**Lambda Function Python Code**

import boto3

import csv

# Initialize the boto3 client

s3\_client = boto3.client('s3')

ses\_client = boto3.client('ses')

def lambda\_handler(event, context):

# Specify the S3 bucket name

bucket\_name = 'ttt-email-marketing' **# Replace with your bucket name**

try:

# Retrieve the CSV file from S3

csv\_file = s3\_client.get\_object(Bucket=bucket\_name, Key='contacts.csv')

lines = csv\_file['Body'].read().decode('utf-8').splitlines()

# Retrieve the HTML email template from S3

email\_template = s3\_client.get\_object(Bucket=bucket\_name, Key='email\_template.html')

email\_html = email\_template['Body'].read().decode('utf-8')

# Parse the CSV file

contacts = csv.DictReader(lines)

for contact in contacts:

# Replace placeholders in the email template with contact information

personalized\_email = email\_html.replace('{{FirstName}}', contact['FirstName'])

# Send the email using SES

response = ses\_client.send\_email(

Source='you@yourdomainname.com', **# Replace with your verified "From" address**

Destination={'ToAddresses': [contact['Email']]},

Message={

'Subject': {'Data': 'Your Weekly Tiny Tales Mail!', 'Charset': 'UTF-8'},

'Body': {'Html': {'Data': personalized\_email, 'Charset': 'UTF-8'}}

}

)

print(f"Email sent to {contact['Email']}: Response {response}")

except Exception as e:

print(f"An error occurred: {e}")

**Lambda Function Test Event**

{

"comment": "Generic test event for scheduled Lambda execution. The function does not use this event data.",

"test": true

}

**IAM Policy for SES and S3 permissions**

**Update the ARN to use your S3 bucket**

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:GetObject"

],

"Resource": **"arn:aws:s3:::ttt-email-marketing/\*"**

},

{

"Effect": "Allow",

"Action": [

"ses:SendEmail",

"ses:SendRawEmail"

],

"Resource": "\*"

}

]

}