**1.** **SSH into each instance**

ssh -i "your-key.pem" ec2-user@<public-ip-of-instance>

**2. Install AWS CLI and JDK:**

sudo yum update -y

sudo yum install java-11-amazon-corretto -y

sudo yum install aws-cli -y

**3. Configure AWS CLI**:

aws configure

**4.Update the System**

Before installing Maven, ensure your system is up-to-date:

sudo yum update -y

**5. Install Maven**

1. **Install Maven**:

sudo yum install maven -y

1. **Verify the Installation**:

mvn -version

3. **You should see output similar to:**

Apache Maven 3.x.x (Red Hat 3.x.x)

Maven home: /usr/share/maven

Java version: 11.0.x, vendor: Amazon.com Inc.

1. **Open the .bash\_profile**:

vi ~/.bash\_profile

1. **Add the following lines**:

export PATH=$PATH:/usr/share/maven/bin

1. **Reload the profile**:

source ~/.bash\_profile

1. **Verify Maven Installation Again**:

mvn -version

**Step 5: Set Up the Java Project**

1. **SSH into EC2 A** and create a Maven project:

mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=aws-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

cd aws-app

1. **Modify the pom.xml** to add AWS dependencies:

nano pom.xml

**Pom.xml**

<dependencies>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-bom</artifactId>

<version>1.12.539</version>

<type>pom</type>

<scope>import</scope>

</dependency>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-s3</artifactId>

</dependency>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-sqs</artifactId>

</dependency>

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-java-sdk-rekognition</artifactId>

</dependency>

</dependencies>

Here are steps you can take to resolve this issue:

**1. Check AWS Credentials Configuration**

Ensure your AWS credentials are correctly configured in the `~/.aws/credentials` file. The file should look like this:

[default]

aws\_access\_key\_id = YOUR\_ACCESS\_KEY\_ID

aws\_secret\_access\_key = YOUR\_SECRET\_ACCESS\_KEY

```

Make sure that:

- The access key ID and secret access key are correct.

- There are no extra spaces or special characters that could cause issues.

2. Check AWS Config File

Similarly, verify the `~/.aws/config` file:

[default]

region = us-east-1

output = json

```

Ensure that there are no invalid entries or formatting issues.

3. Reconfigure AWS CLI

If the configuration files seem fine but you're still encountering issues, try reconfiguring your AWS CLI:

```bash

aws configure

```

This command will prompt you to enter your access key, secret key, region, and output format again.

**Step 6: Write the Java Code**

1. **Create the App.java file**:

vi src/main/java/com/mycompany/app/App.java

**Code A Car detection:**

package com.mycompany.app;

import com.amazonaws.services.rekognition.AmazonRekognition;

import com.amazonaws.services.rekognition.AmazonRekognitionClientBuilder;

import com.amazonaws.services.rekognition.model.\*;

import com.amazonaws.services.s3.AmazonS3;

import com.amazonaws.services.s3.AmazonS3ClientBuilder;

import com.amazonaws.services.sqs.AmazonSQS;

import com.amazonaws.services.sqs.AmazonSQSClientBuilder;

import com.amazonaws.services.sqs.model.SendMessageRequest;

import java.util.List;

public class App {

private static final String S3\_BUCKET = "your-s3-bucket-name"; // Replace with your bucket name

private static final String SQS\_QUEUE\_URL = "https://sqs.us-east-1.amazonaws.com/065870645303/CarImageQueue"; // Replace with your SQS Queue URL

public static void main(String[] args) {

AmazonS3 s3Client = AmazonS3ClientBuilder.standard().withRegion("us-east-1").build();

AmazonRekognition rekognitionClient = AmazonRekognitionClientBuilder.standard().withRegion("us-east-1").build();

AmazonSQS sqsClient = AmazonSQSClientBuilder.standard().withRegion("us-east-1").build();

try {

String[] images = {"1.jpg", "2.jpg", "3.jpg", "4.jpg", "5.jpg", "6.jpg", "7.jpg", "8.jpg", "9.jpg", "10.jpg"};

for (String image : images) {

detectCarAndSendMessage(rekognitionClient, sqsClient, S3\_BUCKET, image);

}

} finally {

s3Client.shutdown();

rekognitionClient.shutdown();

sqsClient.shutdown();

}

}

private static void detectCarAndSendMessage(AmazonRekognition rekognition, AmazonSQS sqsClient, String bucket, String key) {

DetectLabelsRequest request = new DetectLabelsRequest()

.withImage(new Image().withS3Object(new S3Object().withBucket(bucket).withName(key)))

.withMaxLabels(10)

.withMinConfidence(90F);

DetectLabelsResult result = rekognition.detectLabels(request);

List<Label> labels = result.getLabels();

boolean carDetected = false;

for (Label label : labels) {

if (label.getName().equalsIgnoreCase("Car")) {

System.out.printf("Car detected in: %s with confidence: %.2f%%\n", key, label.getConfidence());

sendMessage(sqsClient, key);

carDetected = true;

break;

}

}

if (!carDetected) {

System.out.printf("No car detected in: %s\n", key);

}

}

private static void sendMessage(AmazonSQS sqsClient, String imageName) {

SendMessageRequest sendMsgRequest = new SendMessageRequest()

.withQueueUrl(SQS\_QUEUE\_URL)

.withMessageBody(imageName);

sqsClient.sendMessage(sendMsgRequest);

}

}

**Build and Run the Project**

1. **Compile and Build**:

mvn clean install

1. **Run the Application**:

mvn exec:java -Dexec.mainClass="com.mycompany.app.App"

**Instance B code:**

package com.mycompany.app;

import com.amazonaws.services.sqs.AmazonSQS;

import com.amazonaws.services.sqs.AmazonSQSClientBuilder;

import com.amazonaws.services.sqs.model.Message;

import com.amazonaws.services.sqs.model.ReceiveMessageRequest;

import com.amazonaws.services.s3.AmazonS3;

import com.amazonaws.services.s3.AmazonS3ClientBuilder;

import com.amazonaws.services.rekognition.AmazonRekognition;

import com.amazonaws.services.rekognition.AmazonRekognitionClientBuilder;

import com.amazonaws.services.rekognition.model.\*;

import java.io.FileWriter;

import java.io.IOException;

import java.util.HashSet;

import java.util.List;

public class InstanceB {

public static void main(String[] args) {

// Initialize AWS clients

AmazonRekognition rekognitionClient = AmazonRekognitionClientBuilder.standard()

.withRegion("us-east-1").build();

AmazonS3 s3Client = AmazonS3ClientBuilder.standard()

.withRegion("us-east-1").build();

AmazonSQS sqsClient = AmazonSQSClientBuilder.standard()

.withRegion("us-east-1").build();

String bucketName = "njit-cs-643";

String queueUrl = "https://sqs.us-east-1.amazonaws.com/065870645303/CarImageQueue"; // Your SQS Queue URL

// Use a Set to track processed images and avoid duplicates

HashSet<String> processedImages = new HashSet<>();

try (FileWriter writer = new FileWriter("/home/ec2-user/output1.txt", true)) {

boolean done = false;

while (!done) {

ReceiveMessageRequest receiveRequest = new ReceiveMessageRequest(queueUrl)

.withMaxNumberOfMessages(10) // Get up to 10 messages at once

.withWaitTimeSeconds(10); // Long polling

List<Message> messages = sqsClient.receiveMessage(receiveRequest).getMessages();

if (messages.isEmpty()) {

done = true; // Exit if there are no more messages

} else {

for (Message message : messages) {

String imageKey = message.getBody();

if (imageKey.equals("-1")) {

done = true; // Exit the loop if no more messages

break;

}

// Check if the image has already been processed

if (processedImages.add(imageKey)) {

String detectedText = detectText(rekognitionClient, bucketName, imageKey, writer);

writer.write(imageKey + ": " + detectedText + "\n");

System.out.println("Processed text for: " + imageKey);

}

// Delete the message from SQS after processing

sqsClient.deleteMessage(queueUrl, message.getReceiptHandle());

}

}

}

} catch (IOException e) {

e.printStackTrace();

} finally {

// Properly shut down clients to prevent lingering threads

sqsClient.shutdown();

s3Client.shutdown();

rekognitionClient.shutdown();

}

}

public static String detectText(AmazonRekognition rekognitionClient, String bucketName, String imageKey, FileWriter writer) throws IOException {

// Create an S3Object for the image

S3Object s3Object = new S3Object().withBucket(bucketName).withName(imageKey);

DetectTextRequest request = new DetectTextRequest()

.withImage(new Image().withS3Object(s3Object));

DetectTextResult result = rekognitionClient.detectText(request);

StringBuilder detectedText = new StringBuilder();

// Check if any text detections were found

if (result.getTextDetections() != null && !result.getTextDetections().isEmpty()) {

for (TextDetection textDetection : result.getTextDetections()) {

detectedText.append(textDetection.getDetectedText()).append(" ");

String outputLine = "Detected text: " + textDetection.getDetectedText() +

" with confidence: " + textDetection.getConfidence().toString() + "%";

System.out.println(outputLine);

writer.write(outputLine + "\n"); // Write to the output file

}

} else {

String noTextDetected = "No text detected in " + imageKey;

System.out.println(noTextDetected);

writer.write(noTextDetected + "\n"); // Write to the output file

}

return detectedText.toString().trim();

}

}