

Build using plain JS, should use React for better responsiveness.

Conclusion

Great experience building serverless platform.

Hands on experience with AWS services.

Solidification of cloud computing concepts.

No backend can sell without CSS.

Thank you!

Appendix

- Screenshots of multiple services created
  - Lambdas
  - AWS Cognito
  - API Gateways
  - CloudFront distribution
  - Cognito User Pool

Cognito

Amazon Cognito > User pools > MyAppUserPool

MyAppUserPool Info

Delete user pool

User pool overview

User pool name
MyAppUserPool

User pool ID
us-east-1_vzytzEd8x

ARN
arn:aws:cognito-idp:us-east-1:931772900028:userpool/us-east-1_vzytzEd8x

Estimated number of users
3

Created time
November 26, 2023 at 12:14 PST

Last updated time
November 26, 2023 at 12:14 PST

Create resource

/
/authenticateUser
OPTIONS
POST

/authenticateUser - POST - Method execution

Update documentation | Delete

ARN
arn:aws:execute-api:us-east-1:931772900028:8qcbbyy5s9/*/POST/authenticateUser

Resource ID
lmrkv2

Client → Method request → Integration request → Lambda integration
← Method response ← Integration response Proxy integration ←

Lambda > Functions > userAuth-Cognito

userAuth-Cognito

Function overview Info

Diagram | Template

userAuth-Cognito
Layers (0)

API Gateway

+ Add trigger

CloudFront Distribution and Route 53

CloudFront > Distributions > E1C2JF1HR6YU1M

E1C2JF1HR6YU1M

View metrics

General | Security | Origins | Behaviors | Error pages | Invalidations | Tags

Details

Distribution domain name
d1kn5p6upzfyme.cloudfront.net

ARN
arn:aws:cloudfront::931772900028:distribution/E1C2JF1HR6YU1M

Last modified
December 4, 2023 at 5:43:36 PM UTC

Settings

Edit

Description
-

Price class
Use only North America and Europe

Supported HTTP versions
HTTP/2, HTTP/1.1, HTTP/1.0

Alternate domain names
www.akanksha-cor.me

Custom SSL certificate
www.akanksha-cor.me

Security policy
TLSv1.2_2021

Standard logging
Off

Cookie logging
Off

Default root object
index.html

Route 53 > Hosted zones > www.akanksha-cor.me

Public **www.akanksha-cor.me** Info

Delete zone | Test record | Configure query logging

Hosted zone details

Edit hosted zone

MySQL Database and EC2 instance

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check
ec2-database-connect.	i-0cc6c4cff19483d4c	Running	t2.micro	2/2 checks passed

Databases (1) Group

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size
myprojectdb	Available	Instance	MySQL Community	us-east-1a	db.t3.micro

```
# RDS settings
db_config = {
    'host': 'myprojec[redacted]zonaws.com',
    'user': '[redacted]',
    'password': '[redacted]',
    'database': 'myprojectdb'
}
```

```
# Connect to the database
try:
    conn = pymysql.connect(**db_config)
except pymysql.MySQLError as e:
    print("ERROR: Unexpected error: Could not connect to MySQL instance.")
    print(e)
    sys.exit()
```

API Gateway methods

API Gateway > APIs > ccprojectapi (rri0szbgy2) > Stages

Stages

Stage actions ▼ | Create stage

- dev
  - /
    - /dbconnect
    - /deleteFile
    - /getURL
    - /translate
    - /upload
    - /viewFiles

Stage details Info

Edit

Stage name	Rate Info	Web ACL
dev	-	-
API cache	Burst Info	Client certificate
Inactive	-	-

Invoke URL
https://rri0szbgy2.execute-api.us-east-1.amazonaws.com/dev

Active deployment
vsowib on December 04, 2023, 14:50 (UTC-08:00)

API Gateway > APIs > ccprojectapi (rri0szbgy2) > Authorizers

Authorizers (1) Info

CognitoTokens

Authorizer ID
wxpzkk

Cognito pool
MyAppUserPool - vzytzEd8x (us-east-1)

Token source
Authorization

Token validation - optional
none

fileUploadFunction3	-	Zip	Python 3.8
MyLambdaFunction	-	Zip	Node.js 14.x
fileUploadFunction	-	Zip	Python 3.8
deleteFile	-	Zip	Python 3.8
translateImage	-	Zip	Python 3.8
dbconnectFunction	-	Zip	Python 3.8

Cognito User Pool

Users (3) Info

Delete user | Create user

View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

Property: User name

Search users by attribute

1

User name	Email address	Email verified	Confirmation status	Status
145cc7f5-f852-4dbf-8c...	agupta61@dons.usfca.edu	Yes	Confirmed	Enabled
18bbc9b7-219e-4888-8...	gst.akanksha@gmail.com	Yes	Confirmed	Enabled
222e1bf6-ffea-4abf-950...	agupta9699@gmail.com	Yes	Confirmed	Enabled

Lambda Functions

viewFiles

Layers (0)

API Gateway

+ Add destination

+ Add trigger

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (⌘ P)

Environment Var × lambda_function ×

Environment

viewFiles - /

pymysql

PyMySQL-1.1.0.dist-info

lambda_function.py

```python
def generate_presigned_url(bucket_name, object_name):
    """Generate a presigned URL for an S3 object."""
    s3_client = boto3.client('s3')
    try:
        print("Reached here")
        presigned_url = s3_client.generate_presigned_url('get_object',
                                                         Params={'Bucket': bucket_name,
                                                                 'Key': object_name},
                                                         ExpiresIn=300)

        return presigned_url
    except NoCredentialsError:
        print("Credentials not available")
        return None

def lambda_handler(event, context):
    # Connect to the database
```

translateImage

Layers (0)

API Gateway

+ Add destination

+ Add trigger

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (⌘ P)

lambda_function × Environment Var ×

Environment

translateImage /

lambda_function.py

```python
import base64
import json

def extract_and_join_words(blocks):
    # Filter out the blocks with BlockType 'WORD' and join their text
    return ' '.join([block['Text'] for block in blocks if block['BlockType'] == 'W

def lambda_handler(event, context):
    try:
        # Decode the file content from base64
        file_content = base64.b64decode(event['body'])
        print("File content decoded successfully.")

        # Initialize the Textract and Translate clients
        textract = boto3.client('textract')
        translate = boto3.client(service_name='translate', region_name='us-east-1'
```