Cloud Computing and Big Data -
Homework Assignment 2

Assignment:

In this second assignment you have to implement a Smart Door authentication system. You will learn how to use Kinesis Video Streams and Amazon Rekognition to build a distributed system that authenticates people and provides them with access to a virtual door.

Outline:

This assignment has the following components:

1. **Visitor Vault**
    a. Create a S3 bucket (**B1**) to store the photos of the visitors.
    b. Create a DynamoDB table “passcodes” (**DB1**) that stores temporary access codes to your virtual door and a reference to the visitor it was assigned to.
        i. Use the TTL feature[1] of DynamoDB to expire the records after 5 minutes.
    c. Create a DynamoDB table “visitors” (**DB2**) that stores details about the visitors that your Smart Door system is interacting with.
        i. Index each visitor by the FaceId detected by Amazon Rekognition[2] (more in the next section), alongside the name of the visitor and their phone number. When storing a new face, if the FaceId returned by Rekognition already exists in the database, append the new photo to the existing photos array.

        Use the following schema for the JSON object:

```
{
    “faceId”: “{UUID}”,
    “name”: “Jane Doe”,
```

[1] https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html
[2] https://aws.amazon.com/rekognition/

```
        "phoneNumber": "+12345678901",
        "photos": [
            {
                "objectKey": "my-photo.jpg",
                "bucket": "my-photo-bucket",
                "createdTimestamp":
            "2018-11-05T12:40:02"
            }
        ]
    }
```

2. **Analyze**
    a. Create a Kinesis Video Stream[3] (**KVS1**), that will be used to capture and stream video for analysis.
        i. Download the KVS Producer SDK GStreamer plugin[4]
            - We recommend you use the Docker image to run it, if you're not comfortable with compiling the library yourself.
        ii. Get an IP camera[5] or simulate one on your device to create an RTSP video stream.
        iii. Run one of the GStreamer commands[6] outlined in the GStreamer documentation to stream your RSTP source to Kinesis Video Streams.
    b. Subscribe Rekognition Video[7] to the Kinesis Video Stream (**KVS1**).
    c. Output the Rekognition Video analysis to a Kinesis Data Stream (**KDS1**) and trigger a Lambda function (**LF1**) for every event that Rekognition Video outputs.
    d. For every known[8] face detected by Rekognition, send the visitor an SMS message to the phone number on file. The text message should include a PIN or a One-Time Passcode (OTP) that they can use to open the virtual door.
        i. Store the OTP in the "passcodes" table (**DB1**), with a 5 minute expiration timestamp.

---

[3] https://aws.amazon.com/kinesis/video-streams/
[4] https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/examples-gstreamer-plugin.html
[5] https://www.amazon.com/TENVIS-Wireless-Surveillance-Security-Auto-Cruise/dp/B071DDBT7M
[6] https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/examples-gstreamer-plugin.html#examples-gstreamer-plugin-launch
[7] https://aws.amazon.com/rekognition/video-features/
[8] A known face entails a face detected by Rekognition, whose FaceId can be found in the "visitors" DynamoDB table (DB1).

e. For every unknown face detected by Rekogniton, send an SMS to the "owner" (i.e. yourself or a team member) a photo[9] of the visitor. The text message should also include a link to approve access for the visitor.
    i. If clicked, the link should take you to a simple web page (**WP1**) that collects the name and phone number of the visitor via a web form.
        - Submitting this form should create a new record in the "visitors" table (**DB2**), indexed by the FaceId identified by Rekognition. Note that you will have to build your own API to send information from the form to the backend. Its design and implementation is left up to you.
        - Generate a OTP as in step (d) above and store it in the "passcodes" table (**DB1**), with a 5 minute expiration timestamp.
        - Send the visitor an SMS message to the phone number on file. The text message should include the OTP.

3. **Authorize**
    a. Create a second web page (**WP2**), the "virtual door", that prompts the user to input the OTP.
        i. If the OTP is valid, greet the user by name and present a success message.
        ii. If the OTP is invalid, present a "permission denied" message.
    b. Note that you will have to build your own API to capture and validate the OTP. Its design and implementation is left up to you.

At this point you should be able to:

1. Process streaming video and perform stream analysis to identify faces.
2. Identify known people and provide them with an automatic access code.
3. Trigger an identification workflow that allows or denies access to unknown visitors, as well as adds them to the database for future training.

Acceptance criteria:

1. For a given visitor (ex. the TA), your system should be able to depict their face and email you to allow or deny them access.

[9] You will need to extract the photo from individual fragments of the video stream. Hint: look at the GetMedia API for Kinesis Video Media SDK on AWS. Once you get the video fragment, you'll need to extract frames from it, to obtain an actual image.

2. If allowed access, you should be able to capture their information through a hosted web page. You should then send them an SMS message with a valid OTP that is only valid for a maximum of 5 minutes.
3. The OTP should be valid only once and guaranteed unique across the different visitors.
4. For a returning visitor (ex. the same TA), your system should automatically send them an SMS message with a valid OTP.
5. Given a valid OTP, a visitor should be able to input it and receive a personalized greeting.
6. All other functionality should be working as described above.

ANNEX

Architecture Diagram