AWS Deeplens Project documentation
By: Malik Warren

Security is a top priority in any project.

1. Create an IAM Role that will be used by a Lambda Function
    1. Select: S3FullAccess, AmazonRekognitionReadOnlyAccess, CloudWatchFullAccess, CloudWatchLogsFullAccess, AWSIoTDataAccess
    2. Create an additional role with the following Policies: AmazonS3FullAccess, AWSLambdaFullAccess

2. Create S3 Bucket, it will be used to store dataset images
    # All of this can be done via terminal (cloud9)
    Upload image of the first employee/family member
    Create Rekognition collection
        aws rekognition create-collection —collection-id <name_collection>
    After receiving confirmation message, you can add a face to the Rekogntion collection

```
aws rekognition index-faces \
      --image ‘{“S3Object”:
{“Bucket”:"<name_of_your_database","Name":"malik-
selfie.jpg"}}' \
      --collection-id “<cosc491-proj>“ \
      --detection-attributes "ALL" \
      --external-image-id “<myphoto.jpg>"
```

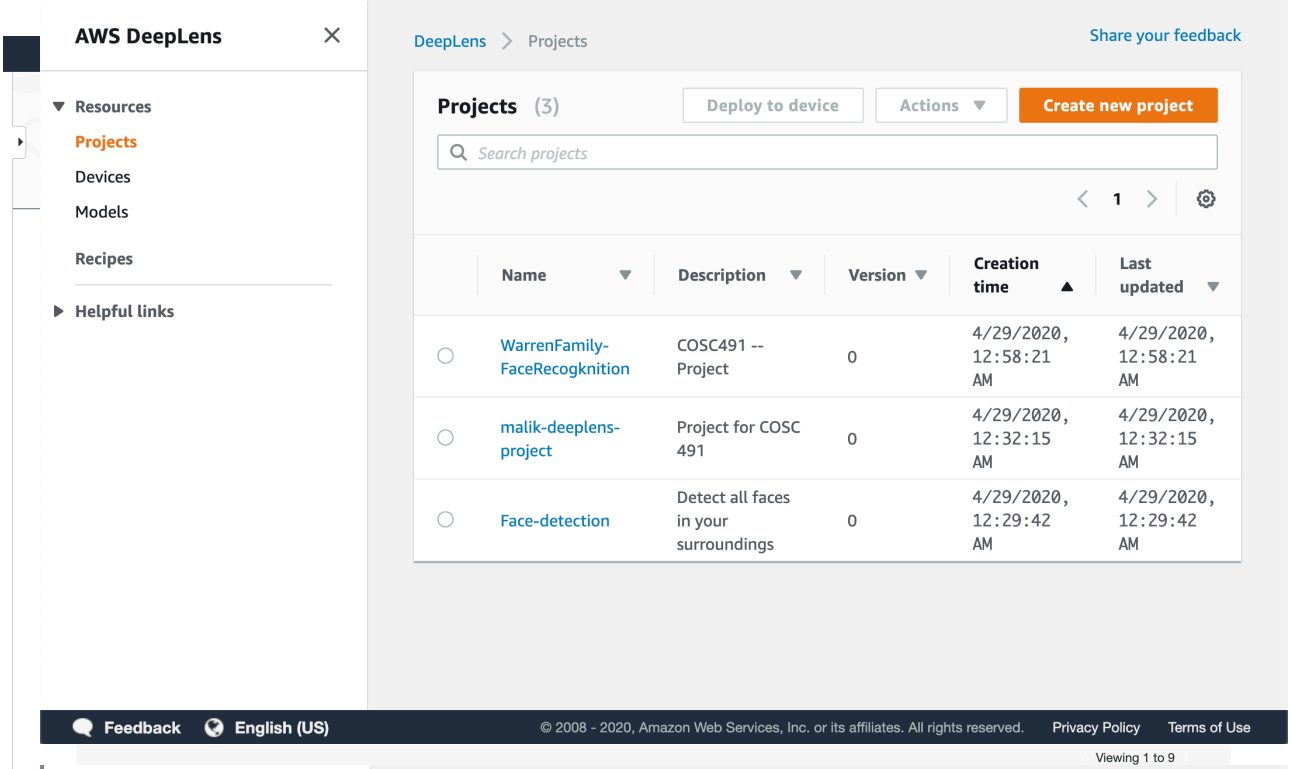
Parse through the results and locate: **Faceid: “xxxxxxxxxxxxxxxxxx”**

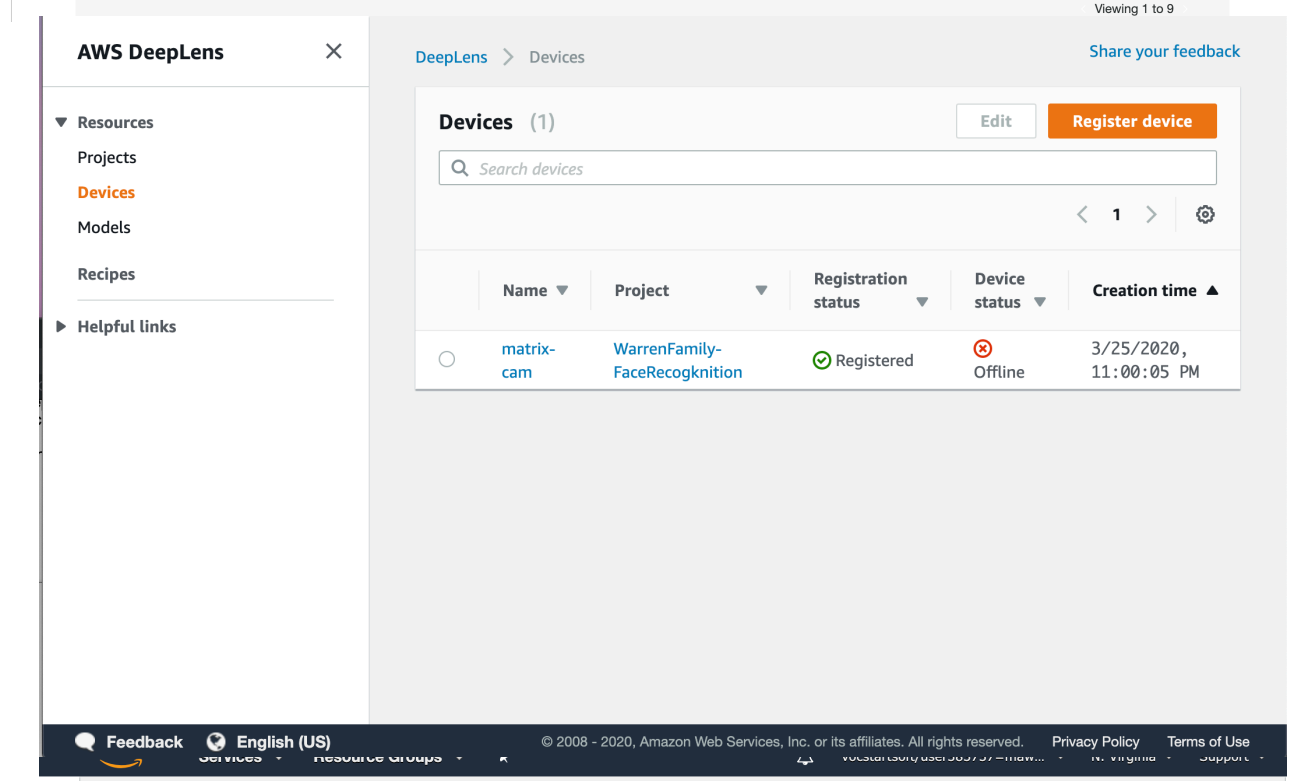
3. Create an employee/family profile using AWS DynamoDB
    1. Create a table using DynamoDB
        define attributes of table as: FaceId, AttributeType= <Additional information you may want>, key, and specify: region as us-east1
    2. Add record in the DynamoDB Table

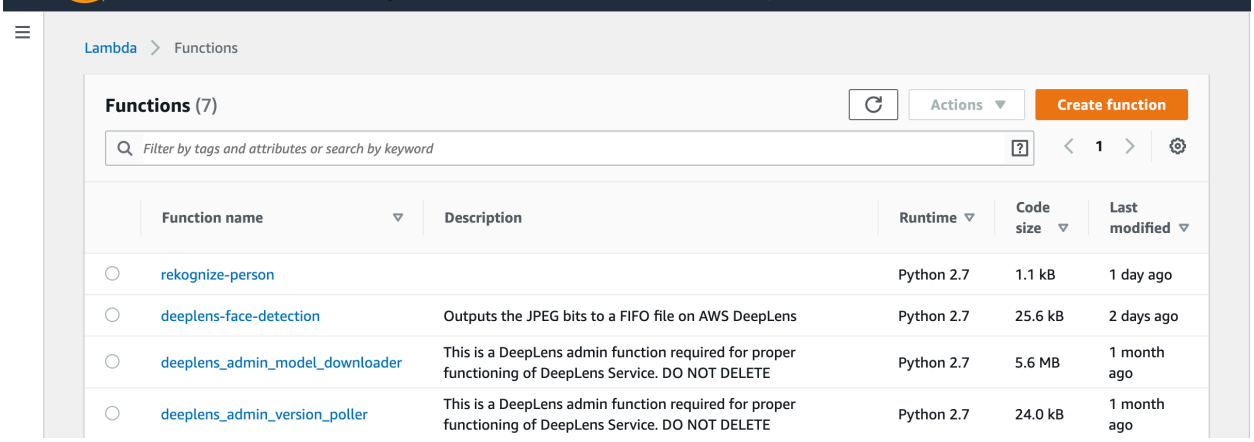
```
aws dynamodb put-item --table-name <“table name”> --item
'{ "FaceId": {"S": “<Faceid_from_previous_step>"},
“<Family/Employee>": {"<AttributeType>": “Hey! It’s
Malik"}}'
```

4. Activate Deeplens
    1. Register device
    2. Deploy a Face detection example model
        (additional resources can be found under **documentation** below

3. Create a new lambda function by going to lambda services in AWS.
    select runtime as Python 2.7
    Set execution role permissions to the role created in (**IAM Role 1.1**)
    Then create environment variable with key: “iot_topic” and value “worker-demo”
    change the timeout under basic settings to **10 secs**


AWS DeepLens
Resources
Projects
Devices
Models
Recipes
Helpful links
DeepLens > Projects
Share your feedback
Projects (3)
Deploy to device
Actions
Create new project
Search projects
1
Name
Description
Version
Creation time
Last updated
WarrenFamily-FaceRecogknition
COSC491 -- Project
0
4/29/2020, 12:58:21 AM
4/29/2020, 12:58:21 AM
malik-deeplens-project
Project for COSC 491
0
4/29/2020, 12:32:15 AM
4/29/2020, 12:32:15 AM
Face-detection
Detect all faces in your surroundings
0
4/29/2020, 12:29:42 AM
4/29/2020, 12:29:42 AM
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Viewing 1 to 9

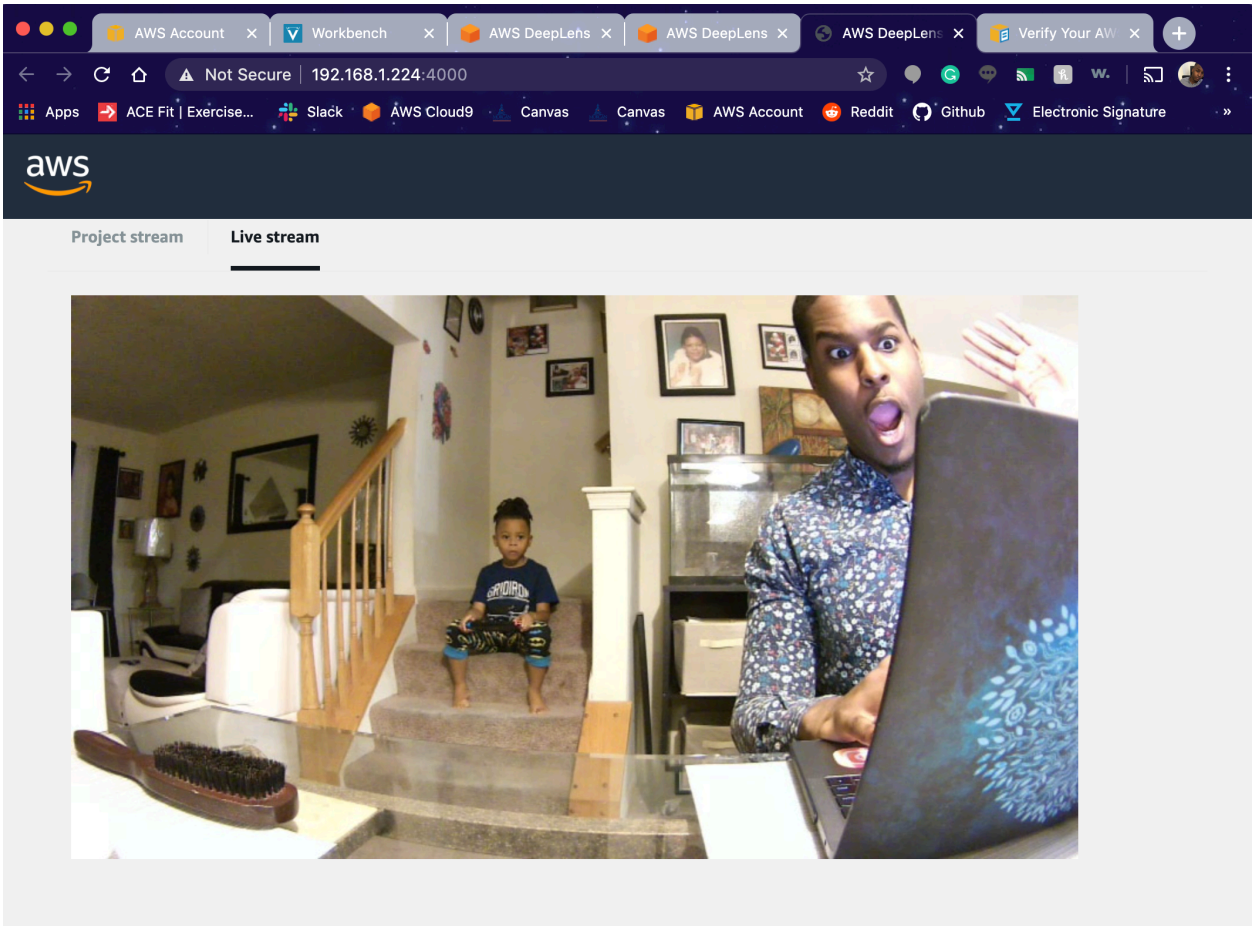

AWS DeepLens
Resources
Projects
Devices
Models
Recipes
Helpful links
DeepLens > Devices
Share your feedback
Devices (1)
Edit
Register device
Search devices
1
Name
Project
Registration status
Device status
Creation time
matrix-cam
WarrenFamily-FaceRecogknition
Registered
Offline
3/25/2020, 11:00:05 PM
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

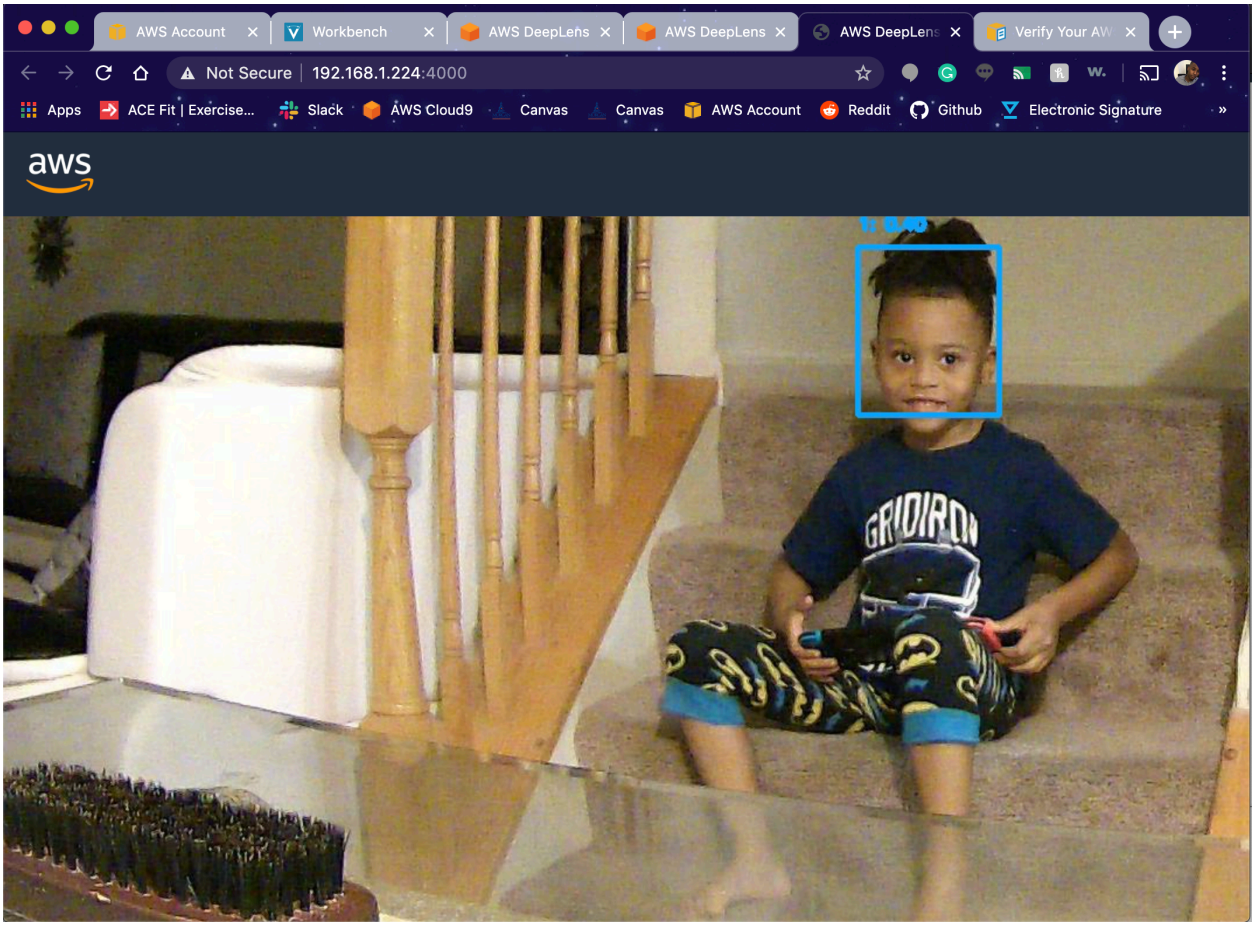

Lambda > Functions
Functions (7)
Actions
Create function
Filter by tags and attributes or search by keyword
1
Function name
Description
Runtime
Code size
Last modified
rekognize-person
Python 2.7
1.1 kB
1 day ago
deeplens-face-detection
Outputs the JPEG bits to a FIFO file on AWS DeepLens
Python 2.7
25.6 kB
2 days ago
deeplens_admin_model_downloader
This is a DeepLens admin function required for proper functioning of DeepLens Service. DO NOT DELETE
Python 2.7
5.6 MB
1 month ago
deeplens_admin_version_poller
This is a DeepLens admin function required for proper functioning of DeepLens Service. DO NOT DELETE
Python 2.7
24.0 kB
1 month ago

4. Listed above are some screenshots of what these difference services after configuration.

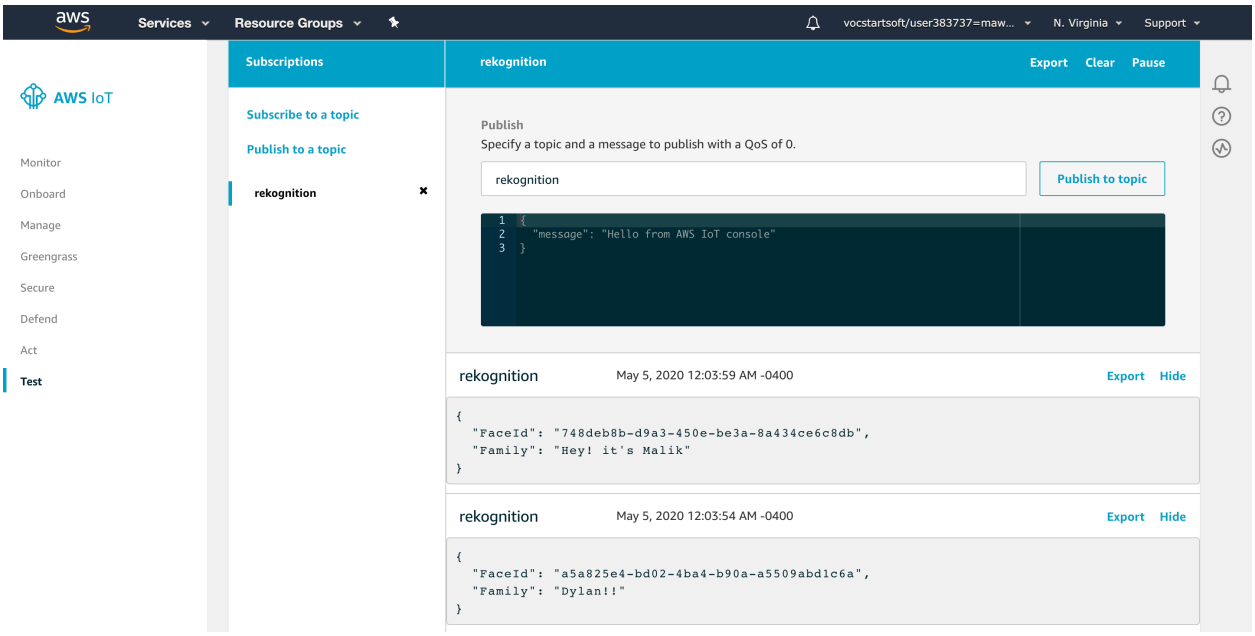
5. After successfully setting up device and the necessary tools go to AWS IoT.
    1. Click on test
    2. Then go to a subscription and add the topic “rekognition”

Original Live stream feed:





This is the feed after the facial detection function is implemented.



This screenshot is shows what happens after the facial recognition function is triggered.

Documentation:

https://www.aws.training/Details/Video?id=27179
https://docs.aws.amazon.com/deeplens/latest/dg/deeplens-inference-lambda-create.html
https://www.aws.training/Details/eLearning?id=32077
https://github.com/mahendrabairagi/DeeplensWorkshop/blob/master/FaceRekognition.md