Using AWS Rekognition with Java in Eclipse

AWS Rekognition is a service that allows you to analyse images and videos to detect objects, text, scenes, and faces. In this task, you'll learn how to integrate **AWS Rekognition** into a Java application using the **Eclipse IDE**.

Step 1: Setting Up Your Java Project in Eclipse

1. **Create a new Maven project in Eclipse**:
   - Open Eclipse → File → New → Maven Project.
   - Select a workspace and choose the **skip archetype** for a simple Java project (e.g., maven-archetype-quickstart).
   - Click **Finish**.
2. **Add AWS SDK for Rekognition to your pom.xml**: Open your pom.xml file and add the following dependency inside the <dependencies> tag:

   ```
   <dependency>
     <groupId>com.amazonaws</groupId>
     <artifactId>aws-java-sdk-rekognition</artifactId>
     <version>1.12.114</version>
   </dependency>
   ```

3. **Download dependencies**: Once you've added this, **Maven** will automatically download the necessary dependencies for AWS Rekognition SDK.

Step 2: AWS Credentials and Utils Class

AWS SDK for Java requires access to your **AWS credentials**. The easiest way to manage credentials locally is by setting them up with the **AWS CLI** or using environment variables. **But for this lab we will add the credentials to our code base**.

2.1 Create a new java class called "**AWSRekognitionUtils**" and set it up with your AWS access credentials (**lecture slide 25**).

Step 3: Implementing AWS Rekognition methods.

1. We are going to be loading images in most of the methods that we will implement, so start of by adding the following method in the AWSRekognitionUtils class:

```
// Utility method to load an image from file into ByteBuffer
public static ByteBuffer loadImage(String imagePath) throws IOException {
    File imageFile = new File(imagePath);
    byte[] imageBytes = Files.readAllBytes(imageFile.toPath());
    return ByteBuffer.wrap(imageBytes);
}
```

2. In the AWSRekognitionUtils, implement a **detect labels** method (**lecture slide 26**).

3. Create a new class with a main method called “AWSRekognitionController” and test the detect label method (**lecture slide 26**).

4. In the AWSRekognitionUtils implement a **detect faces** method and test it in the AWSRekognitionController (**lecture slide 27**). Use an image of your choice or the test images on Moodle.

5. In the AWSRekognitionUtils implement a **detect and annotate faces** method and test it in the AWSRekognitionController (**lecture slide 29**). Use an image of your choice or the test images on Moodle

6. In the AWSRekognitionUtils implement a **detect labels and annotate** method and test it in the AWSRekognitionController (**lecture slide 31**). Use an image of your choice or the test images on Moodle.

7. In the AWSRekognitionUtils implement a **detect text** method and test it in the AWSRekognitionController (**lecture slide 33**). Use an image of your choice or the test images on Moodle.