



**UNIVERSITI TUNKU ABDUL RAHMAN FACULTY OF INFORMATION AND
COMMUNICATION TECHNOLOGY**

UCCD3113 DISTRIBUTED COMPUTER SYSTEMS

Bachelor of Information Technology (Hons) Communications and Networking

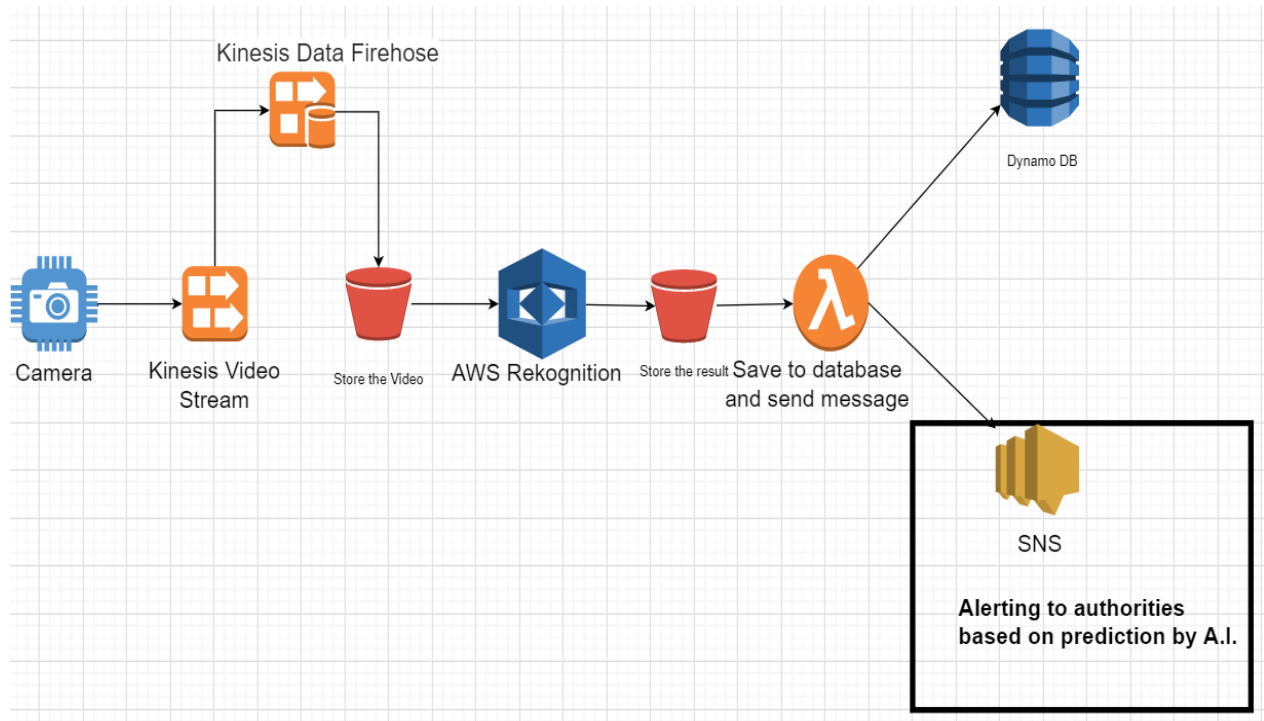
May 2020 Session

Lecturer: Ts Dr. Cheng Wai Khuen

Configuration Manual: Smart Street Light Camera

Name:	ID:	Course
KANNAN A/L DORAJ	1904781	CN
LEE SONG YAO	1905829	CN
VINCENT YIP KAR FAI	1904844	CN
WAN KAR HOU	1607221	CS

Architecture Diagram



Kinesis Video Stream

- Create a Kinesis video stream using the default setting, with the name “Smart_Cam”

The screenshot shows the 'Create a new video stream' page in the AWS Management Console. The page title is 'Create a new video stream' with an 'info' link. Below the title, there is a brief description: 'Create a new video stream and then use the Kinesis Video Streams API to put data into or read data from your video stream. Kinesis Video Streams resources are not covered under the AWS Free Tier, and usage-based charges apply. For more information, see Kinesis Video Streams pricing.' The 'Setup' section contains a 'Video stream name' field with the value 'Smart_Cam'. Below this, there are two radio buttons: 'Default configuration' (selected) and 'Custom configuration'. The 'Default configuration' section shows 'Data retention' set to '1 day(s)' and 'KMS customer master key (CMK)' set to '(Default) aws/kinesisvideo'. A blue information box at the bottom states: 'The KMS customer master key (CMK) used for encryption cannot be changed later. Kinesis Video Streams use the AWS managed CMK (aws/kinesisvideo) to encrypt your data by default.'

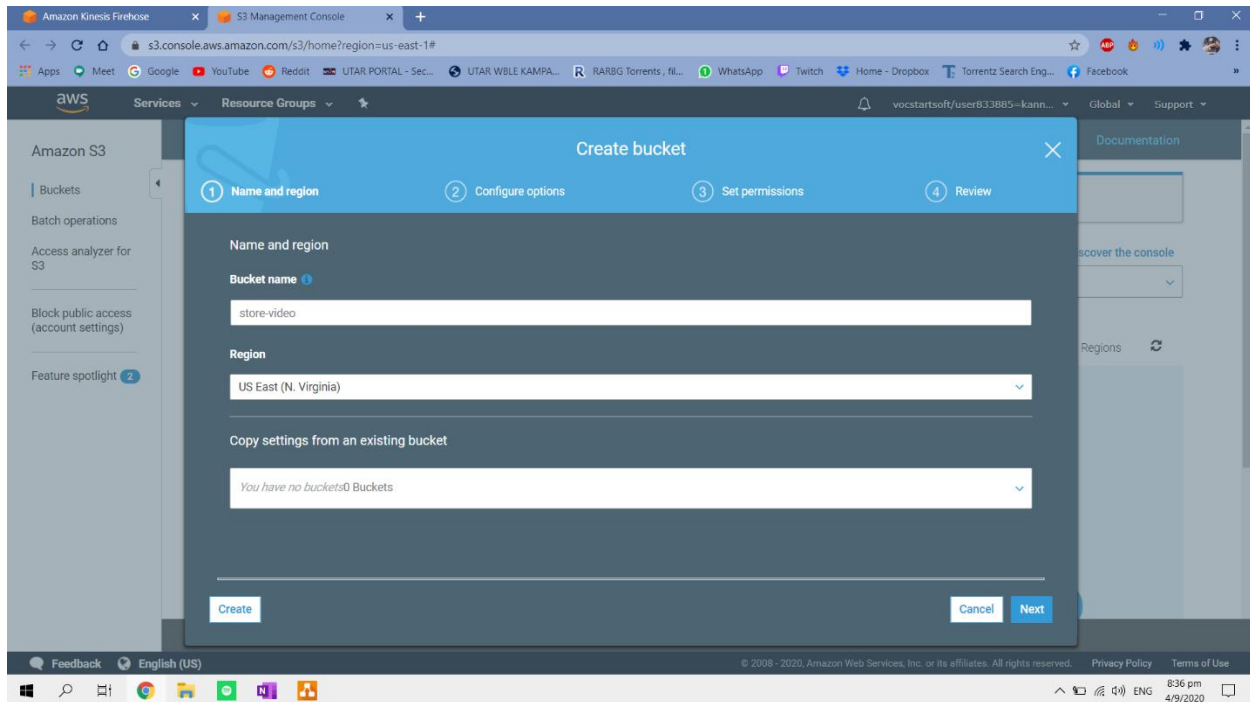
- For the prototype, we use the webcam as our smart camera

The screenshot shows the 'Smart_Cam' video stream details page in the AWS Management Console. The page title is 'Smart_Cam' with an 'info' link and a 'Delete video stream' button. Below the title, there is a description: 'Use this page to view and configure your Kinesis video stream and to download clips from your video stream.' The page is divided into two main sections: 'Connect video stream' and 'Media playback'. The 'Connect video stream' section has a 'Set up your producer' section with a 'Download SDK' button. The 'Media playback' section is currently collapsed. Below these sections, there are tabs for 'Video stream info', 'Monitoring', 'Encryption', 'Data retention', and 'Tags'. The 'Video stream info' tab is active, showing a table with the following information:

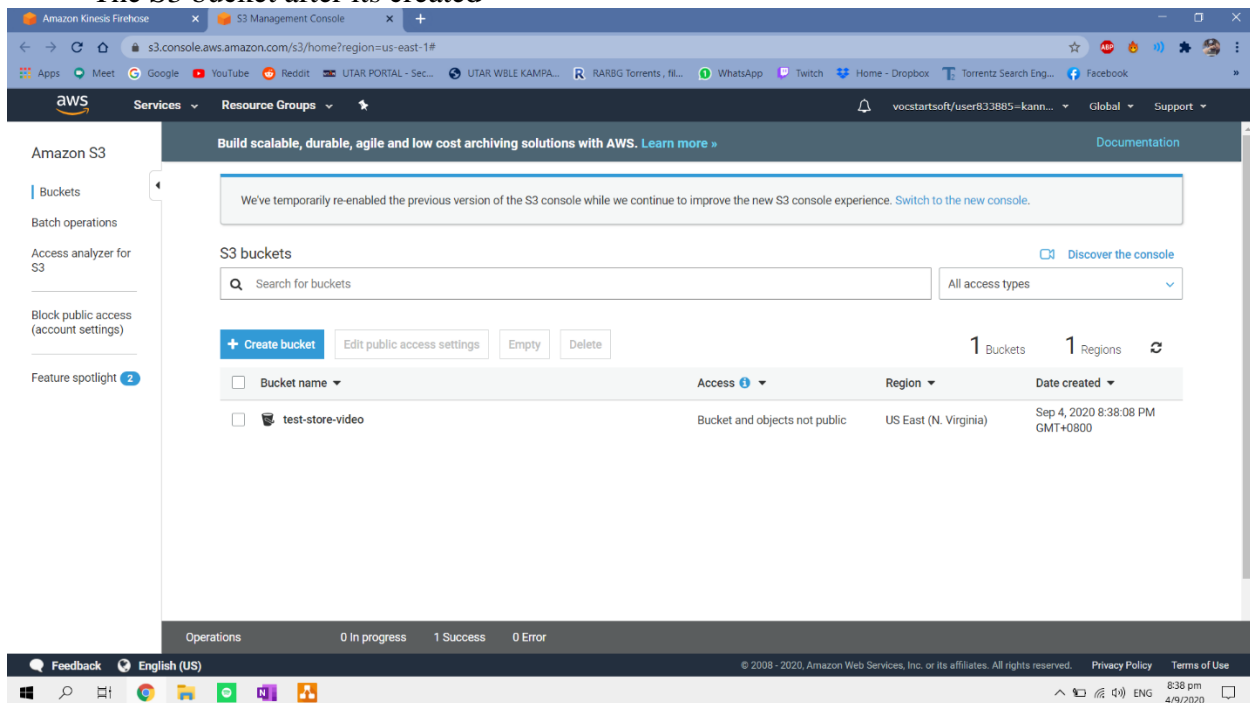
Video stream info	
Video stream name	Status
Smart_Cam	Active
Video stream ARN	Creation time
	Created at: 2020-06-15 10:00:00

Amazon S3 Bucket (Store the Video)

- Using the default settings and naming the S3 bucket as store-video



- The S3 bucket after its created



Kinesis Data Firehose

- Using Firehose as our delivery system to transfer the video files to s3. Using the same name “Smart_Cam”. And select Direct PUT as the source.

Step 1: Name and source

Step 2: Process records

Step 3: Choose a destination

Step 4: Configure settings

Step 5: Review

New delivery stream

Delivery streams load data, automatically and continuously, to the destinations that you specify. Kinesis Data Firehose resources are not covered under the [AWS Free Tier](#), and **usage-based charges apply**. For more information, see [Kinesis Data Firehose pricing](#). [Learn more](#)

Delivery stream name

Smart_Cam

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Choose a source

Choose how you would prefer to send records to the delivery stream.

Firehose data flow overview

```
graph LR; Source[Source] --> Firehose[Firehose delivery stream]; Firehose --> Destination[Destination]; subgraph Firehose; direction LR; SR[Source records] --> PR[Processed records]; end
```

----- Optional

Source

To learn about enabling server-side encryption (SSE), see [Data Protection in Amazon Kinesis Data Firehose](#).

☒ Direct PUT or other sources

Choose this option to send records directly to the delivery stream, or to send records from AWS IoT, CloudWatch Logs, or CloudWatch Events.

- Using standard options given by the AWS firehose

Step 1: Name and source

Step 2: Process records

Step 3: Choose a destination

Step 4: Configure settings

Step 5: Review

Process records

Kinesis Data Firehose can transform records or convert record format before delivery.

Process records data flow overview

```
graph LR; Source[Source records] --> Transform[Transform source records]; Transform --> Convert[Convert record format]; Convert --> Destination[Destination]; subgraph Transform; direction LR; T1[Invoke AWS Lambda function] --> T2[Refer to AWS Glue table for schema]; end
```

----- Optional

Transform source records with AWS Lambda

To return records from AWS Lambda to Kinesis Data Firehose after transformation, the Lambda function you invoke must be compliant with the required record transformation output model. [Learn more](#)

Data transformation

☒ Disabled

☐ Enabled

Convert record format

Data in Apache Parquet or Apache ORC format is typically more efficient to query than JSON. Kinesis Data Firehose can convert your JSON-formatted source records using a schema from a table defined in [AWS Glue](#). For records that aren't in JSON format, create a Lambda function that converts them to JSON in the [Transform source records with AWS Lambda](#) section above. [Learn more](#)

Record format conversion

☒ Disabled

☐ Enabled

If record format conversion is enabled, Kinesis Data Firehose can deliver data to Amazon S3 only. Record format conversion will be configured using the OpenX JSON SerDe. For other options use the [AWS CLI](#).

- Select the destination for the files to be transported to, in our case it's a Amazon s3 storage

Kinesis Data Firehose - Create delivery stream

Step 1: Name and source

Step 2: Process records

Step 3: Choose a destination

Step 4: Configure settings

Step 5: Review

Choose a destination

[Learn more](#)

Destination

- ☒ **Amazon S3**
Object storage built to store and retrieve any amount of data from anywhere.
- ☐ **Amazon Redshift**
An enterprise-level, petabyte scale, fully managed data warehousing service.
- ☐ **Amazon Elasticsearch**
An open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and click stream analytics.
- ☐ **HTTP Endpoint**
A way to deliver data to your custom destination.
- ☐ **Third-party service provider**
Choose from a list of third-party service providers.

S3 destination

Choose a destination in Amazon S3 where your data will be stored. Amazon S3 is object storage built to store and retrieve any amount of data from anywhere.
[Learn more](#)

Firehose to S3 data flow overview

```

graph LR
    Source[Source] --> Firehose[Firehose delivery stream]
    subgraph Firehose
        SR[Source records] --> PR[Processed records (Optional)]
    end
    Firehose --> S3[S3 bucket]
  
```

- And select the Amazon S3 storage that have been created

S3 bucket

test-store-video [View test-store-video in S3 console](#)

[Create new](#)

S3 prefix

By default, Kinesis Data Firehose appends the prefix "YYYY/MM/dd/HH" (in UTC) to the data it delivers to Amazon S3. You can override this default by specifying a custom prefix that includes expressions that are evaluated at runtime.

If your custom prefix doesn't include expressions, Kinesis Data Firehose uses your prefix and appends "YYYY/MM/dd/HH". If your custom prefix includes a Kinesis Data Firehose random string or timestamp expression, Kinesis Data Firehose doesn't append "YYYY/MM/dd/HH". [Learn more](#)

Backup S3 bucket prefix - optional

Enter a prefix

S3 error prefix

You can specify an S3 bucket prefix to be used in error conditions. This prefix can include expressions for Kinesis Data Firehose to evaluate at runtime. [Learn more about the rules for specifying prefix expressions](#)

Backup S3 bucket error prefix - optional

Enter a prefix

- The configure settings will be set to default as given by the Kinesis firehose

Kinesis Data Firehose Create delivery stream

Step 1: Name and source
Step 2: Process records
Step 3: Choose a destination
Step 4: Configure settings
Step 5: Review

Configure settings

Configure buffer, compression, logging, and IAM role settings for your delivery stream. [Learn more](#)

S3 buffer conditions

Kinesis Data Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery will be triggered once either of these conditions has been satisfied. [Learn more](#)

Buffer size
 MiB
 Enter a buffer size between 1 - 128 MiB.

Buffer interval
 seconds
 Enter a buffer interval between 60 - 900 seconds.

S3 compression and encryption

Kinesis Data Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using a KMS master key. [Learn more](#)

S3 compression
☒ Disabled
☐ GZIP
☐ Snappy
☐ Zip

☐ Hadoop-Compatible Snappy

S3 encryption
☒ Disabled
☐ Enabled

Error logging

Kinesis Data Firehose can log record delivery errors to CloudWatch Logs. If enabled, a CloudWatch log group and corresponding log streams are created on your behalf. [Learn more](#)

Error logging
☐ Disabled
☒ Enabled

Tags - optional

You can add tags to organize your AWS resources, track costs, and control access. [Learn more](#)

Key Value - optional

You can add 49 more tag(s)

- Testing the Kinesis Firehose and the S3 by sending demo file

Amazon Kinesis

Kinesis Data Firehose delivery streams > Smart_Cam

Smart_Cam Delete delivery stream

▼ Test with demo data

This test runs a script in your browser to put demo data in your Kinesis Data Firehose delivery stream, which sends to your S3 destination. The format of the demo data is {"ticker_symbol": "QXZ", "sector": "HEALTHCARE", "change": -0.05, "price": 84.51}

Step 1
Start sending demo data to your delivery stream. If you already have data streaming to this destination, demo data is sent along with your source records.

Demo data is being sent to your delivery stream
If the test is successful, you will see demo records in this destination [test-store-video](#)

⏮ Sending demo data

Step 2
Stop sending demo data to your delivery stream after you've concluded your test.

Stop sending demo data

- The demo file from the Kinesis Firehose in S3 bucket

test-store-video

Overview Properties Permissions Management Access points

🔍 Type a prefix and press Enter to search. Press ESC to clear.

Upload + Create folder Download Actions Versions Hide Show US East (N. Virginia)

Viewing 1 to 1			
<input type="checkbox"/> Name	Last modified	Size	Storage class
<input type="checkbox"/> 2020	-	-	-

Viewing 1 to 1

Identity and Access Management

- Create new policy for certain services in IAM

The screenshot shows the AWS IAM 'Create policy' page. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile. The page title is 'Create policy' with two numbered steps (1 and 2). A description states: 'A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)'. Below this are tabs for 'Visual editor' (selected) and 'JSON', and a link for 'Import managed policy'. A sidebar on the left is labeled 'Documentation'. The main content area has a header 'Select a service' with 'Clone' and 'Remove' links. Below this are four sections: 'Service' with a 'Choose a service' link, 'Actions' with the instruction 'Choose a service before defining actions', 'Resources' with 'Choose actions before applying resources', and 'Request conditions' with 'Choose actions before specifying conditions'.

- Allow access to all resources in Amazon S3 in the policy

This screenshot shows the configuration of the policy in the 'Visual editor'. It features two policy statements. The first statement is for 'S3 (All actions)', with 'Clone' and 'Remove' links. It is configured with 'Service' set to 'S3', 'Actions' set to 'Manual actions', and 'Resources' set to 'All resources' (selected via radio button, with a 'close' link). The second statement is for 'SNS (All actions)', also with 'Clone' and 'Remove' links, configured with 'Service' set to 'SNS' and 'Actions' set to 'Manual actions'. A vertical scrollbar is visible on the right side of the configuration area.

- Allow access to all resources in Dynamo DB for the policy

The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'vocstartsoft/user833899=wank...'. A left sidebar contains a 'Documentation' link. The main content area displays the configuration for a policy:

- Request conditions:** A link to 'Specify request conditions (optional)'.
- DynamoDB (All actions):** A section header with 'Clone' and 'Remove' links.
- Service:** Set to 'DynamoDB'.
- Actions:** Set to 'Manual actions'.
- Resources:** Radio buttons for 'Specific' and 'All resources'. The 'All resources' option is selected, and a 'close' link is visible.
- Request conditions:** A link to 'Specify request conditions (optional)'.

At the bottom right of the configuration area, there is a button labeled 'Add additional permissions'.

- Allow access to all resources in Amazon SNS for the policy

The screenshot shows the AWS IAM console interface for configuring a policy for Amazon SNS. The top navigation bar and left sidebar are identical to the previous screenshot. The main content area displays the configuration for a policy:

- Actions:** Set to 'Manual actions'.
- Resources:** Radio buttons for 'Specific' and 'All resources'. The 'All resources' option is selected, and a 'close' link is visible.
- Request conditions:** A link to 'Specify request conditions (optional)'.
- SNS (All actions):** A section header with 'Clone' and 'Remove' links.
- Service:** Set to 'SNS'.
- Actions:** Set to 'Manual actions'.
- Resources:** Radio buttons for 'Specific' and 'All resources'. The 'All resources' option is selected, and a 'close' link is visible.
- Request conditions:** A link to 'Specify request conditions (optional)'.

- Allow access for all resources in CloudWatch for the new Policy, and press “Review policy” button, skip the add tags, enter “s3_json_dynamodb” as the name of policy and press “Create policy” button

Documentation

aws Services Resource Groups

CloudWatch (All actions) Clone Remove

Service CloudWatch

Actions Manual actions

Resources ☐ Specific ☒ All resources close

Request conditions Specify request conditions (optional)

+ Add additional permissions

Character count: 118 of 6,144.

Cancel Review policy

- “s3_json_dynamodb” successfully created

aws Services Resource Groups

Identity and Access Management (IAM)

Policies > s3_json_dynamodb

Summary Delete policy

Policy ARN: arn:aws:iam::836295026856:policy/s3_json_dynamodb

Description

Permissions Policy usage Policy versions Access Advisor

Policy summary {} JSON Edit policy

Filter

Service	Access level	Resource	Request condition
Allow (4 of 238 services) Show remaining 234			
CloudWatch Logs	Full access	All resources	None
DynamoDB	Full access	All resources	None
S3	Full access	All resources	None
SNS	Full access	All resources	None

Dashboard

Access management

Groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCP)

- To create Role for Lambda, go to “Role” under the “access management”, to select Lambda and press “Next Permission” button.

The screenshot shows the 'Create role' page in the AWS IAM console. The page has a header with the AWS logo and navigation links. The main content area is titled 'Create role' and has a progress indicator with four steps: 1 (selected), 2, 3, and 4. The first step is 'Select type of trusted entity'. Below this, there are four options: 'AWS service' (selected), 'Another AWS account', 'Web identity', and 'SAML 2.0 federation'. The 'AWS service' option is highlighted with a blue border. Below the options, there is a section titled 'Choose a use case'. Under 'Common use cases', there are two options: 'EC2' and 'Lambda'. The 'Lambda' option is highlighted with a blue background. Below this, there is a section titled 'Or select a service to view its use cases'. This section contains a grid of service names: API Gateway, AWS Backup, AWS Chatbot, AWS Support, Amplify, AppStream 2.0, CodeDeploy, CodeGuru, CodeStar Notifications, Comprehend, Config, Connect, EMR, ElastiCache, Elastic Beanstalk, Elastic Container Service, Elastic Transcoder, ElasticLoadBalancing, KMS, Kinesis, Lake Formation, Lambda, Lex, License Manager, Rekognition, RoboMaker, S3, SMS, SNS, and SWF. At the bottom of the page, there is a 'Cancel' button and a 'Next: Permissions' button.

- Select “s3_json_dynamodb” “created” as the policy of the role, then press “Next: Tags” button.

The screenshot shows the 'Create role' page in the AWS IAM console, step 2: 'Attach permissions policies'. The page has a header with the AWS logo and navigation links. The main content area is titled 'Create role' and has a progress indicator with four steps: 1, 2 (selected), 3, and 4. The second step is 'Attach permissions policies'. Below this, there is a section titled 'Choose one or more policies to attach to your new role.' There is a 'Create policy' button and a 'Filter policies' dropdown menu. The 'Filter policies' dropdown menu is open, showing a search bar with the text 's3' and a 'Showing 5 results' label. Below the search bar, there is a table with two columns: 'Policy name' and 'Used as'. The table contains five rows: 'AmazonDMSRedshiftS3Role' (Used as: None), 'AmazonS3FullAccess' (Used as: Permissions policy (2)), 'AmazonS3ReadOnlyAccess' (Used as: None), 'QuickSightAccessForS3StorageManagementAnalyticsReadOnly' (Used as: None), and 's3_json_dynamodb' (Used as: Permissions policy (1)). The 's3_json_dynamodb' row is selected with a blue background. Below the table, there is a section titled 'Set permissions boundary'. At the bottom of the page, there is a 'Cancel' button, a 'Previous' button, and a 'Next: Tags' button.

- Ignore this part as it is optional, press “Next: Review” button

The screenshot shows the AWS IAM console 'Create role' page, step 3 of 4. The page title is 'Create role'. Below the title, it says 'Add tags (optional)'. A sub-header explains: 'IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)'. Below this is a table with two columns: 'Key' and 'Value (optional)'. The 'Key' column has a text input field with the placeholder 'Add new key'. The 'Value (optional)' column has an empty text input field. A 'Remove' button is located to the right of the 'Value (optional)' column. Below the table, it says 'You can add 50 more tags.' At the bottom of the page, there are three buttons: 'Cancel', 'Previous', and 'Next: Review'.

- Enter role name “s3_json_dynamodb_role” as Role name and press “Create role” button

The screenshot shows the AWS IAM console 'Create role' page, step 4 of 4. The page title is 'Create role'. Below the title, it says 'Review'. A sub-header says: 'Provide the required information below and review this role before you create it.' Below this is a form with several fields: 'Role name*' with the value 's3_json_dynamodb_role' and a note 'Use alphanumeric and *+,@,_,. characters. Maximum 64 characters.'; 'Role description' with the value 'Allows Lambda functions to call AWS services on your behalf.' and a note 'Maximum 1000 characters. Use alphanumeric and *+,@,_,. characters.'; 'Trusted entities' with the value 'AWS service: lambda.amazonaws.com'; 'Policies' with the value 's3_json_dynamodb' and a link icon; and 'Permissions boundary' with the value 'Permissions boundary is not set'. At the bottom of the page, there are three buttons: 'Cancel', 'Previous', and 'Create role'. A note at the bottom left says 'No tags were added.'

- Role successfully creates

The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with options like Dashboard, Access management, Groups, Users, Roles (selected), Policies, Identity providers, Account settings, Access reports, Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, and Service control policies (SCPs). The main content area is titled 'Roles > s3_json_dynamodb_role Summary'. It displays the role's ARN, description ('Allows Lambda functions to call AWS services on your behalf'), instance profile ARNs, path, creation time, last activity, and maximum session duration. Below this, there are tabs for Permissions, Trust relationships, Tags, Access Advisor, and Revoke sessions. The 'Permissions' tab is active, showing 'Permissions policies (1 policy applied)' and a table with one entry: 's3_json_dynamodb' (Managed policy). There is also a section for 'Permissions boundary (not set)'.

- When json output file of amazon Rekognition passed from amazon Rekognition to s3 bucket and processed by lambda. You can go to Log-groups and select “s3-json-dynamodb” to check for status of it. (Part 1)

The screenshot shows the AWS CloudWatch Logs console. The left sidebar lists various services and log groups. The 'Log groups' section is expanded, showing a list of log groups. The 's3-json-dynamodb' log group is selected. The main area displays a list of log events. Each event includes a timestamp, a message ID, and the log content. The log content shows JSON data representing Rekognition results, including object names, confidence scores, and bounding boxes. For example, one event shows a 'Freeway' with a confidence of 98.89735412597656. Another event shows a 'Solar Panels' with a confidence of 67.94845581054688.

- When json output file of amazon Rekognition passed from amazon Rekognition to s3 bucket and processed by lambda. You can go to Log-groups and select “s3-json-dynamodb” to check for status of it. (Part 2)

The screenshot shows the AWS Management Console interface. On the left, the navigation menu includes sections like CloudWatch, Dashboards, Alarms, and Log groups. The 'Log groups' section is expanded, showing a list of log groups. The 's3-json-dynamodb' log group is selected, and its contents are displayed in the main area. The log events are shown as a table with columns for Time, Message, and Details. The details column contains JSON data representing Rekognition API responses, such as object names (e.g., 'Car', 'Electrical Device', 'Freeway', 'Highway', 'Road', 'Solar Panels', 'Tarmac', 'Transportation', 'Vehicle', 'Asphalt', 'Automobile') and their corresponding confidence scores and bounding boxes.

Amazon S3 Bucket (Store the Results)

- Create a bucket with name “s3-dynamodb-bucket”

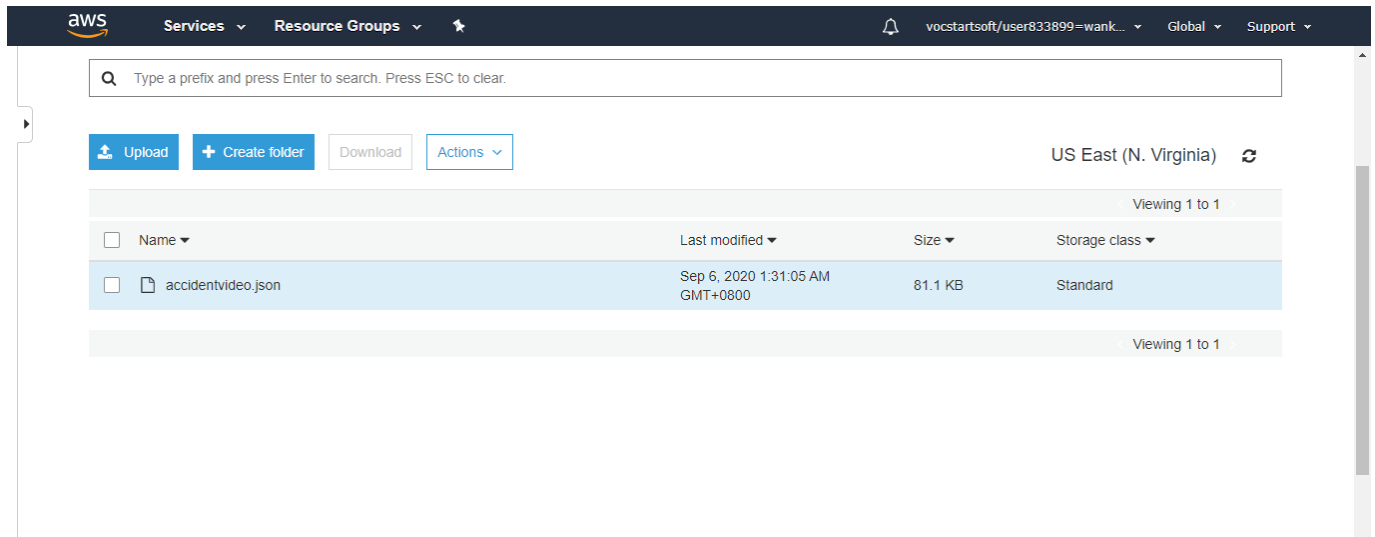
The screenshot shows the AWS 'Create bucket' wizard. The first step, 'Name and region', is active. The 'Bucket name' field contains 's3-dynamodb-bucket'. The 'Region' dropdown is set to 'US East (N. Virginia)'. There is a section for 'Copy settings from an existing bucket' with a dropdown menu. At the bottom, there are 'Create', 'Cancel', and 'Next' buttons.

- “s3-dynamodb-bucket” successfully created

The screenshot shows the AWS S3 console. A search bar at the top allows searching for buckets. Below the search bar, there are buttons for '+ Create bucket', 'Edit public access settings', 'Empty', and 'Delete'. On the right, it shows '5 Buckets' and '1 Regions'. The main table lists the buckets with columns for 'Bucket name', 'Access', 'Region', and 'Date created'.

<input type="checkbox"/>	Bucket name	Access	Region	Date created
<input type="checkbox"/>	image-rekog-lambda2020	Bucket and objects not public	US East (N. Virginia)	Aug 15, 2020 11:59:52 AM GMT+0800
<input type="checkbox"/>	javahome-json-lambda	Bucket and objects not public	US East (N. Virginia)	Aug 25, 2020 10:16:58 AM GMT+0800
<input type="checkbox"/>	rekognition-video-console-demo-iad-836295026856-jg74jvds2u...	Objects can be public	US East (N. Virginia)	Aug 31, 2020 3:35:24 AM GMT+0800
<input type="checkbox"/>	s3-dynamodb-bucket	Bucket and objects not public	US East (N. Virginia)	Sep 6, 2020 1:06:41 AM GMT+0800
<input type="checkbox"/>	video-rekognition-bucket	Bucket and objects not public	US East (N. Virginia)	Sep 5, 2020 8:18:30 PM GMT+0800

- After Amazon Rekognition video finish analyzing the video of accident, it will pass the output of it, which is in json format to “s3-dynamodb-bucket”



- Top view for the complete json output of amazon Rekognition.



- Bottom view for the complete json output of amazon Rekognition.

```
{
  "Name": "Tarmac",
  "Confidence": 73.67108154296875,
  "Instances": [],
  "Parents": [],
  "Timestamp": 28495,
  "Label": {
    "Name": "Trailer Truck",
    "Confidence": 67.05509948730469,
    "Instances": [],
    "Parents": [
      {
        "Name": "Truck",
        "Confidence": 75.09610748291016,
        "Instances": [],
        "Parents": []
      }
    ],
    "Timestamp": 28495,
    "Label": {
      "Name": "Truck",
      "Confidence": 71.2304916381836,
      "Instances": [
        {
          "Width": 0.28215572237968445,
          "Height": 0.5335927605628967,
          "Left": 0.03306322917342186,
          "Top": 0.20724649727344513,
          "Confidence": 57.938514709472656,
          "Parents": [
            {
              "Name": "Vehicle",
              "Confidence": 74.87419128417969,
              "Instances": [],
              "Parents": [
                {
                  "Name": "Transportation",
                  "Confidence": 77.51483917236328,
                  "Instances": [],
                  "Parents": []
                }
              ]
            }
          ]
        }
      ],
      "Timestamp": 28995,
      "Label": {
        "Name": "Automobile",
        "Confidence": 68.29138946533203,
        "Instances": [],
        "Parents": [
          {
            "Name": "Vehicle",
            "Confidence": 86.51390838623047,
            "Instances": [],
            "Parents": [
              {
                "Name": "Freeway",
                "Confidence": 89.65359497070312,
                "Instances": [],
                "Parents": [
                  {
                    "Name": "Road",
                    "Confidence": 99.26083374023438,
                    "Instances": [],
                    "Parents": []
                  }
                ]
              }
            ]
          }
        ],
        "Timestamp": 28995,
        "Label": {
          "Name": "Car",
          "Confidence": 68.29138946533203,
          "Instances": [
            {
              "Width": 0.16644534468650818,
              "Height": 0.14850907027721405,
              "Left": 0.47972798347473145,
              "Top": 0.30837637186050415,
              "Confidence": 68.42882537841797,
              "Parents": [
                {
                  "Name": "Vehicle",
                  "Confidence": 89.65359497070312,
                  "Instances": [],
                  "Parents": [
                    {
                      "Name": "Transportation",
                      "Confidence": 86.51390838623047,
                      "Instances": [],
                      "Parents": [
                        {
                          "Name": "Freeway",
                          "Confidence": 89.65359497070312,
                          "Instances": [],
                          "Parents": [
                            {
                              "Name": "Road",
                              "Confidence": 99.26083374023438,
                              "Instances": [],
                              "Parents": []
                            }
                          ]
                        }
                      ]
                    }
                  ]
                }
              ],
              "Timestamp": 28995,
              "Label": {
                "Name": "Moving Van",
                "Confidence": 67.7796401977539,
                "Instances": [],
                "Parents": [
                  {
                    "Name": "Van",
                    "Confidence": 84.50978088378906,
                    "Instances": [],
                    "Parents": [
                      {
                        "Name": "Transportation",
                        "Confidence": 79.80045318603516,
                        "Instances": [],
                        "Parents": []
                      }
                    ]
                  }
                ],
                "Timestamp": 29496,
                "Label": {
                  "Name": "Automobile",
                  "Confidence": 72.97946166992188,
                  "Instances": [],
                  "Parents": [
                    {
                      "Name": "Vehicle",
                      "Confidence": 58.25628662109375,
                      "Instances": [],
                      "Parents": [
                        {
                          "Name": "Trailer Truck",
                          "Confidence": 58.25628662109375,
                      "Instances": [],
                      "Parents": [
                        {
                          "Name": "Transportation",
                          "Confidence": 88.20211791992188,
                          "Instances": [],
                          "Parents": []
                        }
                      ]
                        }
                      ]
                    }
                  ]
                }
              ],
              "Timestamp": 29496,
              "Label": {
                "Name": "Truck",
                "Confidence": 63.37394332885742,
                "Instances": [
                  {
                    "Width": 0.41914063692092896,
                    "Height": 0.3379736840724945,
                    "Left": 0.45683905482292175,
                    "Top": 0.15388596057891846,
                    "Confidence": 53.25258255004883,
                    "Parents": [
                      {
                        "Name": "Vehicle",
                        "Confidence": 53.15729522705078,
                        "Instances": [],
                        "Parents": [
                          {
                            "Name": "Transportation",
                            "Confidence": 74.12775421142578,
                            "Instances": [],
                            "Parents": [
                              {
                                "Name": "Van",
                                "Confidence": 74.12775421142578,
                                "Instances": [],
                                "Parents": [
                                  {
                                    "Name": "Transportation",
                                    "Confidence": 87.92950439453125,
                                    "Instances": [],
                                    "Parents": []
                                  }
                                ]
                                  }
                                ]
                              }
                            ]
                          }
                        ]
                      }
                    ]
                  }
                ],
                "Timestamp": 29496,
                "Label": {
                  "Name": "Vehicle",
                  "Confidence": 87.92950439453125,
                  "Instances": [],
                  "Parents": []
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

AWS Lambda

- Select “Author from scratch “to create lambda

The screenshot shows the AWS Lambda 'Create function' page. The 'Author from scratch' option is selected. The 'Basic information' section is visible, showing the 'Function name' field with the value 'myFunctionName' and the 'Runtime' dropdown set to 'Node.js 12.x'.

- Enter function name “ s3-json-dynamodb-lambda “, select “Python 3.8 “ as runtime , click “ use existing role “ and select “ s3-json-dynamodb-role “ as the existing role , then press “Create function “ button to create lambda function.

This screenshot shows the 'Create function' page with the following details: Function name is 's3_json_dynamodb_lambda', Runtime is 'Python 3.8', and the 'Use an existing role' option is selected with 's3_json_dynamodb_role' chosen from the dropdown. The 'Create function' button is highlighted in orange at the bottom right.

- Code configuration in lambda. Firstly, get the json output file of amazon Rekognition from the s3-dynamodb-bucket, convert it to python dictionary. After that, remove 'Instances' and 'Parents' from the 'Labels' in the python dictionary, and add in 'Timestamp' and 'id' to the 'Labels'. At this point, there are 'id', 'Name', 'Label', 'Timestamp' at the 'Labels'. Store all 'Labels' to DynamoDB- 'LabelDetected'. Finally, all these 'Labels' data will be sent to specific email via SNS.

```

1 import boto3
2 import json
3 import math
4 s3_client=boto3.client('s3')
5 client=boto3.client('sns')
6 dynamodb = boto3.resource('dynamodb')
7 def lambda_handler(event, context):
8     bucket = event['Records'][0]['s3']['bucket']['name']
9     json_file_name=event['Records'][0]['s3']['object']['key']
10    json_object= s3_client.get_object(Bucket=bucket,key=json_file_name)
11    jsonFileReader=json_object['Body'].read()
12    jsonDict = json.loads(jsonFileReader)
13
14    for i in range(25):
15        print(jsonDict['Labels'][i]['Timestamp'])
16        print(jsonDict['Labels'][i]['Label'])
17        jsonDict['Labels'][i]['Label']['Confidence'] = math.floor( jsonDict['Labels'][i]['Label']['Confidence'])
18        del jsonDict['Labels'][i]['Label']['Instances']
19        del jsonDict['Labels'][i]['Label']['Parents']
20        jsonDict['Labels'][i]['Label']['Timestamp']=jsonDict['Labels'][i]['Timestamp']
21        jsonDict['Labels'][i]['Label']['id']=i+1
22        table= dynamodb.Table('LabelDetected')
23        table.put_item(item=jsonDict['Labels'][i]['Label'])
24
25    tosend =""
26    string_concat=""
27    for i in range(25):
28        print(jsonDict['Labels'][i]['Label'])
29        tosend+=str(jsonDict['Labels'][i]['Label'])
30
31    message=client.publish(TargetArn='arn:aws:sns:us-east-1:836295026856:video-lambda-sns',
32    Message=tosend,Subject='Detected Labels in Video')
33    return 'Hello from Lambda'

```

- Add in Amazon s3 bucket – “s3-dynamodb- bucket” as trigger for “s3-json-dynamodb-lambda”, select “All object create events “and input “. json “for Suffix-optional.

- Tick the Recursive invocation and press “Add” button to create the trigger

The screenshot shows the 'Add trigger' dialog in the AWS Lambda console. The 'Prefix - optional' field contains 'e.g. images/' and the 'Suffix - optional' field contains '.json'. The 'Enable trigger' checkbox is checked. Below this, there is a section for 'Recursive invocation' with an information icon and a warning: 'If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more'. A checkbox below this warning is also checked, stating: 'I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.' At the bottom right of the dialog are 'Cancel' and 'Add' buttons.

- The trigger was successfully created.

The screenshot shows the configuration page for the Lambda function 's3_json_dynamodb_lambda'. The 'Configuration' tab is selected. In the 'Designer' section, there is a visual representation of the function's configuration. It shows a box for 's3_json_dynamodb_lambda' with a 'Layers' section below it containing '(0)'. To the left of this box is a box for 'S3' with a '+ Add trigger' button below it. To the right of the 's3_json_dynamodb_lambda' box is a '+ Add destination' button. At the top right of the configuration area are buttons for 'Throttle', 'Qualifiers', 'Actions', 'Select a test event', 'Test', and 'Save'.

DynamoDB

- Create Table ‘LabelDetected’ with id (Number) as primary key in DynamoDB, press “Create” button to create the table.

The screenshot shows the 'Create DynamoDB table' wizard in the AWS Management Console. The 'Table name' is 'LabelDetected'. The 'Primary key' is 'id' with a data type of 'Number'. The 'Add sort key' checkbox is unchecked. Under 'Table settings', 'Use default settings' is selected. A message states: 'You do not have the required role to enable Auto Scaling by default. Please refer to documentation.' The 'Create' button is visible at the bottom right.

- “LabelDetected” table successfully created (PART 1)

The screenshot shows the AWS Management Console with the 'LabelDetected' table selected. The 'Items' tab is active, showing a search bar and a 'Start search' button. A message states: 'An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key. More info'. The table is currently empty, showing '0 to 0 items'.

- “LabelDetected” table successfully created (PART 2)

The screenshot shows the AWS Management Console interface for the 'LabelDetected' table in DynamoDB. The left sidebar contains navigation links for DynamoDB, Tables, Backups, Reserved capacity, Preferences, DAX, Dashboard, Clusters, Subnet groups, Parameter groups, and Events. The main content area displays the 'LabelDetected' table details, including Stream details and Table details.

Stream details:

- Stream enabled: No
- View type: -
- Latest stream ARN: -
- Manage Stream button

Table details:

- Table name: LabelDetected
- Primary partition key: id (Number)
- Primary sort key: -
- Point-in-time recovery: DISABLED [Enable](#)
- Encryption Type: DEFAULT [Manage Encryption](#)
- KMS Master Key ARN: Not Applicable
- Encryption Status: -
- CloudWatch Contributor Insights: DISABLED [Manage Contributor Insights](#) **NEW**
- Time to live attribute: DISABLED [Manage TTL](#)
- Table status: Active
- Creation date: September 6, 2020 at 1:20:01 AM UTC+8
- Read/write capacity mode: Provisioned
- Last change to on-demand mode: -
- Provisioned read capacity units: 5 (Auto Scaling Error)
- Provisioned write capacity units: 5 (Auto Scaling Error)
- Last decrease time: -
- Last increase time: -
- Storage size (in bytes): 0 bytes
- Item count: 0 [Manage live count](#)
- Region: US East (N. Virginia)
- Amazon Resource Name (ARN): am:aws:dynamodb:us-east-1:836295026856:table/LabelDetected

- After the lambda finish extract important data from json output file of amazon Rekognition, it will store it in the DynamoDB table – “LabelDetected”. (PART 1)

The screenshot shows the AWS Management Console interface for the 'LabelDetected' table in DynamoDB, displaying the 'Items' tab. The left sidebar contains navigation links for DynamoDB, Tables, Backups, Reserved capacity, Preferences, DAX, Dashboard, Clusters, Subnet groups, Parameter groups, and Events. The main content area displays the 'LabelDetected' table items, including a search bar and a table of items.

Items tab:

- Scan: [Table] LabelDetected: id
- Viewing 1 to 25 items
- Search bar: Scan [Table] LabelDetected: id
- Add filter button
- Start search button

Id	Confidence	Name	Timestamp
1	71	Asphalt	0
2	86	Automobile	0
3	86	Car	0
4	67	Electrical Device	0
5	98	Freeway	0

- After the lambda finish extract important data from json output file of amazon Rekognition, it will store it in the DynamoDB table – “LabelDetected”. (PART 2)

The screenshot shows the AWS Management Console interface for the 'LabelDetected' table. The left sidebar lists various services, and the main panel displays the table's 'Items' tab. The table contains the following data:

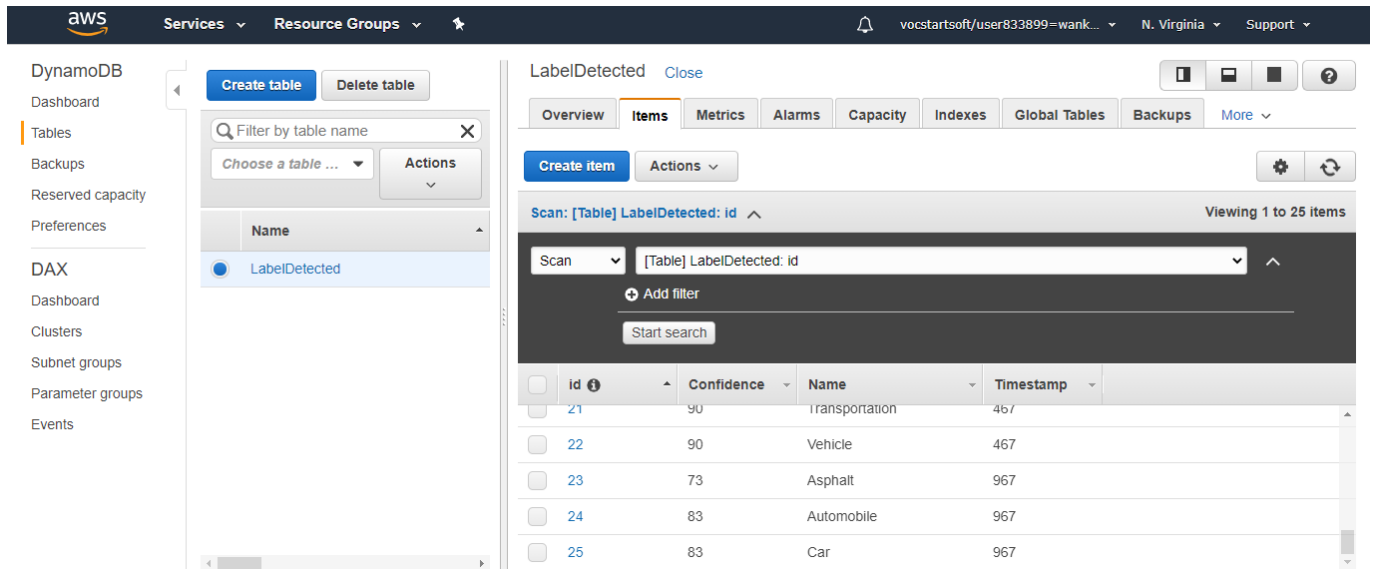
id	Confidence	Name	Timestamp
7	99	Road	0
8	67	Solar Panels	0
9	71	Tarmac	0
10	86	Transportation	0

- After the lambda finish extract important data from json output file of amazon Rekognition, it will store it in the DynamoDB table – “LabelDetected”. (PART 3)

The screenshot shows the AWS Management Console interface for the 'LabelDetected' table, displaying a different set of items. The table contains the following data:

id	Confidence	Name	Timestamp
12	65	Asphalt	467
13	90	Automobile	467
14	90	Car	467
15	67	Electrical Device	467
16	98	Freeway	467

- After the lambda finish extract important data from json output file of amazon Rekognition, it will store it in the DynamoDB table – “LabelDetected”. (PART 4)



The screenshot displays the AWS Management Console interface for a DynamoDB table named 'LabelDetected'. The left-hand navigation pane shows the 'DynamoDB' section with options like 'Tables', 'Backups', and 'Reserved capacity'. The main console area is divided into two panes. The left pane shows the 'LabelDetected' table with a 'Create table' button and a list of tables. The right pane shows the 'Items' tab, which displays a table of data. The table has four columns: 'id', 'Confidence', 'Name', and 'Timestamp'. The data is as follows:

id	Confidence	Name	Timestamp
21	90	Transportation	467
22	90	Vehicle	467
23	73	Asphalt	967
24	83	Automobile	967
25	83	Car	967

Amazon Simple Notification Services (Amazon SNS)

- Create Topic in Amazon SNS. Enter “video-lambda-sns” as name, press the “Create topic” button to create the topic.

The screenshot shows the 'Create topic' page in the Amazon SNS console. The left sidebar contains navigation links for 'Dashboard', 'Topics', 'Subscriptions', and 'Mobile'. The main content area is titled 'Details' and contains the following fields and sections:

- Name:** A text input field containing 'video-lambda-sns'. Below it, a note states: 'Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).'.
- Display name - optional:** A text input field containing 'My Topic'. Below it, a note states: 'To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)'. A sub-note below the field states: 'Maximum 100 characters, including hyphens (-) and underscores (_).'.
- Encryption - optional:** A section with a header 'Encryption - optional' and a description: 'Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.'
- Access policy - optional:** A section with a header 'Access policy - optional' and a description: 'This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. [Info](#)'.
- Delivery retry policy (HTTP/S) - optional:** A section with a header 'Delivery retry policy (HTTP/S) - optional' and a description: 'The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section. [Info](#)'.
- Delivery status logging - optional:** A section with a header 'Delivery status logging - optional' and a description: 'These settings configure the logging of message delivery status to CloudWatch Logs. [Info](#)'.
- Tags - optional:** A section with a header 'Tags - optional' and a description: 'A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#)'.

At the bottom right of the 'Details' section, there are two buttons: 'Cancel' and 'Create topic'.

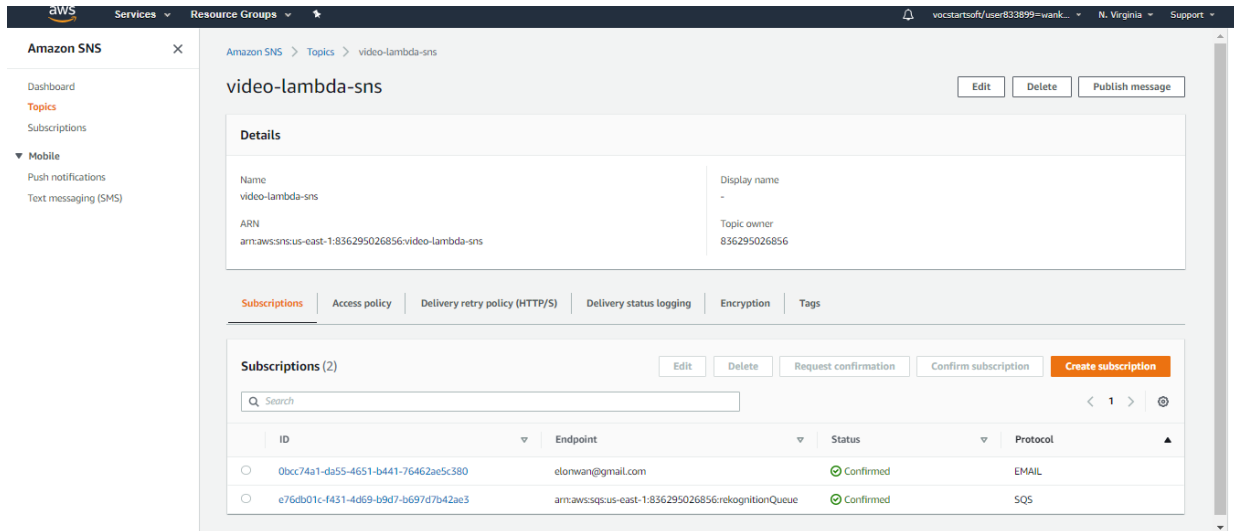
- Create subscription for topic “ video-lambda-sns” , select the ARN for video-lambda-sns for Topic ARN , select “Email” as protocol , and input specific email “elonwan@gmail.com” as Endpoint . Finally press “Create subscription” button to create subscription for the topic.

The screenshot shows the 'Create subscription' page in the Amazon SNS console. The left sidebar contains navigation links for 'Dashboard', 'Topics', 'Subscriptions', and 'Mobile'. The main content area is titled 'Create subscription' and contains the following fields and sections:

- Topic ARN:** A text input field containing 'arn:aws:sns:us-east-1:836295026856:video'.
- Protocol:** A dropdown menu with 'Email' selected.
- Endpoint:** A text input field containing 'elonwan@gmail.com'. Below it, a note states: 'An email address that can receive notifications from Amazon SNS.'
- Confirmation:** A blue box with a checkmark icon and the text: 'After your subscription is created, you must confirm it. [Info](#)'.
- Subscription filter policy - optional:** A section with a header 'Subscription filter policy - optional' and a description: 'This policy filters the messages that a subscriber receives. [Info](#)'.
- Redrive policy (dead-letter queue) - optional:** A section with a header 'Redrive policy (dead-letter queue) - optional' and a description: 'Send undeliverable messages to a dead-letter queue. [Info](#)'.

At the bottom right of the 'Create subscription' section, there are two buttons: 'Cancel' and 'Create subscription'.

- Topic “video-lambda-sns” and Subscription successfully created.



- After lambda extract important data from json output file of amazon Rekognition , it will send those data to specific email- elonwan@gmail.com via SNS

