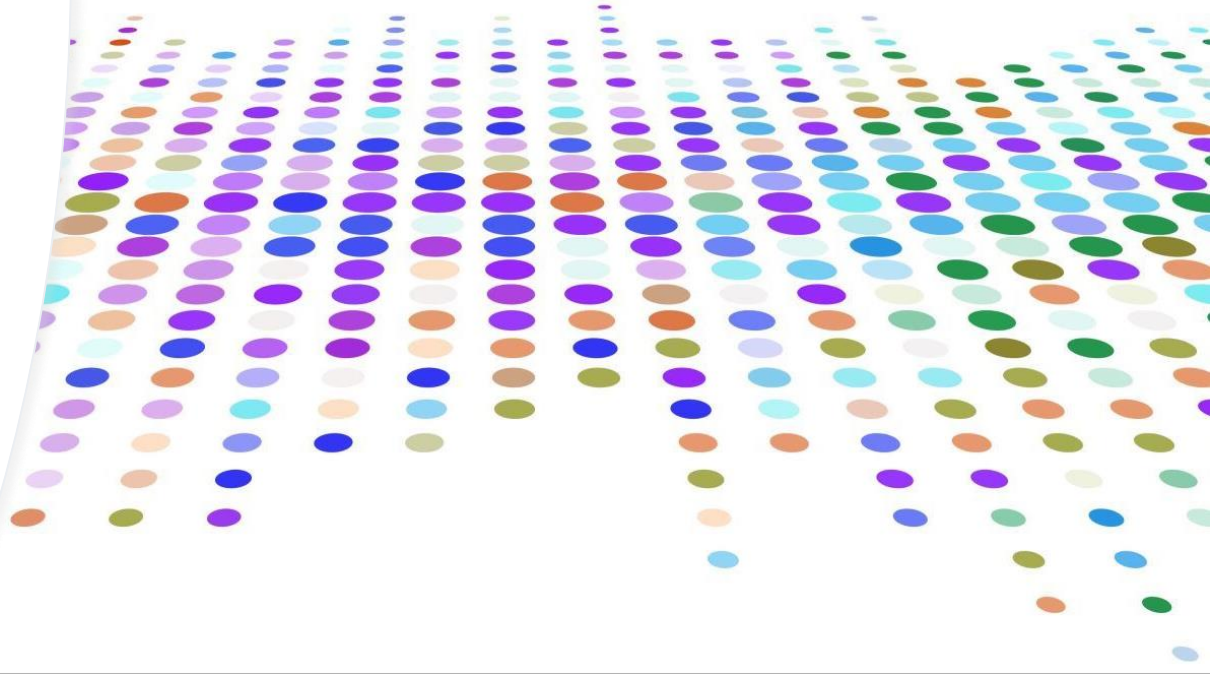
AWS - Serverless Object Detection application

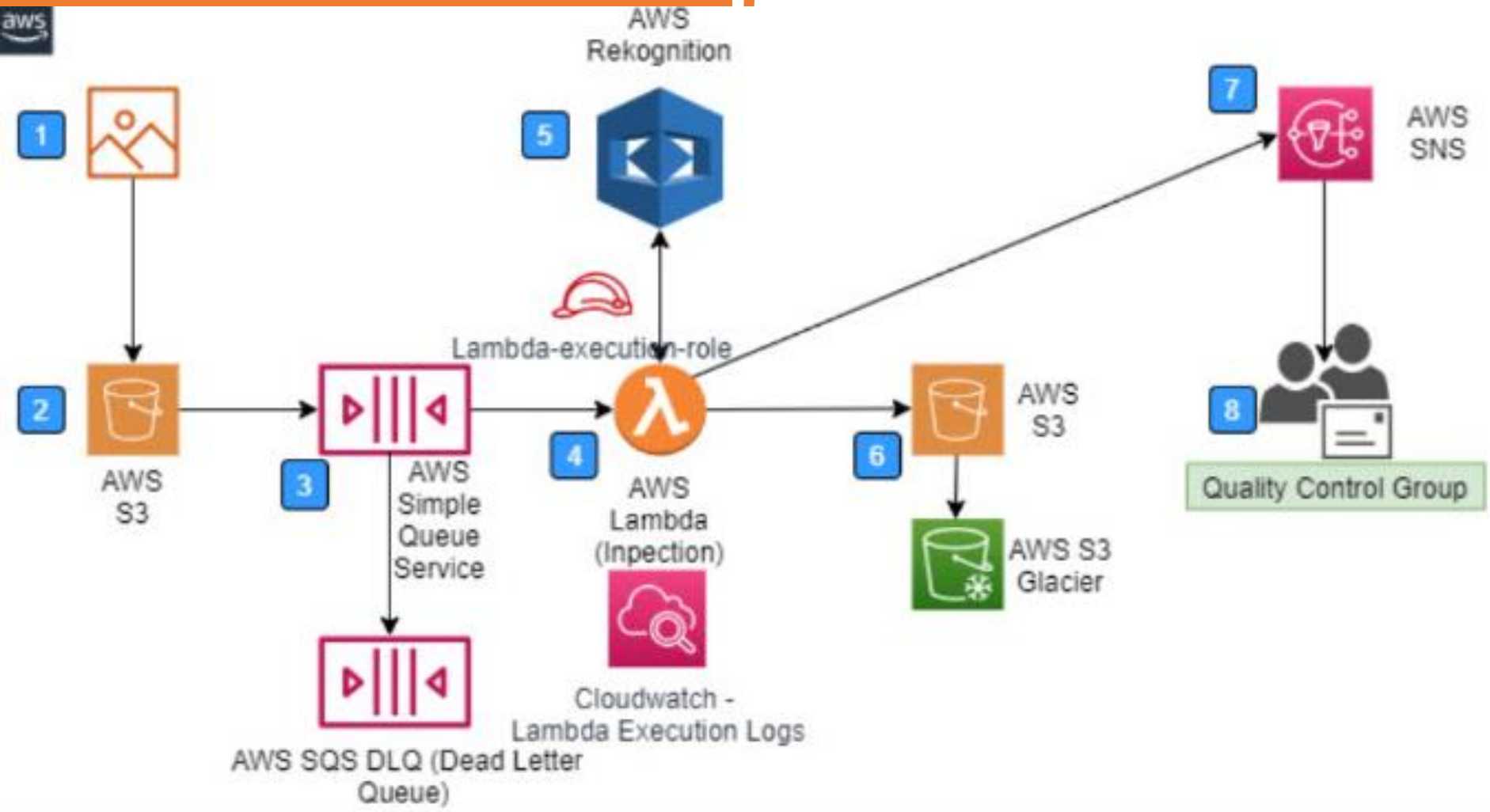
- ITC6450 - Group 4
- Sarina Poth
- Sandeep Samsani
- Jui Ying Lee



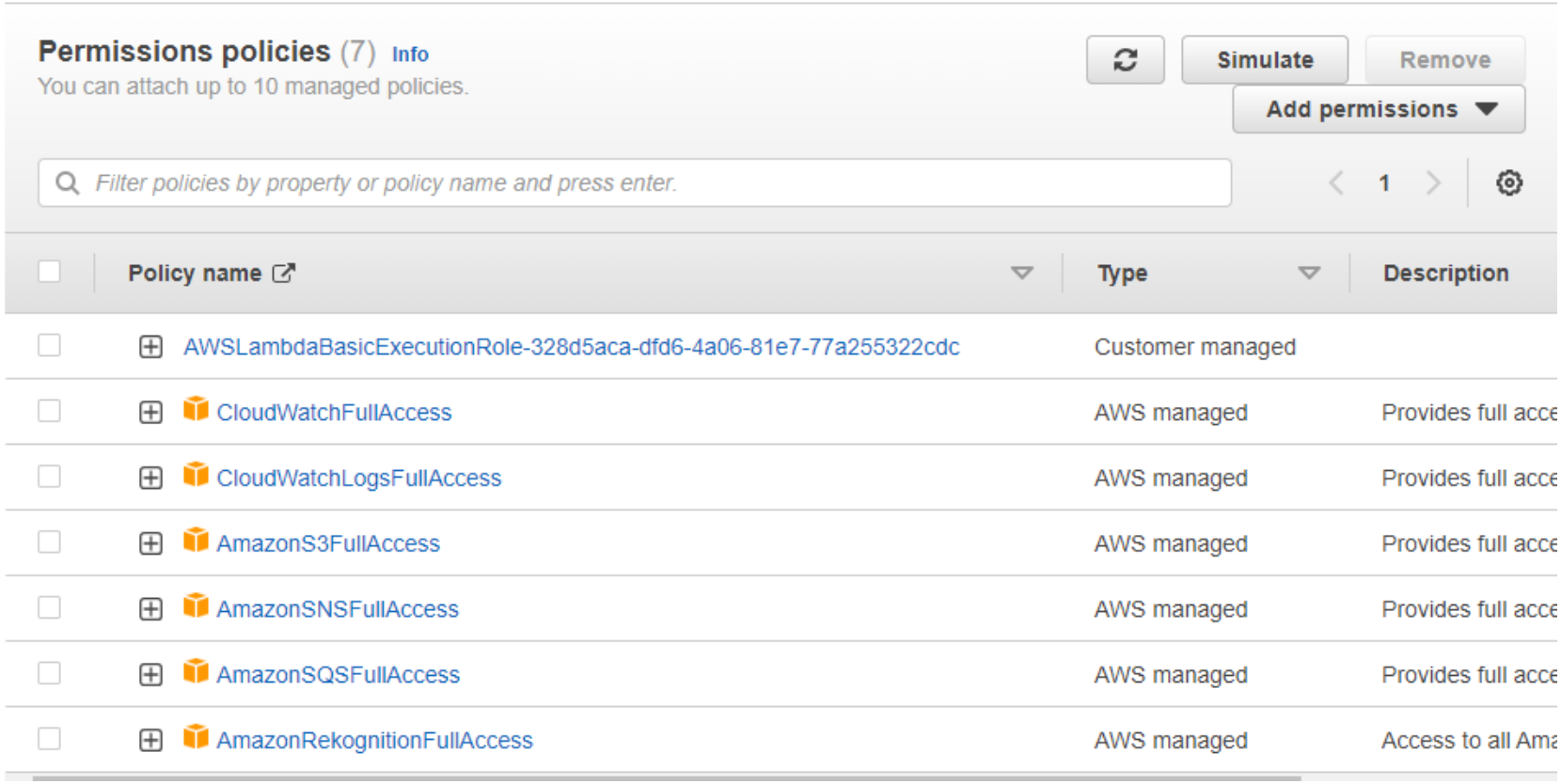
Agenda

- Explanation of architecture configuration and components:
  - IAM
  - S3
  - SQS
  - Lambda
  - Rekognition
  - CloudWatch
  - SNS
- Demonstration

Overview

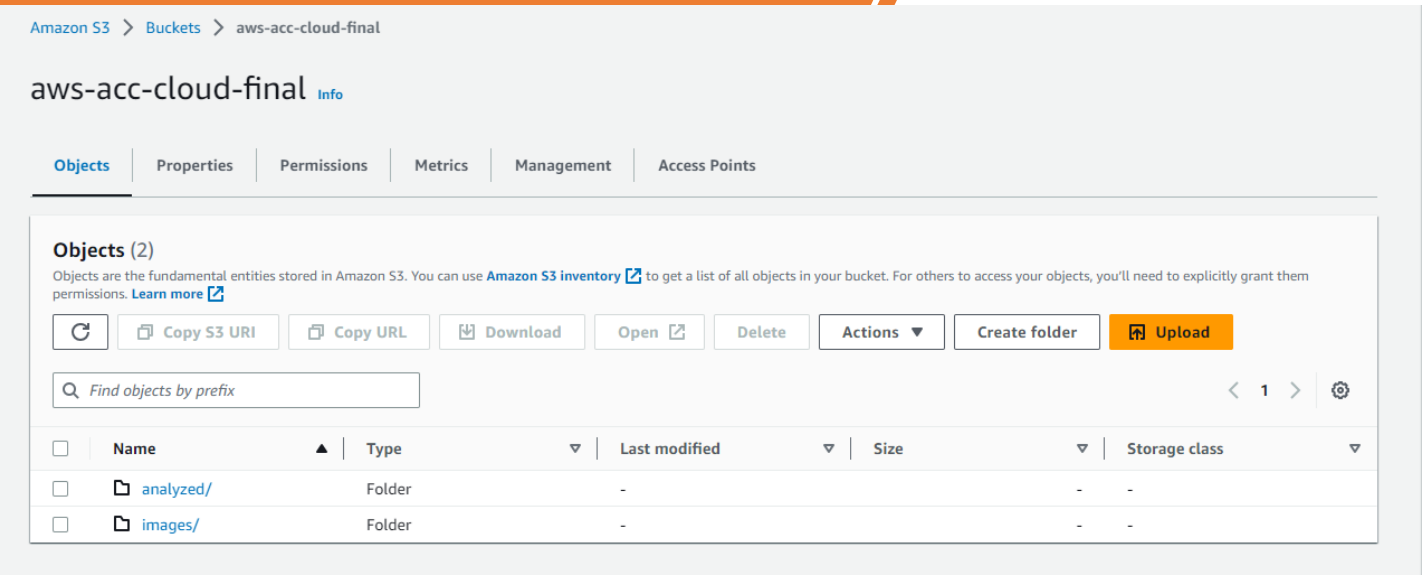

AWS
Rekognition
1
5
7
AWS
SNS
Lambda-execution-role
2
AWS
S3
3
AWS
Simple
Queue
Service
4
AWS
Lambda
(Inpection)
6
AWS
S3
8
Quality Control Group
AWS S3
Glacier
Cloudwatch -
Lambda Execution Logs
AWS SQS DLQ (Dead Letter
Queue)

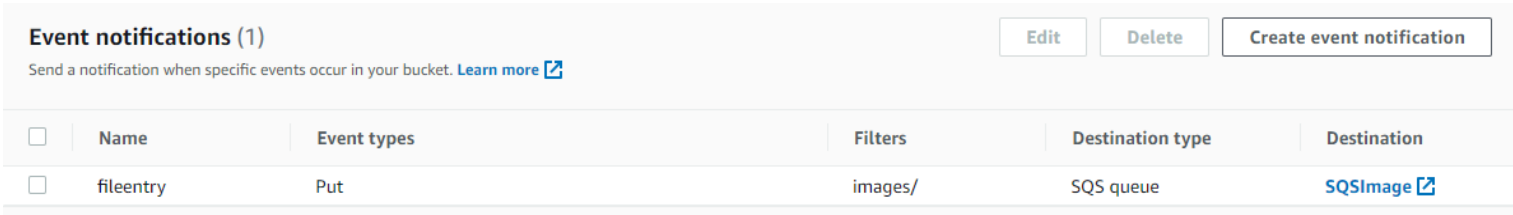
IAM Lambda Role



S3

- One bucket with an images folder and an analyzed folder
- Event notification configured to send messages to SQS queue when image is uploaded to the images folder





AWS SQS

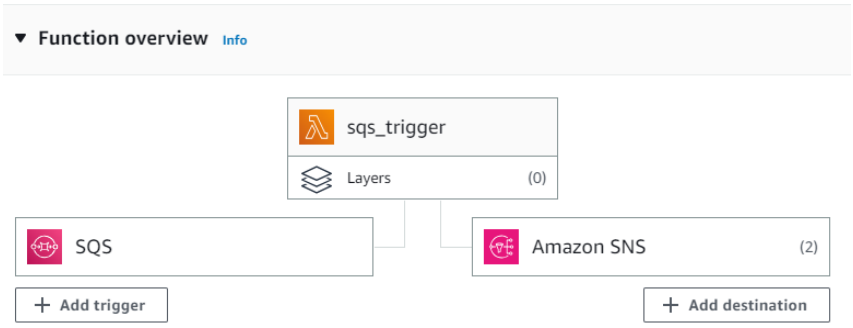
Access policy (Permissions) Info

Define who can access your queue.


```
{
  "Version": "2012-10-17",
  "Id": "Policy1687642161023",
  "Statement": [
    {
      "Sid": "Stmt1687642134863",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:us-east-1:323708512535:SQSImage",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:s3:::aws-acc-cloud-final"
        }
      }
```


- Primary queue and dead letter queue
- Messages from S3 are sent to the primary queue for processing
- Access policy grants the S3 bucket permission to send messages to SQS
- SQS triggers the Lambda function to process the images via Rekognition

AWS Lambda



- Focus on application code rather than infra. management.
- Lambda function is invoked in response to the trigger event made by SQS.
- Function executes in a runtime environment provided by AWS Lambda.
- Lambda execution enables serverless computing with easy scalability, cost efficiency, and abstraction of infrastructure.
- Embracing Lambda execution can unlock the benefits of serverless computing in our applications.

AWS Rekognition

```python
import json
import boto3

s3_client = boto3.client('s3')
rekognition_client = boto3.client('rekognition')
sns_client = boto3.client('sns')
sqs_queue_url = 'https://sqs.us-east-1.amazonaws.com/323708512535/SQSImage'
sns_topic_arn = 'arn:aws:sns:us-east-1:323708512535:notify_users'

def lambda_handler(event, context):

  # Print the event to check its structure

    for record in event['Records']:
        record= json.loads(record['body'])
        # Access the S3 bucket and object key from the record
        s3_bucket = record['Records'][0]['s3']['bucket']['name']
        s3_object_key = record['Records'][0]['s3']['object']['key']
        try:
            # Perform image analysis using AWS Rekognition
            response = rekognition_client.detect_labels(
                Image={
                    'S3Object': {
                        'Bucket': s3_bucket,
                        'Name': s3_object_key
                    }
                },
                MaxLabels=10,
                MinConfidence=80
            )

            # Determine if the analysis is successful based on desired labels and confidence threshold
            desired_labels = ['Widgets','Smartphones']  # Add your desired labels here
            confidence_threshold = 80  # Set your desired confidence threshold

            labels = [label['Name'] for label in response['Labels']]
            confidences = [label['Confidence'] for label in response['Labels']]
```

- Rekognition is an AWS service that utilizes advanced ML algorithms to analyze images and videos, enabling detection of objects, faces, and text, as well as performing other visual recognition tasks.
- Receive input through the trigger and passes input to the Lambda function.
- Invoke Rekognition APIs for image and video analysis.
- Process and return results or take actions based on analysis. (i.e., Labels, Confidence).
- Benefits of this service includes Scalability, cost efficiency, advanced analysis.

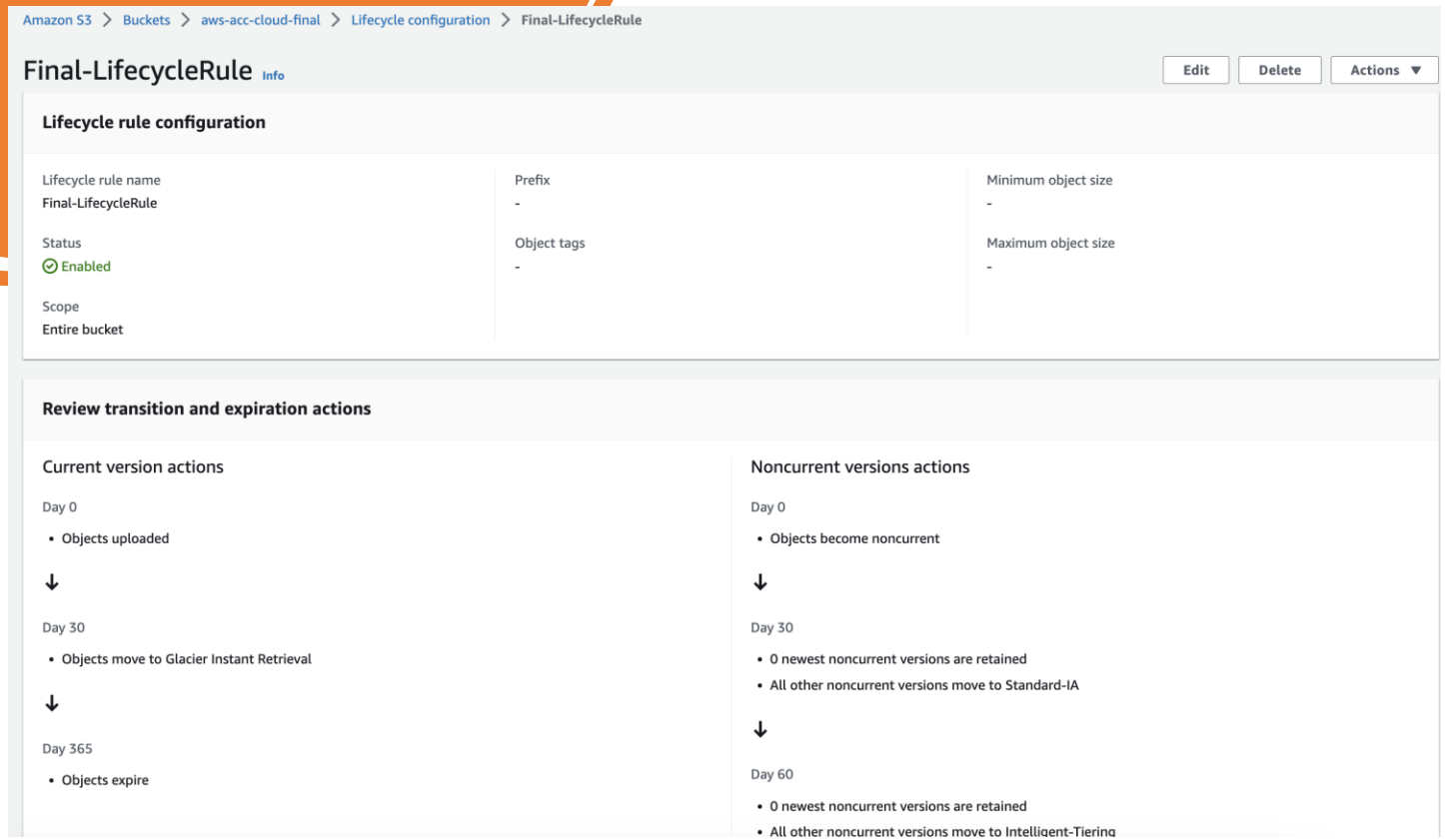
CloudWatch

- Monitors the Lambda function to see if it runs properly
- Logs any errors during the execution
- Shows which labels were detected in the uploaded images

Timestamp	Message
	No older events at this moment. *Retry*
2023-06-25T14:25:10.572-04:00	INIT_START Runtime Version: python:3.9.v23 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:b…
2023-06-25T14:25:10.989-04:00	START RequestId: b7866a55-7e42-5340-857a-3747f6f95998 Version: $LATEST
2023-06-25T14:25:12.263-04:00	Labels: ['Electronics', 'Phone', 'Mobile Phone', 'Iphone']
2023-06-25T14:25:12.263-04:00	Confidences: [99.99984741210938, 99.99984741210938, 99.99979400634766, 99.82115173339844]
2023-06-25T14:25:12.263-04:00	Success: False
2023-06-25T14:25:12.686-04:00	Image moved to: s3://aws-acc-cloud-final/analyzed/failure/images/phone.jpg
2023-06-25T14:25:12.736-04:00	Image deleted: s3://aws-acc-cloud-final/images/phone.jpg
2023-06-25T14:25:12.945-04:00	Analysis results published to SNS
2023-06-25T14:25:12.948-04:00	END RequestId: b7866a55-7e42-5340-857a-3747f6f95998
2023-06-25T14:25:12.948-04:00	REPORT RequestId: b7866a55-7e42-5340-857a-3747f6f95998 Duration: 1958.78 ms Billed Duration: 1959 m…

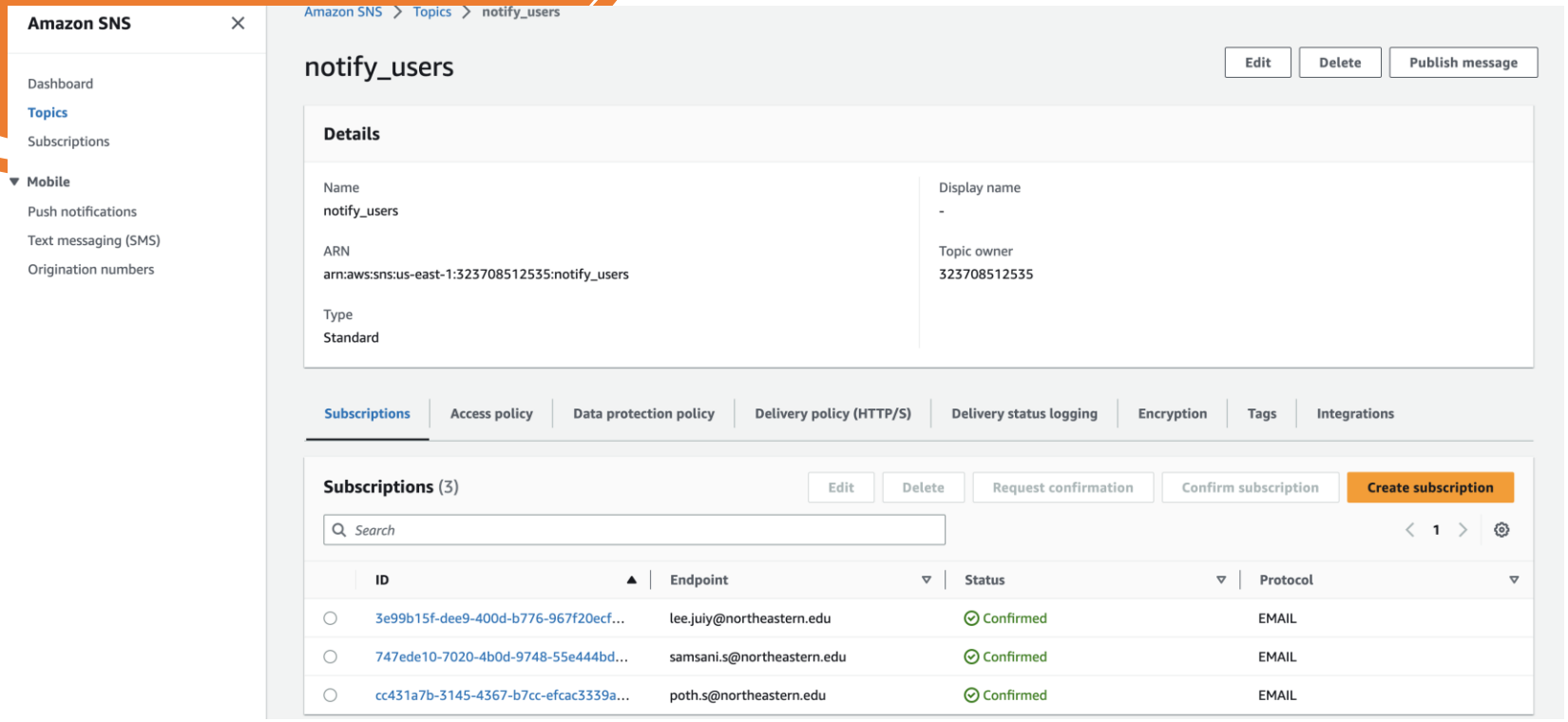
S3 –> S3 Glacier

- Creating Lifecycle Policies and transition :
  - Provide a cost-effective solution for managing data throughout its lifecycle.
  - Automate transitions and actions on objects in S3 buckets.
  - Streamline storage, reduce costs, and meet compliance requirements effortlessly.



Lambda -> SNS

- Build event-driven architectures and enable async communication between different components of the application



Live Implementation

- Let’s dive into our presentation.

Thank you!!