**AWS SES**

Amazon Simple Email Service (SES) is a cloud-based email sending and receiving service provided by Amazon Web Services (AWS).

It allows you to send and receive emails using a highly scalable and reliable infrastructure.

SES can be used for various purposes, such as sending transactional emails, marketing campaigns, notifications, and more. Here are some key features and concepts associated with AWS SES:

**1. Sending and Receiving Emails:** SES enables you to send emails programmatically or through the AWS Management Console. You can also receive incoming emails and process them using SES.

**2. Email Sending Options:**

- SMTP Interface: You can use the SMTP interface to integrate SES with your applications and send emails.

- API: SES provides an API that allows you to send emails using the AWS SDKs or RESTful API calls.

**3. Email Content:** SES supports various email formats, including text, HTML, and multipart messages. You can also include attachments.

**4. Sending Domains and Verification:** To send emails through SES, you need to verify the domains or email addresses you intend to use. This helps prevent abuse and ensures that your emails are delivered reliably.

**5. Bounces and Complaints:** SES provides mechanisms for handling bounces (undeliverable emails) and complaints (when recipients mark your emails as spam). Managing these helps maintain a good sending reputation.

**6. Pricing:** AWS SES offers a flexible pricing model based on the number of emails you send. There are also different pricing tiers depending on whether you're sending within the same AWS region or cross-region.

**Code for sending emails using the JavaMail API with AWS SES SMTP:-**

**Step 1: Set Up AWS SES SMTP Credentials**

1. Go to the AWS Management Console and navigate to the SES service.

2. In the "SMTP Settings" section, you'll find your SMTP credentials. Note down the SMTP server name, port, username, and password.

**Step 2: Create a Maven Project**

Create a new Maven project or use an existing one. Make sure you have Maven and Java installed on your system.

**Step 3: Add JavaMail and AWS SDK Dependencies**

Open your `pom.xml` file and add the JavaMail and AWS SDK dependencies:

**<properties>**

**<maven.compiler.source>11</maven.compiler.source>**

**<maven.compiler.target>11</maven.compiler.target>**

**</properties>**

**<dependencies>**

**<dependency>**

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

**<artifactId>ses</artifactId>**

**<version>2.20.127</version>**

**</dependency>**

**<dependency>**

**<groupId>com.sun.mail</groupId>**

**<artifactId>javax.mail</artifactId>**

**<version>1.6.2</version>**

**</dependency>**

**</dependencies>**

**Step 4: Write Java Code**

**A) Using AWS SES CLIENT**

**package com.app.raghu;**

**import java.io.ByteArrayOutputStream;**

**import java.io.IOException;**

**import java.nio.ByteBuffer;**

**import java.util.Properties;**

**import javax.mail.MessagingException;**

**import javax.mail.Session;**

**import javax.mail.internet.InternetAddress;**

**import javax.mail.internet.MimeMessage;**

**import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;**

**import software.amazon.awssdk.core.SdkBytes;**

**import software.amazon.awssdk.regions.Region;**

**import software.amazon.awssdk.services.ses.SesClient;**

**import software.amazon.awssdk.services.ses.model.RawMessage;**

**import software.amazon.awssdk.services.ses.model.SendRawEmailRequest;**

**public class SesSmtpSender {**

**public static void main(String[] args) throws MessagingException, IOException {**

**String sender = "javabyraghu@gmail.com";**

**String recipient = "javabyraghu@gmail.com";**

**String subject = "Hello from AWS SES SMTP";**

**String bodyText = "This is a test email sent using AWS SES SMTP.";**

**Region region = Region.AP\_SOUTH\_1; // Change to your desired region**

**SesClient sesClient = SesClient.builder()**

**.credentialsProvider(DefaultCredentialsProvider.create())**

**.region(region)**

**.build();**

**// Create a JavaMail MIME message**

**Properties props = new Properties();**

**Session session = Session.getDefaultInstance(props, null);**

**MimeMessage mimeMessage = new MimeMessage(session);**

**mimeMessage.setFrom(new InternetAddress(sender));**

**mimeMessage.setRecipient(javax.mail.Message.RecipientType.TO, new InternetAddress(recipient));**

**mimeMessage.setSubject(subject);**

**mimeMessage.setText(bodyText);**

**// Convert MimeMessage to raw email**

**ByteArrayOutputStream outputStream = new ByteArrayOutputStream();**

**mimeMessage.writeTo(outputStream);**

**ByteBuffer rawEmailByteBuffer = ByteBuffer.wrap(outputStream.toByteArray());**

**// Send the raw email using SES**

**SendRawEmailRequest emailRequest = SendRawEmailRequest.builder()**

**.rawMessage(RawMessage.builder().data(SdkBytes.fromByteBuffer(rawEmailByteBuffer)).build())**

**.build();**

**sesClient.sendRawEmail(emailRequest);**

**System.out.println("Email sent successfully!");**

**sesClient.close();**

**}**

**}**

**B) USING JAVA MAIL API**

**package com.app.raghu;**

**import java.util.Properties;**

**import javax.mail.Authenticator;**

**import javax.mail.Message;**

**import javax.mail.MessagingException;**

**import javax.mail.PasswordAuthentication;**

**import javax.mail.Session;**

**import javax.mail.Transport;**

**import javax.mail.internet.InternetAddress;**

**import javax.mail.internet.MimeMessage;**

**public class SesSmtpJavaMailSender {**

**public static void main(String[] args) {**

**String sender = "javabyraghu@gmail.com";**

**String recipient = "javabyraghu@gmail.com";**

**String subject = "Hello from AWS SES SMTP with JavaMail";**

**String bodyText = "This is a test email sent using AWS SES SMTP and JavaMail.";**

**// SMTP server settings (Replace with your SES SMTP settings)**

**String smtpHost = "email-smtp.ap-south-1.amazonaws.com";**

**int smtpPort = 587;**

**String smtpUsername = "AWS SDK - IAM USER ACCESS KEY";**

**String smtpPassword = "AWS SDK - IAM USER SECRET KEY";**

**// Setup JavaMail properties**

**Properties properties = new Properties();**

**properties.put("mail.smtp.auth", "true");**

**properties.put("mail.smtp.starttls.enable", "true");**

**properties.put("mail.smtp.host", smtpHost);**

**properties.put("mail.smtp.port", smtpPort);**

**Session session = Session.getInstance(properties, new Authenticator() {**

**protected PasswordAuthentication getPasswordAuthentication() {**

**return new PasswordAuthentication(smtpUsername, smtpPassword);**

**}**

**});**

**try {**

**// Create a new MimeMessage**

**Message message = new MimeMessage(session);**

**message.setFrom(new InternetAddress(sender));**

**message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(recipient));**

**message.setSubject(subject);**

**message.setText(bodyText);**

**// Send the message using the Transport class**

**Transport.send(message);**

**System.out.println("Email sent successfully!");**

**} catch (MessagingException e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**Step 5: Replace SES SMTP Settings**

Replace the placeholders in the code with your actual SES SMTP server settings (SMTP host, port, username, and password).

**Step 6: Build and Run**

Compile and run your project using Maven: