Serverless-Architecture-Deployment

1. Project Description

This project implements a serverless video streaming solution using AWS services — **API Gateway**, **AWS Lambda**, and **Amazon S3**. When the API Gateway endpoint is accessed, it triggers a Lambda function that generates a **pre-signed S3 URL** to securely stream a video file. This allows clients to view videos stored in S3 without exposing the actual S3 URL or making the bucket public.

---

2. Architecture Overview

css

CopyEdit

Client Browser

│

▼

API Gateway (HTTP Endpoint)

│

▼

AWS Lambda (Triggered on Request)

│

▼

Generates Pre-signed S3 URL (GET)

│

▼

Redirects to Video in S3

▼

Video Streams to Client

---

3. Components Used

- **Amazon S3**: Stores the video file(s)
- **AWS Lambda**: Generates a temporary access URL (pre-signed GET request)
- **API Gateway**: Exposes a secure HTTP endpoint for external clients

- **IAM Role**: Grants Lambda permission to read from S3

---

4. Implementation Steps

Step 1: Upload Video to S3

- Created an S3 bucket (sahbuck786)
- Uploaded a video file (e.g., WhatsApp Video 2025-05-20 at 10.15.33 AM.mp4)
- Ensured the bucket is **private** for security

Step 2: Create Lambda Function

- Wrote a Python-based Lambda function to:
    - Accept API calls
    - Generate a pre-signed S3 URL
    - Redirect the user to this URL for streaming

Step 3: Configure API Gateway

- Created a REST API in API Gateway
- Set up a resource path (e.g., /stream)
- Added a GET method and integrated it with the Lambda function

Step 4: Connect and Test

- Called the endpoint using Postman or a browser
- The client was redirected to the secure S3 URL
- Video streamed directly in the browser without exposing the bucket

---

5. Lambda Code (Generate Pre-Signed URL & Redirect)

python

CopyEdit

```
import boto3
from botocore.exceptions import ClientError


s3 = boto3.client('s3')
```

```
BUCKET_NAME = 'sahbuck786'
VIDEO_KEY = 'WhatsApp Video 2025-05-20 at 10.15.33 AM.mp4'

def lambda_handler(event, context):
    try:
        presigned_url = s3.generate_presigned_url(
            'get_object',
            Params={
                'Bucket': BUCKET_NAME,
                'Key': VIDEO_KEY
            },
            ExpiresIn=3600  # 1 hour
        )

        return {
            'statusCode': 302,
            'headers': {
                'Location': presigned_url
            },
            'body': ''
        }

    except ClientError as e:
        return {
            'statusCode': 500,
            'body': f"S3 Error: {str(e)}"
        }
```

---

6. IAM Role Policy for Lambda

Ensure the Lambda role has permission to generate pre-signed URLs:

json

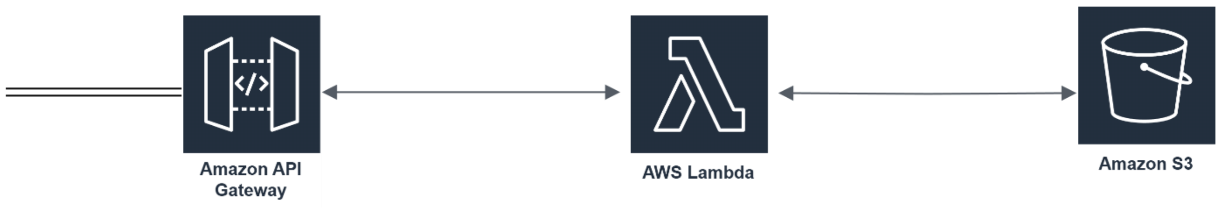
CopyEdit

```
{
  "Effect": "Allow",
  "Action": ["s3:GetObject"],
  "Resource": "arn:aws:s3:::sahbuck786/*"
}
```

---

7. Summary

This project demonstrates a secure and scalable **serverless video streaming** setup using AWS. By combining API Gateway, Lambda, and S3, we enable access to private media without exposing S3 objects directly or requiring server management.


Amazon API Gateway
AWS Lambda
Amazon S3