**Mass Emailing System using AWS Services**

*Prepared in the partial fulfilment of the**Summer Internship Program on AWS*

AT



*Under the guidance of*

# *Mr. Gopi Raju, APSSDC Mr. Anil Kumar, APSSDC*

*Submitted by*

**22B81A0566- IBBA SANJEEV KUMAR**

SIR C R REDDY COLLEGE OF ENGINEERING, AP

(July-2024)

# ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of my summer internship project at **Andhra Pradesh Skill Development Corporation (APSSDC)**. This opportunity has been an enriching and transformative experience for me, and I am truly thankful for the support, guidance, and encouragement I have received along the way.

First and foremost, I extend my sincere regards to Mr.Gopiraju and Mr.Anil Kumar, my supervisors and mentors, for providing me with valuable insights, constant guidance, and unwavering support throughout the duration of the internship. Their expertise and encouragement have been instrumental in shaping the direction of this project.

I would like to thank the entire team at **Andhra Pradesh Skill Development Corporation (APSSDC)** for fostering a collaborative and innovative environment. The camaraderie, knowledge sharing, and feedback I received from my colleagues significantly contributed to the development and success of this project.

In conclusion, I am honoured to have been a part of this internship program, and I look forward to leveraging the skills and knowledge gained to contribute positively to future endeavours.

Thank you.

Sincerely,

**IBBA SANJEEV KUMAR**,

**22B81A0566** , i.sanjeevkumar352@gmail.com

# ABSTRACT

The "Automated Mass Emailing System using AWS Services" project is designed to streamline and automate large-scale email communication for businesses and organizations. Leveraging the power of Amazon Web Services (AWS), this project integrates AWS Lambda, Simple Email Service (SES), CloudWatch, and IAM to create a robust and scalable solution. The core objective of this project is to provide a reliable platform for sending mass emails efficiently, whether for marketing campaigns, newsletters, or important announcements. Mass emailing is a common requirement for businesses, organizations, and individuals alike. However, sending a large number of emails manually can be timeconsuming and error-prone. One way to automate this process is by using Amazon Web Services. AWS Lambda is a serverless computing service that allows you to run code without provisioning or managing servers.

CloudWatch-CloudWatch, it is a monitoring service that provides data and actionable insights for AWS resources.

IAM (Identity and Access Management) is a service that helps you securely control access to AWS resources.IAM (Identity and Access Management) is a service that helps you securely control access to

AWS resources.

Amazon Simple Email Service (SES) is a cloud-based email sending and receiving service provided by Amazon Web Services (AWS). It offers a reliable and scalable solution for businesses and developers to send and receive email messages.

The "Automated Mass Emailing System using AWS Services" project not only simplifies the process of sending mass emails but also enhances the reliability, scalability, and security of email communication. It provides organizations with a cost-effective solution for reaching their target audience effectively and efficiently, while adhering to best practices in email delivery and management.This project empowers businesses to harness the capabilities of AWS to optimize their email communication strategies, ultimately contributing to improved customer engagement and outreach.

This project empowers businesses to harness the capabilities of AWS to optimize their email communication strategies, ultimately contributing to improved customer engagement and outreach.

# TABLE OF CONTENTS

1. INTRODUCTION..................................................................................................... 5-6

2.METHODOLOGY......................................................................................................7-8

3.3.ARCHITECTURE...................................................................................................9-10

4.IMPLEMENTATION.................................................................................................11-18

5.RESULT......................................................................................................................19

6.CONCLUSION...........................................................................................................20

# 1.INTRODUCTION

In the digital age, effective communication through email is pivotal for businesses and organizations to reach their target audience, whether it's for marketing, newsletters, or important announcements. However, managing and sending mass emails at scale can be a complex and resource-intensive task. The " Mass Emailing using AWS Services" project addresses this challenge by harnessing the power of Amazon Web Services (AWS) to create an efficient and automated solution.

This project aims to streamline and simplify the process of sending mass emails while ensuring security, scalability, and reliability. By integrating key AWS services such as AWS Lambda, Simple Email Service (SES), CloudWatch, and IAM, this system offers a comprehensive solution for organizations seeking to enhance their email communication strategies.

Here are the primary objectives of this project:

**Efficiency:**

By automating the email sending process, organizations can save time and resources that would otherwise be spent on manual email campaigns. This system allows for the quick and efficient dissemination of information to large recipient lists.

**Scalability:**

The project leverages the scalability of AWS services, enabling organizations to send emails to thousands or even millions of recipients without worrying about infrastructure limitations.

**Reliability:**

AWS SES, known for its high deliverability rates, ensures that emails reach their intended recipients' inboxes, minimizing the risk of emails being marked spam. Additionally, CloudWatch monitoring provides insights into system performance, allowing for proactive issue resolution.

**Security:**

IAM roles and permissions are carefully managed to guarantee the security of sensitive email data and AWS resources. This system adheres to industry best practices for secure email communication.

**Automation:**

CloudWatch Events and AWS Lambda enable the automation of email sending tasks, such as scheduling email campaigns or triggering emails based on specific events, reducing manual intervention.

By implementing the "Automated Mass Emailing System using AWS Services," organizations can not only optimize their email communication but also ensure that their messages are delivered reliably and securely. This project empowers businesses to focus on content and strategy while AWS handles the technical aspects of email delivery, ultimately contributing to improved customer engagement and outreach.



***Figure 1Services used in this project***

# 2.METHODOLOGY

The development and implementation of the " Mass Emailing System using AWS Services" project involve several key phases and steps. This methodology outlines the systematic approach to building and deploying the system effectively**.**

**2.1Requirements Gathering:**

The requirement gathering phase for the "Automated Mass Emailing System using AWS Services" project involves identifying the key needs and specifications of the system. This includes determining the types of emails to be sent, recipient list management, scheduling requirements, integration with AWS services like SES and CloudWatch, security and compliance considerations, and performance expectations. Additionally, understanding the user roles and access control requirements, as well as any regulatory compliance needs is crucial. Clear documentation of these requirements will guide the project's development and ensure that the final system aligns with the organization's goals for efficient, secure, and scalable mass email communication.

**2.2Design:**

The project, "Automated Mass Emailing System using AWS Services," is designed to automate and optimize large-scale email communication. It leverages Amazon Web Services (AWS) components such as Lambda, Simple Email Service (SES), CloudWatch, and IAM. AWS Lambda hosts the email-sending function, while SES handles email delivery with high deliverability rates. CloudWatch monitors performance, and IAM ensures secure access. The system enables efficient email composition, recipient list management, and scheduling through CloudWatch Events. Its scalable, reliable, and secure architecture enhances email communication while adhering to compliance requirements, making it an invaluable tool for businesses and organizations seeking efficient and effective mass email outreach**.**

**2.3Implementation:**

The implementation of the "Automated Mass Emailing System using AWS Services" involves:

* Create an AWS account and IAM roles for secure access.
* Develop a Lambda function for email composition, recipient management, SES integration, and error handling.
* Verify sender domains, configure SES, and ensure compliance.
* Set up CloudWatch alarms and logs for monitoring.
* Create rules to trigger the Lambda function automatically.
* Thoroughly test the system for email campaigns, deliverability, and error handling.
* Deploy to the production environment and configure scaling.
* Document setup, maintenance, and provide training.
* Manage IAM roles and permissions, ensuring compliance.
* Continuously monitor and optimize system performance and efficiency**.** 
  1. **Testing**

Testing detects the faults in the process, therefore by the knowledge of the faults in the process and recovering the project from them can help us to proceed further in the development.

* 1. **Deployment**

We have deployed Python code in a way that it should return an email and we have written the messages which needs to be placed in the email.

* 1. **Iterative Refinement**

Throughout the development lifecycle, an iterative approach was adopted to incorporate booking details, city details, and refine the system. Regular interactions with artificial intelligence provided valuable insights that guided refinements and enhancements.

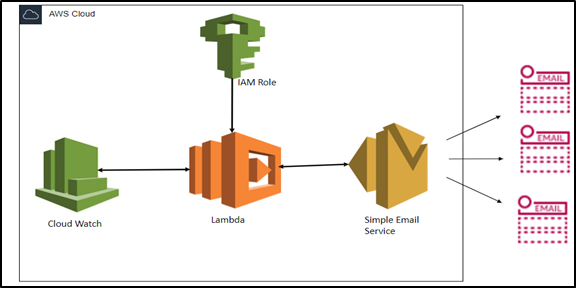
The methodology employed in the development of the Mass Emailing creation facilitated a comprehensive and user-centric approach, ensuring the creation of a functional, secure, and intuitive platform for customers to interact with the agents directly to use email.

"Automated Mass Emailing System using AWS Services," is designed to automate and optimize largescale email communication. It leverages Amazon Web Services (AWS) components such as Lambda, Simple Email Service (SES), CloudWatch, and IAM. AWS Lambda hosts the email-sending function, while SES handles email delivery with high deliverability rates.

# 3.ARCHIETURE

The architecture for a mass emailing project using AWS Lambda and SES (Simple Email Service) involves several components that work together to enable scalable and cost-effective email sending.

1. **AWS Lambda Functions:** You create AWS Lambda functions to handle various aspects of your email sending process. These functions are event-driven and can be triggered by different sources like API Gateway, S3, or a custom trigger. Lambda functions will play a central role in processing and sending emails.
2. **Email List and Data Source:** You need a data source (e.g., a database or CSV file in Amazon S3) containing the email addresses and other relevant information of your recipients. Lambda functions will fetch and process this data.
3. **SES Integration:** AWS SES is the core email delivery service. You'll configure SES to handle the email delivery. SES provides features for bounce and complaint handling, email authentication (DKIM, SPF, and DMARC), and email templates.
4. **Lambda-SES Integration:** Your Lambda functions will integrate with SES to send emails. When triggered, these functions will generate personalized email content and use SES to deliver the messages. SES ensures high deliverability and compliance with email sending standards.
5. **DynamoDB for Tracking:** Optionally, you can use Amazon DynamoDB to track email sending status and delivery outcomes. This allows you to monitor which emails were successfully delivered and handle any issues or retries for failed deliveries.
6. **Logging and Monitoring:** Implement AWS CloudWatch for logging and monitoring Lambda function performance, SES sending statistics, and any error handling. This provides visibility into your email sending operations.
7. **Security and IAM Roles:** Ensure proper IAM (Identity and Access Management) roles and permissions for Lambda functions to interact with SES and other AWS resources securely. Follow AWS best practices for security.
8. **Cost Control:** AWS offers a pay-as-you-go pricing model. Properly configure billing alarms in CloudWatch to monitor your email sending costs and stay within budget.
9. **Scaling and Load Balancing:** AWS Lambda and SES automatically scale with your email sending needs. No need to worry about provisioning or managing servers.
10. **Testing and Deployment:** Before deploying your mass emailing solution, thoroughly test it with a smaller dataset to ensure that everything works as expected. You can also use AWS CodePipeline for continuous integration and deployment.



***Figure 2 Architecture diagram of mass emailing***

The architecture diagram above shows how an event is triggered from CloudWatch, captured by a lambda function, and then passed to Amazon SES, which sends emails to the Users.

This architecture leverages the serverless capabilities of AWS Lambda and the robust email delivery features of SES to create a scalable, reliable, and cost-efficient solution for mass emailing while ensuring high email deliverability rates and compliance with best practices.

# 4.IMPLEMENTATION

***SERVICES USED:***

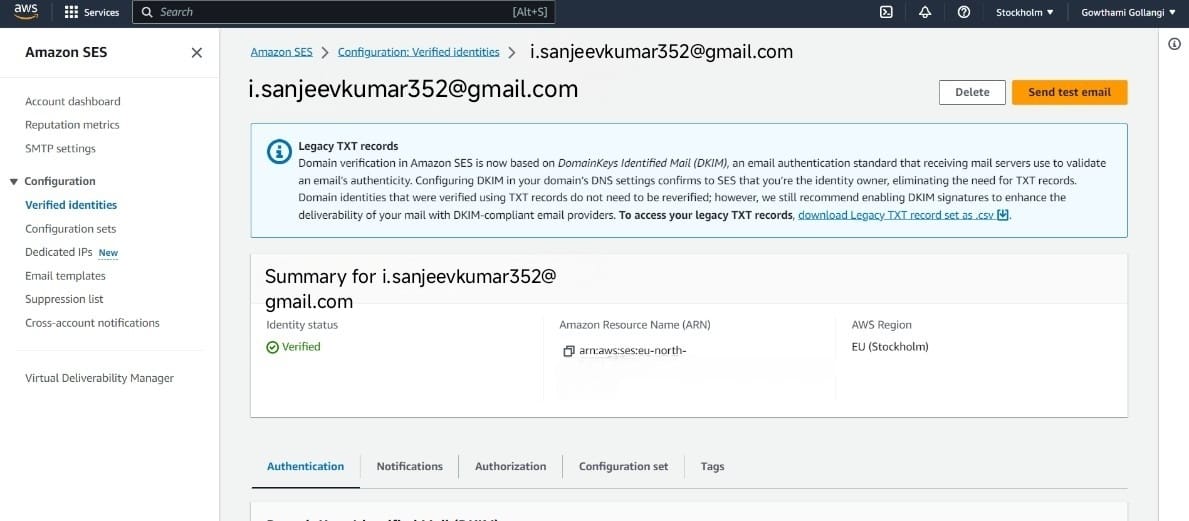
* Identity and Access Management (IAM)
* Lambda function
* Simple Email Service (SES)
* CloudWatch

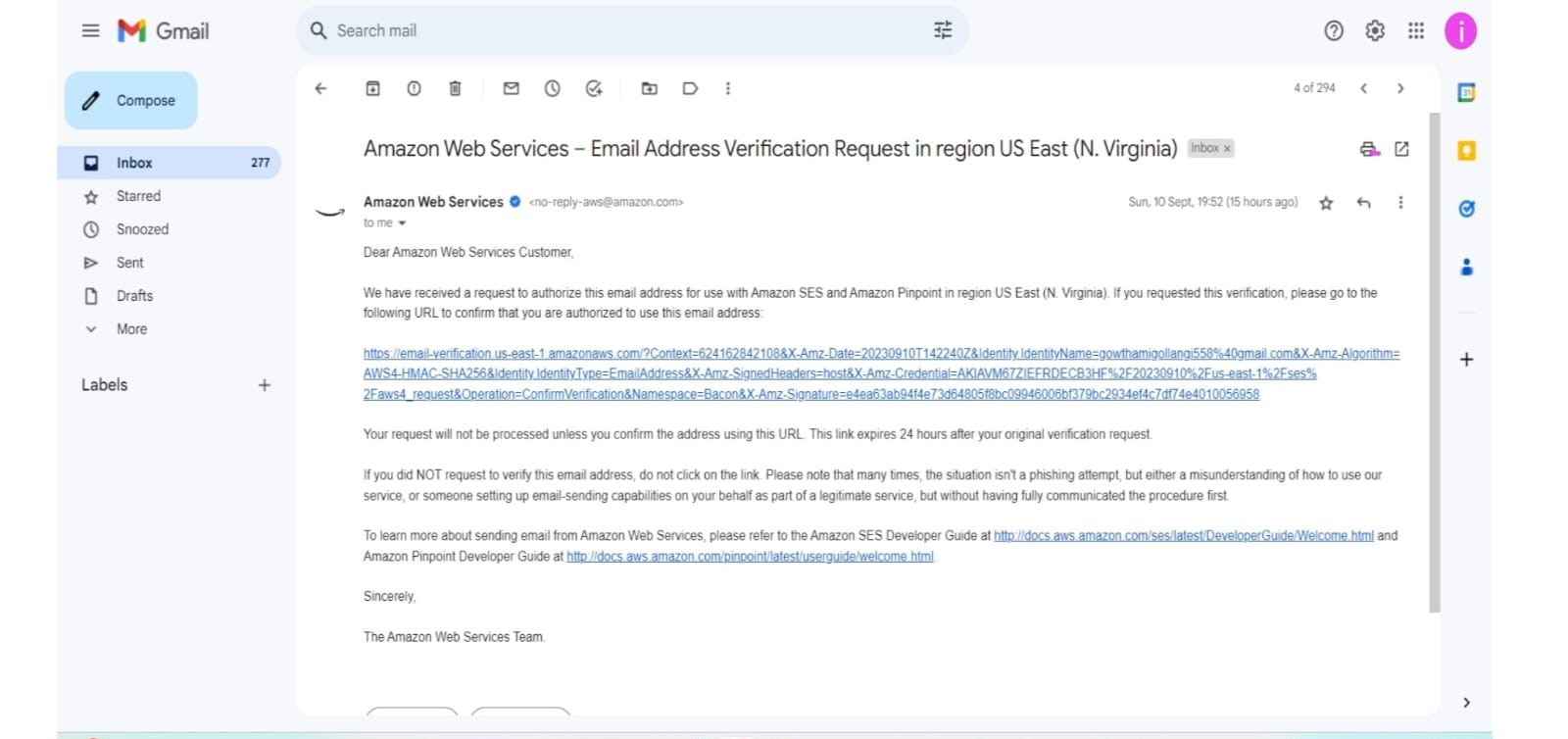
***IMPLEMENTENTATION STEPS:***

**STEP 1:**

**CREATING IDENTITY IN SES (Simple Email Service)**

* For sending email, we must have a verified identity so click on create identity.
* Select email address and give your specific email which need to be verified.
* After this click on create identity at bottom.
* Check mail inbox to see a mail received from Amazon web services.
* Click on the link provided in the mail then it will be redirected to AWS such that it gives info that the mail has been verified.
* By refreshing the page, we can see identity status as verified.

 **Figure 3 Identity Status of E-mail**

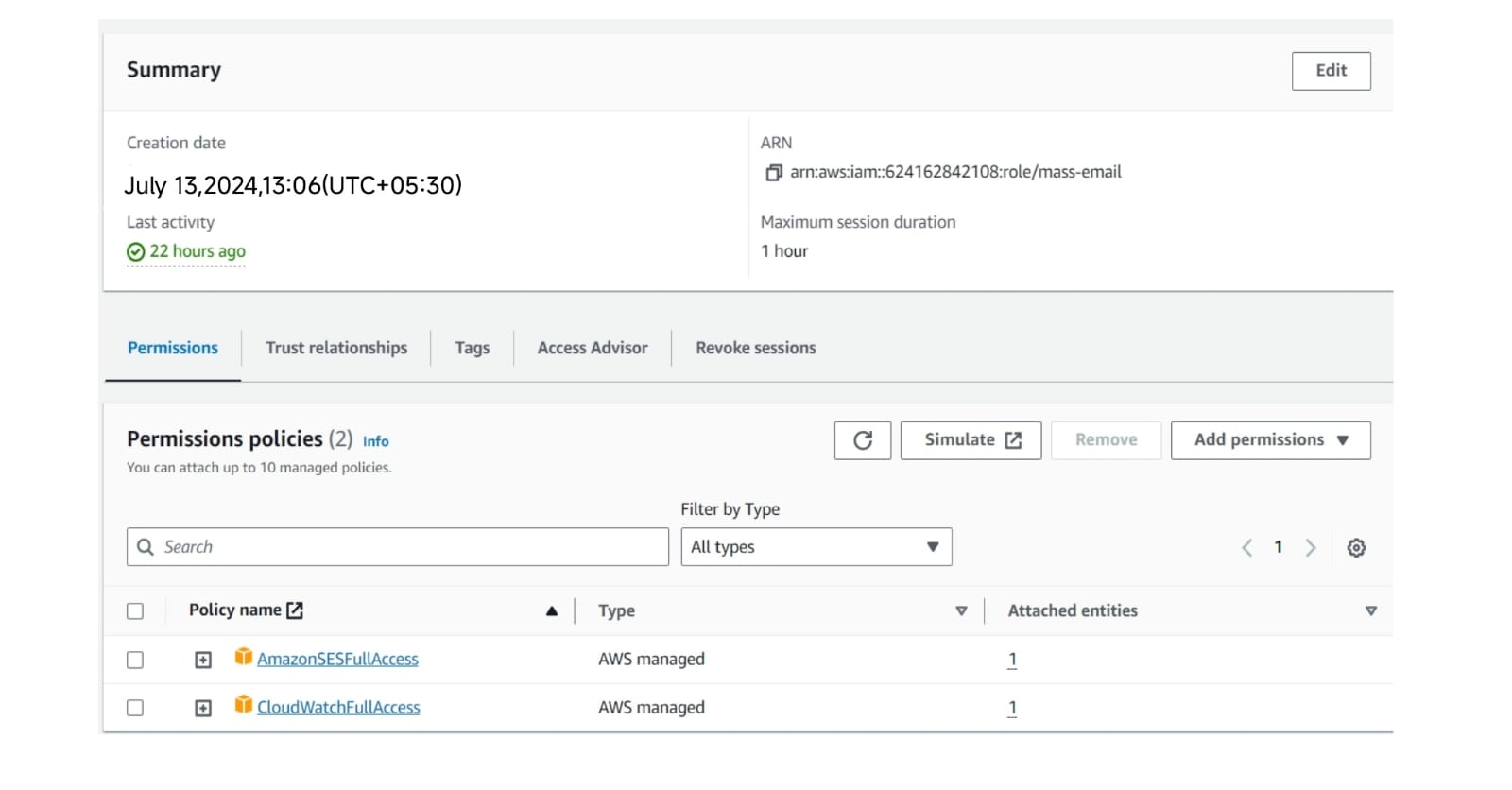


## Figure 4 Verification mail received from AWS

**STEP 2:**

**CREATING IAM (Identity Access Management) ROLE**

* In the AWS console search for IAM in the search bar and select the service
* In that select roles and click on create role
* Select use cases as Lambda and click on next
* In permission policies choose CloudWatch full access, and SES full access and then click on next
* Give a suitable name for the role and click on create role
* Then role will be created.



## Figure 5 Permissions to create IAM role

**STEP 3:**

**CREATING A LAMBDA FUNCTION**

* In the AWS console search for Lambda in the search bar and select the service.
* Provide the name of the function.
* In the position of runtime, we must choose the language that we want. Here, I am choosing the most recent Python 3.9 version.
* Choose whether to create a new or existing role as the executing role in the following step. I’m choosing the role that I already created in the previous phase. • Rest everything. We can keep it optional
* Select Create a Function.
* Then click on code and write the code over there.
* Next in the source mail give your image ID i.e., verified mail and change destination mail which is your verified E-mail.
* Now save and deploy it.
* Test the code for checking errors

**THE CODE DEPLOYED IN AWS LAMBDA:**

import boto3 from botocore.exceptions import ClientError def send\_email():

SENDER = "youremail@gmail.com" # must be verified in AWS SES Email

RECIPIENT = "youremail@gmail.com" # must be verified in AWS SES Email

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.

AWS\_REGION = "us-east-1"

# The subject line for the email.

SUBJECT = "This is test email for testing purpose..!!"

# The email body for recipients with non-HTML email clients.

BODY\_TEXT = ("Hey Hi...\r\n"

"This email was sent with Amazon SES using the "

"AWS SDK for Python (Boto)."

)

# The HTML body of the email.

BODY\_HTML = """<html>

<head></head>

<body>

<h1>Hey Hi...</h1>

<p>This email was sent with

<a href='https://aws.amazon.com/ses/'>Amazon SES CQPOCS</a> using the

<a href='https://aws.amazon.com/sdk-for-python/'>

AWS SDK for Python (Boto)</a>.</p>

</body>

</html>

"""

# The character encoding for the email.

CHARSET = "UTF-8"

# Create a new SES resource and specify a region.

client = boto3.client('ses',region\_name=AWS\_REGION)

# Try to send the email.

try:

#Provide the contents of the email.

response = client.send\_email(

Destination= {

'ToAddresses': [

RECIPIENT,

],

},

Message= {

'Body': {

'Html': {

'Data': BODY\_HTML

},

'Text': {

'Data': BODY\_TEXT

},

},

'Subject': {

'Data': SUBJECT

},

},

Source=SENDER

)

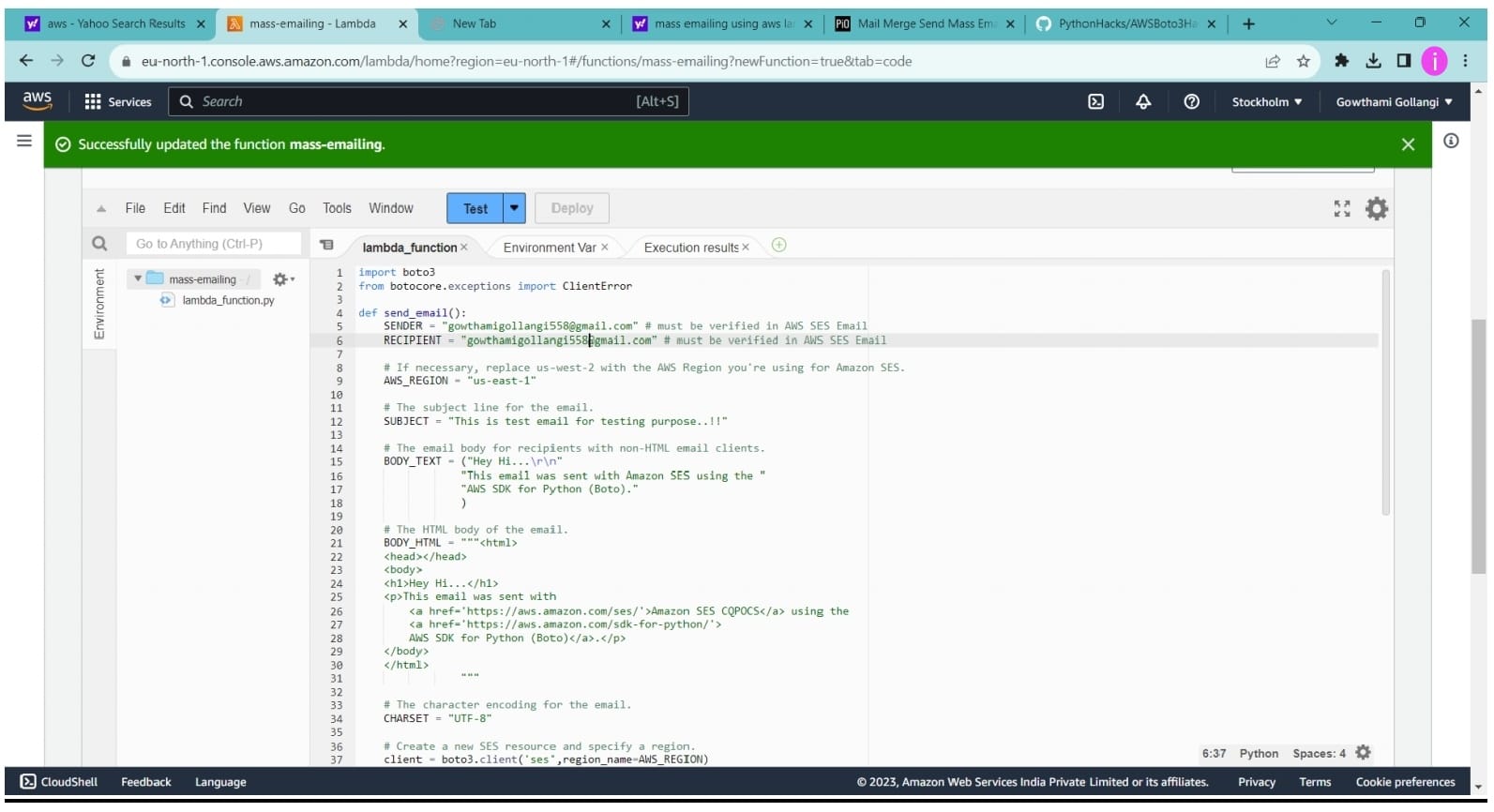
# Display an error if something goes wrong. except ClientError as e:

print(e.response['Error']['Message'])

else:

print ("Email sent! Message ID:"), print(response['MessageId']) def lambda\_handler(event, context): # TODO implement

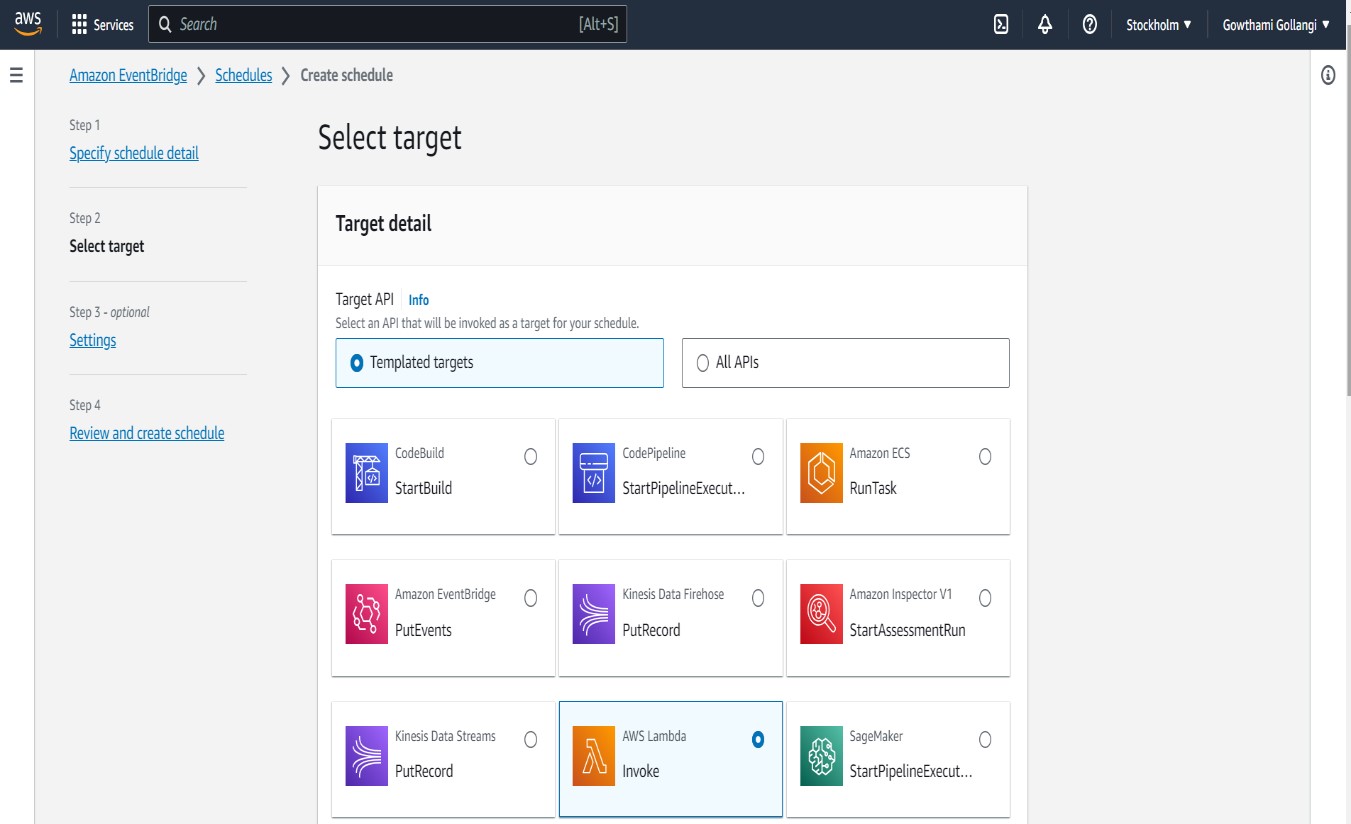
send\_email()



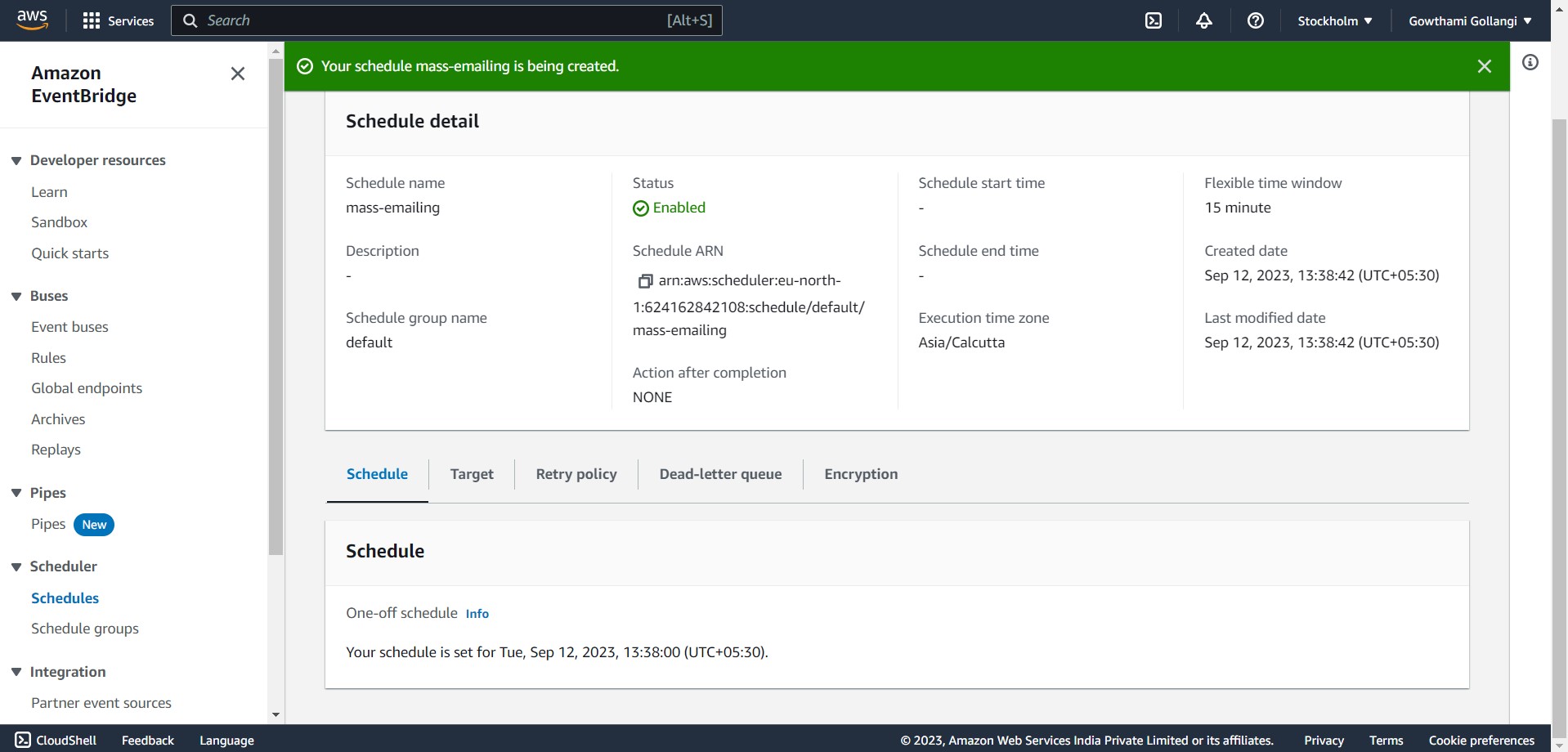
**STEP 4:**

**CREATING CLOUDWATCH EVENT**

* In the AWS console search for CloudWatch in the search bar and select the service.
* In the left navigation pane, select Event in that select rules and then click on create rule.
* Next, give a name to the rule, select schedule and click on continue Event Bridge schedular.
* Next, choose the schedule and set it to a specific time.
* Select add target and choose the lambda function I’m choosing the function that I already created in the previous phase.
* Click on configure details.
* Click on Create the rule.



## Figure 7 Selecting AWS lambda as target



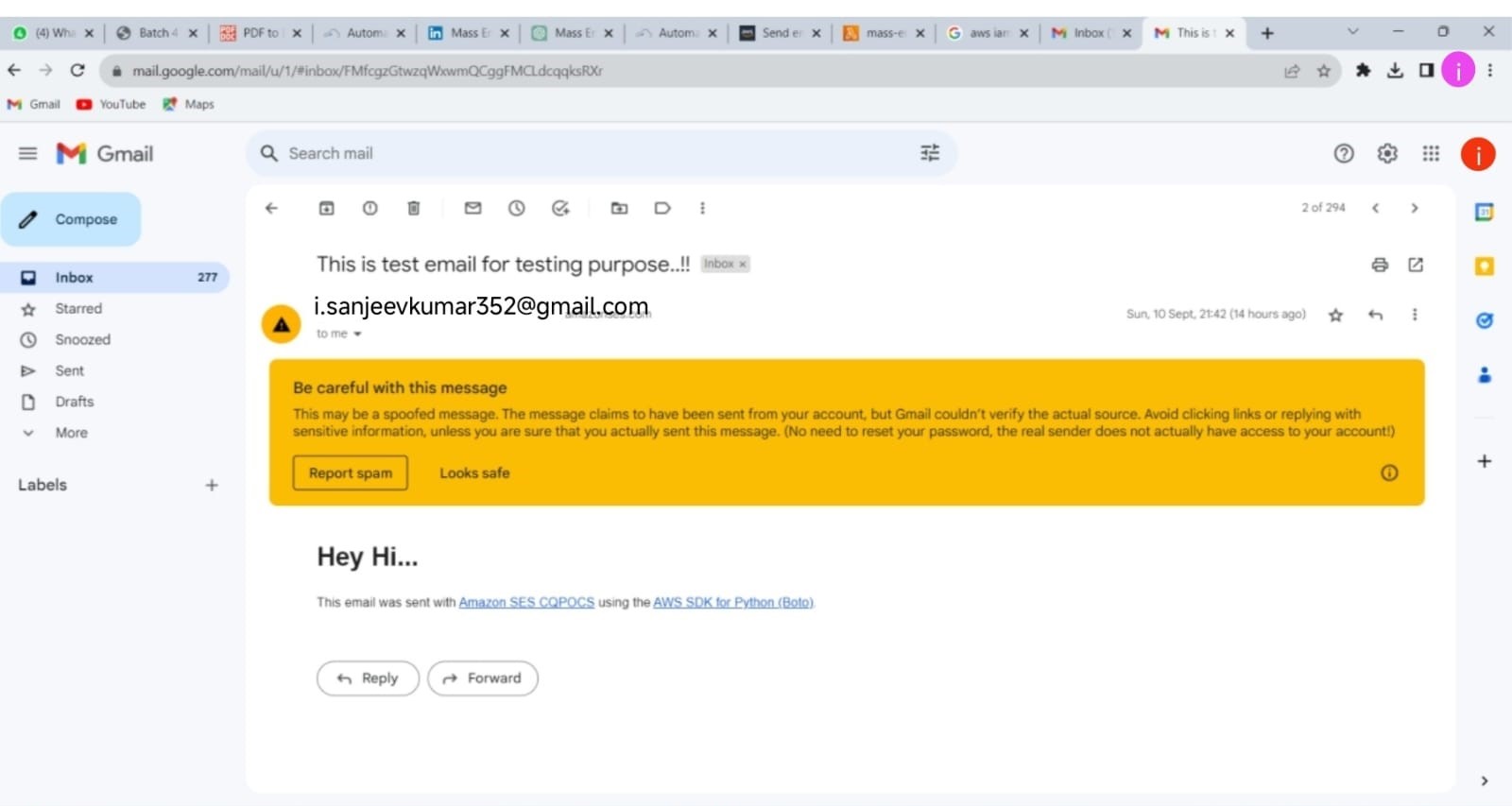
***Figure 8 Scheduling the CloudWatch to send E-mail***

## 5.RESULT

Implementing a mass emailing project using AWS Lambda and SES offers a highly efficient and scalable solution for organizations seeking to engage with their audience through email campaigns. This architecture combines the event-driven power of AWS Lambda with the robust email delivery capabilities of SES, resulting in a system that can handle various email sending scenarios. With AWS Lambda, you can dynamically generate personalized email content, ensuring that each recipient receives tailored messages. SES enhances email deliverability through authentication mechanisms like DKIM, SPF, and DMARC, minimizing the chances of emails ending up in spam folders. The architecture's serverless nature means you can effortlessly scale to meet changing email volumes without the complexities of managing servers.

Additionally, SES's automated bounce and complaint handling streamline the process, maintaining your sender reputation. Detailed metrics and monitoring through CloudWatch provide insights into campaign performance, ensuring you can fine-tune your email strategy effectively. This architecture promotes cost efficiency with its pay-as-you-go model, making it accessible to businesses of all sizes. Overall, AWS Lambda and SES together form a powerful, reliable, and cost-effective solution for mass emailing, helping organizations deliver their messages effectively while maintaining the highest standards of email deliverability and security.

• Lambda is triggered and sent an email to the users at the scheduled time



***Figure 9 E-mail received as a result***

### 6.CONCLUSION

*In concluding this project, using AWS Lambda, IAM role, CloudWatch, and SES can be an efficient and reliable way to send mass emails. Lambda can be used to execute code that triggers email sending, while IAM roles can provide the necessary permissions to access SES.*

CloudWatch can be used to monitor and log the email-sending process, ensuring its successful completion. SES can be used to send bulk emails while providing features like email validation, suppression lists, and bounce handling. Overall, using these services in conjunction can provide a scalable and cost-effective solution for sending mass emails.

Leveraging AWS Lambda and SES for a mass emailing project provides an efficient, scalable, and costeffective solution for organizations of all sizes. This architecture harnesses the power of serverless computing through Lambda, allowing for event-driven email generation and delivery. SES, with its robust features like email authentication, bounce handling, and delivery tracking, ensures high deliverability and maintains sender reputation. With the ability to dynamically personalize email content and handle large volumes of recipients, this approach caters to the needs of marketing campaigns, transactional emails, and various other email use cases. Additionally, AWS's pay-as-you-go pricing model enables cost control and flexibility, making it an attractive choice for businesses. In essence, the integration of AWS Lambda and SES streamlines the mass emailing process, offering a reliable and scalable solution that empowers organizations to reach their audience effectively and efficiently while adhering to industry best practices.