To achieve sending an email when someone submits a contact form using AWS services and Lambda, you can follow these steps:

1. **Create an AWS SES (Simple Email Service) to send emails.**
2. **Create an AWS Lambda function to handle the form submission and send an email using SES.**
3. **Set up an API Gateway to trigger the Lambda function from your contact form submission.**
4. **Update your contact form to call the API Gateway endpoint.**

Here's a detailed step-by-step guide:

1. Set up AWS SES

1. **Verify your email address:**
   - Go to the SES Console.
   - In the SES dashboard, click on "Email Addresses" under the "Identity Management" section.
   - Click "Verify a New Email Address" and follow the prompts to verify the email address from which you'll send emails.
2. **Move out of the SES sandbox (optional but recommended for production):**
   - By default, SES starts in a sandbox mode with restrictions. To move out, you need to request production access.

2. Create a Lambda Function

1. **Create a new Lambda function:**
   - Go to the Lambda Console.
   - Click "Create function".
   - Choose "Author from scratch".
   - Set a function name, e.g., `ContactFormHandler`.
   - Choose a runtime, e.g., Python 3.8 or Node.js 14.x.
   - Create the function.

2. **Set up the Lambda function code:**
    - Replace the default code with the following (Python example):

```
python
Copy code
import json
import boto3
import os

ses = boto3.client('ses')

def lambda_handler(event, context):
    # Extract data from the event
    body = json.loads(event['body'])
    name = body['name']
    email = body['email']
    message = body['message']

    # Email parameters
    subject = f"New Contact Form Submission from {name}"
    recipient = os.environ['RECIPIENT_EMAIL']
    sender = os.environ['SENDER_EMAIL']
    body_text = f"Name: {name}\nEmail: {email}\nMessage: {message}"

    # Send the email
    response = ses.send_email(
        Source=sender,
        Destination={
            'ToAddresses': [recipient]
        },
        Message={
            'Subject': {
                'Data': subject,
                'Charset': 'UTF-8'
            },
            'Body': {
                'Text': {
                    'Data': body_text,
                    'Charset': 'UTF-8'
                }
            }
        }
    )

    return {
        'statusCode': 200,
        'body': json.dumps('Email sent successfully')
    }
```

3. **Set environment variables:**
    - In the Lambda function configuration, set the environment variables `RECIPIENT_EMAIL` and `SENDER_EMAIL` to your recipient and verified sender email addresses, respectively.

4. **Set up permissions:**
    - Attach the AWS managed policy `AmazonSESFullAccess` to the Lambda function's execution role so it can send emails using SES.

3. Set up API Gateway

1. **Create a new API:**
    - Go to the API Gateway Console.
    - Click "Create API".
    - Choose "HTTP API" or "REST API" (HTTP API is simpler and preferred for this use case).
2. **Create an endpoint:**
    - Create a new resource and method (e.g., POST) that will trigger your Lambda function.
    - Integrate the POST method with your Lambda function.
3. **Deploy the API:**
    - Deploy the API to a stage and note the endpoint URL.

4. Update Your Contact Form

Update your contact form HTML/JavaScript to submit the form data to the API Gateway endpoint. Example using JavaScript:

```
html
Copy code
<form id="contactForm">
  <input type="text" name="name" placeholder="Your Name" required>
  <input type="email" name="email" placeholder="Your Email" required>
  <textarea name="message" placeholder="Your Message" required></textarea>
  <button type="submit">Send</button>
</form>

<script>
document.getElementById('contactForm').addEventListener('submit', async
function(event) {
  event.preventDefault();

  const formData = new FormData(this);
  const data = {
    name: formData.get('name'),
    email: formData.get('email'),
    message: formData.get('message')
  };

  const response = await fetch('https://YOUR_API_GATEWAY_ENDPOINT', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
```

```
    body: JSON.stringify(data)
  });

  const result = await response.json();
  alert(result);
});
</script>
```

Replace `https://YOUR_API_GATEWAY_ENDPOINT` with the actual endpoint URL you obtained from the API Gateway.

Summary

By following these steps, you can set up a system where submitting a contact form triggers an AWS Lambda function, which sends an email using AWS SES. This approach ensures that your email sending functionality is scalable and secure.