Q1. Answers:-

Input:-

A computer screen shot of a computer screen

AI-generated content may be incorrect.



Output:-

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Athena query :-

A computer screen shot of a computer screen

AI-generated content may be incorrect.

Number of logins per region:-

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Q2. Answers:-

Input:-

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Lamda function:-

A screenshot of a computer

AI-generated content may be incorrect.

import json

import boto3

import logging

from datetime import datetime

logger = logging.getLogger()

logger.setLevel(logging.INFO)

s3\_client = boto3.client('s3')

sns\_client = boto3.client('sns')

SNS\_TOPIC\_ARN = "arn:aws:sns:us-east-1:430168628440:daily\_reports\_alerts"

def lambda\_handler(event, context):

    try:

        logger.info(f"Received event: {json.dumps(event)}")

        # Extract bucket & object info

        for record in event['Records']:

            bucket\_name = record['s3']['bucket']['name']

            object\_key = record['s3']['object']['key']

            upload\_time = record['eventTime']

            # Get object metadata

            response = s3\_client.head\_object(Bucket=bucket\_name, Key=object\_key)

            file\_size = response['ContentLength']

            message = {

                "FileName": object\_key,

                "Bucket": bucket\_name,

                "Size (bytes)": file\_size,

                "UploadTime": upload\_time

            }

            logger.info(f"Publishing message: {message}")

            # Publish to SNS

            sns\_client.publish(

                TopicArn=SNS\_TOPIC\_ARN,

                Subject="New Daily Report Uploaded",

                Message=json.dumps(message)

            )

        return {"statusCode": 200, "body": "Message published successfully."}

    except Exception as e:

        logger.error(f"Error processing S3 event: {e}", exc\_info=True)

        raise e

SNS screenshot:-

A screenshot of a computer

AI-generated content may be incorrect.

Cloudwatch logs:-

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Q3. Answers:-

Input:-

A screenshot of a computer

AI-generated content may be incorrect.

Lambda:-

A computer screen with a message box

AI-generated content may be incorrect.

Code:-

import json

import boto3

import time

import os

athena = boto3.client('athena')

sns = boto3.client('sns')

cloudwatch = boto3.client('cloudwatch')

ATHENA\_DB = 'athena1'

ATHENA\_OUTPUT = 's3://orders12345/athena\_results/'

SNS\_TOPIC\_ARN = 'arn:aws:sns:us-east-1:430168628440:data-quality-alerts'

def lambda\_handler(event, context):

    s3\_record = event['Records'][0]['s3']

    bucket = s3\_record['bucket']['name']

    key = s3\_record['object']['key']

    print(f"Triggered by file: s3://{bucket}/{key}")

    # Only process files from your target prefix

    if not key.startswith(INCOMING\_PREFIX):

        print(f"Skipping file outside prefix: {key}")

        return {"status": "skipped"}

    query = """

        SELECT COUNT(\*) AS bad\_records

        FROM orders

        WHERE order\_id IS NULL OR price < 0;

    """

    start = time.time()

    response = athena.start\_query\_execution(

        QueryString=query,

        QueryExecutionContext={'Database': ATHENA\_DB},

        ResultConfiguration={'OutputLocation': ATHENA\_OUTPUT}

    )

    Athena Query:-

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Q4. Answer:-

Input:-

A screenshot of a computer

AI-generated content may be incorrect.

For using redshift cluster:-

Showed access denied