To develop a notification system using AWS SDK V3, SNS, and SQS with a JavaScript-based AWS Lambda function, follow these steps:

**1. Set Up AWS Account and Services**

* Ensure you have an AWS account set up and that you have configured AWS CLI with the appropriate IAM permissions for SNS, SQS, and Lambda.
* Create an SNS topic and an SQS queue in the AWS Management Console.

**2. Create and Configure the SQS Queue**

* Go to the **SQS Console**, create a new queue, and configure the access policy to allow messages from your SNS topic.
* Ensure the SQS queue is subscribed to the SNS topic.

**3. Create and Configure the SNS Topic**

* Navigate to the **SNS Console** and create a new topic.
* Note down the ARN of the SNS topic.
* Subscribe the SQS queue to the SNS topic through the **SNS Console** or **SQS Console** by adding the subscription.

**4. Create an IAM Role for Lambda**

* Create an IAM role that has the necessary permissions to:
  + Publish messages to SNS.
  + Read from SQS.
* Attach the **AWSLambdaBasicExecutionRole** and additional policies to handle SNS and SQS actions.

**5. Set Up Lambda Function**

* Go to the **AWS Lambda Console**, create a new Lambda function, and choose Node.js as the runtime.
* Assign the IAM role created in the previous step to the Lambda function.

**6. Write the Lambda Function Code**

* Use the following code snippet to develop your Lambda function using AWS SDK V3:

javascript

Copy code

// Import required AWS SDK clients and commands for Node.js

const { SNSClient, PublishCommand } = require("@aws-sdk/client-sns");

const { SQSClient, ReceiveMessageCommand, DeleteMessageCommand } = require("@aws-sdk/client-sqs");

// Create SNS and SQS client

const snsClient = new SNSClient({ region: "us-east-1" }); // Replace with your region

const sqsClient = new SQSClient({ region: "us-east-1" }); // Replace with your region

// Define SNS topic ARN and SQS queue URL

const topicArn = "arn:aws:sns:us-east-1:123456789012:YourTopic"; // Replace with your topic ARN

const queueUrl = "https://sqs.us-east-1.amazonaws.com/123456789012/YourQueue"; // Replace with your queue URL

exports.handler = async (event) => {

try {

// Publish a message to the SNS topic

const publishParams = {

TopicArn: topicArn,

Message: "Hello from Lambda using AWS SDK V3!"

};

await snsClient.send(new PublishCommand(publishParams));

console.log("Message published to SNS.");

// Receive messages from the SQS queue

const receiveParams = {

QueueUrl: queueUrl,

MaxNumberOfMessages: 1,

WaitTimeSeconds: 5

};

const data = await sqsClient.send(new ReceiveMessageCommand(receiveParams));

if (data.Messages) {

console.log("Message received from SQS:", data.Messages[0].Body);

// Process the message and delete it from the queue

const deleteParams = {

QueueUrl: queueUrl,

ReceiptHandle: data.Messages[0].ReceiptHandle

};

await sqsClient.send(new DeleteMessageCommand(deleteParams));

console.log("Message deleted from SQS.");

} else {

console.log("No messages received.");

}

} catch (error) {

console.error("Error:", error);

}

};

**7. Deploy the Lambda Function**

* Deploy the function by uploading the code through the AWS Management Console or by using the AWS CLI or an Infrastructure as Code tool (e.g., AWS CDK or Serverless Framework).
* Test the function by triggering it manually or by setting up a CloudWatch Event rule.

**8. Test the Integration**

* Publish a message to the SNS topic to verify that it gets delivered to the SQS queue and that the Lambda function processes it.
* Monitor the Lambda function logs in **Amazon CloudWatch** to ensure it is executing as expected and handling messages properly.

**9. Verify and Monitor**

* Check the **SQS Console** to see if messages are being dequeued after being processed by the Lambda function.
* Use **Amazon CloudWatch** for monitoring, logs, and troubleshooting any issues with the Lambda execution.

**10. Cleanup (Optional)**

* If this is a test project, delete the resources (Lambda, SNS, SQS, IAM roles) to avoid incurring additional costs.

By following these steps, you will have a functioning JavaScript Lambda function using AWS SDK V3 integrated with SNS and SQS for a notification system.