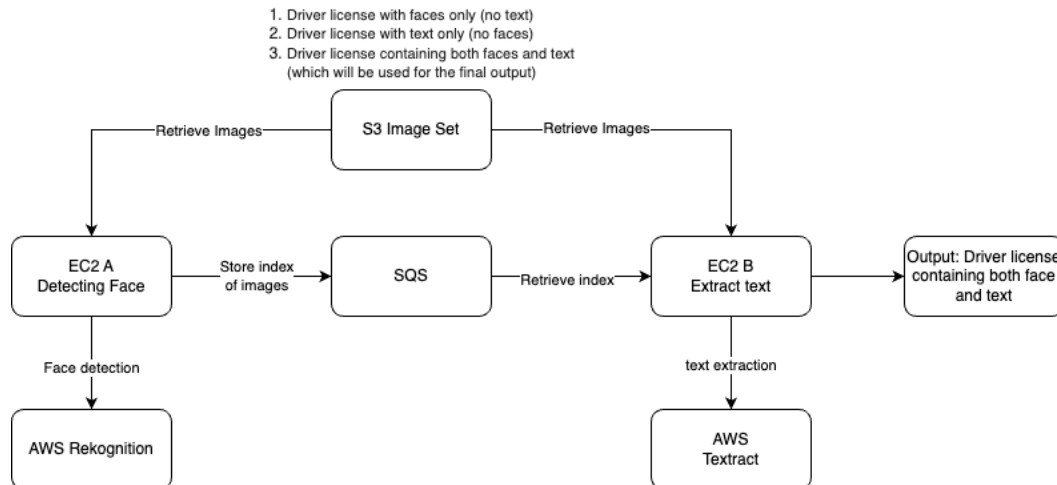


CS 643, Cloud Computing - Programming Assignment 1

Goal: The purpose of this individual assignment is to learn how to use the Amazon AWS cloud platform and how to develop an AWS application that uses existing cloud services. Specifically, you will learn: (1) how to create VMs (EC2 instances) in the cloud; (2) how to use cloud storage (S3) in your applications; (3) how to communicate between VMs using a queue service (SQS); (4) how to program distributed applications in Java on Linux VMs in the cloud; and (5) how to use a machine learning service (AWS Rekognition, AWS Textract) in the cloud. You are allowed to use [ChatGPT](#) or other AI programming copilots for help with programming, but if you do, you need to report how exactly you used these tools and how useful they were.

Description: You have to build an image recognition pipeline in AWS, using two EC2 instances, S3, SQS, Rekognition, and Textract. The assignment must be done in Java on Amazon Linux VMs. For the rest of the description, you should refer to the figure below:

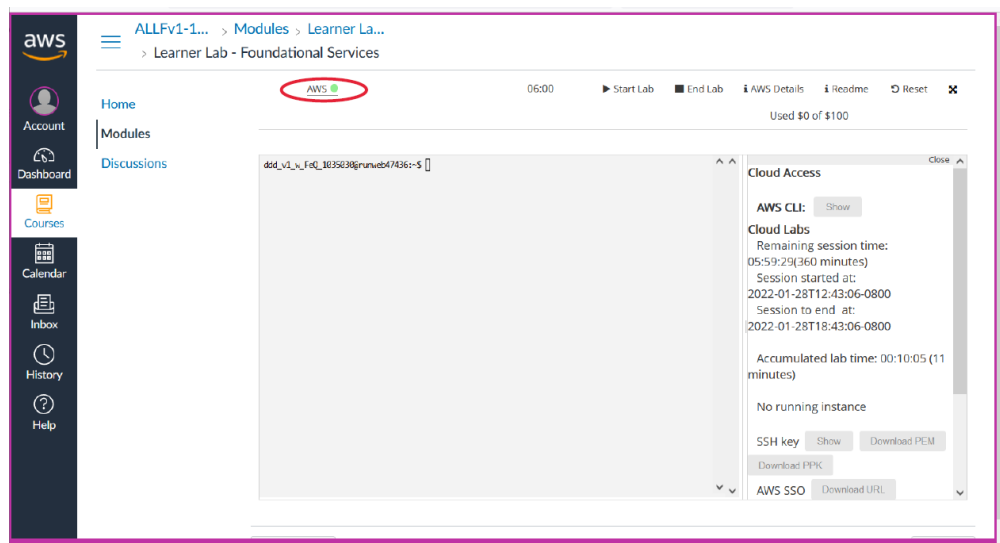


You have to create two EC2 instances (EC2 A and EC2 B in the figure) with Amazon Linux AMI, running in parallel, each executing a Java application. Instance A will read 10 images from an S3 bucket that we created (public bucket name: cs643-sp25-project1) and perform face detection using AWS Rekognition. If a face is detected with a confidence higher than 75%, the index of that image (e.g., 2.jpg) is stored in SQS. Meanwhile, Instance B will read image indexes from SQS as soon as they become available, retrieve the corresponding images from S3, and use AWS Textract for text recognition. The instance records both the image index and the extracted text. The dataset includes images containing only faces, images containing only text, and images containing both faces and text. These two instances work in parallel; for example, while Instance A is processing image_3, Instance B may be processing image_1, which was previously identified as containing a face by Instance A. When Instance A completes its processing, it adds an index of -1 to SQS to signal to Instance B that no more images will be processed, and Instance B stops processing when it detects -1 in the queue. The final output is written to output.txt in the EBS volume associated with Instance B, listing only the indexes of images that contain both a face and text, along with the extracted

text. This implementation ensures that the system functions correctly regardless of which instance starts first.

AWS Learner Lab for Project

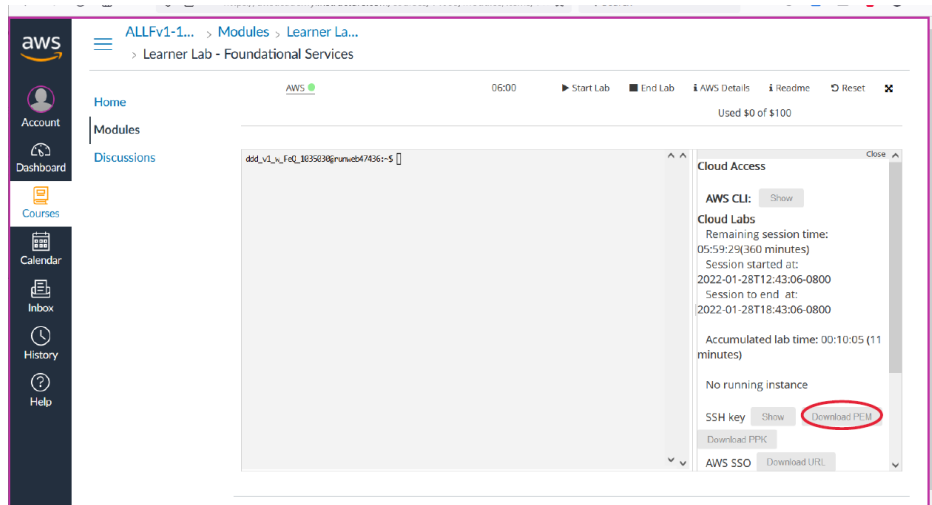
To work on this project, you will need to log into your AWS Learner Lab course: <https://awsacademy.instructure.com/courses/93346>). To start working, you need to go to Modules -> Learner Lab -> Learner Lab -> Start Lab (in the top menu). You can connect to the AWS Management Console by choosing the AWS link above the terminal window (as shown below).



Additional information:

1. EC2 Management

To get the keys for the EC2 instances, click on AWS Details (top menu, after you start the lab). You should use the same .pem key for both instances to ssh into them (see the ssh command in lecture 2). You can download the .pem key as shown below (.pem key is for Linux/Mac; download .ppk key for use with PuTTY in Windows):



- When launching instances, you should select the key that already exists in the EC2 dashboard->Key pairs (vockey).
- You must configure the Security Group well to prevent any attacks. In the Security Group tab, there is a column called "Source" which tells from which IP address this instance can be accessed: you should select "MYIP" from the drop box. You should open just three ports: SSH, HTTP, HTTPS.
- Be sure to terminate your instances after finishing your jobs. Otherwise, you will be charged for every hour of running your instances up to the session limit associated with AWS Learner Labs.

2. Programmer's keys

To code with AWS SDKs, you need a pair of keys (access-id, secret-key) which will be created uniquely for your IAM account. You should follow these instructions to work with these keys: <https://docs.aws.amazon.com/rekognition/latest/dg/setup-awscli-sdk.html>. Please be aware that you shall skip the step of creating IAM user, because the learner lab accounts already have different IAM users. You can access the AWS credentials, SSH key, etc. by clicking "AWS detail". These keys work for all AWS programming services which you intend to connect through AWS SDKs. Therefore, the same keys can be used for Rekognition, SQS, and S3.

3. Temporary credentials in Amazon EC2 instances

Please beware that EC2 instances created in the AWS Learner Lab require temporary credentials to access the S3 bucket from your Java program: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp_use-resources.html. Please pay attention to "Using temporary credentials in Amazon EC2 instances"->"The recommended way to get credentials is to use roles for Amazon EC2..." Alternatively, you can explicitly get the temporary security credentials as well. For AWS Learner Lab, you can go to AWS Details -> "AWS CLI: Copy and paste the following into ~/.aws/credentials", the temporary credentials "aws_session_token" is also there, and you can include it in your EC2 ~/.aws/credentials.

Submission: You will submit in Canvas (<https://njit.instructure.com/courses/46948>), under Programming Assignment 1, the following files:

- The Java code of your two applications, for car recognition and for text recognition (which includes all the communication code with the other services).
- An AI-copilot file in which you describe how you used AI-copilot(s) (if you didn't use one, just say so in the file) and how useful they were.
- A README file that contains:
 - o A step-by-step description of how to set-up the cloud environment and run the application.
 - o A public youtube link that shows a demo of you application (compile & run)
 - o If you used an AI-copilot, you should describe how you used and how useful it was.

Grading:

- Code for face recognition – 30 points
- Code for text extraction – 30 points
- Code for communication with S3, SQS, Rekognition and Textract – 40 points
- Extra-credit – 20 points (best assignment to be anonymized & posted)