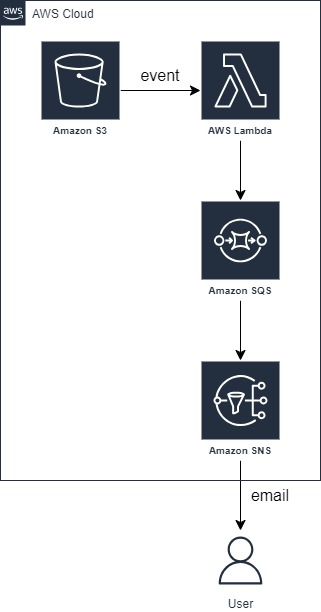
AWS S3 File Upload Notification System

Objective: Develop an automated notification system that sends email alerts whenever a new file is uploaded to an Amazon S3 bucket.

**The project involved the following components:**

* Amazon S3: Monitored for file uploads.
* AWS Lambda: Implemented to process S3 events and trigger notifications.
* Amazon SNS: Created a topic for sending email notifications.
* Amazon SQS: Configured a queue to handle event messages.

# Architecture:



# Project Steps:

**Create an Amazon S3 Bucket:**

* Set up an S3 bucket to store files.

**Create an SNS Topic:**

* Create an Amazon SNS (Simple Notification Service) topic.
* Set up an email subscription to the SNS topic for email notifications.

**Create an SQS Queue:**

* Set up an Amazon SQS (Simple Queue Service) queue to handle event messages.

**Create a Lambda Function:**

* Develop an AWS Lambda function using Python to process S3 events, send messages to the SQS queue, and publish notifications to the SNS topic.

**Lambda code:**

import json

import boto3

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

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

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

def lambda\_handler(event, context):

sns\_topic\_arn = ‘sns\_arn’

sqs\_queue\_url = 'sqs\_url'

# Process S3 event records

for record in event['Records']:

print(event)

# Extract S3 bucket and object information

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

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

# Example: Sending metadata to SQS

metadata = {

'bucket': s3\_bucket,

'key': s3\_key,

'timestamp': record['eventTime']

}

# Send metadata to SQS

sqs\_response = sqs\_client.send\_message(

QueueUrl=sqs\_queue\_url,

MessageBody=json.dumps(metadata)

)

# Example: Sending a notification to SNS

notification\_message = f"New file uploaded to S3 bucket '{s3\_bucket}' with key '{s3\_key}'"

sns\_response = sns\_client.publish(

TopicArn=sns\_topic\_arn,

Message=notification\_message,

Subject="File Upload Notification"

)

return {

'statusCode': 200,

'body': json.dumps('Processing complete')

}