Name: Abishek Vellineni
Email Address: av739@njit.edu
Subject : CS643852-Cloud Computing
GitHub - https://github.com/Abishek183/cloud_project_1

Module 03 Assignment 03: Programming Assignment 1

Introduction

This project is a demonstration of distributed computing on cloud, therefore we will need two instances on the cloud. One for recognizing the car images, and one for getting the text out of the image.

Initial steps for AWS Credentials setup

The credentials created here will be used by our application for connecting to Rekognition service adn SQS.

1. Create an AWS Account with NJIT email.
2. Access the AWS Account and look for IAM there (For setting up the credentials for the application)
Goto "IAM" -> "Access Management" -> "Policies" -> "Create policy"(eg: module 03 Assignment
3). Choose the following services with full access:

- Rekognition
- S3
- SQS

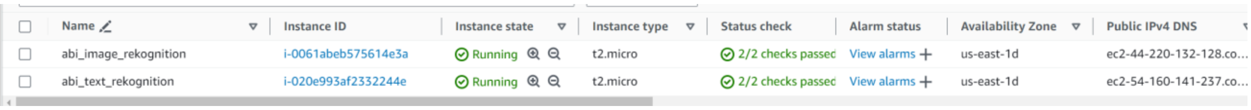
This will allow application to access all the services mentioned above.

Creating EC2 Instances:

Two EC2 instances will be spun up while the cloud **environment** is set up. All of this will be accomplished using the 12-month free tier plan.

1. In the search bar, type "EC2," and then select the first result.
2. Select "Instances" from the navigation menu on the left.
3. Select "Launch Instances." 3.
4. Choose the "Amazon linux 2 AMI - free tier eligible" option. Choose "Next: Configure Security Group" in step 5.
5. When you click "Add Rule," the following will automatically fill in for you:
6. HTTP, HTTPS, SSH
7. Choose "My IP" from the "Source" drop-down for each rule.
8. put up a login "key pair"
9. Create the "Key pair"; after creation, the key file will be downloaded. Save it with caution.

10.To start the instance, click “Launch instance”.

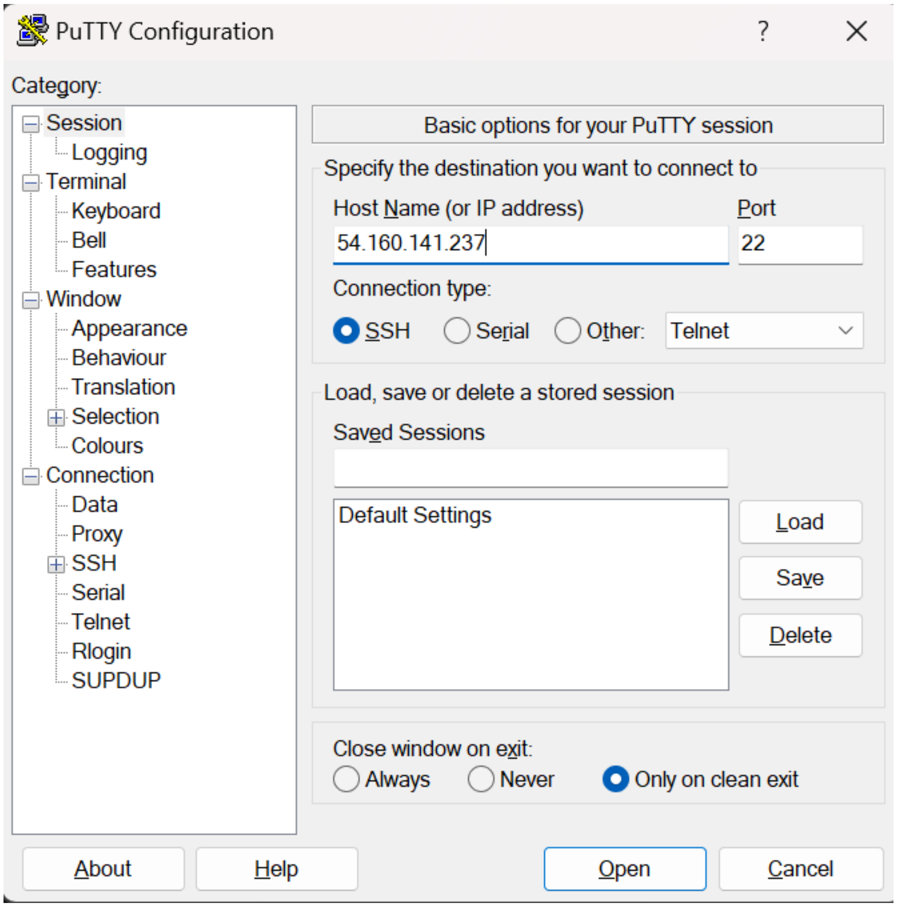


□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	abi_image_rekognition	i-0061abeb575614e3a	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1d	ec2-44-220-132-128.co...
□	abi_text_rekognition	i-020e993af2332244e	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-1d	ec2-54-160-141-237.co...

Connecting to the EC2 instance using ssh:

I have used putty to connect the EC2 instance. Below are the steps to connect:

1) Open Putty and provide the public IP address of the instance.
2) Configure the .PPK file that was assigned to the EC2 instance while creation to the putty.
3) Click on Connect.
4) Enter “ec2-user” when asked for the username.
5) Perform aws Configure each time to validate the session details with aws.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 54.160.141.237

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Once you have connected to the EC2 instance use the below commands to Install JAVA.

```
sudo yum update
sudo yum install java-1.8.0-openjdk -y
sudo amazon-linux-extras install java-openjdk11 -y
```

Setting up Amazon SQS:

1. Navigate to the SQS section in the AWS Management Console.
2. Click on "Create a New Queue."
3. Opt for the "FIFO" queue type.
4. During the configuration process:
5. Provide a distinctive name that ends with ".fifo" for the queue. I have given “av739.fifo” for my queue.
6. Turn on both "Content-Based Deduplication" and "High Throughput" features.
7. Finalize by creating the queue. Once done, you can view it within the SQS dashboard.

Running Application in EC2 instance:

1) Compile the java code to generate the .jar file.
2) Use winscp to transfer the jar file to the EC2 instances.
3) Then run the below commands to perform execution.

First : Run ImageRekognition JAR in EC2_ImageRekognition instance -> java -jar image-rekognition-1.0-SNAPSHOT.jar

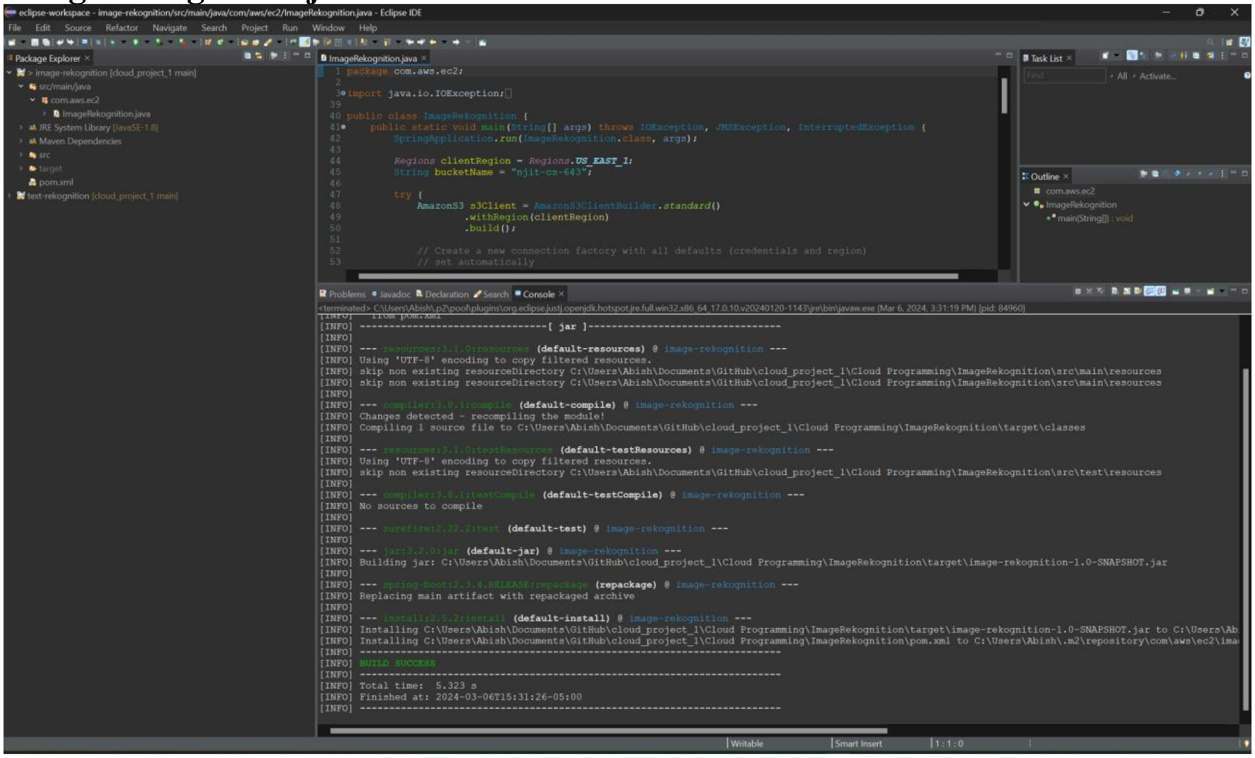
Second : Run TextRekognition JAR in EC2_TextRekognition instance -> java -jar text-rekognition-1.0-SNAPSHOT.jar

Workflow in Below Images:

1) Started by executing abi_Image_Rekognition to process the car images and push the results to the SQS Queue.
2) Next, executed abi_Text_Rekognition to listen to the SQS Queue and process the images for text recognition.
3) abi_Text_Rekognition fetches the processed images and recognizes the text present in the images.
4) Finally, it prints the output of the text recognition process.

Screenshots for building the JAVA application:

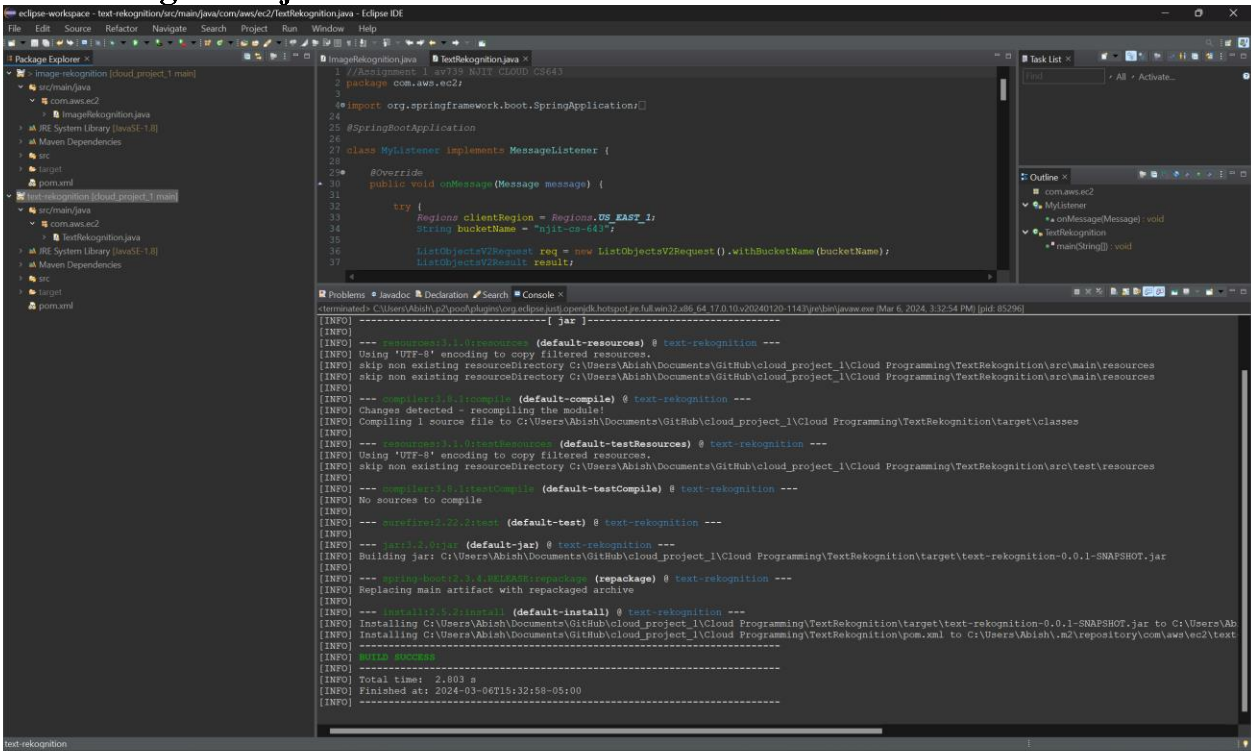
ImageRekognition.java


eclipse-workspace - image-rekognition/src/main/java/com/aws/ec2/ImageRekognition.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
> image-rekognition [cloud_project_1 main]
src/main/java
com.aws.ec2
ImageRekognition.java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
pom.xml
text-rekognition [cloud_project_1 main]
ImageRekognition.java
Task List
Find
All
Activate...
Outline
com.aws.ec2
ImageRekognition
main(String[]) : void
Problems
Javadoc
Declaration
Search
Console
Writable
Smart Insert
1 : 1 : 0

```java
package com.aws.ec2;

import java.io.IOException;

public class ImageRekognition {
    public static void main(String[] args) throws IOException, JMSException, InterruptedException {
        SpringApplication.run(ImageRekognition.class, args);

        Regions clientRegion = Regions.US_EAST_1;
        String bucketName = "njit-cs-643";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .build();

            // Create a new connection factory with all defaults (credentials and region)
            // set automatically
```

```
<terminated> C:\Users\Abish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Mar 6, 2024, 3:31:19 PM) [pid: 84960]
[INFO]    from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.1.0:resources (default-resources) @ image-rekognition ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\ImageRekognition\src\main\resources
[INFO] skip non existing resourceDirectory C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\ImageRekognition\src\main\resources
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ image-rekognition ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\ImageRekognition\target\classes
[INFO]
[INFO] --- resources:3.1.0:testResources (default-testResources) @ image-rekognition ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\ImageRekognition\src\test\resources
[INFO]
[INFO] --- compiler:3.8.1:testCompile (default-testCompile) @ image-rekognition ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:2.22.2:test (default-test) @ image-rekognition ---
[INFO]
[INFO] --- jar:3.2.0:jar (default-jar) @ image-rekognition ---
[INFO] Building jar: C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\ImageRekognition\target\image-rekognition-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot:2.3.4.RELEASE:repackage (repackage) @ image-rekognition ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- install:2.5.2:install (default-install) @ image-rekognition ---
[INFO] Installing C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\ImageRekognition\target\image-rekognition-1.0-SNAPSHOT.jar to C:\Users\Ab
[INFO] Installing C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\ImageRekognition\pom.xml to C:\Users\Abish\.m2\repository\com\aws\ec2\ima
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.323 s
[INFO] Finished at: 2024-03-06T15:31:26-05:00
[INFO] ------------------------------------------------------------------------
```

TextRekognition.java



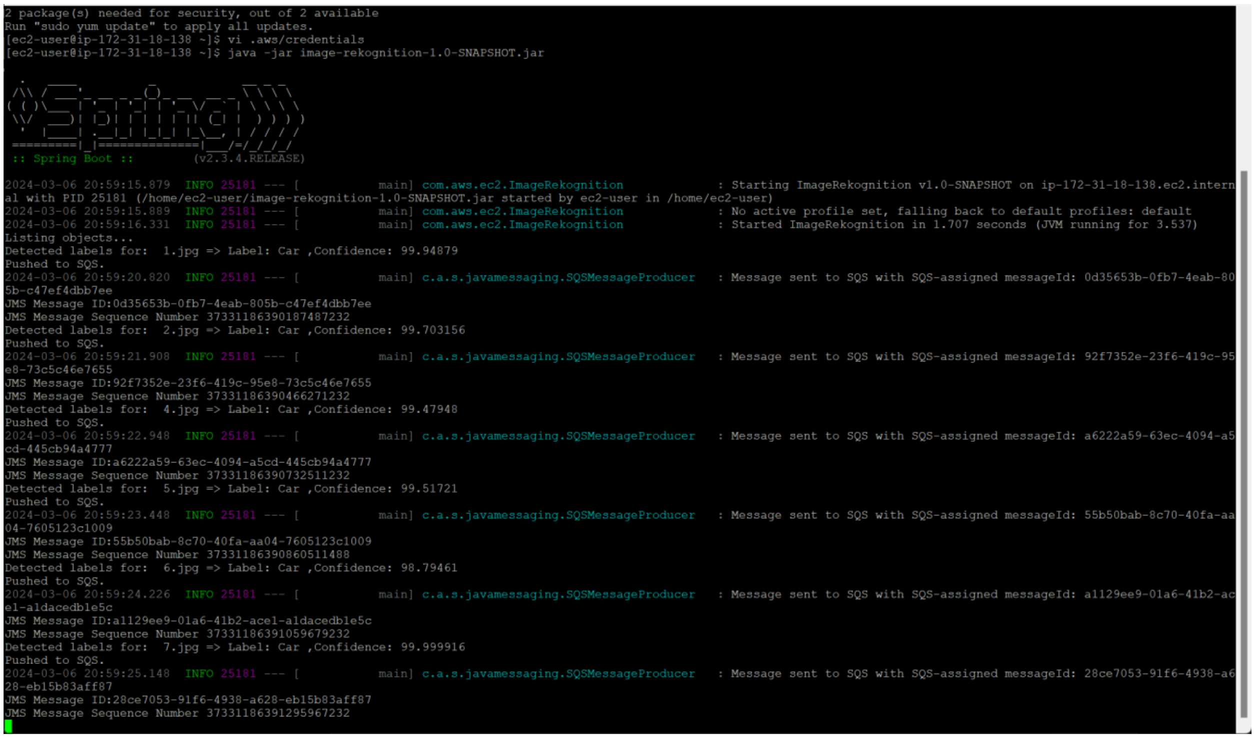
```
//Assignment 1 av739 NJIT CLOUD CS643
package com.aws.ec2;

import org.springframework.boot.SpringApplication;

@SpringBootApplication

class MyListener implements MessageListener {

    @Override
    public void onMessage(Message message) {

        try {
            Regions clientRegion = Regions.US_EAST_1;
            String bucketName = "njit-cs-643";

            ListObjectsV2Request req = new ListObjectsV2Request().withBucketName(bucketName);
            ListObjectsV2Result result;
```

```
<terminated> C:\Users\Abish\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Mar 6, 2024, 3:32:54 PM) [pid: 85296]
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- resources:3.1.0:resources (default-resources) @ text-rekognition ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\TextRekognition\src\main\resources
[INFO] skip non existing resourceDirectory C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\TextRekognition\src\main\resources
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ text-rekognition ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\TextRekognition\target\classes
[INFO]
[INFO] --- resources:3.1.0:testResources (default-testResources) @ text-rekognition ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\TextRekognition\src\test\resources
[INFO]
[INFO] --- compiler:3.8.1:testCompile (default-testCompile) @ text-rekognition ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:2.22.2:test (default-test) @ text-rekognition ---
[INFO]
[INFO] --- jar:3.2.0:jar (default-jar) @ text-rekognition ---
[INFO] Building jar: C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\TextRekognition\target\text-rekognition-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot:2.3.4.RELEASE:repackage (repackage) @ text-rekognition ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- install:2.5.2:install (default-install) @ text-rekognition ---
[INFO] Installing C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\TextRekognition\target\text-rekognition-0.0.1-SNAPSHOT.jar to C:\Users\Ab
[INFO] Installing C:\Users\Abish\Documents\GitHub\cloud_project_1\Cloud Programming\TextRekognition\pom.xml to C:\Users\Abish\.m2\repository\com\aws\ec2\text
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.803 s
[INFO] Finished at: 2024-03-06T15:32:58-05:00
[INFO] ------------------------------------------------------------------------
```

Output on execution of JAR files.

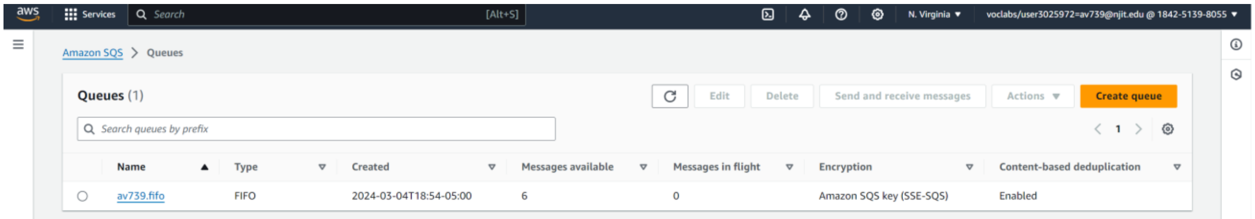
abi_Image_Rekognition:

- This component is responsible for processing images using Amazon Rekognition.
- It starts by fetching car images from a public S3 bucket (https://njit-cs-643.s3.us-east-1.amazonaws.com).
- Once an image is fetched, it performs object recognition on the image using Amazon
Rekognition.
- After processing, it prints the list of objects detected in the image.
- Finally, it pushes the processed image and its results to an SQS Queue.



```
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-18-138 ~]$ vi .aws/credentials
[ec2-user@ip-172-31-18-138 ~]$ java -jar image-rekognition-1.0-SNAPSHOT.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.3.4.RELEASE)

2024-03-06 20:59:15.879  INFO 25181 --- [           main] com.aws.ec2.ImageRekognition             : Starting ImageRekognition v1.0-SNAPSHOT on ip-172-31-18-138.ec2.intern
al with PID 25181 (/home/ec2-user/image-rekognition-1.0-SNAPSHOT.jar started by ec2-user in /home/ec2-user)
2024-03-06 20:59:15.889  INFO 25181 --- [           main] com.aws.ec2.ImageRekognition             : No active profile set, falling back to default profiles: default
2024-03-06 20:59:16.331  INFO 25181 --- [           main] com.aws.ec2.ImageRekognition             : Started ImageRekognition in 1.707 seconds (JVM running for 3.537)
Listing objects...
Detected labels for:  1.jpg => Label: Car ,Confidence: 99.94879
Pushed to SQS.
2024-03-06 20:59:20.820  INFO 25181 --- [           main] c.a.s.javamessaging.SQSMessageProducer   : Message sent to SQS with SQS-assigned messageId: 0d35653b-0fb7-4eab-80
5b-c47ef4dbb7ee
JMS Message ID:0d35653b-0fb7-4eab-805b-c47ef4dbb7ee
JMS Message Sequence Number 37331186390187487232
Detected labels for:  2.jpg => Label: Car ,Confidence: 99.703156
Pushed to SQS.
2024-03-06 20:59:21.908  INFO 25181 --- [           main] c.a.s.javamessaging.SQSMessageProducer   : Message sent to SQS with SQS-assigned messageId: 92f7352e-23f6-419c-95
e8-73c5c46e7655
JMS Message ID:92f7352e-23f6-419c-95e8-73c5c46e7655
JMS Message Sequence Number 37331186390466271232
Detected labels for:  4.jpg => Label: Car ,Confidence: 99.47948
Pushed to SQS.
2024-03-06 20:59:22.948  INFO 25181 --- [           main] c.a.s.javamessaging.SQSMessageProducer   : Message sent to SQS with SQS-assigned messageId: a6222a59-63ec-4094-a5
cd-445cb94a4777
JMS Message ID:a6222a59-63ec-4094-a5cd-445cb94a4777
JMS Message Sequence Number 37331186390732511232
Detected labels for:  5.jpg => Label: Car ,Confidence: 99.51721
Pushed to SQS.
2024-03-06 20:59:23.448  INFO 25181 --- [           main] c.a.s.javamessaging.SQSMessageProducer   : Message sent to SQS with SQS-assigned messageId: 55b50bab-8c70-40fa-aa
04-7605123c1009
JMS Message ID:55b50bab-8c70-40fa-aa04-7605123c1009
JMS Message Sequence Number 37331186390860511488
Detected labels for:  6.jpg => Label: Car ,Confidence: 98.79461
Pushed to SQS.
2024-03-06 20:59:24.226  INFO 25181 --- [           main] c.a.s.javamessaging.SQSMessageProducer   : Message sent to SQS with SQS-assigned messageId: a1129ee9-01a6-41b2-ac
e1-a1dacedb1e5c
JMS Message ID:a1129ee9-01a6-41b2-ace1-a1dacedb1e5c
JMS Message Sequence Number 37331186391059679232
Detected labels for:  7.jpg => Label: Car ,Confidence: 99.999916
Pushed to SQS.
2024-03-06 20:59:25.148  INFO 25181 --- [           main] c.a.s.javamessaging.SQSMessageProducer   : Message sent to SQS with SQS-assigned messageId: 28ce7053-91f6-4938-a6
28-eb15b83aff87
JMS Message ID:28ce7053-91f6-4938-a628-eb15b83aff87
JMS Message Sequence Number 37331186391295967232
```

SQS queue:

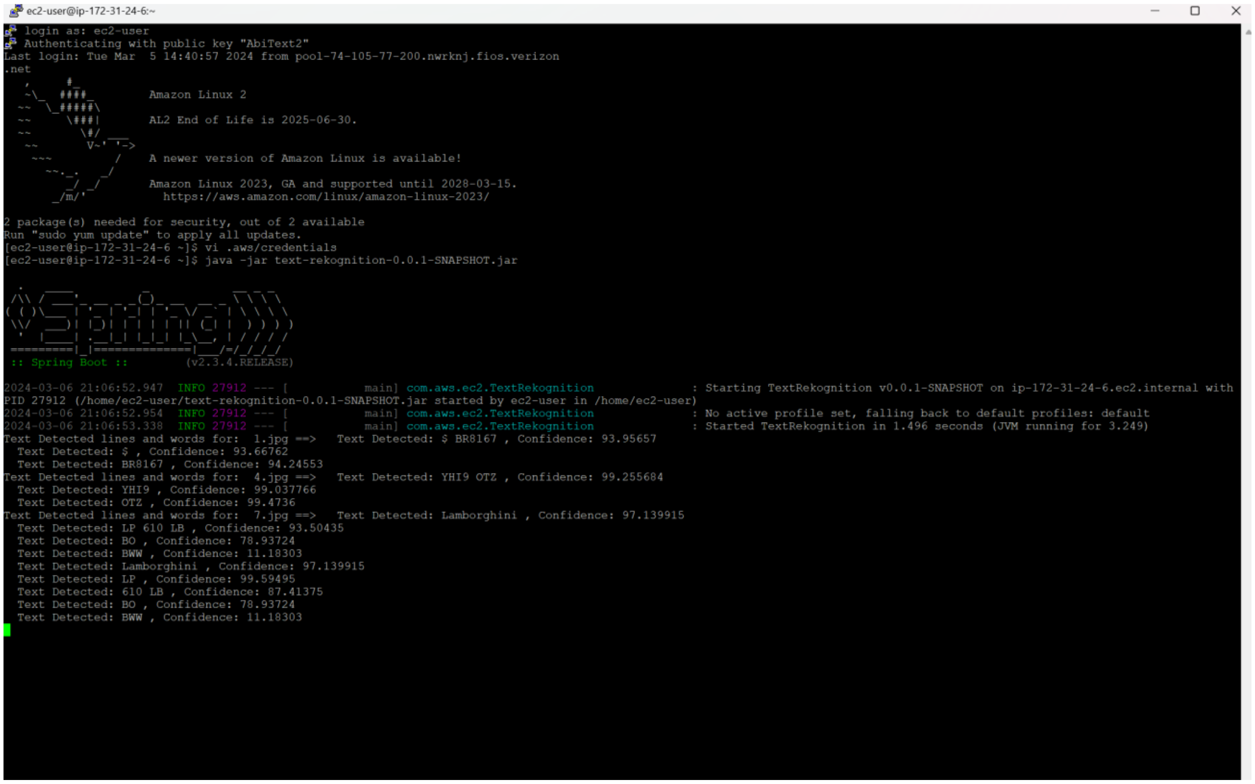


Queues (1)

	Name ▲	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication
○	av739.fifo	FIFO	2024-03-04T18:54-05:00	6	0	Amazon SQS key (SSE-SQS)	Enabled

abi_Text_Rekognition:

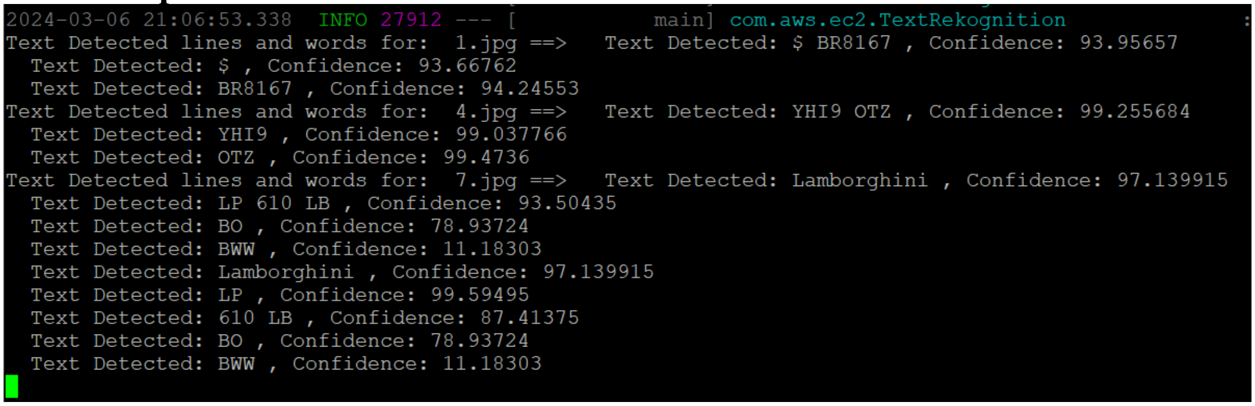
- This component is responsible for processing text from car images.
- It listens to the SQS Queue where the processed images are pushed by abi_Image_Rekognition.
- When an image is received from the queue, it fetches the respective image.
- Then, it performs text recognition on the image using Amazon Rekognition.
- After processing, it recognizes the text present in the image and prints the output.



ec2-user@ip-172-31-24-6:~

```
 login as: ec2-user
 Authenticating with public key "AbiText2"
Last login: Tue Mar  5 14:40:57 2024 from pool-74-105-77-200.nwrknj.fios.verizon
.net
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-24-6 ~]$ vi .aws/credentials
[ec2-user@ip-172-31-24-6 ~]$ java -jar text-rekognition-0.0.1-SNAPSHOT.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.3.4.RELEASE)

2024-03-06 21:06:52.947  INFO 27912 --- [           main] com.aws.ec2.TextRekognition              : Starting TextRekognition v0.0.1-SNAPSHOT on ip-172-31-24-6.ec2.internal with
PID 27912 (/home/ec2-user/text-rekognition-0.0.1-SNAPSHOT.jar started by ec2-user in /home/ec2-user)
2024-03-06 21:06:52.954  INFO 27912 --- [           main] com.aws.ec2.TextRekognition              : No active profile set, falling back to default profiles: default
2024-03-06 21:06:53.338  INFO 27912 --- [           main] com.aws.ec2.TextRekognition              : Started TextRekognition in 1.496 seconds (JVM running for 3.249)
Text Detected lines and words for:  1.jpg ==>   Text Detected: $ BR8167 , Confidence: 93.95657
  Text Detected: $ , Confidence: 93.66762
  Text Detected: BR8167 , Confidence: 94.24553
Text Detected lines and words for:  4.jpg ==>   Text Detected: YHI9 OTZ , Confidence: 99.255684
  Text Detected: YHI9 , Confidence: 99.037766
  Text Detected: OTZ , Confidence: 99.4736
Text Detected lines and words for:  7.jpg ==>   Text Detected: Lamborghini , Confidence: 97.139915
  Text Detected: LP 610 LB , Confidence: 93.50435
  Text Detected: BO , Confidence: 78.93724
  Text Detected: BWW , Confidence: 11.18303
  Text Detected: Lamborghini , Confidence: 97.139915
  Text Detected: LP , Confidence: 99.59495
  Text Detected: 610 LB , Confidence: 87.41375
  Text Detected: BO , Confidence: 78.93724
  Text Detected: BWW , Confidence: 11.18303
```

Final Output:



```
2024-03-06 21:06:53.338  INFO 27912 --- [           main] com.aws.ec2.TextRekognition              :
Text Detected lines and words for:  1.jpg ==>   Text Detected: $ BR8167 , Confidence: 93.95657
  Text Detected: $ , Confidence: 93.66762
  Text Detected: BR8167 , Confidence: 94.24553
Text Detected lines and words for:  4.jpg ==>   Text Detected: YHI9 OTZ , Confidence: 99.255684
  Text Detected: YHI9 , Confidence: 99.037766
  Text Detected: OTZ , Confidence: 99.4736
Text Detected lines and words for:  7.jpg ==>   Text Detected: Lamborghini , Confidence: 97.139915
  Text Detected: LP 610 LB , Confidence: 93.50435
  Text Detected: BO , Confidence: 78.93724
  Text Detected: BWW , Confidence: 11.18303
  Text Detected: Lamborghini , Confidence: 97.139915
  Text Detected: LP , Confidence: 99.59495
  Text Detected: 610 LB , Confidence: 87.41375
  Text Detected: BO , Confidence: 78.93724
  Text Detected: BWW , Confidence: 11.18303
```