

**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

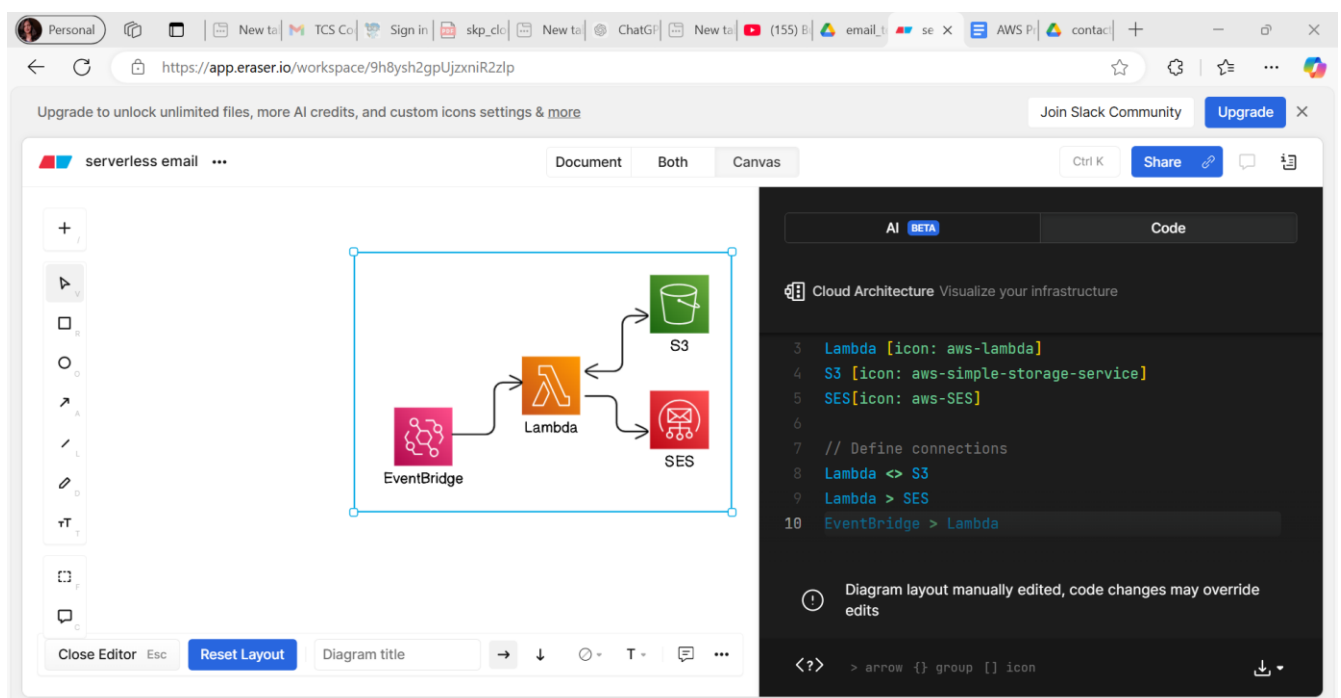
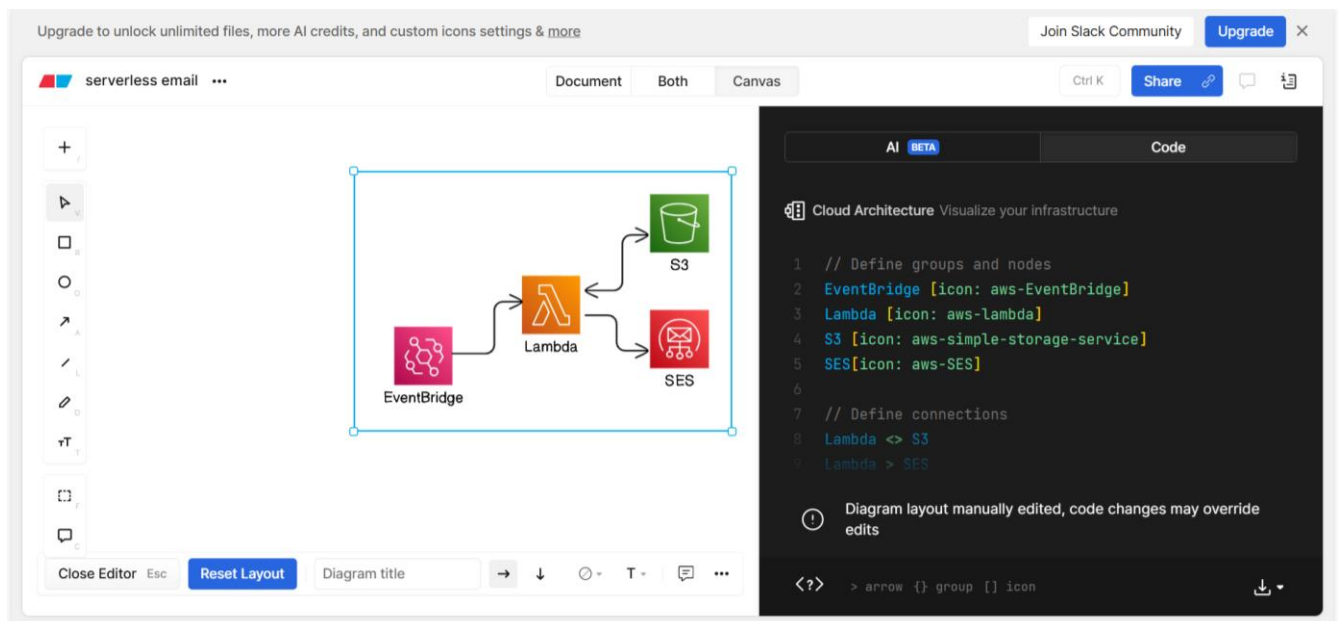
1. A piece to store email templates and list of contacts

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

### 3. Cloud Architecture



## 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

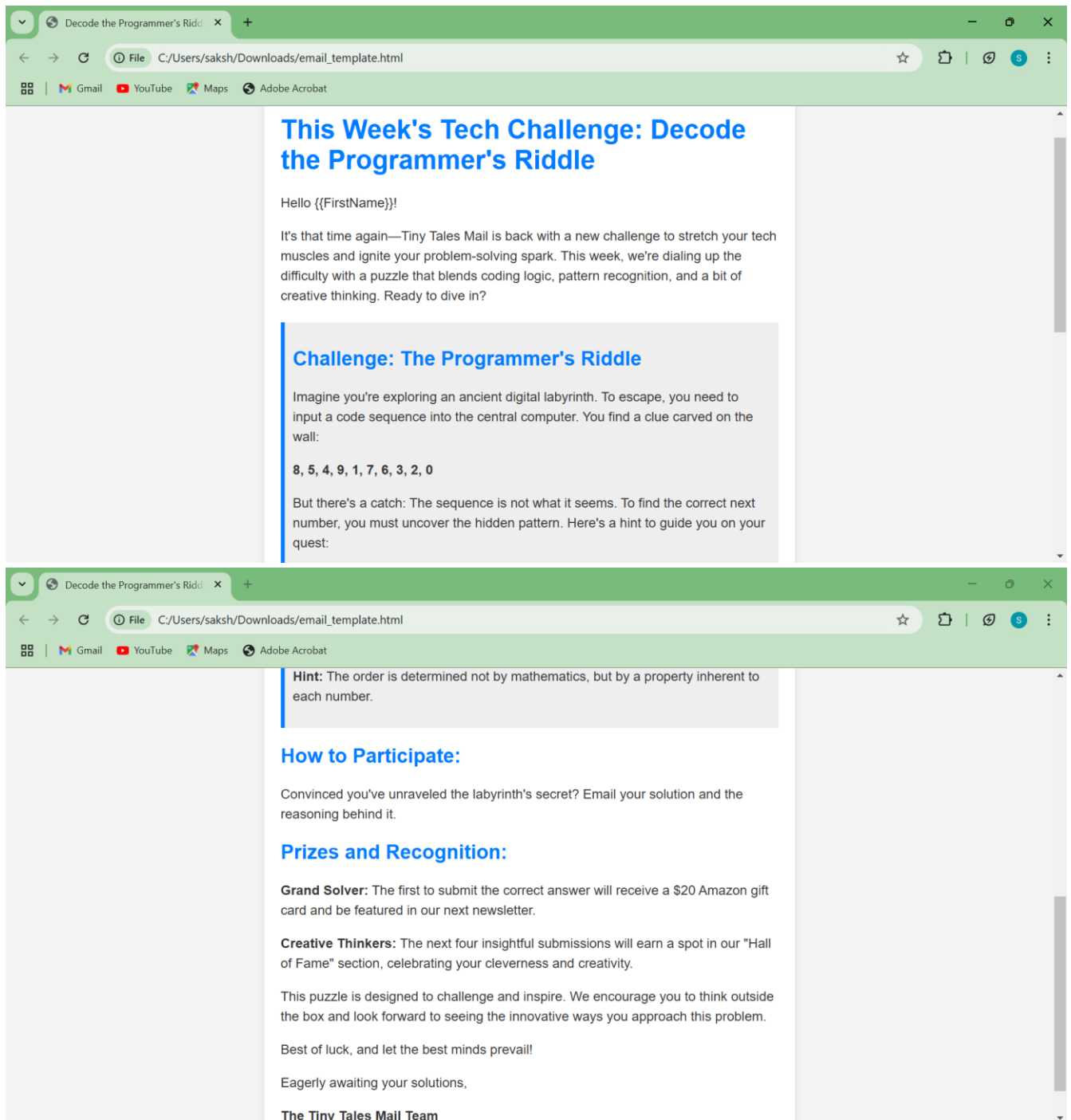
Choose the region for your bucket and open view details

The screenshot shows the AWS S3 console interface. At the top, a green banner indicates "Successfully created bucket 'ttt-email-marketing10'" with a "View details" button. Below this, a section for "Account snapshot" is visible. The main area is titled "General purpose buckets (1)" and includes a search bar, a table of buckets, and a "Create bucket" button. The table lists the bucket "ttt-email-marketing10" in the "US East (N. Virginia) us-east-1" region, with a creation date of "October 4, 2024, 15:20:07 (UTC+05:30)".

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">ttt-email-marketing10</a>	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	October 4, 2024, 15:20:07 (UTC+05:30)

## 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



## Code for HTML Template

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Decode the Programmer's Riddle - Tiny Tales Mail</title>
<style>
```

```

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 20px;
  background-color: #f4f4f4;
  color: #333;
}
.container {
  max-width: 600px;
  margin: auto;
  background: #ffffff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
h1, h2 {
  color: #007bff;
}
p {
  font-size: 16px;
  line-height: 1.5;
}
.challenge-details {
  background-color: #eee;
  border-left: 5px solid #007bff;
  padding: 10px;
  margin: 20px 0;
}
</style>
</head>
<body>
<div class="container">
  <h1>This Week's Tech Challenge: Decode the Programmer's Riddle</h1>
  <p>Hello {{FirstName}}!</p>
  <p>It's that time again—Tiny Tales Mail is back with a new challenge to stretch your tech muscles
  and ignite your problem-solving spark. This week, we're dialing up the difficulty with a puzzle that
  blends coding logic, pattern recognition, and a bit of creative thinking. Ready to dive in?</p>

  <div class="challenge-details">
    <h2>Challenge: The Programmer's Riddle</h2>
    <p>Imagine you're exploring an ancient digital labyrinth. To escape, you need to input a code
    sequence into the central computer. You find a clue carved on the wall:</p>
    <p><strong>8, 5, 4, 9, 1, 7, 6, 3, 2, 0</strong></p>
  </div>
  </div>
</body>
</html>

```

<p>But there's a catch: The sequence is not what it seems. To find the correct next number, you must uncover the hidden pattern. Here's a hint to guide you on your quest:</p>

<p><strong>Hint:</strong> The order is determined not by mathematics, but by a property inherent to each number.</p>

</div>

<h2>How to Participate:</h2>

<p>Convinced you've unraveled the labyrinth's secret? Email your solution and the reasoning behind it.</p>

<h2>Prizes and Recognition:</h2>

<p><strong>Grand Solver:</strong> The first to submit the correct answer will receive a \$20 Amazon gift card and be featured in our next newsletter.</p>

<p><strong>Creative Thinkers:</strong> The next four insightful submissions will earn a spot in our "Hall of Fame" section, celebrating your cleverness and creativity.</p>

<p>This puzzle is designed to challenge and inspire. We encourage you to think outside the box and look forward to seeing the innovative ways you approach this problem.</p>

<p>Best of luck, and let the best minds prevail!</p>

<p>Eagerly awaiting your solutions,</p>

<p><strong>The Tiny Tales Mail Team</strong></p>

</div>

</body>

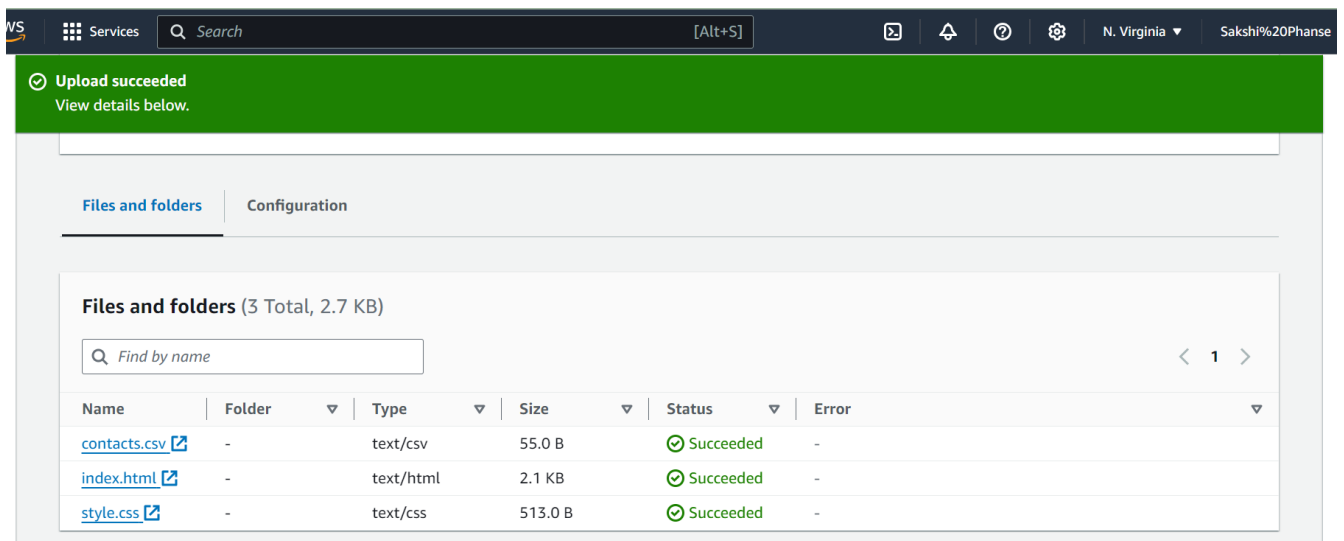
</html>

## 6. The CSV file with contact information to email to

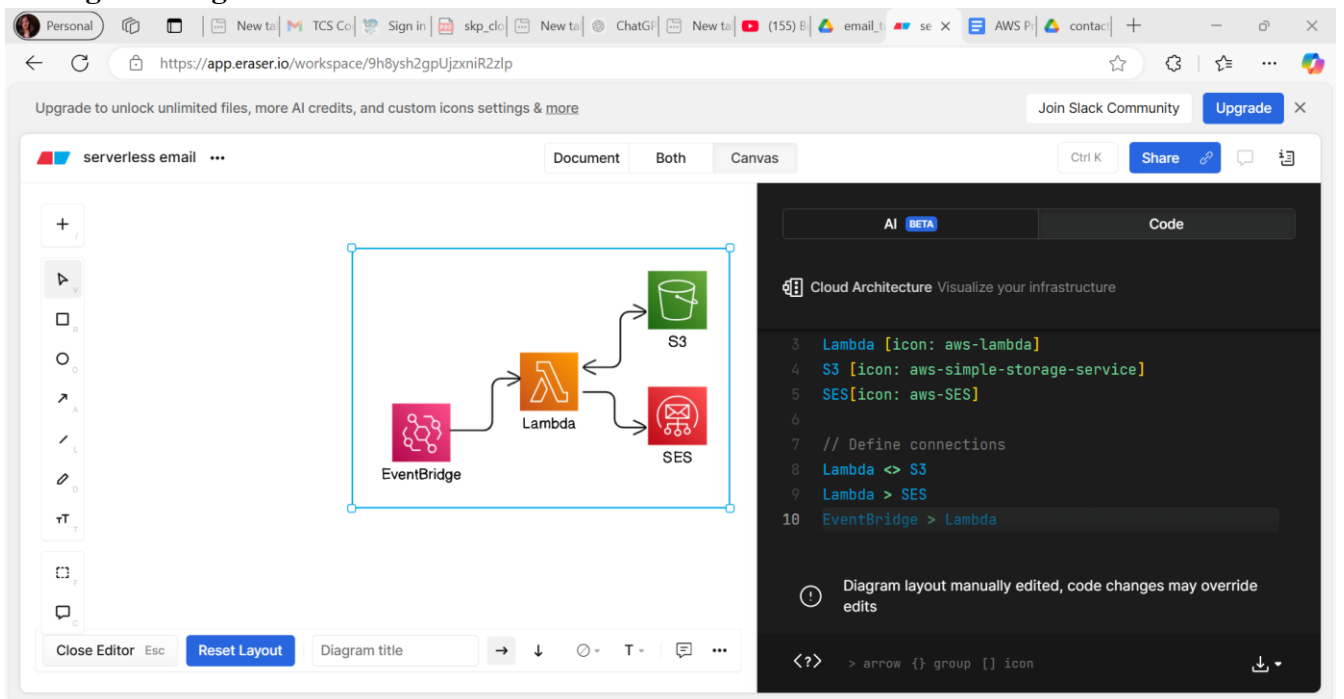
	A	B	C	D	E
1	FirstName	Email			
2	sakshi	<a href="mailto:sakshiphanse22@gmail.com">sakshiphanse22@gmail.com</a>			
3	phanse	<a href="mailto:phansesakshi22@gmail.com">phansesakshi22@gmail.com</a>			
4					

## 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 (sakshipanse22@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	Daily sending quota	Maximum send rate
 <a href="#">Sandbox</a>	200 emails per 24-hour period	1 email per second





### Verify emails to send the emails

#### Amazon Web Services – Email Address Verification Request in region US East (N. Virginia)

Inbox x



**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&X-Amz-Credential=AKIAVM67ZIEFRDECB3HF%2F20241004%2Fus-east-1%2Fses%2Faws4\\_request&Operation=ConfirmVerification&Namespace=Bacon&X-Amz-Signature=a297641d1c7e96189268160daa4a320284504d495b62f96c131656341edb0449](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&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!

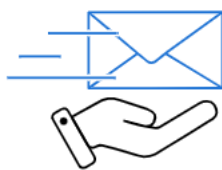
You have successfully verified an email address. You can now start sending email from this address.

DKIM records		
DKIM-signed messages help receiving mail servers validate that a message was not forged or altered in transit. <a href="#">Learn more</a>		
Type	Name	Value
CNAME	utjei2w5hykxb225dvpfjydyj3kyswnh._domainkey.sakshiphanse22.com	utjei2w5hykxb225dvpfjydyj3kyswnh.dkim.amazonses.com
CNAME	ndajo7dhnj6rsdb5ktpmxtjkvegiuwaq._domainkey.sakshiphanse22.com	ndajo7dhnj6rsdb5ktpmxtjkvegiuwaq.dkim.amazonses.com
CNAME	ayldr6xxvgihwjcvp2viqsbenkvaks6._domainkey.sakshiphanse22.com	ayldr6xxvgihwjcvp2viqsbenkvaks6.dkim.amazonses.com

MAIL FROM records		
When using a custom MAIL FROM domain, messages sent through Amazon SES will be marked as originating from your domain instead of a subdomain of amazon.com. <a href="#">Learn more</a>		
Type	Name	Value
No MAIL FROM records found		
To configure MAIL FROM records go to <a href="#">sakshiphanse22.com</a> .		

## 11. Sending a test email using Amazon SES

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



✓ Verified

### Verify email address

**sakshiphanse22@gmail.com**

Check your inbox to verify ownership of this email address.

[View all identities](#)

---

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

- Email address
- send email ([phansesakshi22@gmail.com](mailto:phansesakshi22@gmail.com))

Action required

To verify ownership of this identity, check your inbox for a verification request email and click the link provided.

Resend

Amazon SES

>

Configuration: Identities

>

phansesakshi22@gmail.com

phansesakshi22@gmail.com

Delete

Send test email

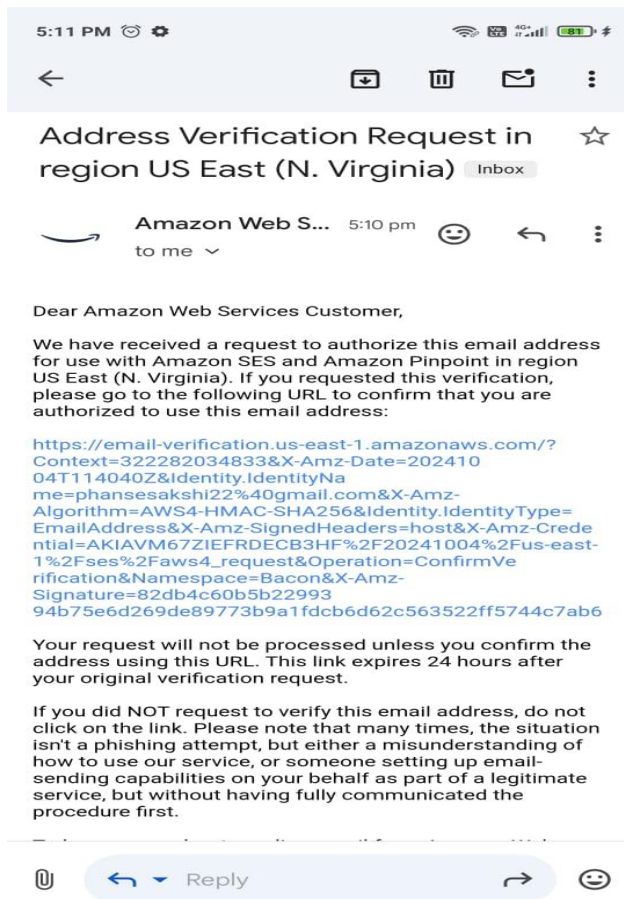
Summary for phansesakshi22@gmail.com

<div>Identity status</div> <div><div></div>Verification pending</div>	<div>Amazon Resource Name (ARN)</div> <div><div></div>arn:aws:ses:us-east-1:322282034833:identity/phansesaks hi22@gmail.com</div>	<div>AWS Region</div> <div>US East (N. Virginia)</div>
---	---	--

Recommendations for phansesakshi22@gmail.com

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

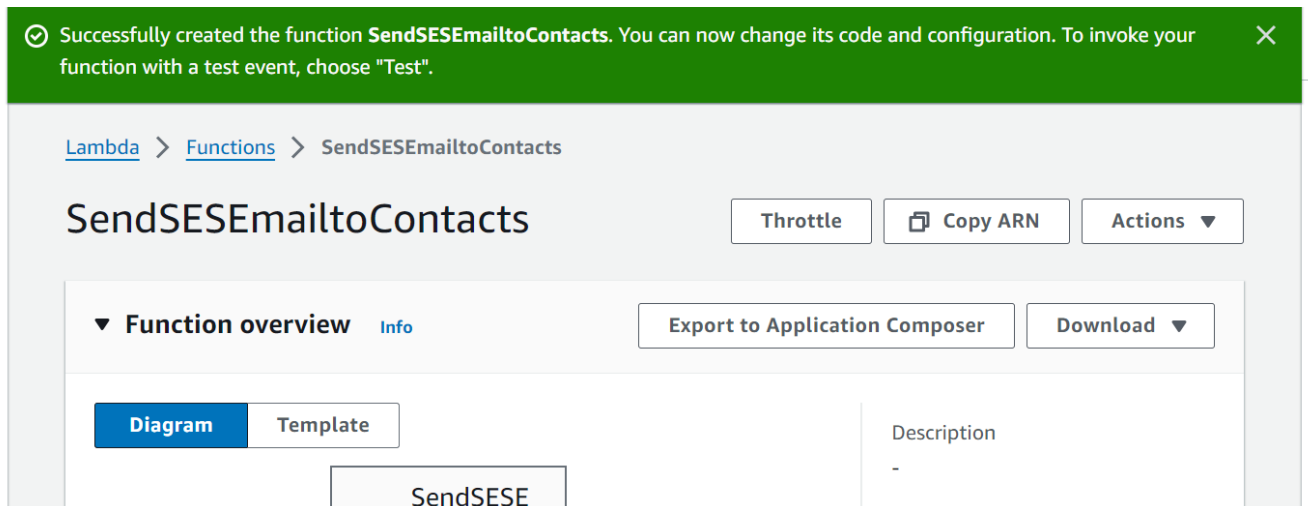
- Open Mail verification
- go to identities again
- check the verified mail



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

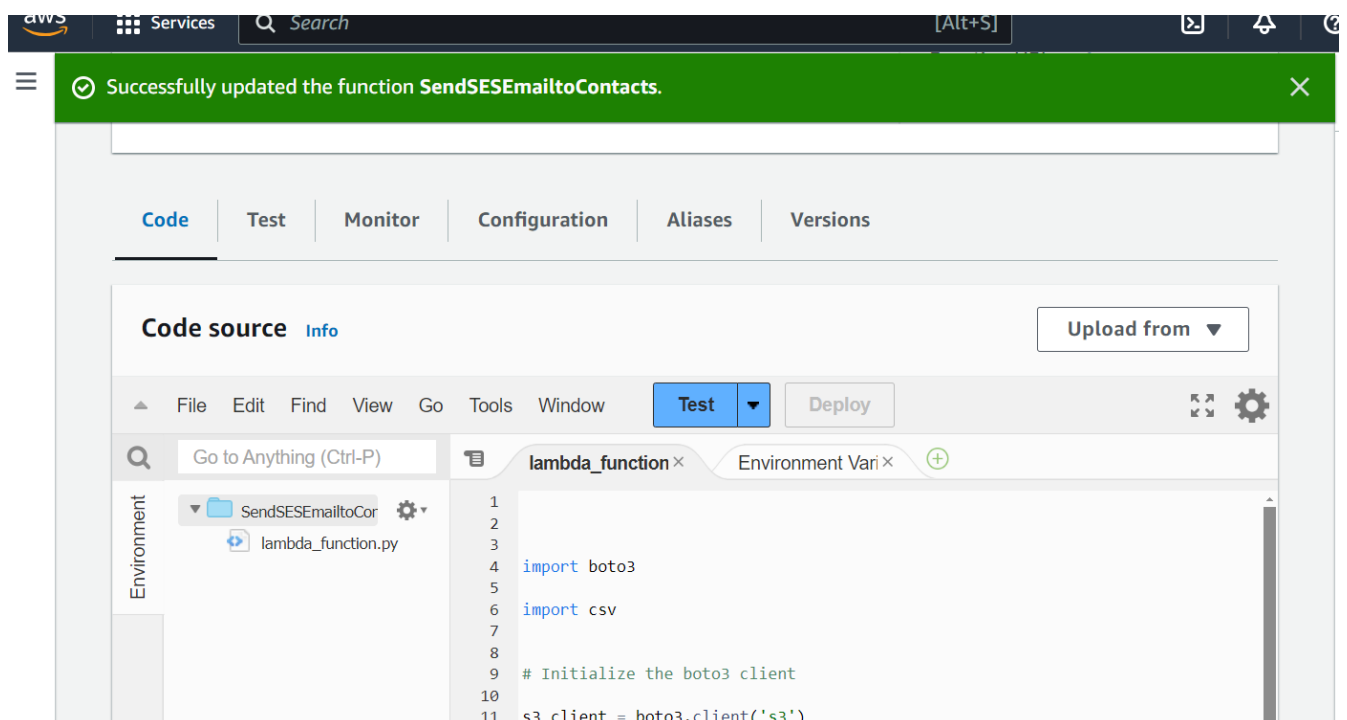
#### 15. Creating a new Python Lambda function for email logic

- 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 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-marketing10' # 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()

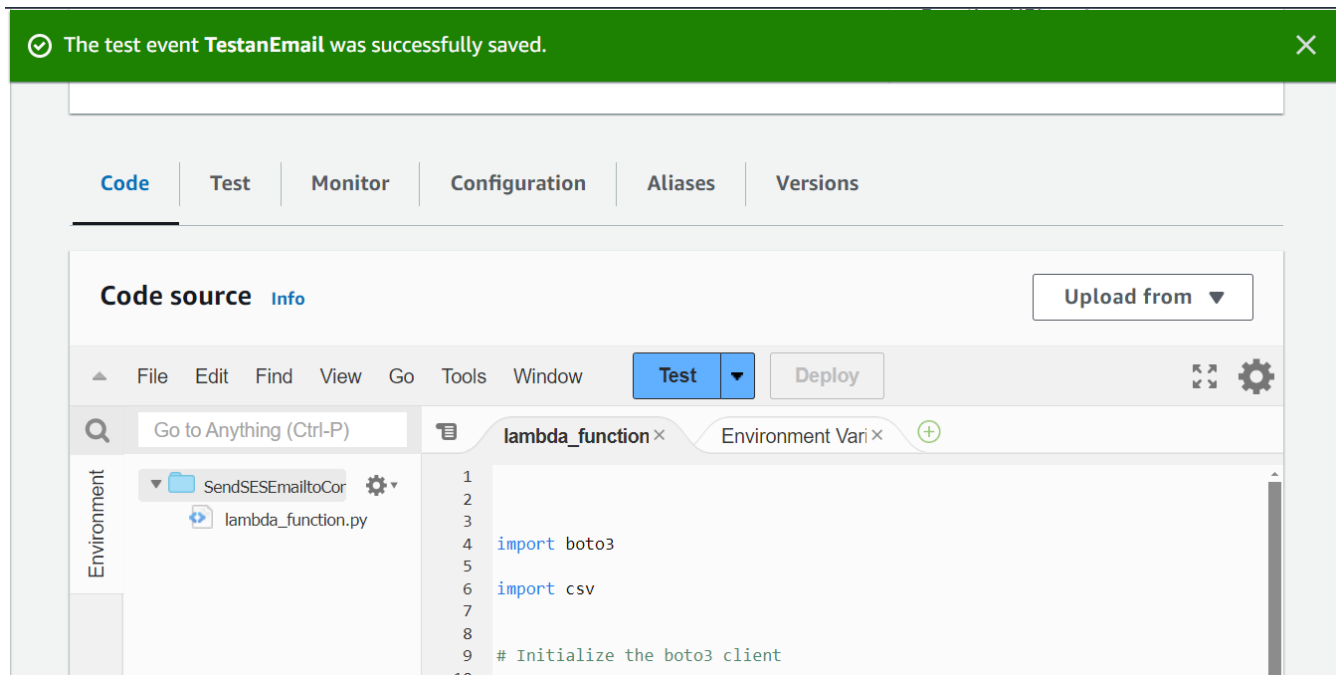
        # Retrieve the HTML email template from S3
        email_template = s3_client.get_object(Bucket=bucket_name, Key='email_template.html')
        email_html = email_template['Body'].read().decode('utf-8')

        # 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='sakshiphanse22@gmail.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}')
```

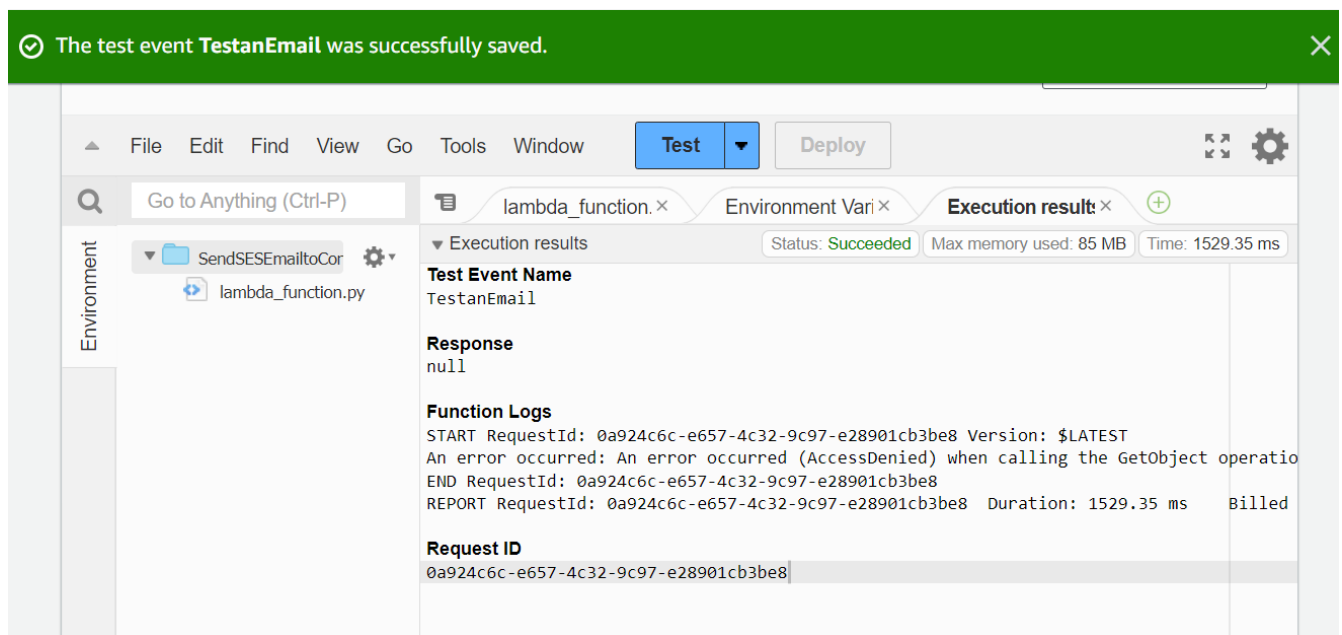
## Lambda test event



## Lambda Function Test Event

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

## 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
- JSON Editor

### Paste code-

### 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-marketing10/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
    }
  ]
}
```

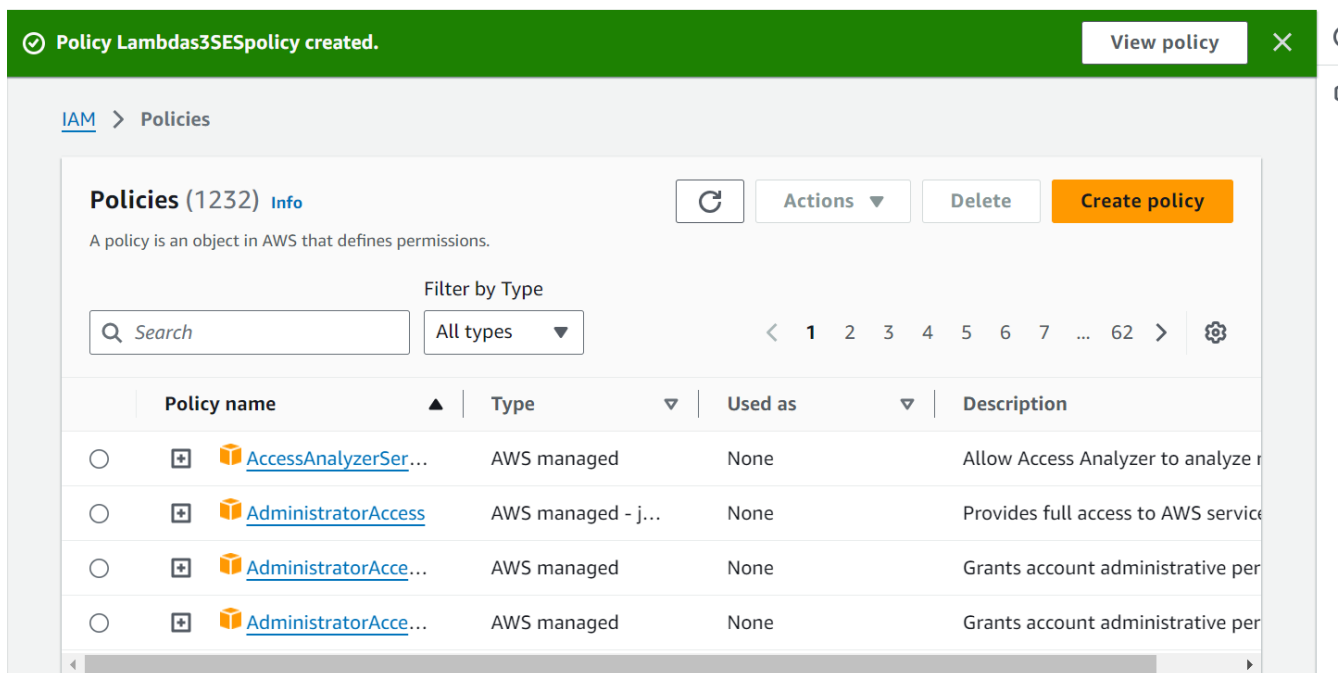
```

    "Resource": "*"
  }
]
}

```

## 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 (sendsemailtocontacts) -next -action name -next
- connect schedule



 Policy was successfully attached to role.

SendSESEmailtoContacts-role-yrsoc8k9
[Info](#)

Delete

Summary

Edit

<div>Creation date</div> <div>October 04, 2024, 17:30 (UTC+05:30)</div>	<div>ARN</div> <div>  arn:aws:iam::322282034833:role/service-role/SendSESEmailtoContacts-role-yrsoc8k9 </div>
<div>Last activity</div> <div>  33 minutes ago </div>	<div>Maximum session duration</div> <div>1 hour</div>

### 24. Testing the EventBridge schedule for sending emails

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

✔ Your schedule sendweeklyemail2 is being created.

[Amazon EventBridge](#) > [Schedules](#) > sendweeklyemail2

## sendweeklyemail2

Disable

Edit

Delete

### Schedule detail


Schedule name  
sendweeklyemail2

Status  
✔ Enabled

Schedule start time  
-

Flexible time window  
-

Description  
-

Schedule ARN  
 arn:aws:scheduler:us-east-1:322282034833:schedule/default/sendweeklyemail2

Schedule end time  
-

Created date  
Oct 04, 2024, 18:57:00 (UTC+05:30)

Schedule group name  
default

Execution time zone  
Asia/Calcutta

Last modified date  
Oct 04, 2024, 18:57:00 (UTC+05:30)

-View cloudswatch  
Scroll-first link-click.

**25. CHALLENGE: Ideas to enhance the application!**

**26. IMPORTANT! Deleting your resources**