

PROGRAMMING ASSIGNMENT 1

CS 643851: Cloud Computing

Name : Deva Rohinth Alagu Arumugam

NJIT ID : 31623674

UCID : da585

Email : da585@njit.edu

Phone : +1 862-279-4190

AWS Image and Text Recognition Pipeline

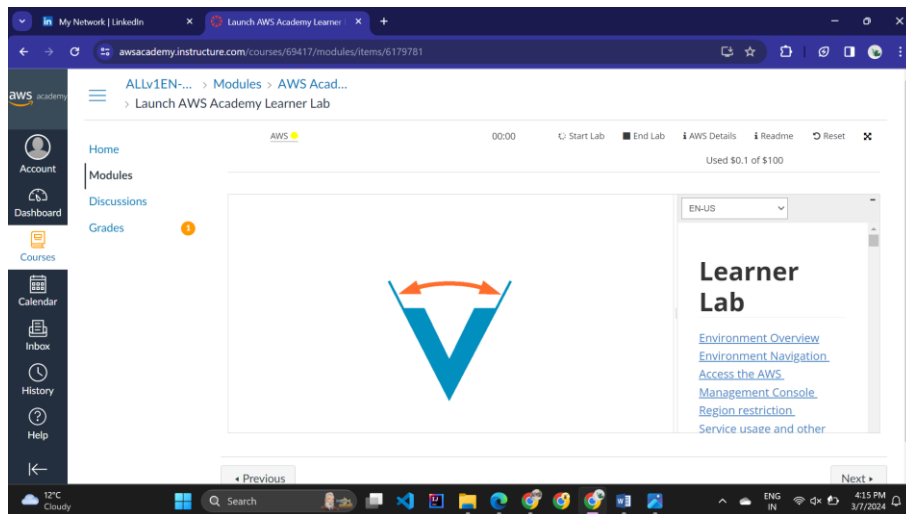
Introduction:

This project demonstrates cloud-based distributed computing, two cloud instances are required. One for text extraction from the picture, and another for identifying the automobile photos. Amazon Web Services (AWS) services, such as S3, SQS, and Recognition, are used by the pipeline.

Initial steps for AWS Credentials setup:

Our application will connect to SQS and the Recognition Service using the credentials you establish here.

1. Navigate to the AWS Academy course and sign up using your NJIT email.
2. If you don't have an Amazon Web Services (AWS) account, create one using the links provided in the Invitation.



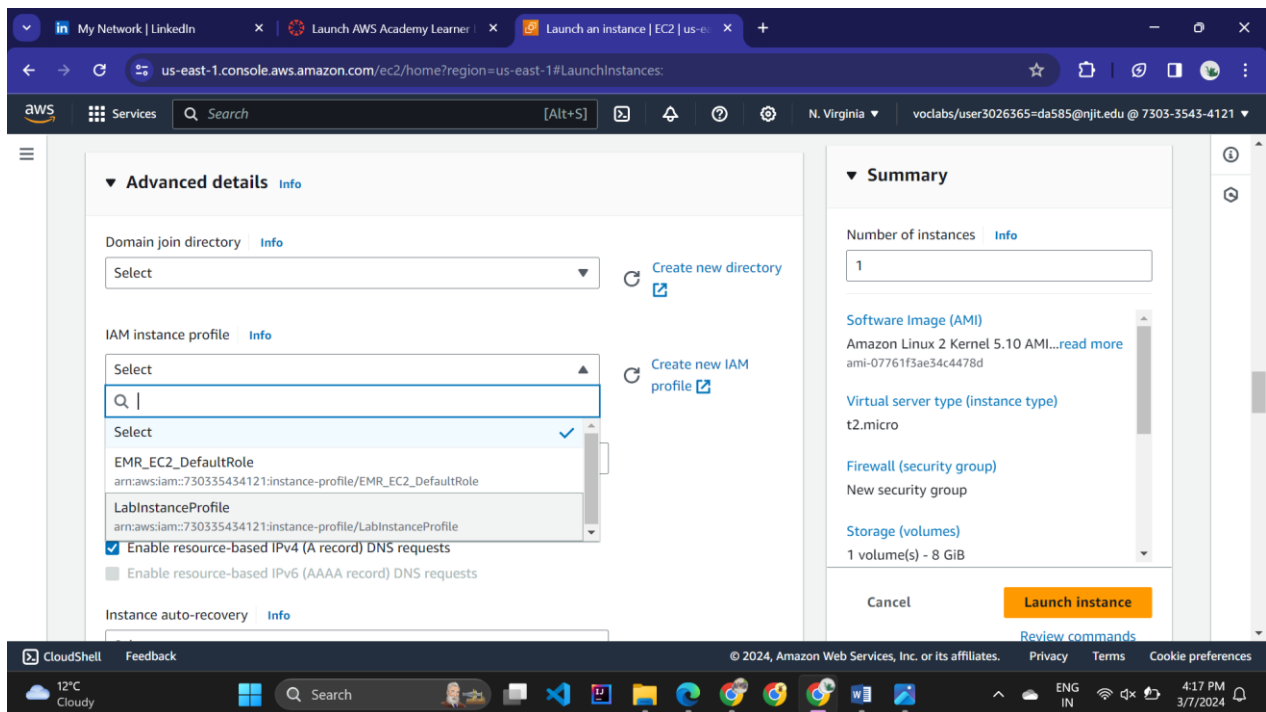
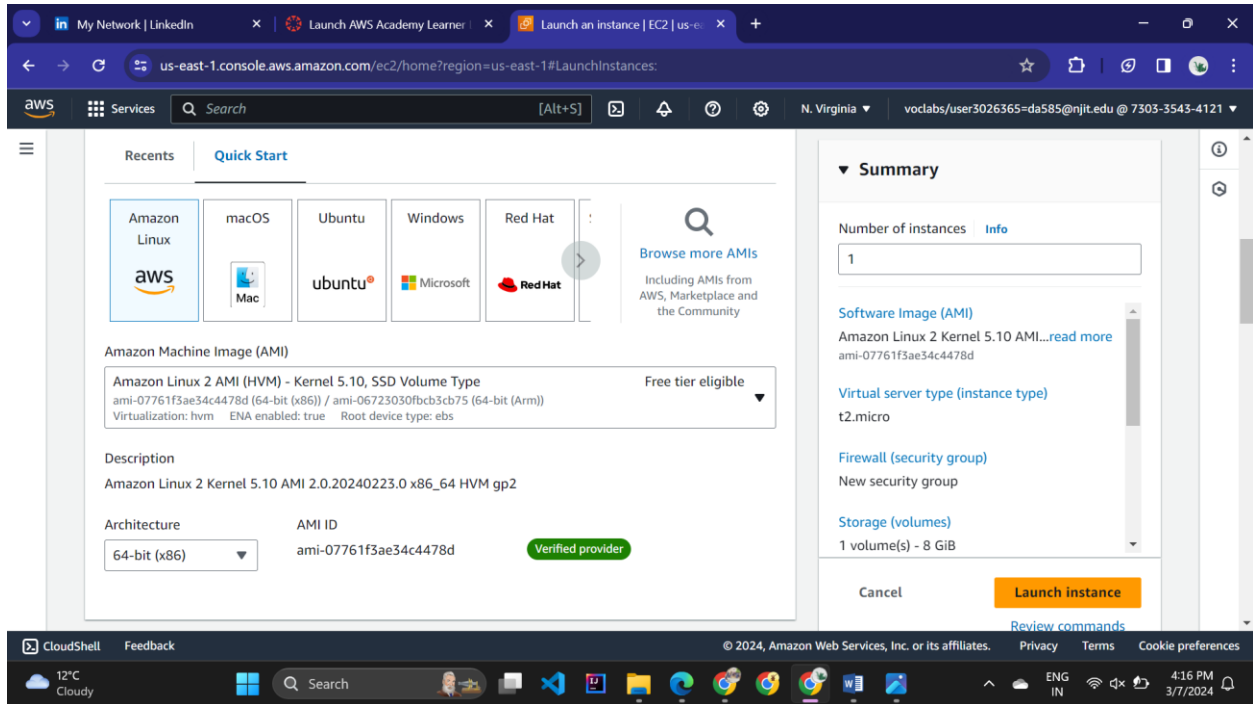
Required:

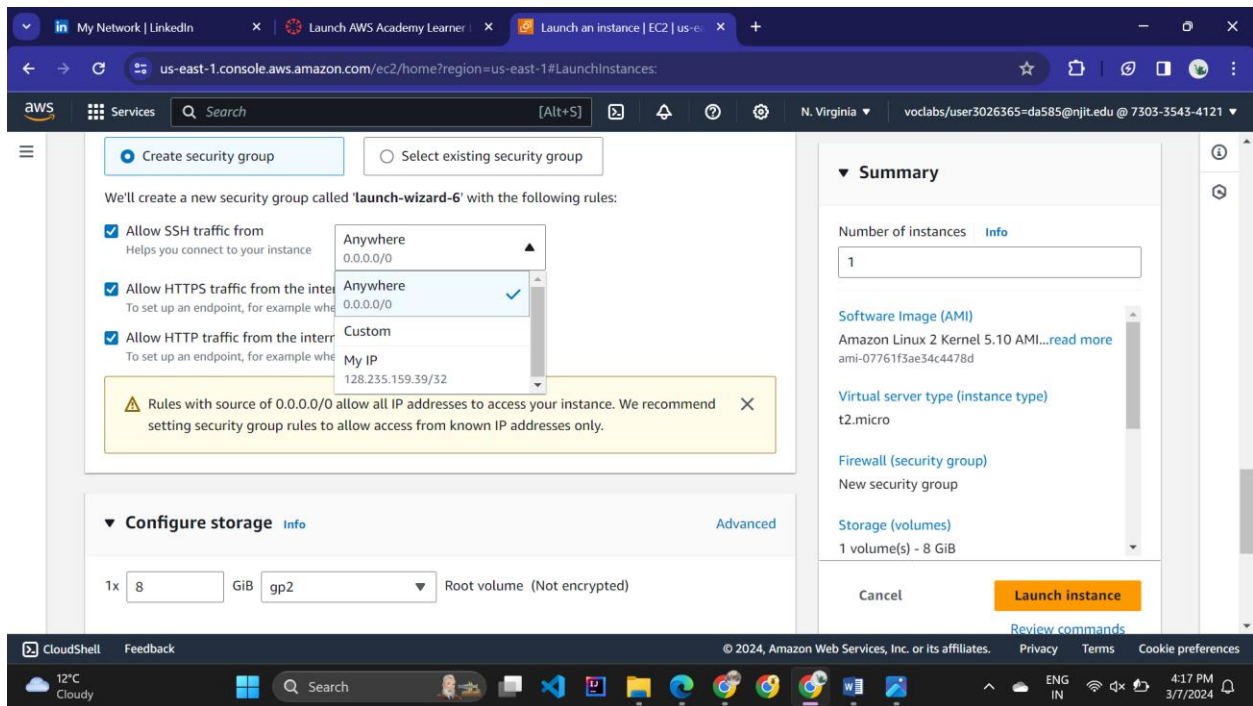
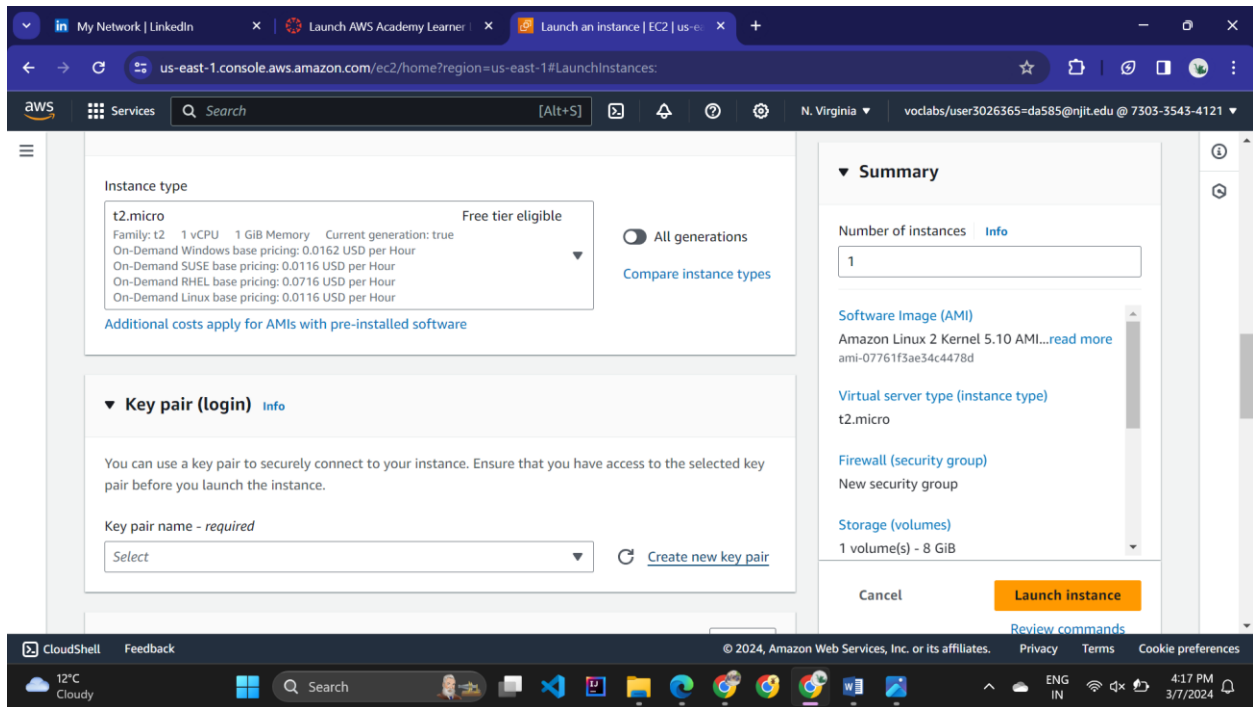
1. Wincp
2. Putty
3. Maven
5. Java

Steps to spin up EC2 instances:

1. Click on "Launch instance."
2. Under "AMI," select "Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type." and choose the "t2.micro" instance type.
3. Select "create new key-pair" as the Key-Pair, and type key name and select ".ppk" then create key.
4. In the "Network Settings" section, create a security group and configure it to allow the following:
 - a. SSH traffic from your IP address.
 - b. HTTPS and HTTP traffic from the internet.
5. Set to "My IP" to restrict traffic to your anywhere.

6. Follow these steps to create two instances. You can name your instances on your own, for example, I have did like “Ec1” and “Ec2”.





Accessing Instance: Once the instance is launched, you can access it using SSH. Use the public IP address or public DNS provided by AWS to connect to your instance.

- Use Putty or WinSCP from windows.
- Select SFTP which is a file transfer protocol to upload to EC2 via cyberduck.
- Enter Public DNS and then attach the .pem file of the respective EC2 wanted to connect.
- Once connected, drag and drop the JAR files of the projects separately to respective EC2s.

Connecting to the EC2 instance using ssh:

The below steps to be done to both EC2s created

To Install Java in linux:

Execute the below 2 commands

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

To check if java is successfully installed

-> java -version

In the Vocareum, our AWS credentials are present in the AWS Details Tab.so open the tab using the below commands,create aws credentials file.

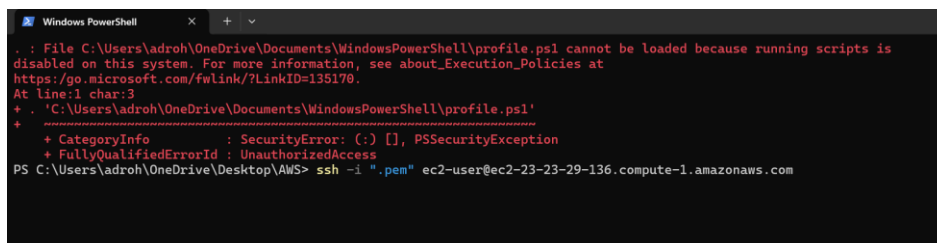
- To create directory -> mkdir home/ec2-user/.aws
- To create a file -> touch home/ec2-user/.aws/credentials
- To write in the credentials -> vi ~/.aws/credentials
-> Paste the credentials and press ESC key and enter **:wq** to save the file

Then,

- 1.Download the “.pem” key file and placed it in folder for the both instance.
- 2.Open the folder and type putty in the addressbar,it will prompt a putty window.
- 3.In the putty window navigate to connections->ssh->Auth->credentials provide the required information like , HostIP address(use the AWS instance address) and select the downloaded key.
- 4.it will connect to the instance
- 5.Use the ssh command to connect to your EC2 instance,open the folder in which both the keys are present and open it in windows powershell using the command:

Syntax: ssh -i your-key.pem ec2-user@your-ec2-instance-ip

6. You will be prompted to confirm the connection by typing "yes" and then press Enter.
7. You are now connected to your EC2 instance via SSH and putty.



```
Windows PowerShell
. : File C:\Users\adroh\OneDrive\Documents\WindowsPowerShell\profile.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:3
+ . 'C:\Users\adroh\OneDrive\Documents\WindowsPowerShell\profile.ps1'
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

PS C:\Users\adroh\OneDrive\Desktop\AWS> ssh -i ".pem" ec2-user@ec2-23-23-29-136.compute-1.amazonaws.com
```

Java Project Explanation and Execution Flow:

To accomplish this project, we need to create two Java projects to perform the following tasks:

Creating JAR Files:

To execute these programs on EC2 instances, you'll need to create executable JAR files for each task. Ensure that you have the executable JAR files for both programs, as you'll upload them to the respective EC2 instance

Commands to Execute JAR files:

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

Object Detection Program (Ec1):

1. This program is executed on Ec1, is to detect car objects within images. It will read 10 images from a specific S3 bucket (e.g., <https://njit-cs-643.s3.us-east-1.amazonaws.com>) and perform object detection.
2. When the program identifies a car in an image with a confidence level exceeding 90%, it will record the index of that image. Subsequently, this index will be pushed to SQS for further processing.

Text Recognition Program (Ec2):

1. This program is executed on Ec2. It reads image indexes from the SQS queue as soon as they become available in the queue.
2. The program then downloads the corresponding images from the S3 bucket one by one and perform text recognition using AWS Rekognition.
3. It's important to highlight that both Ec1 and Ec2 operate in parallel. For instance, while Ec1 is processing image 3, Ec2 may simultaneously work on image 1, previously identified as containing a car by Ec1.
4. Once Ec1 completes its image processing, it signals to Ec2 by adding an index of -1 to the queue, indicating that no more indexes will be received.
5. When Ec2 finishes its tasks, it generates a file within its associated Elastic Block Store (EBS) volume. This file contains the indexes of images that exhibit both cars and text, along with the actual text associated with each image index.


```
ec2-user@ip-172-31-37-120:~$
62 kernel-5.15 available [ ~stable ]
63 postgresql14 available [ ~stable ]
64 firefox available [ ~stable ]
65 lustre available [ ~stable ]
66 tpmp1.1 available [ ~stable ]
67 awsl11 available [ ~stable ]
68 php8.2 available [ ~stable ]
69 dnsmasq available [ ~stable ]
70 unbound1.17 available [ ~stable ]
72 collectd-python3 available [ ~stable ]
† Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-172-31-37-120 ~]$
[ec2-user@ip-172-31-37-120 ~]$
[ec2-user@ip-172-31-37-120 ~]$
[ec2-user@ip-172-31-37-120 ~]$
[ec2-user@ip-172-31-37-120 ~]$
[ec2-user@ip-172-31-37-120 ~]$
[ec2-user@ip-172-31-37-120 ~]$ ls
[ec2-user@ip-172-31-37-120 ~]$ cd .aws
[ec2-user@ip-172-31-37-120 ~]$ cd .aws
[ec2-user@ip-172-31-37-120 .aws]$ vi credentials
[ec2-user@ip-172-31-37-120 .aws]$ cd..
-bash: cd.: command not found
[ec2-user@ip-172-31-37-120 .aws]$ cd..
-bash: cd.: command not found
[ec2-user@ip-172-31-37-120 .aws]$ cd ..
[ec2-user@ip-172-31-37-120 ~]$ java -jar text-rekognition-0.0.1-SNAPSHOT.jar

Spring Boot (v2.3.4.RELEASE)

2024-03-06 00:05:11.763 INFO 6197 --- [main] com.aws.ec2.TextRekognition : Starting TextRekognition v0.0.1-SNAPSHOT on ip-172-31-37-120.ec2.internal with PID 6197 (/home/ec2-user/text-rekognition-0.0.1-SNAPSHOT.jar started by ec2-user in /home/ec2-user)
2024-03-06 00:05:11.775 INFO 6197 --- [main] com.aws.ec2.TextRekognition : No active profile set, falling back to default profiles: default
2024-03-06 00:05:12.115 INFO 6197 --- [main] com.aws.ec2.TextRekognition : Started TextRekognition in 1.422 seconds (JVM running for 2.778s)
```

OUTPUT:

After executing all the code Finally, it prints the output of the text recognition process.

```
Image Index: 1.jpg
Text Detected: $ BR8167
Text Detected: $
Text Detected: BR8167
Image Index: 4.jpg
Text Detected: YHI9 OTZ
Text Detected: YHI9
Text Detected: OTZ
Image Index: 7.jpg
Text Detected: Lamborghini
Text Detected: LP 610 LB
Text Detected: B0
Text Detected: BMW
Text Detected: Lamborghini
Text Detected: LP
Text Detected: 610 LB
Text Detected: B0
Text Detected: BMW
```