**AI 620 Emerging Topics in Artificial Intelligence**

**HOS02A Pictorial Translator Application**

02/19/2023 Developed by Yared Shewarade

09/11/2024 Reviewed by Naveena Moddu

09/16/2024 Updated by Anh Nguyen

School of Technology and Computing (STC) @City University of Seattle (CityU)

**Before You Start**

1. The directory path shown in screenshots may be different from yours.
2. Some steps are not explained in the tutorial**.** If you are not sure what to do:
3. Consult the resources listed below.
4. If you cannot solve the problem after a few tries, ask a courses student worker for help.

**Learning Outcomes**

Students will be able to learn:

1. Introduction to Pictorial Translator
2. Setting up pictorial translator application project application

**Resources**

* Tripuraneni, S., & Song, C. (2019). Hands-on artificial intelligence on amazon web services: Decrease the time to market for AI and ML applications with the power of AWS (1st ed.). Packt.

# Introduction to Pictorial Translator Application

The pictorial translator application is a tool that provides a sequential process for uploading the image, viewing the uploaded images, and seeing the translated text. It’s interaction with the upload image endpoint and storage service is straightforward. The users' requests are essentially passed through to AWS S3 and back in a chain that is shielded from both the endpoint and the application through the layers of abstraction. The photos will be stored in an S3 bucket, and the text detection and translation will be performed from the S3 bucket.

The translate image text endpoint simplifies some business logic from the pictorial translator application. Pictorial Translator is only sending the image ID to the translate image text endpoint and then receiving the translation for every line of text in the image. The translate image text endpoint does a couple of things behind the scenes. This endpoint is calling detect\_text() in recognition service on the entire image, and then calling translate\_text() in translation service multiple times for the lines of detected text. The endpoint will only call the translation service if the detected line of text meets a minimum confidence threshold.

Graphical user interface

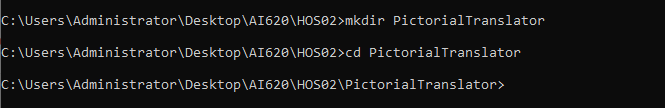
Description automatically generated with low confidence

# Setup project structure

Note: For submission, take the screenshot for all steps and save it in your local repository.

1. Go to your HOS02 repository folder. Type the following to create root project directory in your terminal/cmd.

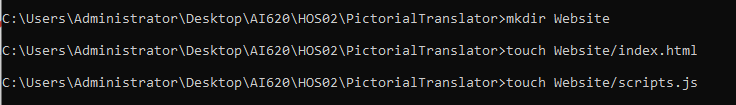
Text

Description automatically generated

1. Type the following to create placeholders for creating a directory named Website, two files index.html, scripts.js inside website.

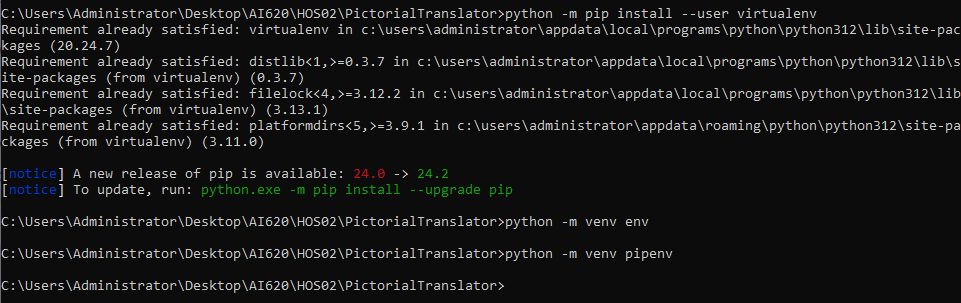
Text

Description automatically generated



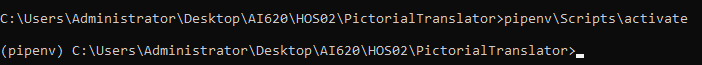
1. Type the following to create a Python virtual environment

|  |
| --- |
| python -m pip install --user virtualenv  python -m venv pipenv |



1. Type the following to activate the virtual environment
2. Windows

|  |
| --- |
| pipenv\Scripts\activate |



1. OSX:

|  |
| --- |
| pipenv/bin/activate |

In case you face the permission denied issue while using the above command. Please run the below commands. Below are images showing the above command and ways to handle the permission denied issue by using any of the below commands.

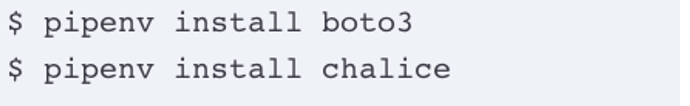
|  |
| --- |
| . pipenv/bin/activate |

(Note: *space* between dot and ‘venv/bin/activate’)

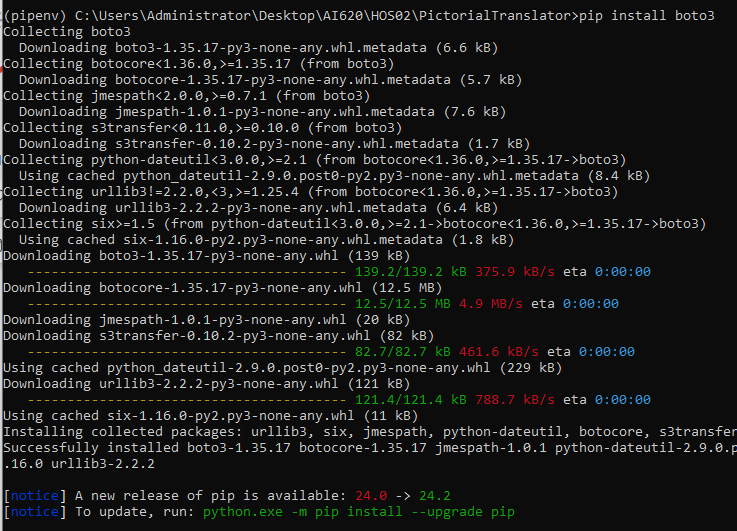
Or

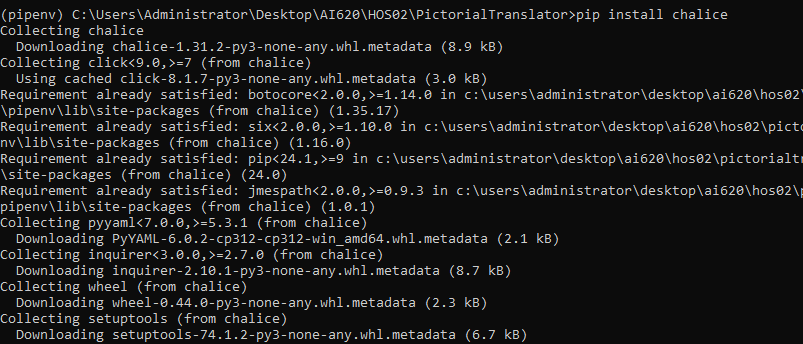
|  |
| --- |
| source pipenv/bin/activate |

1. Type the following to install boto3 and chalice



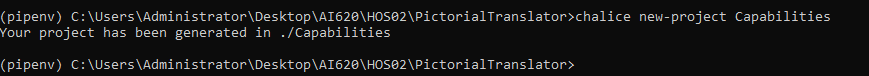
Note: - if pipenv doesn’t work, you can use pip/pip3





1. Type the following to create orchestration layer

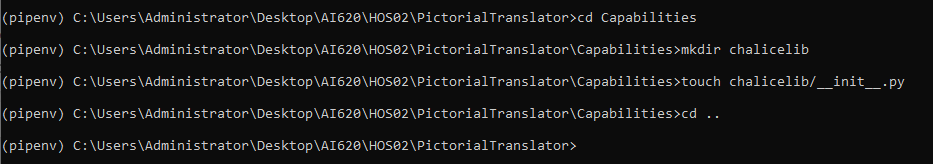




1. Type the following to create the chalicelib python package

A picture containing text

Description automatically generated



Note: - The project structure for pictorial translator should look like the following:

Diagram

Description automatically generated with medium confidence

# Recognition service – text detection

Let’s leverage the Amazon Rekognition service to detect text in an image of German street sign.



Source:  <https://www.freeimages.com/photo/german-one-way-street-sign-3-1446112>.

1. Download the above image and upload to your S3 bucket

Note:

1. You can create a new S3 bucket or use the bucket from HOS01.

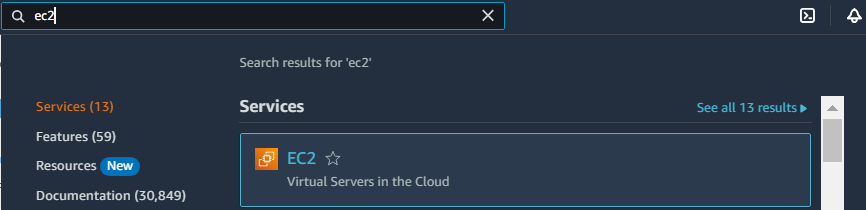
For example, this document will use the bucket with the name “contentsawsai”. You will need to create a **UNIQUE** bucket name as the bucket names across Amazon S3 are global.

1. Keep note of your S3 bucket region.

Graphical user interface, text, application, Word

Description automatically generated

1. Create EC2 instance on AWS. Login to AWS Console. Type “EC2” in the Find service search bar and click the EC2 keyword from the result.

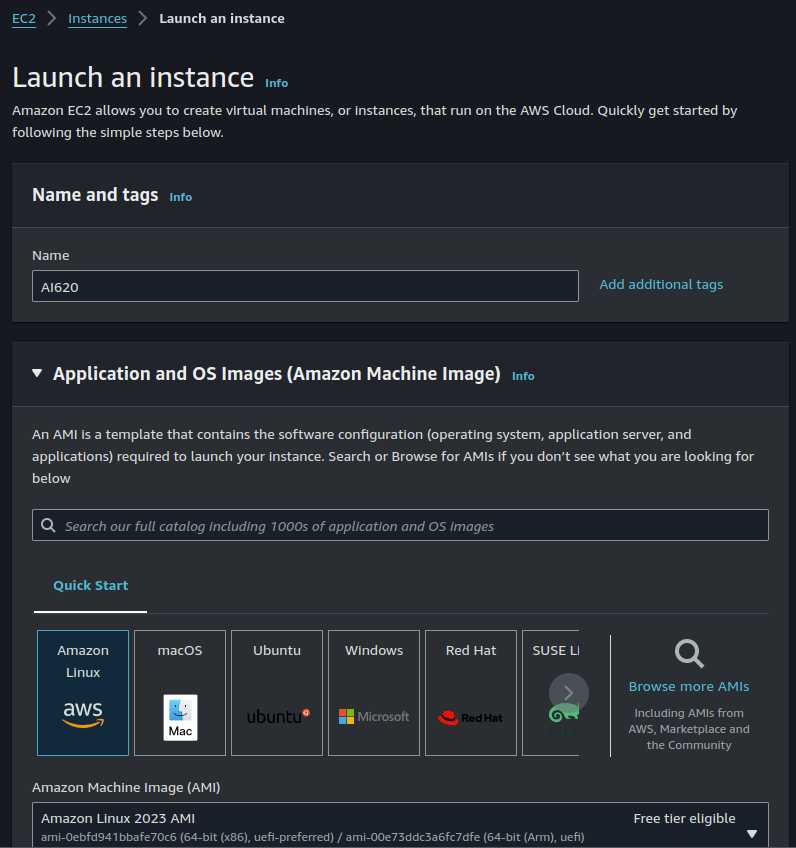


1. Click the “Launch Instances” button and click the “Launch Instance” option from the sub menu, and follow all the default steps to create instance.

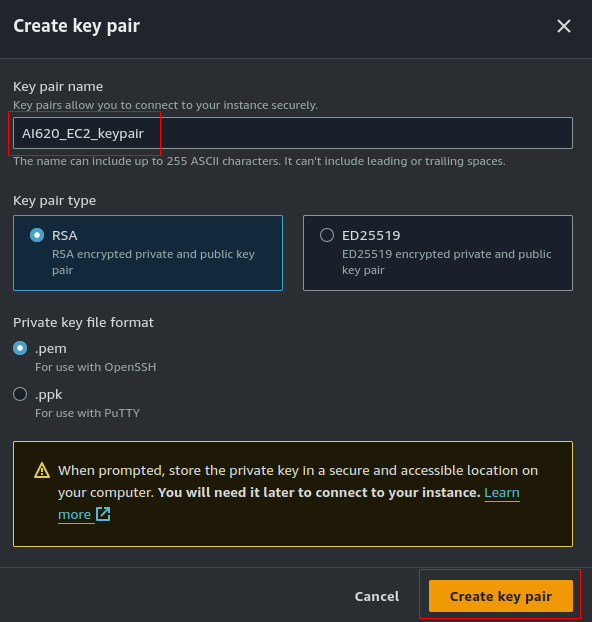
Graphical user interface, text, application, email

Description automatically generated

1. Name and tags: AI620
2. Application and OS Images (Amazon Machine Image): Amazon Linux 2023 AMI
3. Instance type: t2.micro



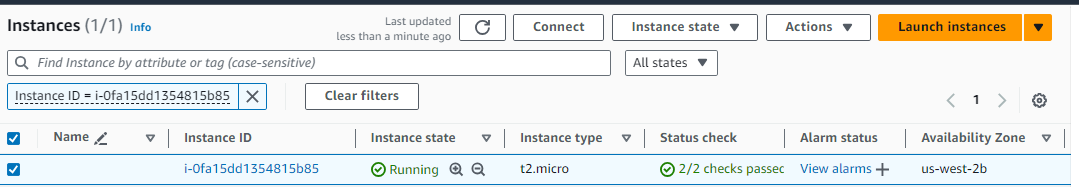
1. Key pair (login):
2. Click “Create new key pair”
3. Key pair name: AI620\_EC2\_keypair
4. Key pair type: RSA
5. Private key file format: .pem
6. Click “Create key pair”. Save this key pair in any folder on your local machine. You need to remember where you save this file.

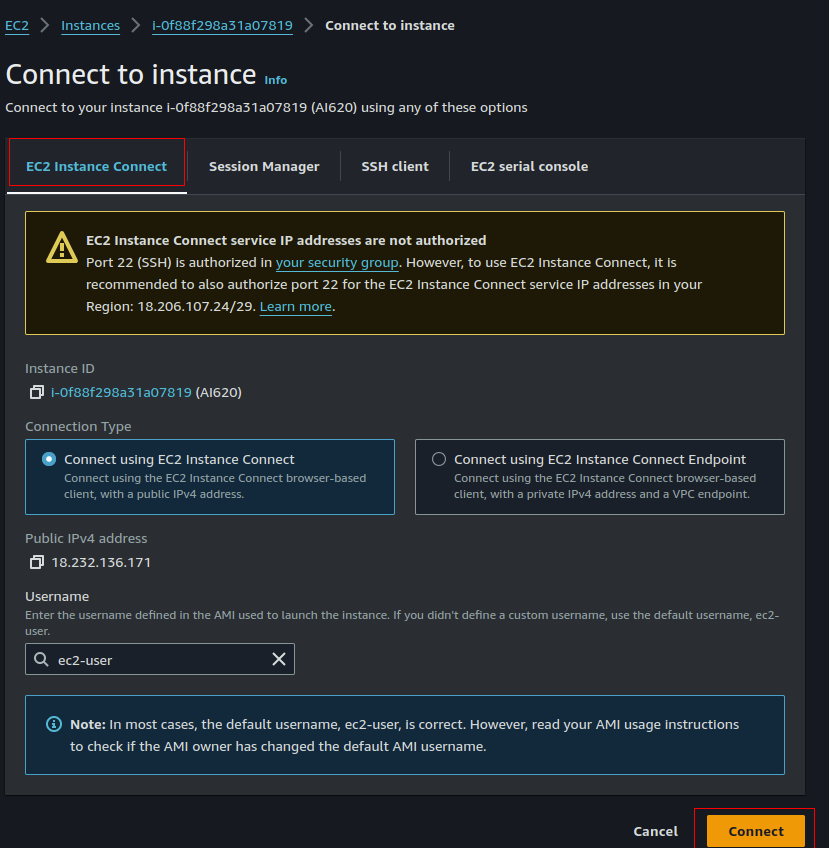


For the rest of the configuration, keep the defaults and click”Launch instance”

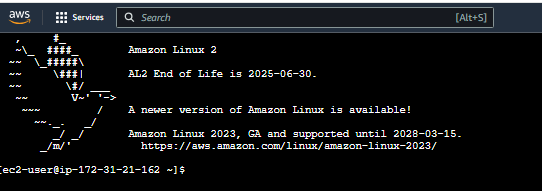
Wait until your instance’s Status check shows 2/2 checks passed

1. Click the “Connect” button. Change to “EC2 Instance Connect” tab.





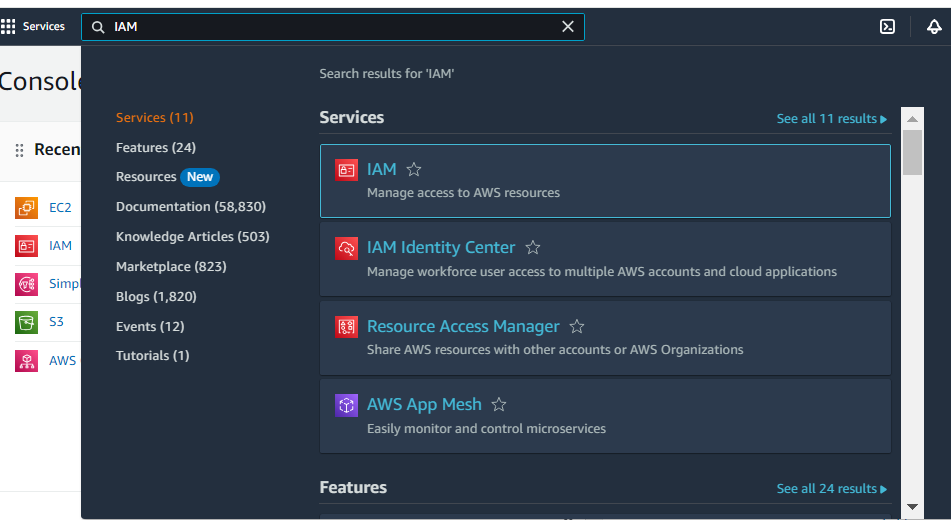
Click on the connect button. The AWS CLI open like this



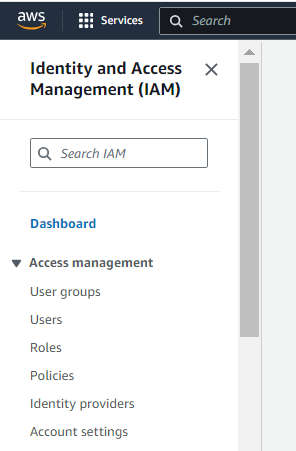
1. Before advancing further, we need to create Access Key ID and Secrete Access Key from AWS account and download it as .csv file. You can skip the next section if you have already created Access Key ID and Secrete Access Key.

# Steps to create Access Key ID and Secrete Access Key:

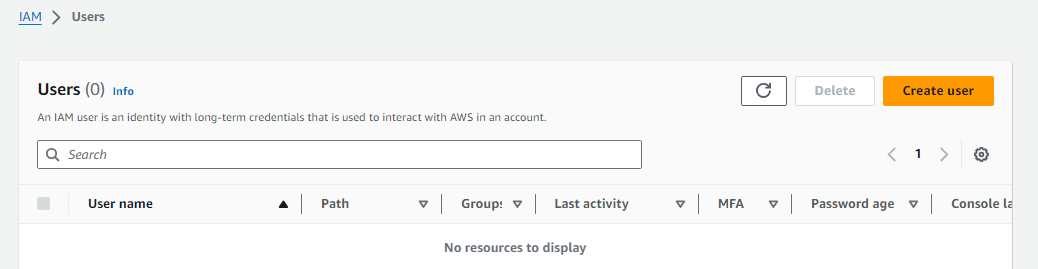
1. In the AWS Management Console, type “IAM” in the Find services search bar and   
   click IAM when the search result shows up.



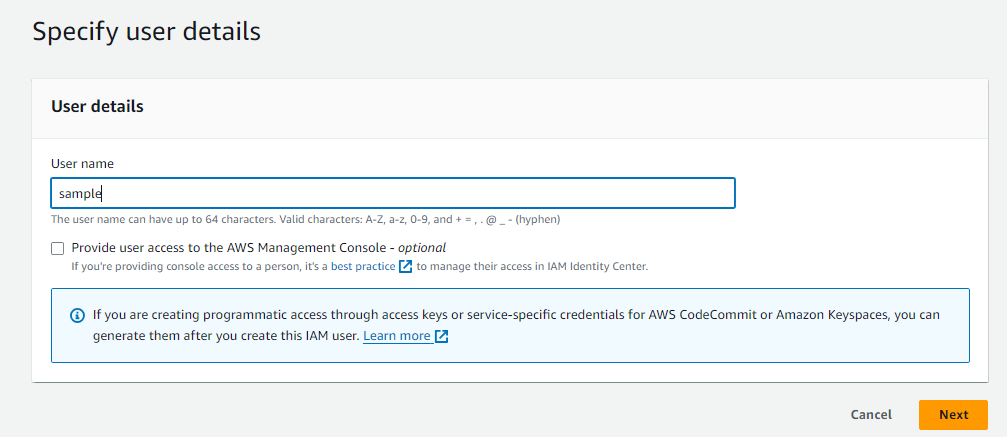
1. In the IAM interface, click “Users” button in the left-pane menu



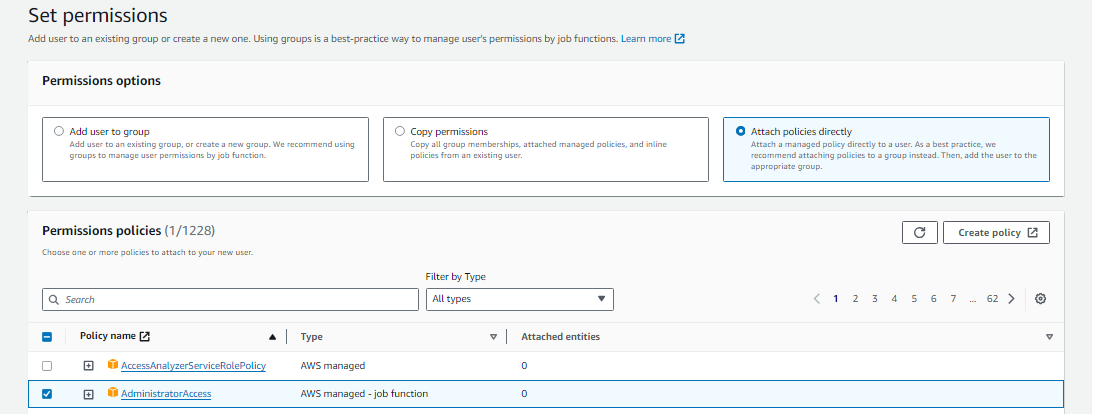
1. Click “Create user” button.



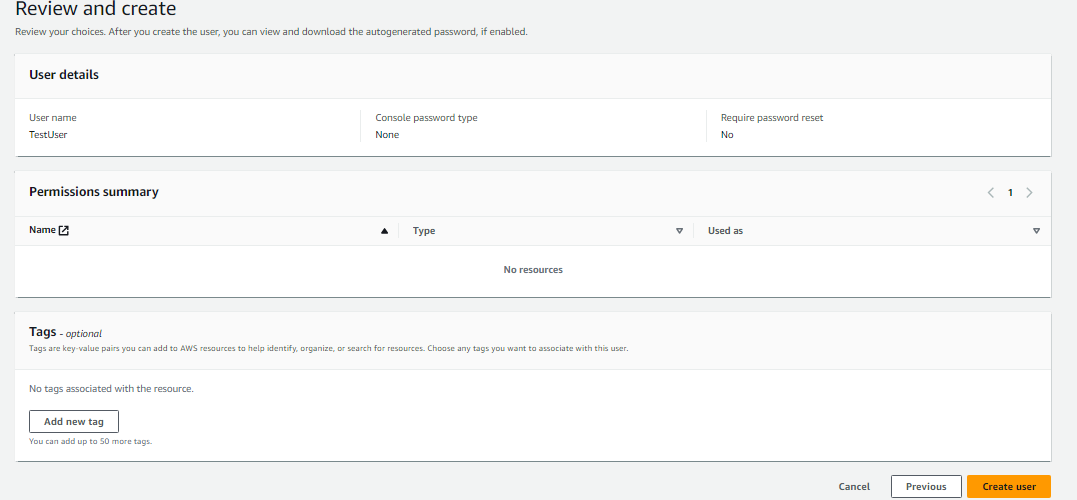
1. Fill out the Username (i.e., johndoe or janedoe). In this case, we used “sample” for a username.



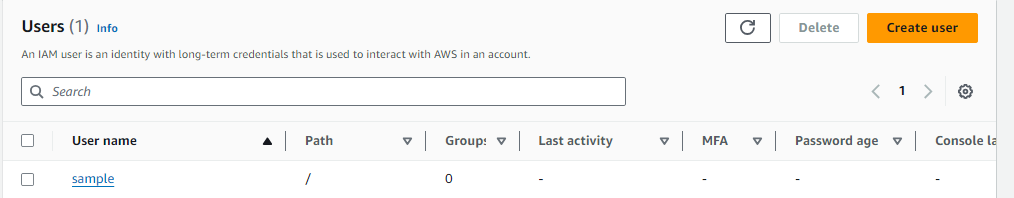
1. Click “next” button then set permissions to the user. Select “attach policies directly” and choose “AdministratorAccess” policy and click on next.



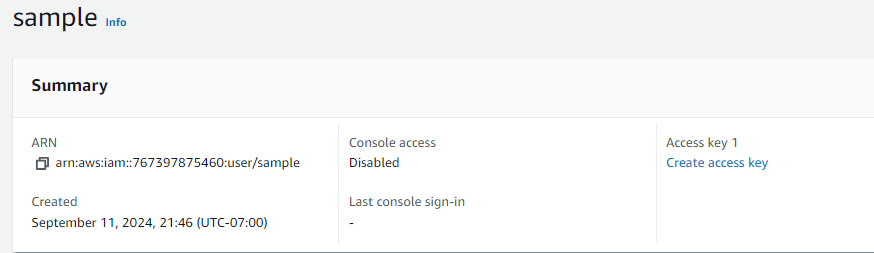
1. Click the “Create user” button.



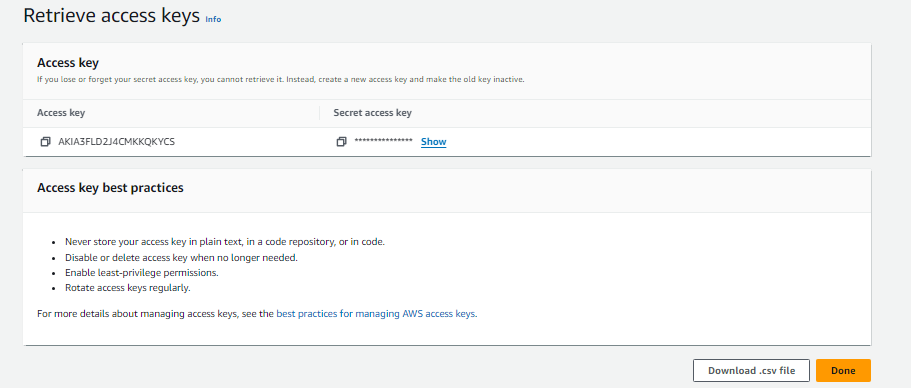
User created successfully.



1. Open the created sample and click on “Create access key “.



1. Choose CLI and select checkbox and click next



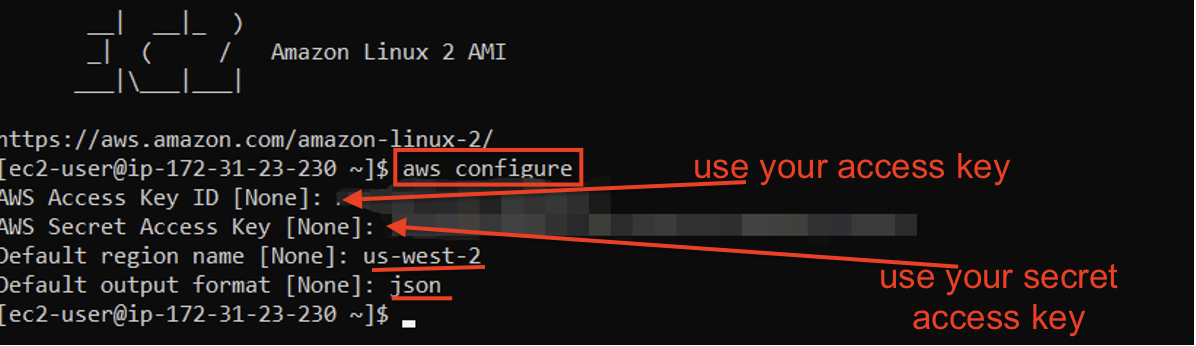
1. Click “Download.csv” and keep the file somewhere safe. Remember where you store this file as it will be used in next HOS.

You can keep this file in your HOS02 repository folder, but DO NOT commit this file.

1. Go back to your EC2 Instance Connect tab. Type the following to configure AWS.

|  |
| --- |
| aws configure |

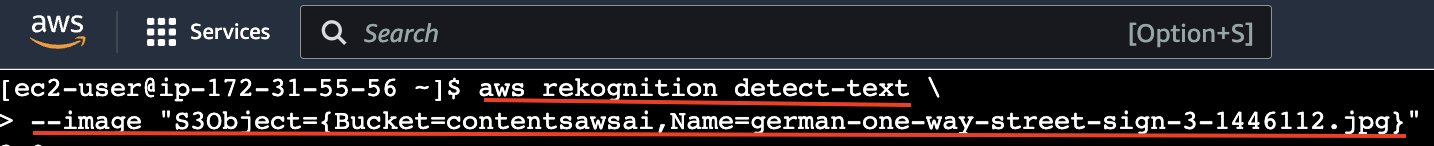
1. AWS Access Key ID: Your Access Key ID
2. AWS Secret Access Key: Your Secret Access Key
3. Default region name: Enter your S3 bucket region. Eg, us-west-2
4. Default output format: json
5. Press Enter to save all of the above configuration.



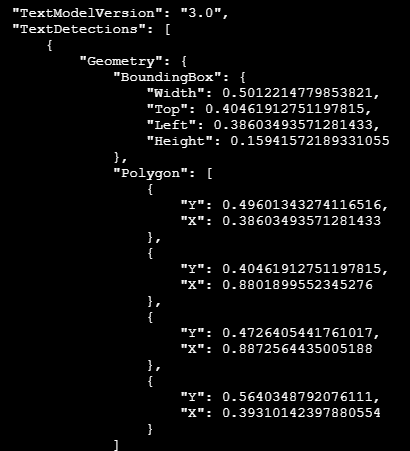
1. Type the following in AWS CLI to detect the uploaded images. If you uploaded the image in the HOS01 bucket, change the Bucket name in red.

|  |
| --- |
| aws rekognition detect-text –-image “S3Object={Bucket=contentsawsai,Name=german-one-way-street-sign-3-1446112.jpg}” |

Note – Type your bucket name.



The Output is:



# Clean up

On your AWS management console, go to EC2 > Instances. Check the running instances in the region. Click Instance state > Stop instance to turn off your instance.

# HOS submission instructions

1. Please install the GitHub Desktop: <https://cityuseattle.github.io/docs/git/github_desktop/>

2. Clone, organize, and submit your work through GitHub Desktop: <https://cityuseattle.github.io/docs/hoporhos>