Name - Phanse Sakshi Krishandev

PRN No. - 121A2034

Year - BE

Branch - EXTC

CLOUD COMPUTING PROJECT REPORT FH2024

PROBLEM STATEMENT: BUILD A SERVERLESS EMAIL MARKETING APPLICATION

Introduction:

This project focuses on creating a serverless email marketing application using AWS services to automate personalized email campaigns. It reduces infrastructure management by utilizing:

- Amazon SES for sending emails.

- AWS Lambda to process email templates and contact data.

- Amazon S3 for storing email templates and contact lists.

- Amazon Event Bridge to schedule email campaigns.

- AWS IAM for managing security and access controls.

The goal is to streamline email marketing, allowing users to store and manage email templates, personalize messages, and send bulk emails efficiently, all while leveraging the scalability and cost-effectiveness of serverless architecture.

2. Services we have used?

Services will be using Amazon Simple Email Service(SES), AWS Lambda, Amazon Simple Storage service(S3), Amazon Event Bridge, AWS Identity and Access Management (IAM)

How much will this cost?

using eraser.io to create checklist

2. What you need to following along?

1. email address (to send to)that you can validate (can be personal email addresses)

2. An email address (to send from) that you can valid date (needs to be from your own domain)

3. basic knowledge of AWS

4. A text editor/ somewhere to write HTML code

3. High level requirements for application

1. A piece to store email templates and list of contacts

2. A way to send email

3. A way to merge email/ templates with contacts and send them to the email service.

4.A way to Trigger sending of emails and a schedule clean up

4. Creating an S3 bucket to store email templates and contacts

Step 1: Create an S3 Bucket

- Sign in to the AWS Management Console:

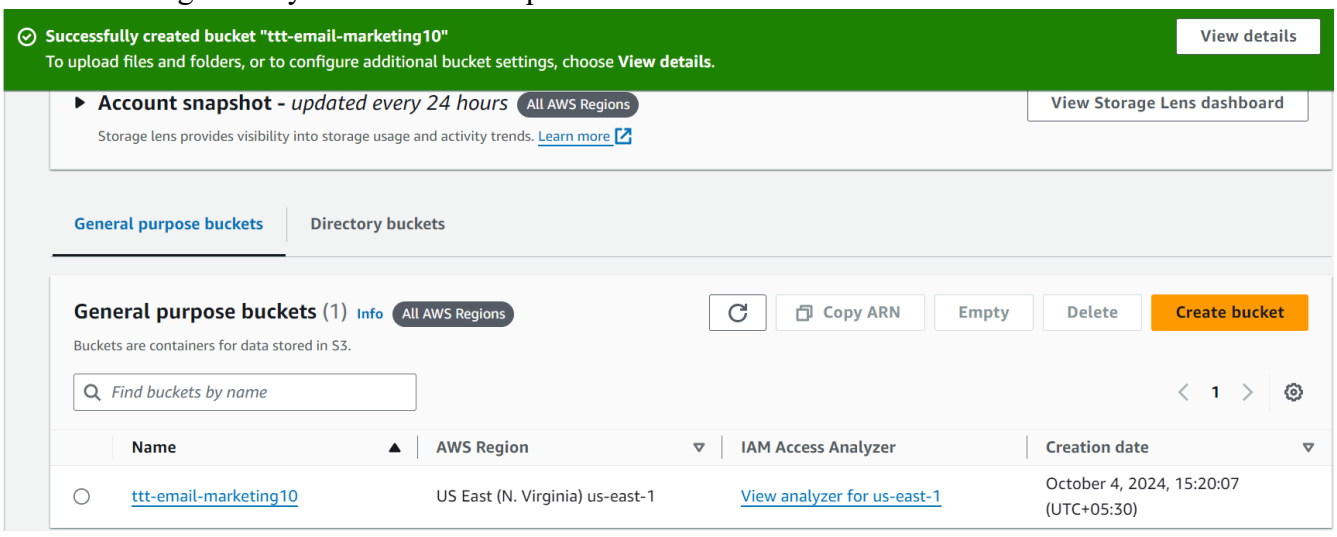
Open the AWS Management Console and sign in to your AWS account.

Navigate to S3:

- Go to the S3 service by selecting it from the AWS Management Console.
- Create a new bucket:
- Click the "Create bucket" button.

Enter a globally unique name for your bucket like ttt-email-marketing

Choose the region for your bucket and open view details



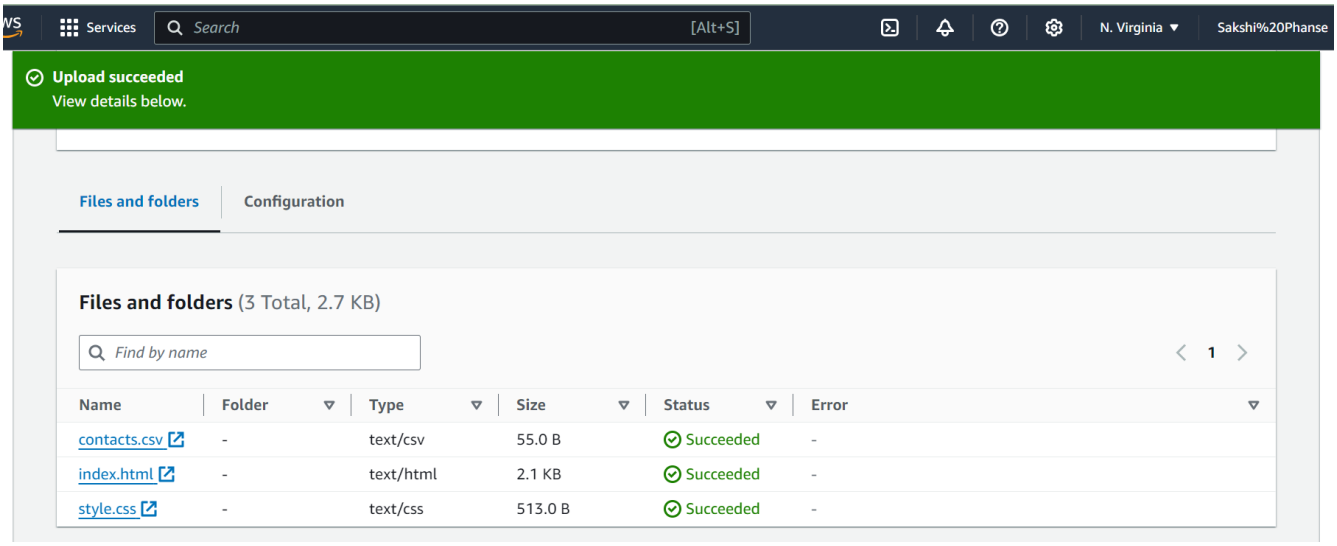
5. The HTML email template we'll be using to send

Now create HTML template. we have written the Hello {first name} in HTML file

6. The CSV file with contact information to email to

7. Uploading the email template and contact list to the S3 bucket

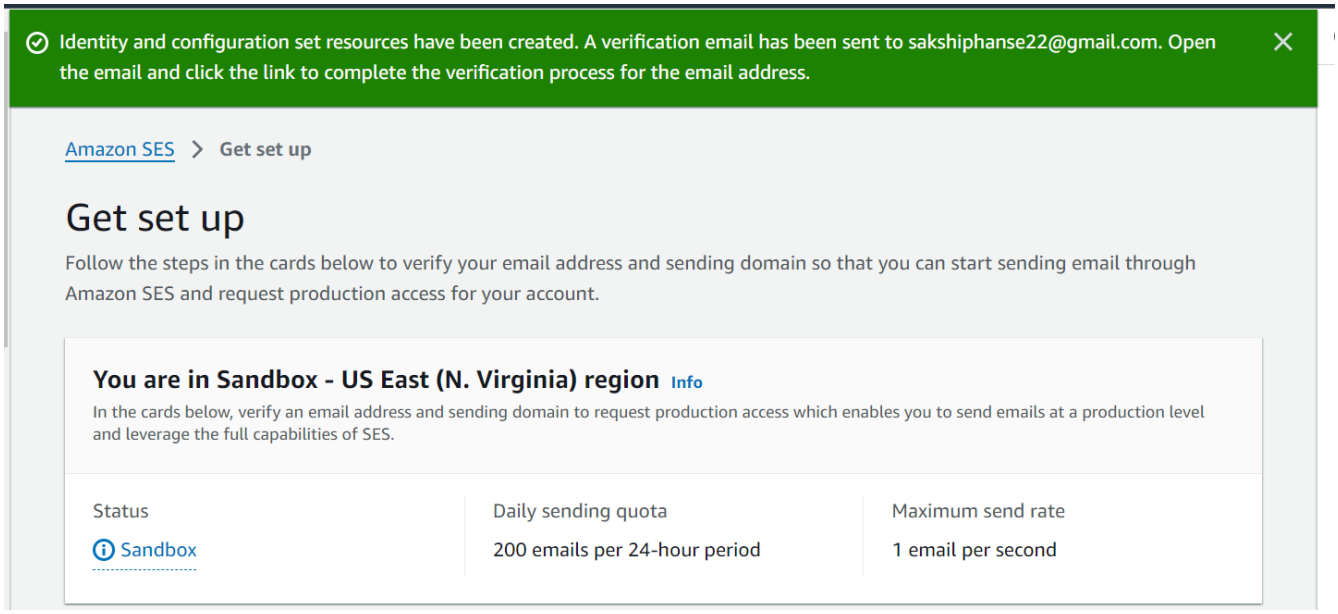
- Now back to S3 bucket
- object tab
- upload the contact.csv file and email file
- drag that file and upload successfully
- close

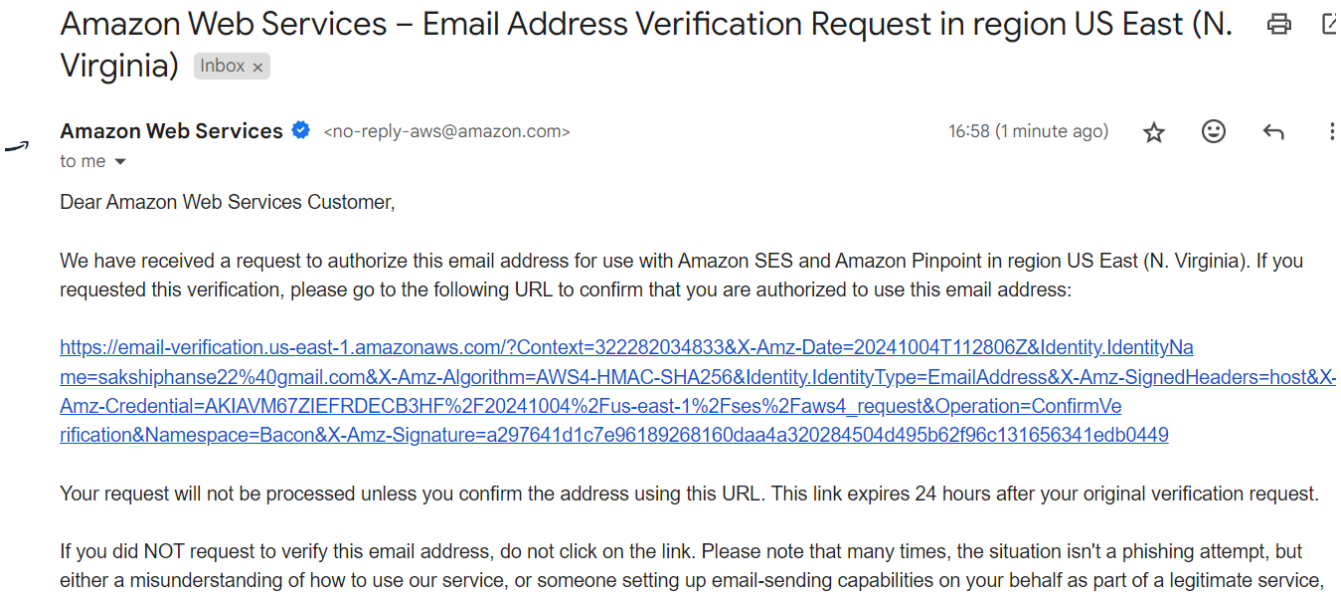


8. Diagramming as code with Eraser.io

9. Setting up Amazon Simple Email Service (SES) with an email address and domain
- Open link in new tab from aws logo
- search SES
- Enter email address (sakshiphanse22@gmail.com)

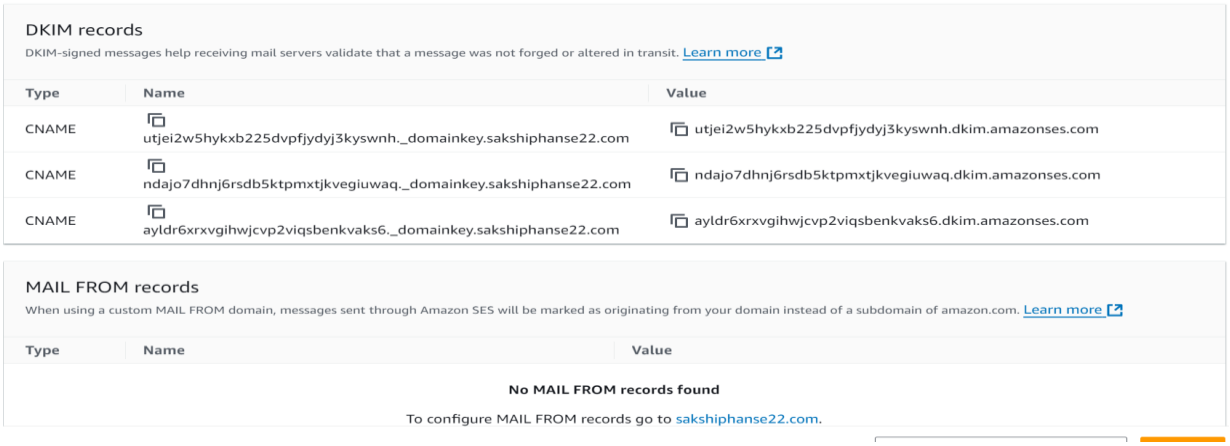
10. The SES sandbox, limitations, and how to move it to production


Identity and configuration set resources have been created. A verification email has been sent to sakshiphanse22@gmail.com. Open the email and click the link to complete the verification process for the email address.
Amazon SES > Get set up
Get set up
Follow the steps in the cards below to verify your email address and sending domain so that you can start sending email through Amazon SES and request production access for your account.
You are in Sandbox - US East (N. Virginia) region Info
In the cards below, verify an email address and sending domain to request production access which enables you to send emails at a production level and leverage the full capabilities of SES.
Status
Sandbox
Daily sending quota
200 emails per 24-hour period
Maximum send rate
1 email per second


Amazon Web Services – Email Address Verification Request in region US East (N. Virginia)
Inbox ×
Amazon Web Services <no-reply-aws@amazon.com>
16:58 (1 minute ago)
to me
Dear Amazon Web Services Customer,
We have received a request to authorize this email address for use with Amazon SES and Amazon Pinpoint in region US East (N. Virginia). If you requested this verification, please go to the following URL to confirm that you are authorized to use this email address:
https://email-verification.us-east-1.amazonaws.com/?Context=322282034833&X-Amz-Date=20241004T112806Z&Identity.IdentityName=sakshiphanse22%40gmail.com&X-Amz-Algorithm=AWS4-HMAC-SHA256&Identity.IdentityType=EmailAddress&X-Amz-SignedHeaders=host&X-Amz-Credential=AKIAVM67ZIEFRDECB3HF%2F20241004%2Fus-east-1%2Fses%2Faws4_request&Operation=ConfirmVerification&Namespace=Bacon&X-Amz-Signature=a297641d1c7e96189268160daa4a320284504d495b62f96c131656341edb0449
Your request will not be processed unless you confirm the address using this URL. This link expires 24 hours after your original verification request.
If you did NOT request to verify this email address, do not click on the link. Please note that many times, the situation isn't a phishing attempt, but either a misunderstanding of how to use our service, or someone setting up email-sending capabilities on your behalf as part of a legitimate service,

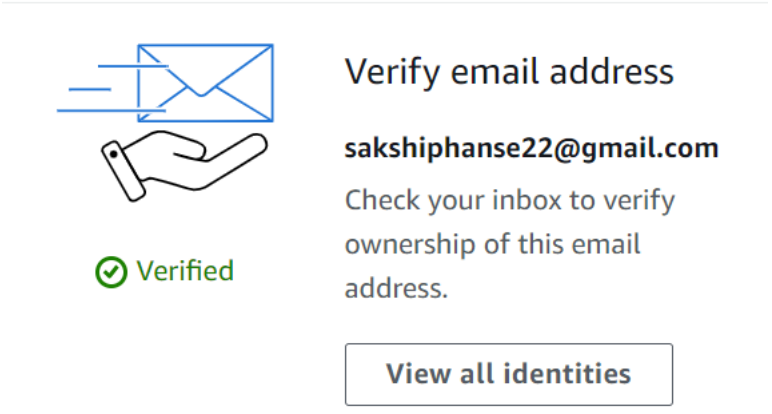
Congratulations!

You have successfully verified an email address. You can now start sending email from this address.



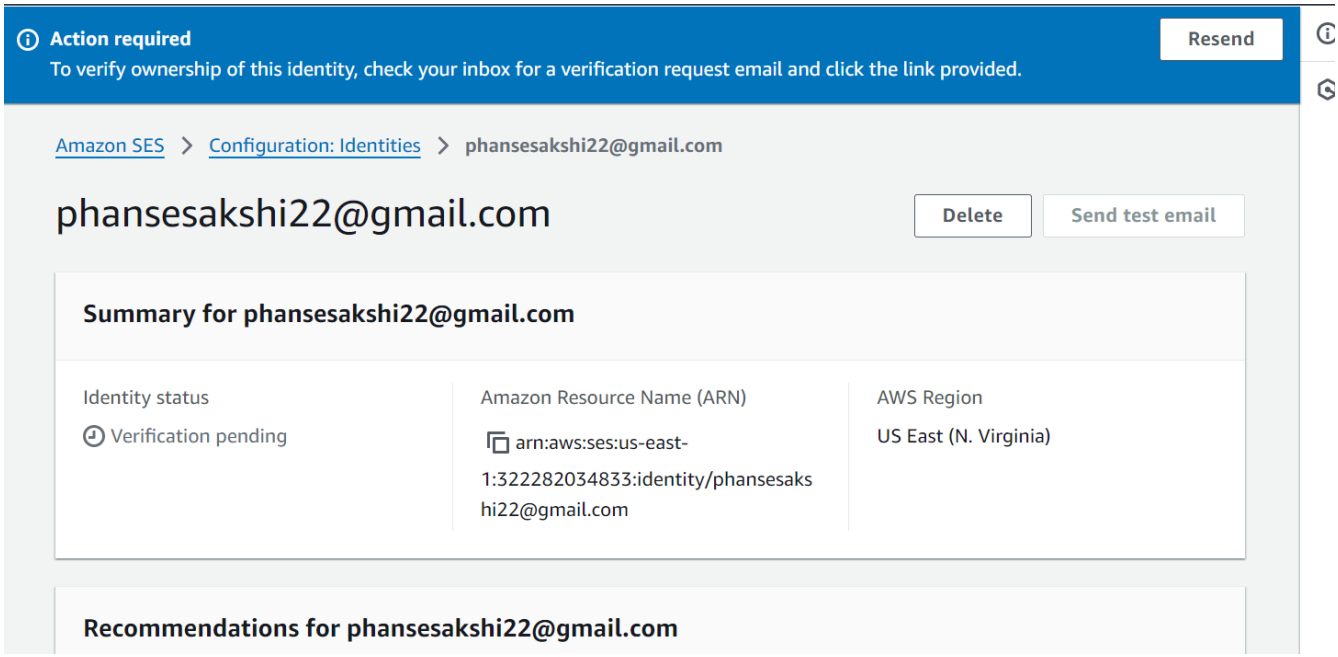
11. Sending a test email using Amazon SES

- Raw, successful delivery
- send test email
- go to view identities



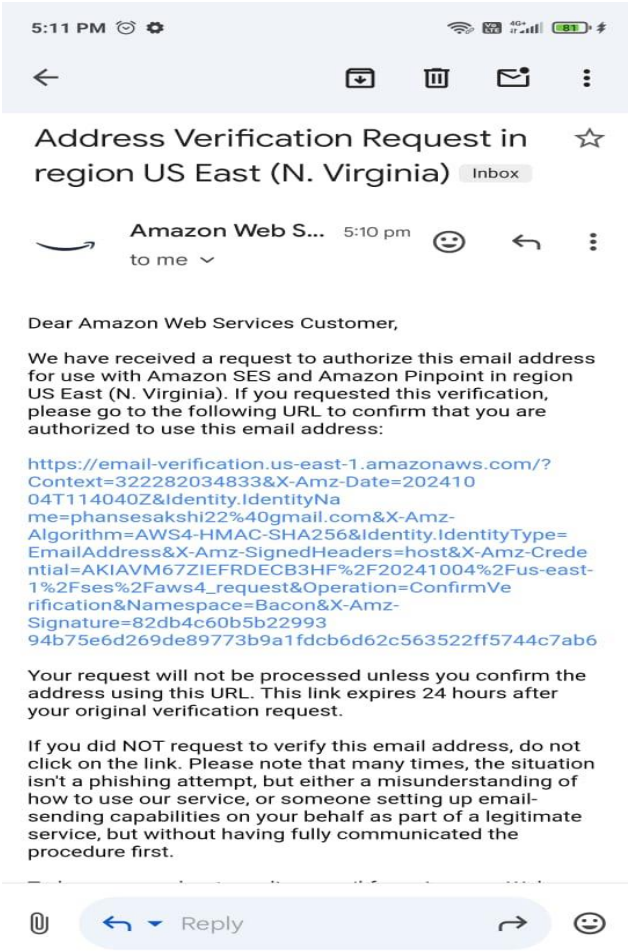
12. Validating email addresses to send to (a requirement for the sandbox)

- Email address
- send email (phansesakshi22@gmail.com)



13. Using Lambda to create and send personalized emails to SES

- Open Mail verification
- go to identities again
- check the verified mail



14. Adding connections between nodes in Eraser.io(diagram)

15. Creating a new Python Lambda function for email logic

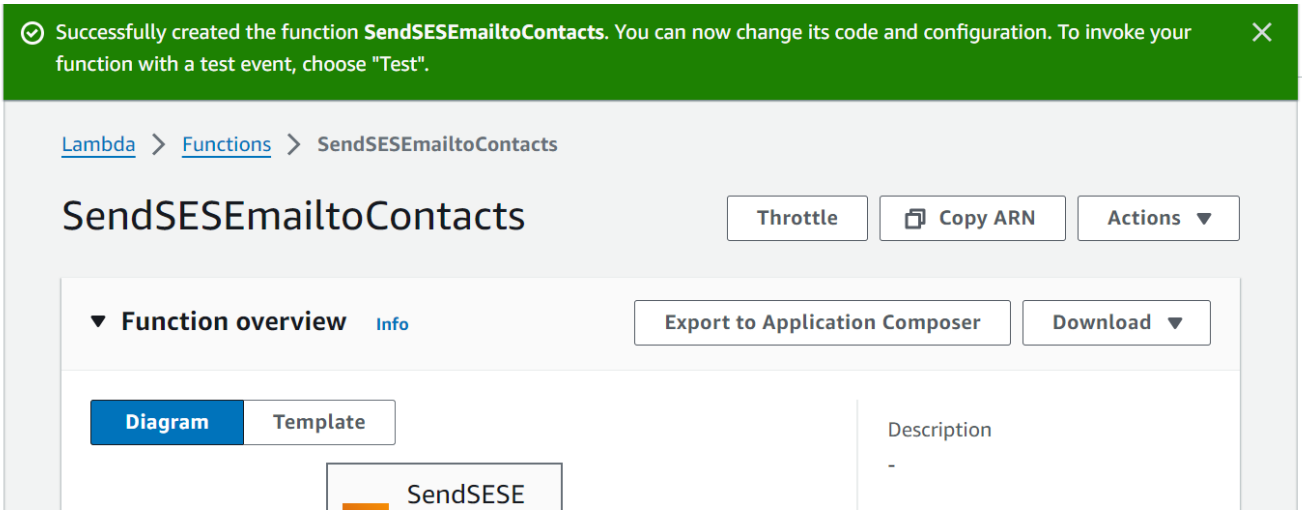
- open new tab
- Go to Lambda
- create new function

1. author from scratch
2. function name - sendSESemailtocontacts
3. use Python 3.1

- change default

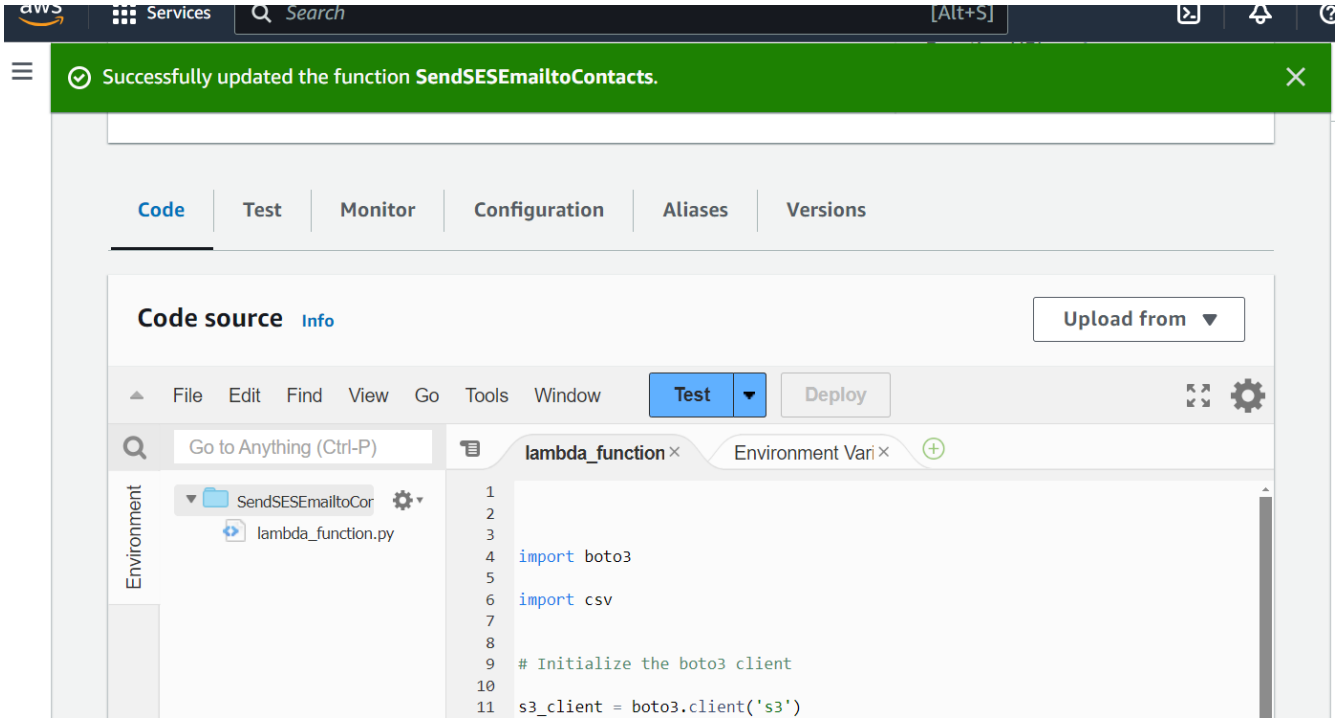
1. create new role
2. create function

- scroll down to scroll section
- from the file take Lambda code list page and paste it in that code
- change the mail address in source
- deploy

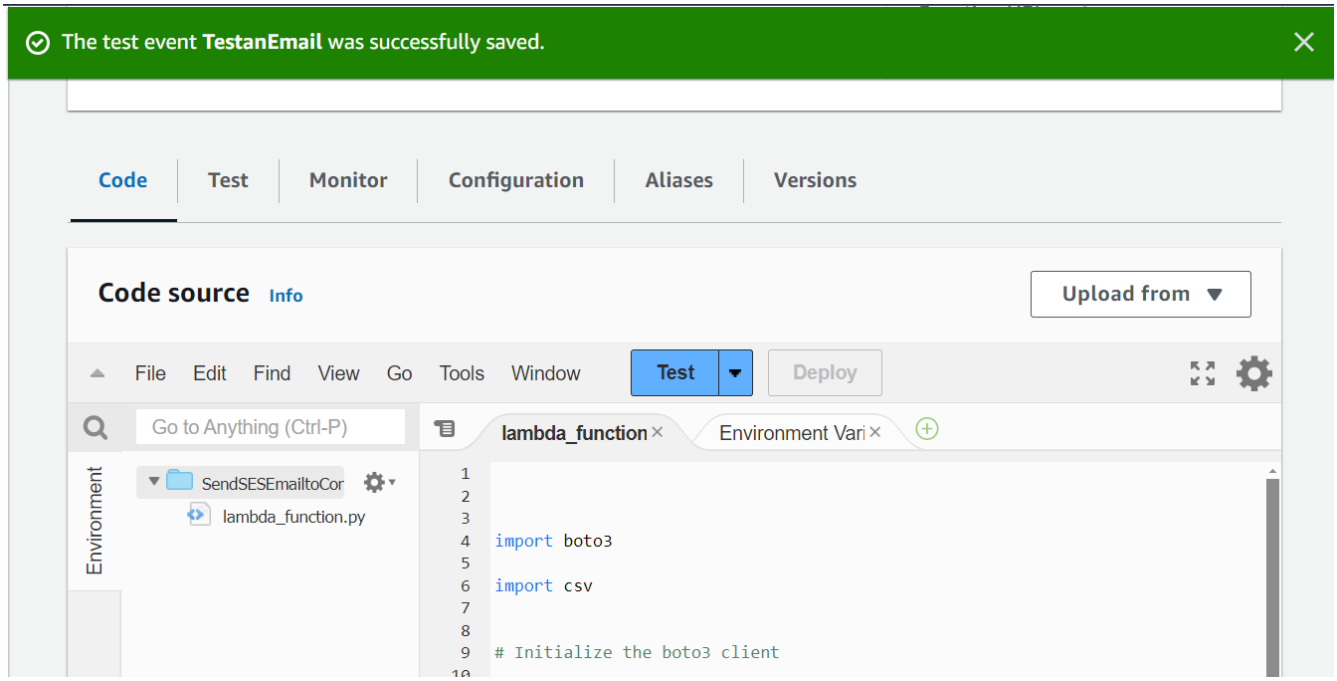


16. Configuring and running a test event for the Lambda function

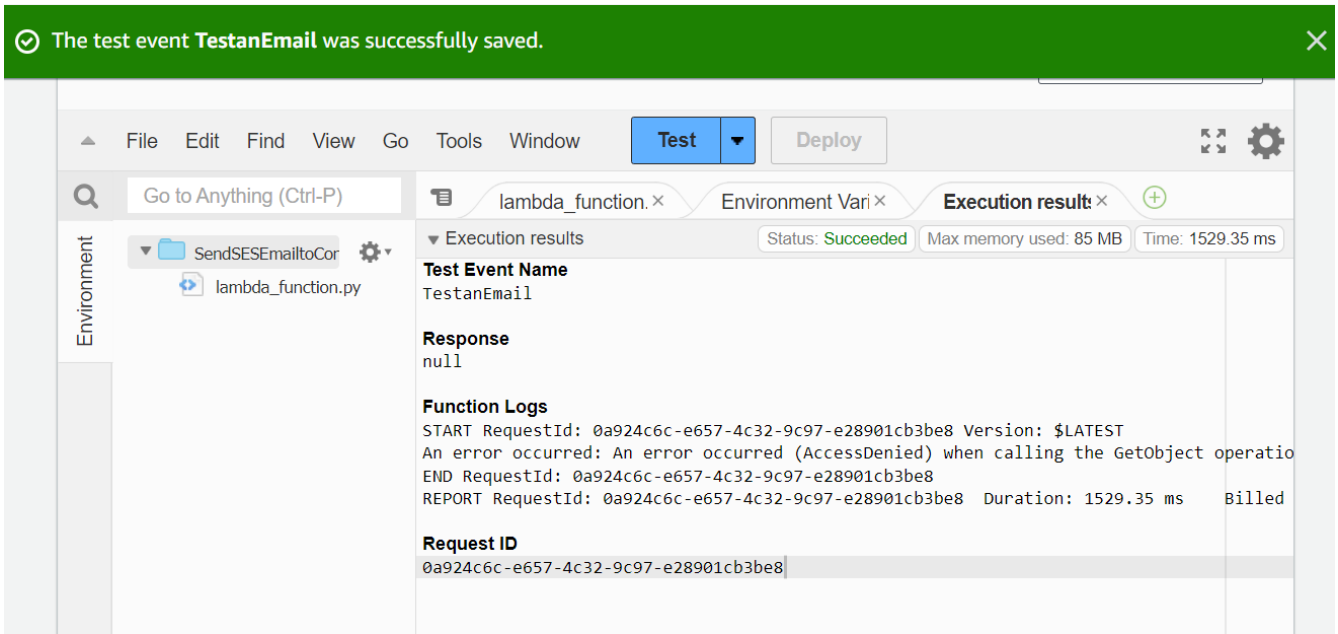
- Test – Configure
- Create new event
- Test an Email
- sharing private
- JSON code – from the file lambda function test event code and paste it.



Lambda test event



17. Get Object operation: Access Denied error when trying to retrieve from the S3 bucket from Lambda

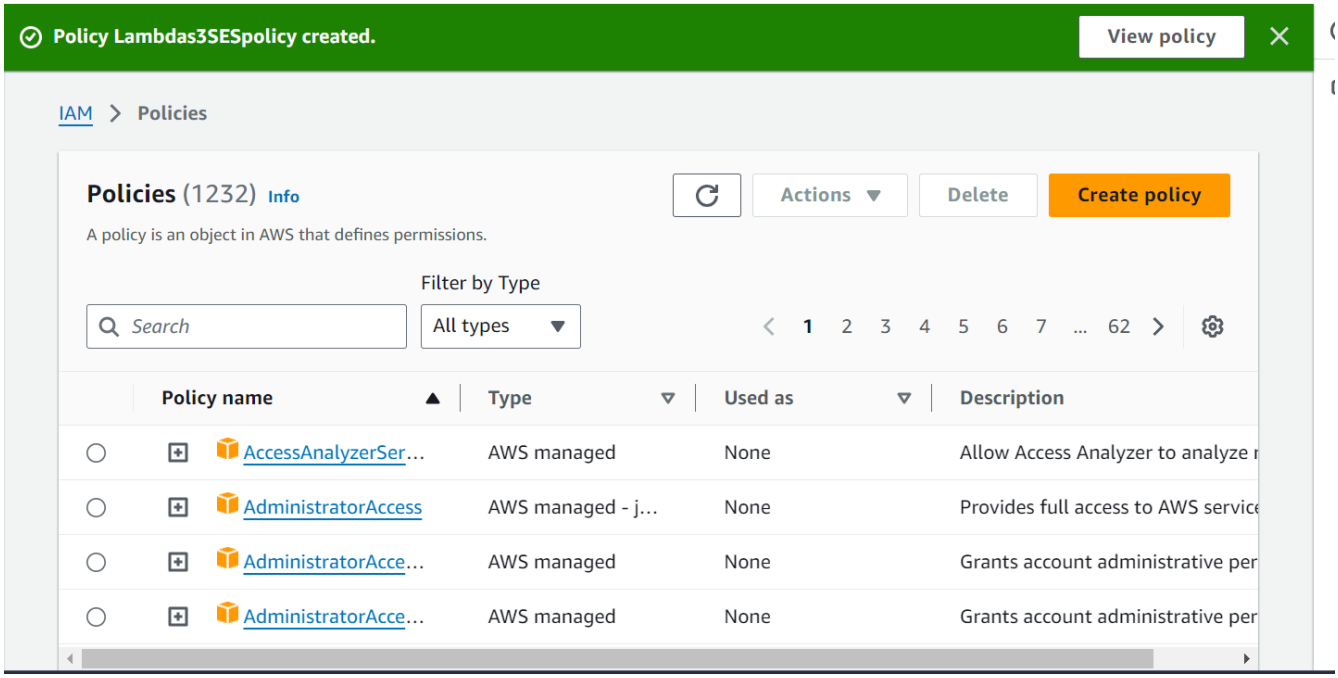


18. – Updating the Lambda execution role with appropriate permissions

- Configuration of Taste
- permissions
- role name

- click on link (tab)
- scroll to policy -click -code

19. Creating a new policy with permissions for S3 and SES
- Policies
- open new (tab)
- create new policy
- JSON and update code of IAM policy for SES and S3 permissions
- update bucket name – (ttt-email-marketing10) in code
- next
- policy name- lamdaS3SESpolicy
- create



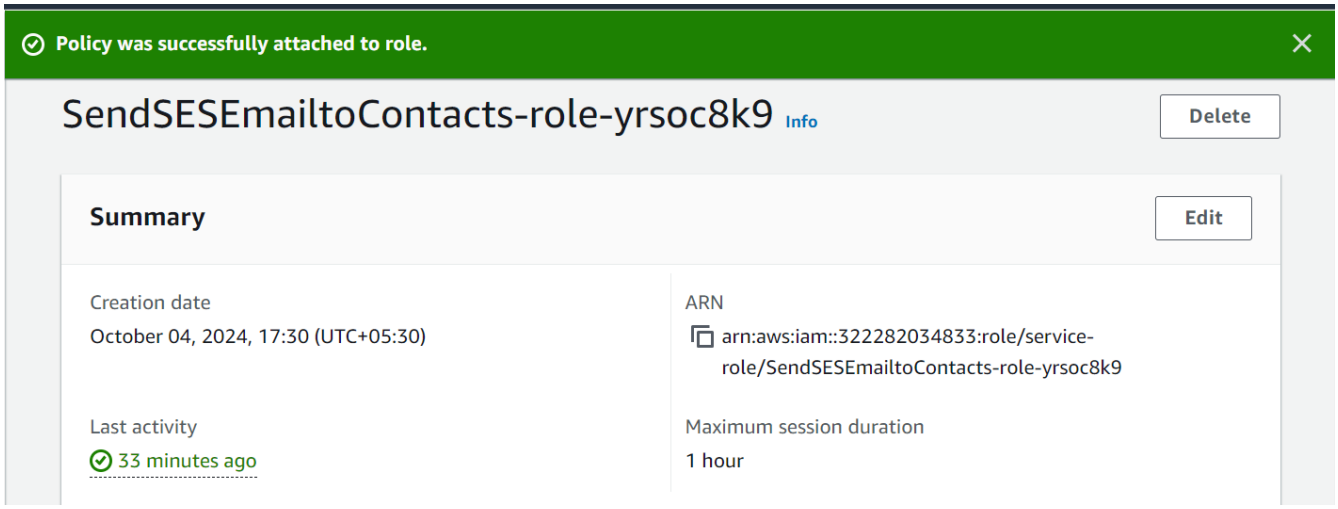
20. Attaching the policy to the Lambda execution role
- Go back to execution role
- Add permission
- Add policies (right corner) refresh
- search Lambda
- type -customer managed
- select s3sespolicylambda -all permissions

21. Testing the Lambda function again with updated permissions
- Go back to Lambda function
- code tab
- remove environment variable and execution result test
(check html page)

22. How to trigger the Lambda function to send emails on a schedule

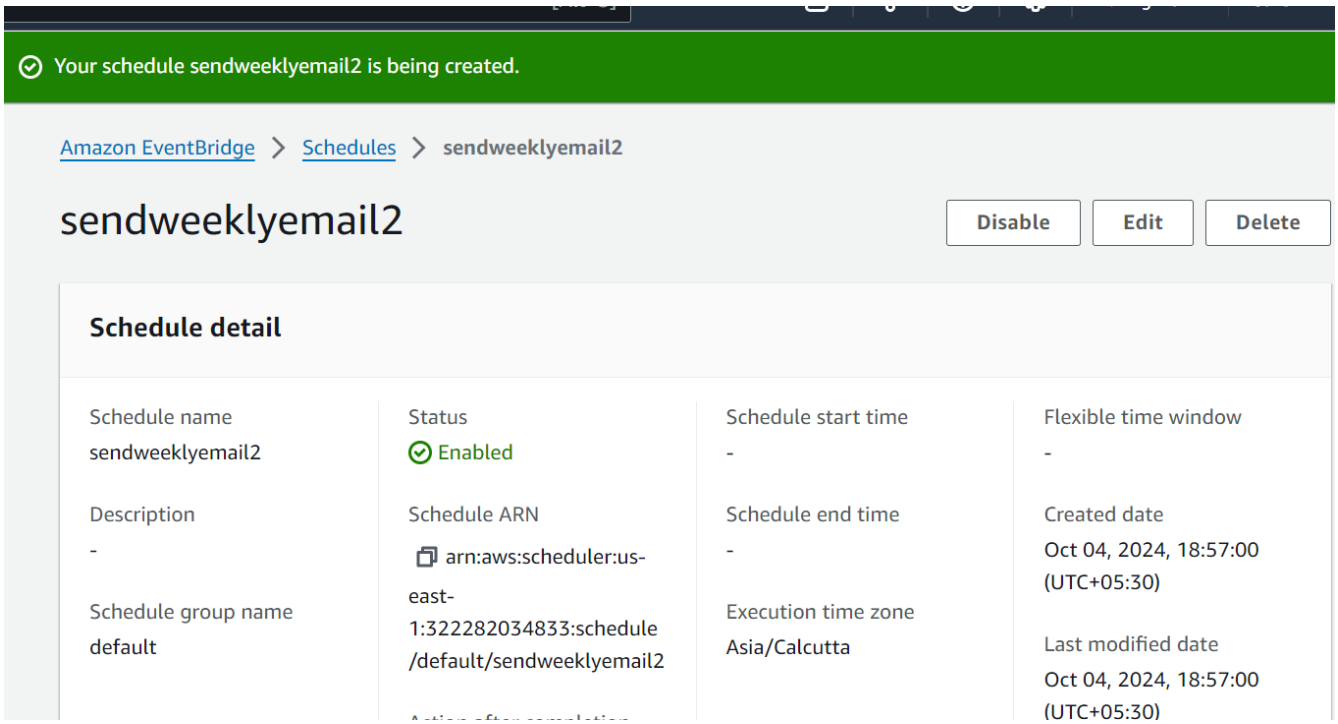
23. Creating a new schedule using EventBridge.

- Policies
- search Amazon event Bridge -schedule -Open name –(sendweeklyemail2)
- recurring schedule -select today's date 04/10/24 (15:28)
- select next Lambda invoke
- select (sendsesemailtocontacts) -next -action name -next
- connect schedule



24. Testing the EventBridge schedule for sending emails

- scroll -target – link -open in new tab
- monitor tab
- timezone-local



-View cloudswatch
Scroll-first link-click.

25. CHALLENGE: Ideas to enhance the application!
26. IMPORTANT! Deleting your resources