Here is a **Java code example** for using **AWS SNS (Simple Notification Service) with Subscriptions**. This code demonstrates:

1. **Creating an SNS Topic**
2. **Subscribing an Email to the Topic**
3. **Publishing a Message to the Topic**

**1. Add AWS SDK Dependency**

If you are using **Maven**, add this dependency to your pom.xml:

<dependency>

<groupId>software.amazon.awssdk</groupId>

<artifactId>sns</artifactId>

<version>2.20.0</version>

</dependency>

**2. Java Code for AWS SNS with Subscription**

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;

import software.amazon.awssdk.regions.Region;

import software.amazon.awssdk.services.sns.SnsClient;

import software.amazon.awssdk.services.sns.model.\*;

public class SnsExample {

public static void main(String[] args) {

// Set AWS Region

Region region = Region.US\_EAST\_1; // Change as needed

// Create SNS Client

try (SnsClient snsClient = SnsClient.builder()

.region(region)

.credentialsProvider(ProfileCredentialsProvider.create())

.build()) {

// 1. Create SNS Topic

String topicArn = createSNSTopic(snsClient, "MyTestTopic");

// 2. Subscribe Email or SMS to the Topic

subscribeToTopic(snsClient, topicArn, "email", "your-email@example.com"); // Change to your email

// subscribeToTopic(snsClient, topicArn, "sms", "+1234567890"); // Use for SMS subscription

// 3. Publish Message to SNS Topic

publishToTopic(snsClient, topicArn, "Hello! This is a test message from AWS SNS using Java.");

} catch (SnsException e) {

System.err.println("Error: " + e.awsErrorDetails().errorMessage());

}

}

// Method to Create SNS Topic

public static String createSNSTopic(SnsClient snsClient, String topicName) {

CreateTopicRequest request = CreateTopicRequest.builder().name(topicName).build();

CreateTopicResponse response = snsClient.createTopic(request);

System.out.println("SNS Topic Created: " + response.topicArn());

return response.topicArn();

}

// Method to Subscribe Email or SMS to the Topic

public static void subscribeToTopic(SnsClient snsClient, String topicArn, String protocol, String endpoint) {

SubscribeRequest request = SubscribeRequest.builder()

.topicArn(topicArn)

.protocol(protocol)

.endpoint(endpoint)

.build();

SubscribeResponse response = snsClient.subscribe(request);

System.out.println("Subscription Request Sent. Subscription ARN: " + response.subscriptionArn());

}

// Method to Publish Message to SNS Topic

public static void publishToTopic(SnsClient snsClient, String topicArn, String message) {

PublishRequest request = PublishRequest.builder()

.topicArn(topicArn)

.message(message)

.build();

PublishResponse response = snsClient.publish(request);

System.out.println("Message Published. Message ID: " + response.messageId());

}

}

**How This Code Works**

1. **Creates an SNS Topic** → "MyTestTopic"
2. **Subscribes an Email (or SMS) to the Topic**
   * Email subscribers must **confirm the subscription** by clicking the link in the email.
3. **Publishes a Message to the Topic**

**Run the Code**

* Make sure your AWS credentials are **configured properly** in ~/.aws/credentials.
* Update the **AWS region, email, or phone number** in the code.
* Run the Java program.

This will create an **SNS topic, subscribe an email, and send a message** using AWS SNS in Java. 🚀