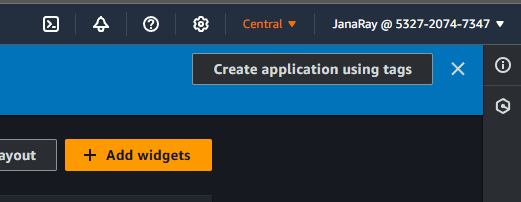
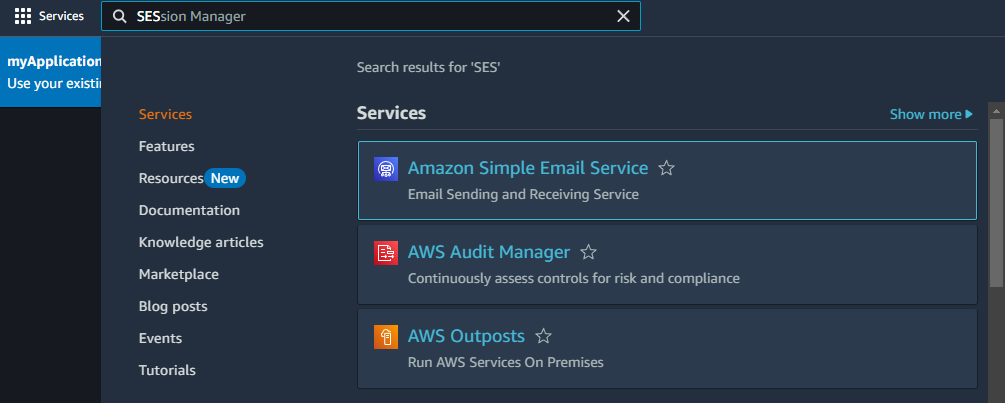
**AWS SES: A scalable and reliable emailing service that can be used to send different types of emails to customers.**

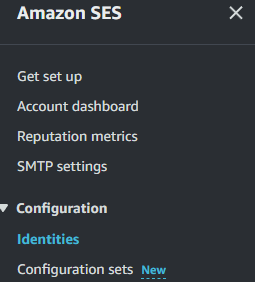
**Steps to set up AWS SES:**

Step 1: Select Region: Currently selected Central Canada but can be a region that better reflects your usage.  


Step 2: Look up Amazon SES



Step 3: On the left side menu, Click Identities

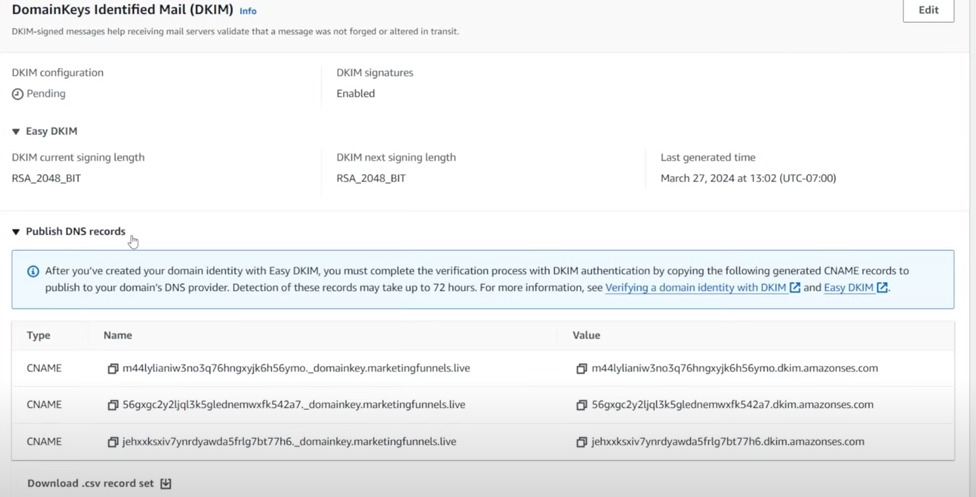


Step 4: Click Create Identity



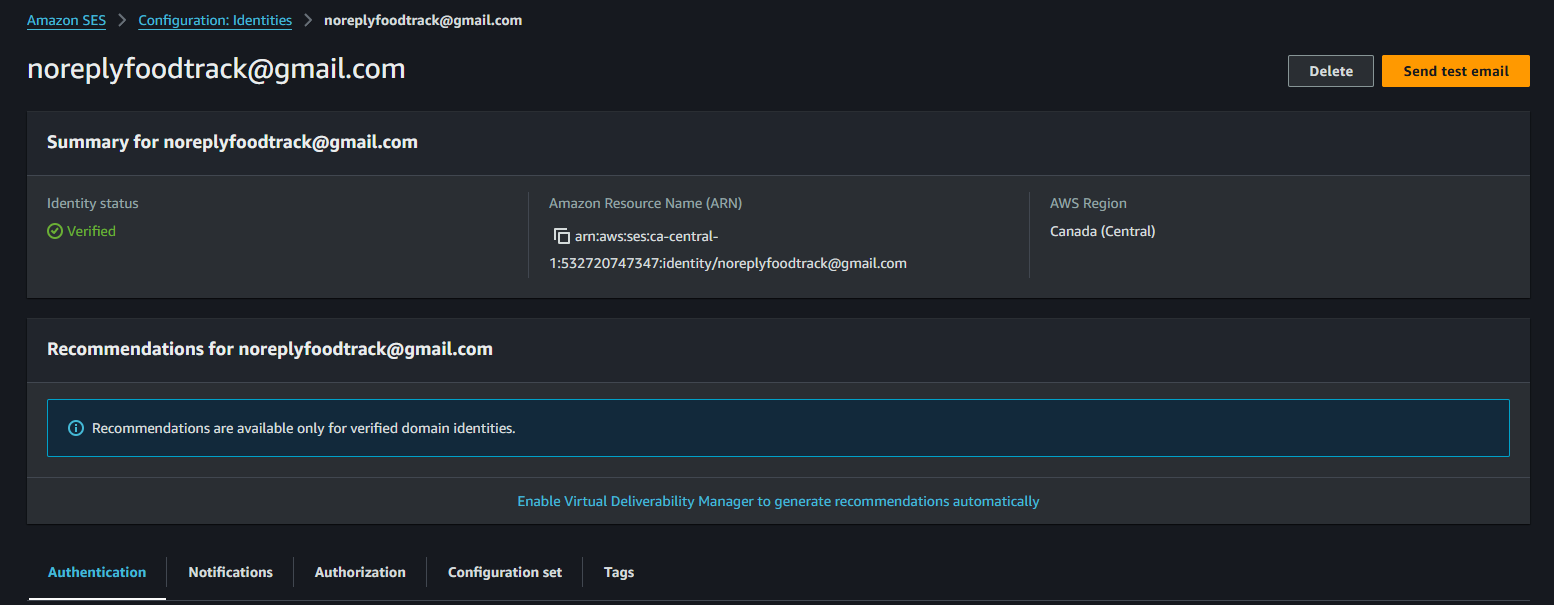
Step 5: Choose Domain or Email Address then fill out the boxes.

Step 6: If you used the domain, validate the identity by going into the ‘Manage DNS records of your domain’ section. Then create new records using the ‘CName’ category type, Name, and Target(Value) when you scroll down on the AWS SES domain identity page. The domain may take up to 72 hours to verify.

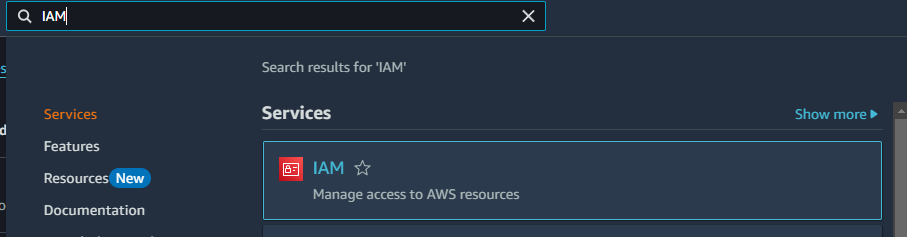


For those that used an email instead, check the email for that account and follow the verification steps there.

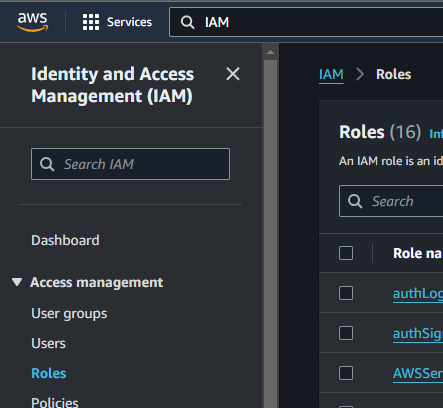
Step 7: The domain or email should be all setup. You will be able to test it using the Send test email button when you click into the SES identity.



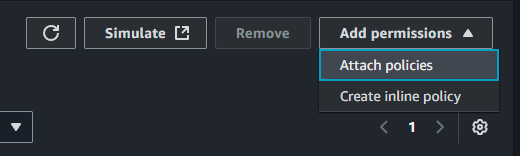
Step 8: To use SES in AWS Lambda, we need to attach the ‘AmazonSESFullAccess’ Permission policy to the lambda role. Regardless if we are just attaching this permission to an existing lambda or creating a new one, go to the IAM dashboard.



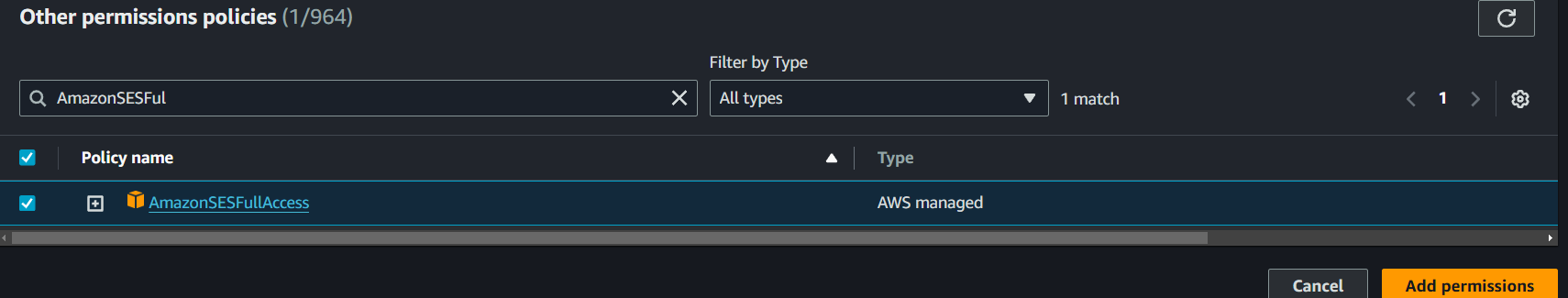
Step 9: Click roles on the left side bar menu.

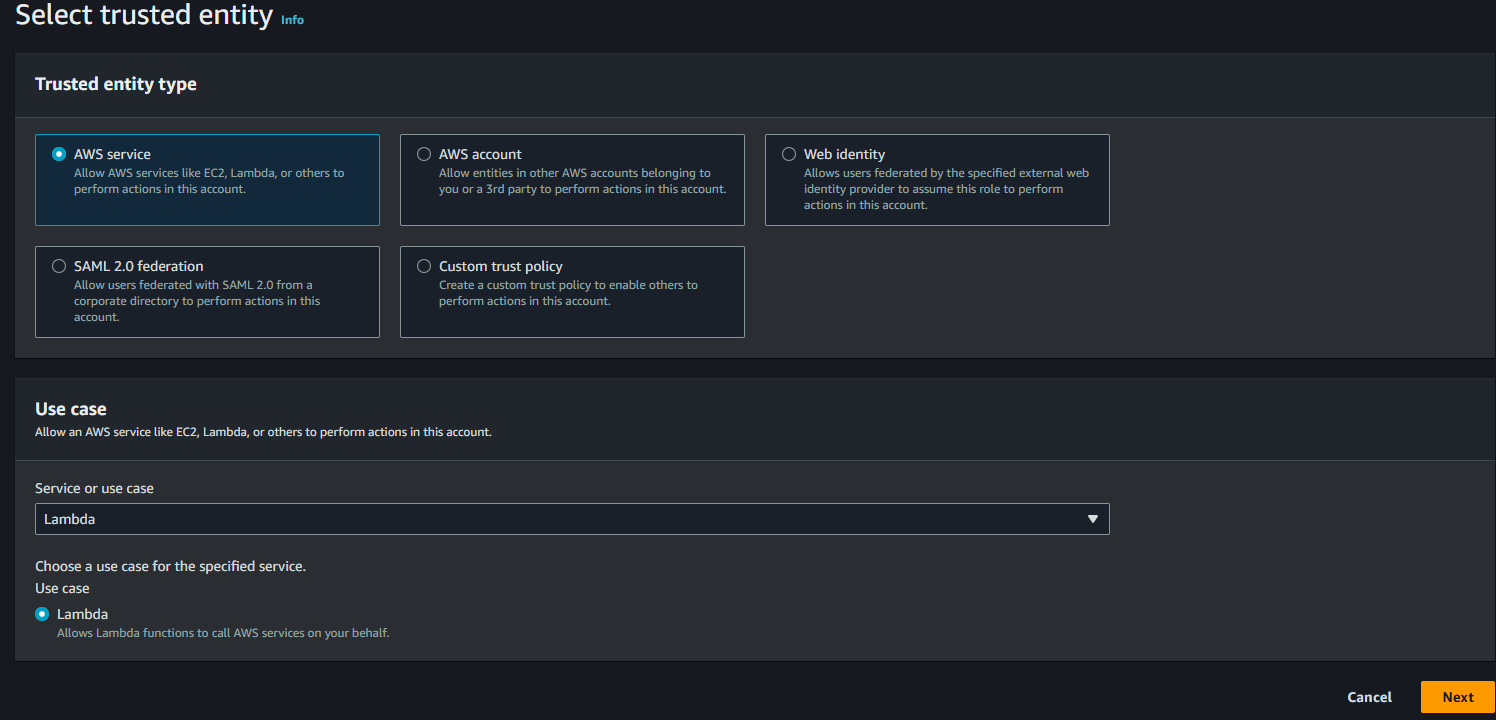


Step 10: To update an existing lambda role, click the role name, Add permissions, then Attach policies.

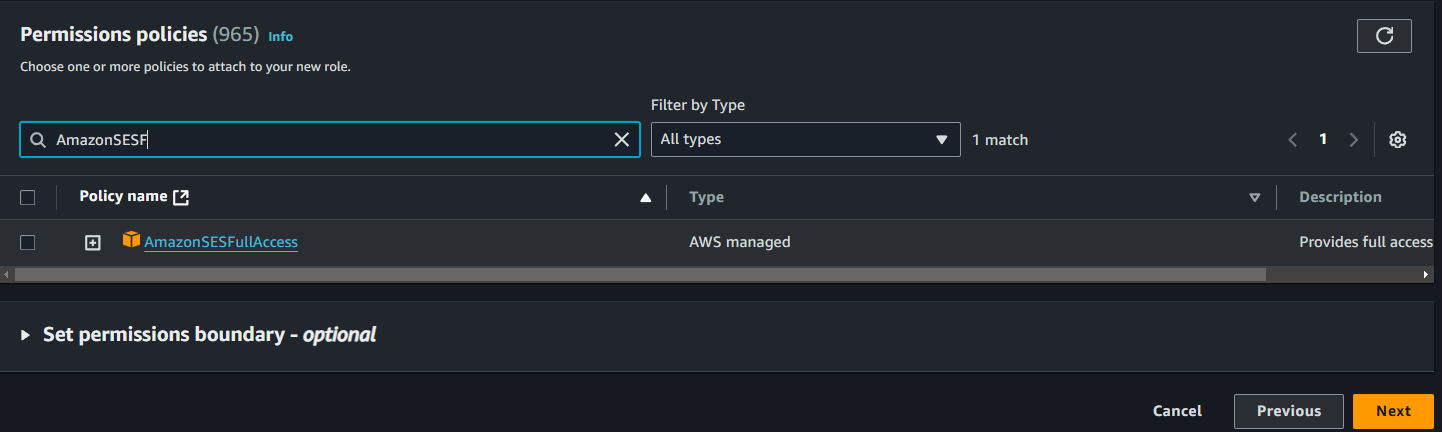


Search and select the AmazonSESFullAccess then click Add permissions.

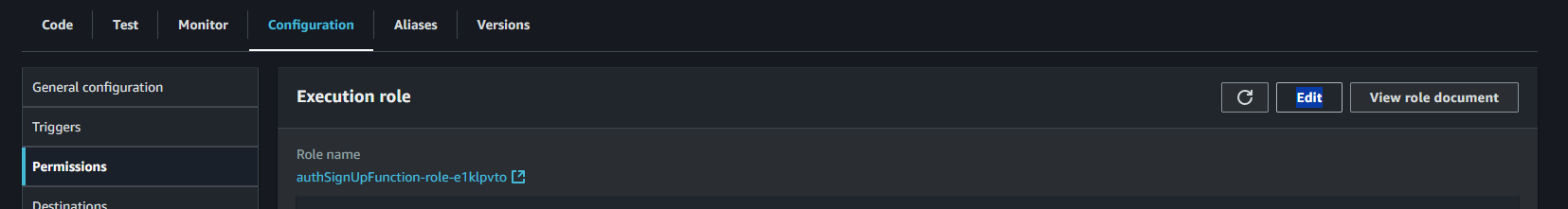


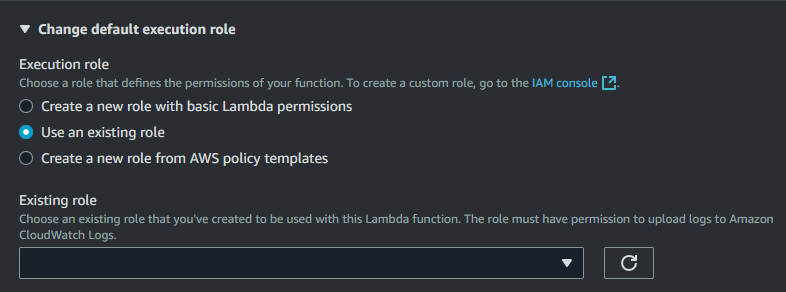
For a new lambda role, click create role, then AWS service and choose a Lambda Use Case.

After hitting next, search and select the AmazonSESFullAccess Permission policy and hit Next.



After that, give the role a name and click Create Role.

Step 11: Ensure that the Lambda functions that are created map to the correct roles. To check or edit it, got to the Lambda functions dashboard, click a lambda function and navigate to Configuration and then Permissions. If it is wrong, hit edit.

If you are creating a new function, expand the ‘Change default execution role’ section and select Use an existing role. Click the existing role to map the lambda function to.  


Here is an example script written by chatGPT to send an email using the SES in Lambda functions:

const AWS = require('aws-sdk');

const ses = new AWS.SES({ region: 'us-east-1' }); // Update to your SES region

exports.handler = async (event) => {

const senderEmail = "noreplyfoodtrack@gmail.com"; // Your verified Gmail address

const recipientEmail = "recipient@example.com"; // Replace with the recipient's email

const subject = "Test Email from AWS Lambda and SES";

const bodyText = "Hello,\nThis is a test email sent from AWS Lambda using Amazon SES.";

const bodyHtml = `

<html>

<head></head>

<body>

<h1>Hello!</h1>

<p>This is a test email sent from AWS Lambda using <a href='https://aws.amazon.com/ses/'>Amazon SES</a>.</p>

</body>

</html>

`;

const params = {

Source: senderEmail,

Destination: {

ToAddresses: [recipientEmail],

},

Message: {

Subject: { Data: subject },

Body: {

Text: { Data: bodyText },

Html: { Data: bodyHtml }

}

}

};

try {

const data = await ses.sendEmail(params).promise();

return {

statusCode: 200,

body: JSON.stringify({ message: "Email sent successfully!", data: data }),

};

} catch (error) {

return {

statusCode: 400,

body: JSON.stringify({ message: "Failed to send email", error: error.message }),

};

}

};