BECHFAM LLC

Serverless Contact Form – Implementation Guide

This guide walks you through deploying a secure, serverless contact form using Amazon API Gateway, AWS Lambda, and Amazon Simple Email Service (SES).

Prerequisites

- AWS account with AdministratorAccess (for deployment).
- A sending domain you control (e.g., bechfam.io).
- Recommended region: us■west■2 (Oregon) for U.S. workloads.

Architecture Summary

HTTPS POST requests hit API Gateway → invoke a Lambda function → Lambda validates inputs and calls Amazon SES to send an email to your chosen recipient.

Step 1 — Verify Domain in Amazon SES

1 Open Amazon SES → Verified identities → Verify a new domain.

2 Add DNS records for DKIM/SPF as shown by SES to your DNS host.

3 If your account is in SES Sandbox, request Production access (Use case: contact form notifications).

Step 2 — Create the Lambda Function

Runtime: Python 3.11 (or Node.js 20). Configure the following environment variables:

• RECIPIENT_EMAIL (e.g., info@yourdomain.com)
• SENDER_EMAIL (verified in SES, e.g., no-reply@yourdomain.com)
• SUBJECT_PREFIX (e.g., "Contact Form: ")

Sample Python handler (minimal):

```
import os, json, boto3
ses = boto3.client('ses')
RECIPIENT = os.environ['RECIPIENT_EMAIL']
SENDER = os.environ['SENDER_EMAIL']
SUBJECT_PREFIX = os.environ.get('SUBJECT_PREFIX', 'Contact Form: ')

def lambda_handler(event, context):
body = json.loads(event.get('body') or '{}')
name = (body.get('name') or '').strip()
email = (body.get('email') or '').strip()
message = (body.get('message') or '').strip()

if not name or not email or not message:
return {'statusCode': 400, 'headers': {'Content-Type': 'application/json'}, 'body':
json.dumps({'error': 'Missing fields'})}
```

```
text = f"""New contact form submission:
Name: {name}
Email: {email}

Message:
{message}
"""

ses.send_email(
Source=SENDER,
Destination={'ToAddresses': [RECIPIENT]},
Message={
'Subject': {'Data': SUBJECT_PREFIX + name},
'Body': {'Text': {'Data': text}}
}
)

return {'statusCode': 200, 'headers': {'Content-Type': 'application/json'}, 'body':
json.dumps({'ok': True})}
```

IAM Permissions for Lambda

Attach a policy allowing SES SendEmail:

```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": ["ses:SendEmail", "ses:SendRawEmail"],
"Resource": "*"
}
]
}
```

Step 3 — Create API Gateway (HTTP API)

1. Create an HTTP API with a single route: POST /contact
2. Integration: Lambda function created in Step 2.
3. Enable CORS for your website origins (e.g., https://www.yourdomain.com).
4. Deploy the API and note the invoke URL.

Step 4 — Optional: Google reCAPTCHA Verification

Add a reCAPTCHA token field to your form and verify in Lambda using Google’s siteverify endpoint before sending email.

Step 5 — Test & Handover

1. Submit a test message and confirm email delivery from SES.
2. Provide the client with API endpoint, Lambda env var values, and a sample HTML form snippet.
3. Enable CloudWatch log retention and set basic alarms (5XX on API, Lambda errors).

Cost Estimate (Typical)

Under light traffic, monthly AWS costs are often below $1 (API Gateway + Lambda + SES free/low tiers).

Operations & Security Best Practices

- Use least■privilege IAM on the Lambda role.
- Set request size limits and timeouts.
- Rotate SES sender addresses/credentials as needed.
- Enable CloudWatch alarms for error spikes.

Support

For assistance, contact info@bechfam.io. Standard support: 14 days post■deployment.