Smart Door Authentication System Walkthrough

This walkthrough document describes the steps to implement a Smart Door Authentication System leveraging AWS services such as Kinesis Video Streams, Amazon Rekognition, DynamoDB, and S3. The system authenticates visitors and provides access to a virtual door.

Overview

The Smart Door Authentication System aims to:

- Process video streams to identify faces.
- Authenticate known visitors by sending an OTP for access.
- Manage unknown visitors by sending notifications and capturing their details.
- Provide a user-friendly interface for OTP-based door access.

Architecture

The system comprises three main components:

- **Visitor Vault**
- **Stream Analysis**
- **Access Authorization**

Implementation Steps

1. Visitor Vault

a. S3 Bucket

- Create an S3 bucket (`B1`) to store visitor photos.

b. DynamoDB Tables

1. **Passcodes Table (`DB1`)**:
   - Schema: `{"visitorId": STRING, "passcode": STRING, "TTL": NUMBER}`
   - Use the DynamoDB TTL feature to expire records after 5 minutes.
2. **Visitors Table (`DB2`)**:

```
{
    "faceId": "{UUID}",
    "name": "{Name}",
    "phoneNumber": "{Phone}",
    "photos": [
        {
            "objectKey": "{PhotoKey}",
            "bucket": "{BucketName}",
            "createdTimestamp": "{Timestamp}"
        }
    ]
}
```

   - Index by `FaceId` for efficient lookups.
   - Append new photos to the `photos` array for existing `FaceId` entries.

---

2. Stream Analysis

a. Kinesis Video Stream

- Create a Kinesis Video Stream (`KVS1`) to capture video input.
- Use the KVS Producer SDK GStreamer plugin to stream video from an IP camera or simulated source.

b. Rekognition Video Integration

- Subscribe Amazon Rekognition Video to `KVS1` to analyze the video.

- Output analysis results to a Kinesis Data Stream (`KDS1`).
- Trigger a Lambda function (`LF1`) for every event from Rekognition Video.

c. Notifications

1. **Known Faces**:
   - Send SMS to visitors with a one-time passcode (OTP).
   - Store the OTP in `DB1` with a 5-minute expiration.
2. **Unknown Faces**:
   - Send SMS to the system owner with a photo of the visitor.
   - Include a link to approve or deny access.

---

3. Access Authorization

a. Web Page 1: Visitor Approval

- Create a simple web page (`WP1`) for approving unknown visitors.
- Capture visitor details (name and phone number) via a form.
- Store the details in `DB2` and generate an OTP for the visitor.
- Send the OTP to the visitor via SMS.

b. Web Page 2: Virtual Door

- Create a web page (`WP2`) for OTP entry.
- Validate the OTP against `DB1`:
  - If valid, greet the visitor and display a success message.
  - If invalid, display a "permission denied" message.

Acceptance Criteria

- Identify and notify system owners of unknown visitors with images and approval links.

- Generate and send valid OTPs for approved visitors.
- Ensure OTPs are unique, valid for 5 minutes, and usable only once.
- Automatically authenticate returning visitors with new OTPs.
- Allow visitors to access the virtual door using their OTPs.

Tools and Resources

- **AWS Services**: Kinesis Video Streams, Rekognition, DynamoDB, S3, Lambda, and SNS.
- **SDKs**: Kinesis Video Streams Producer SDK, AWS SDK for APIs.
- **Web Development**: HTML, CSS, JavaScript, and REST API integration.

Setup Instructions

1. **AWS Configuration**:
   - Set up required AWS services with appropriate IAM permissions.
   - Configure DynamoDB TTL and Rekognition Video event subscriptions.
2. **Code Deployment**:
   - Use AWS Lambda for backend logic.
   - Host web pages (`WP1` and `WP2`) using AWS S3 or other hosting solutions.
3. **Testing**:
   - Simulate video streams to verify Rekognition analysis.
   - Test OTP workflows for known and unknown visitors.

This document provides a comprehensive guide for implementing the Smart Door Authentication System. Clone the repository and follow the instructions to deploy the solution.