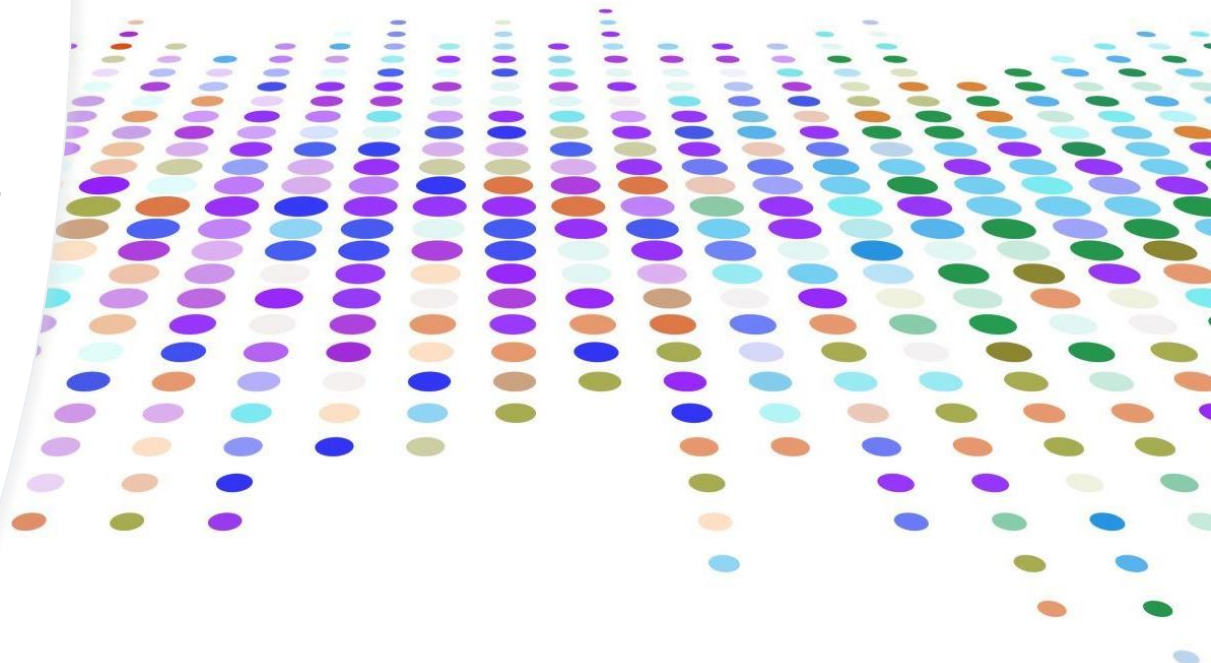




AWS - Serverless Object Detection application

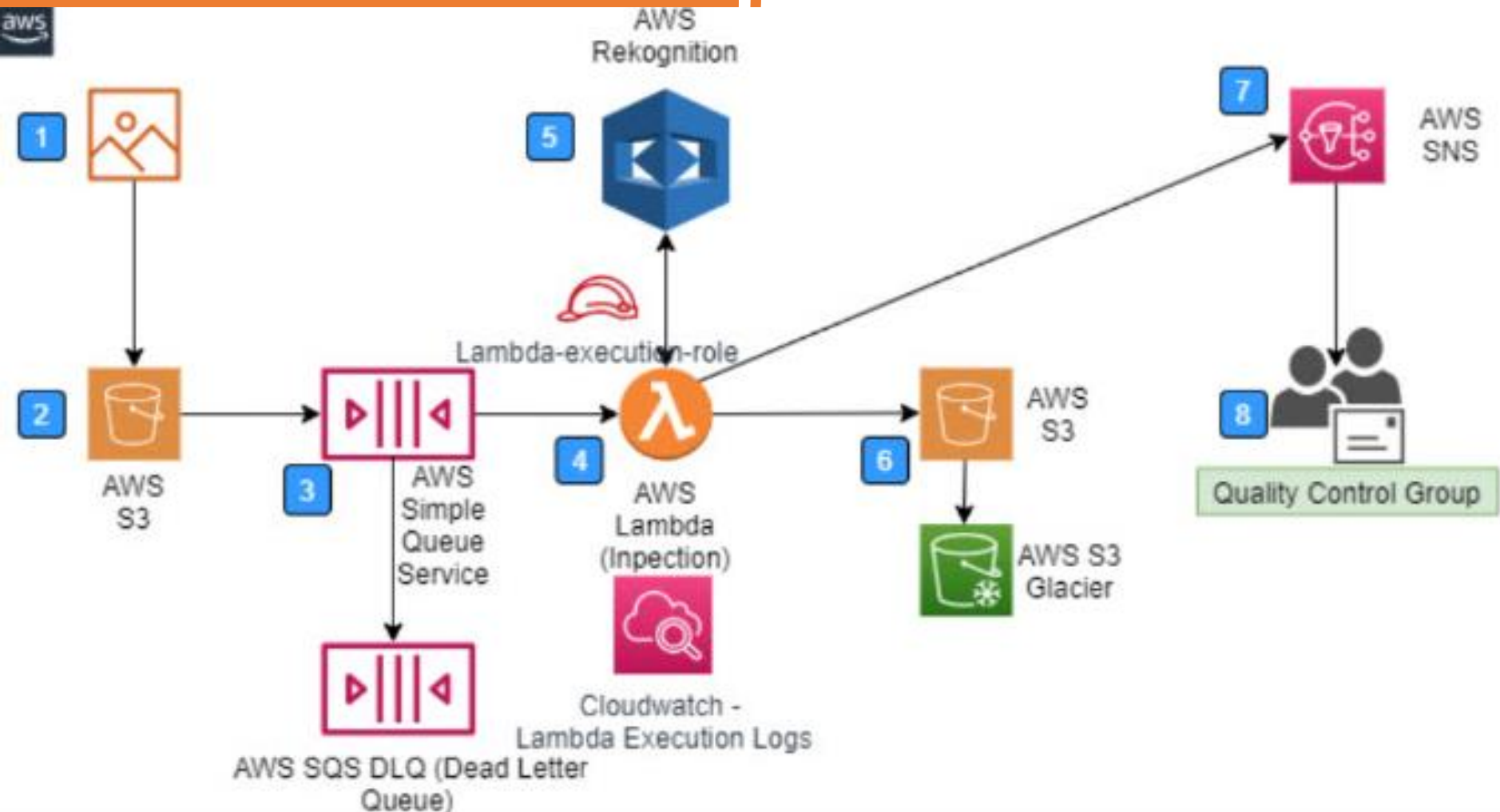
- ITC6450 - Group 4
- Sarina Poth
- Sandeep Samsani
- Jui Ying Lee



Agenda


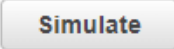


- Explanation of architecture configuration and components:
 - IAM
 - S3
 - SQS
 - Lambda
 - Rekognition
 - CloudWatch
 - SNS
- Demonstration


Overview

















IAM Lambda Role

Permissions policies (7) [Info](#)
You can attach up to 10 managed policies.

< 1 > 

<input type="checkbox"/>	Policy name 	Type	Description
<input type="checkbox"/>	 AWSLambdaBasicExecutionRole-328d5aca-dfd6-4a06-81e7-77a255322cdc	Customer managed	
<input type="checkbox"/>	  CloudWatchFullAccess	AWS managed	Provides full access to CloudWatch logs and metrics.
<input type="checkbox"/>	  CloudWatchLogsFullAccess	AWS managed	Provides full access to CloudWatch Logs.
<input type="checkbox"/>	  AmazonS3FullAccess	AWS managed	Provides full access to Amazon S3.
<input type="checkbox"/>	  AmazonSNSFullAccess	AWS managed	Provides full access to Amazon SNS.
<input type="checkbox"/>	  AmazonSQSFullAccess	AWS managed	Provides full access to Amazon SQS.
<input type="checkbox"/>	  AmazonRekognitionFullAccess	AWS managed	Access to all Amazon Rekognition APIs and endpoints.

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

Amazon S3 > Buckets > aws-acc-cloud-final

aws-acc-cloud-final [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	analyzed/	Folder	-	-	-
<input type="checkbox"/>	images/	Folder	-	-	-

Event notifications (1)

Send a notification when specific events occur in your bucket. [Learn more](#)

[Edit](#) [Delete](#) [Create event notification](#)

<input type="checkbox"/>	Name	Event types	Filters	Destination type	Destination
<input type="checkbox"/>	fileentry	Put	images/	SQS queue	SQSImage

AWS SQS

Access policy (Permissions) [Info](#)

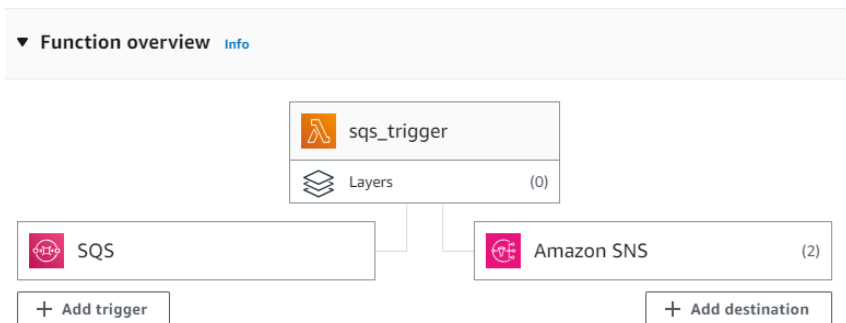
Define who can access your queue.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1687642161023",
  "Statement": [
    {
      "Sid": "Stmt1687642134863",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:*",
      "Resource": "arn:aws:sqs:us-east-1:323708512535:SQSIImage",
      "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

- 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.

```
1 import json
2 import boto3
3
4 s3_client = boto3.client('s3')
5 rekognition_client = boto3.client('rekognition')
6 sns_client = boto3.client('sns')
7 sqs_queue_url = 'https://sqs.us-east-1.amazonaws.com/323708512535/SQSImage'
8 sns_topic_arn = 'arn:aws:sns:us-east-1:323708512535:notify_users'
9
10 def lambda_handler(event, context):
11
12     # Print the event to check its structure
13
14     for record in event['Records']:
15         record = json.loads(record['body'])
16         # Access the S3 bucket and object key from the record
17         s3_bucket = record['Records'][0]['s3']['bucket']['name']
18         s3_object_key = record['Records'][0]['s3']['object']['key']
19         try:
20             # Perform image analysis using AWS Rekognition
21             response = rekognition_client.detect_labels(
22                 Image={
23                     'S3Object': {
24                         'Bucket': s3_bucket,
25                         'Name': s3_object_key
26                     }
27                 },
28                 MaxLabels=10,
29                 MinConfidence=80
30             )
31
32             # Determine if the analysis is successful based on desired labels and confidence threshold
33             desired_labels = ['Widgets', 'Smartphones'] # Add your desired labels here
34             confidence_threshold = 80 # Set your desired confidence threshold
35
36             labels = [label['Name'] for label in response['Labels']]
37             confidences = [label['Confidence'] for label in response['Labels']]
```


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.

Amazon S3 > Buckets > aws-acc-cloud-final > Lifecycle configuration > Final-LifecycleRule

Final-LifecycleRule [Info](#)

[Edit](#) [Delete](#) [Actions](#) ▼

Lifecycle rule configuration

Lifecycle rule name Final-LifecycleRule	Prefix -	Minimum object size -
Status 🟢 Enabled	Object tags -	Maximum object size -
Scope Entire bucket		

Review transition and expiration actions

Current version actions Day 0 <ul style="list-style-type: none">• Objects uploaded ↓ Day 30 <ul style="list-style-type: none">• Objects move to Glacier Instant Retrieval ↓ Day 365 <ul style="list-style-type: none">• Objects expire	Noncurrent versions actions Day 0 <ul style="list-style-type: none">• Objects become noncurrent ↓ Day 30 <ul style="list-style-type: none">• 0 newest noncurrent versions are retained• All other noncurrent versions move to Standard-IA ↓ Day 60 <ul style="list-style-type: none">• 0 newest noncurrent versions are retained• All other noncurrent versions move to Intelligent-Tiering
--	---

Lambda -> SNS

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

The screenshot displays the Amazon SNS console interface. On the left, a sidebar menu includes 'Dashboard', 'Topics', 'Subscriptions', and a 'Mobile' section with 'Push notifications', 'Text messaging (SMS)', and 'Origination numbers'. The main content area is titled 'Amazon SNS > Topics > notify_users'. At the top right of this section are buttons for 'Edit', 'Delete', and 'Publish message'. Below this is a 'Details' section with the following information:

Name	notify_users	Display name	-
ARN	arn:aws:sns:us-east-1:323708512535:notify_users	Topic owner	323708512535
Type	Standard		

Below the details are tabs for 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery policy (HTTP/S)', 'Delivery status logging', 'Encryption', 'Tags', and 'Integrations'. The 'Subscriptions' tab is active, showing a list of 3 subscriptions. At the top of this tab are buttons for 'Edit', 'Delete', 'Request confirmation', 'Confirm subscription', and 'Create subscription'. A search bar is present above the table. The table has columns for ID, Endpoint, Status, and Protocol.

ID	Endpoint	Status	Protocol
3e99b15f-dee9-400d-b776-967f20ecf...	lee.juiy@northeastern.edu	Confirmed	EMAIL
747ede10-7020-4b0d-9748-55e444bd...	samsani.s@northeastern.edu	Confirmed	EMAIL
cc431a7b-3145-4367-b7cc-efcac3339a...	poth.s@northeastern.edu	Confirmed	EMAIL

Live Implementation

- Let's dive into our presentation.

Thank you!!