**SEVEN WEEKS SUMMER TRAINING REPORT**

**on**

# AWS Cloud Computing training

Submitted by

**KATTAMURI VENKATESH**

**Registration No: 12213214**

**Program Name: Btech. CSE (3rd Year)**

Under the Guidance of:

**Muhammad Samir**

**Akhtar and Mr. Yash Raj**

**Gokboru Tech Pvt. Ltd.,**

**Noida (India)**

(June-July,2024)

## DECLARATION

I hereby declare that I have completed my seven weeks summer training at Gokboru Pvt. Ltd. from June 02,2024 to July 14, 2024, under the guidance of Muhammad Samir Akhtar and Mr. Yash Raj. I declare that I have worked full dedication during their 7 weeks of training and my learning outcomes fulfill the requirements of training for the award of degree of B.tech CSE, Lovely Professional University, Phagwara.

Name of Student –

Kattamuri Venkatesh

Registration no:

12213214

## ACKNOWLEDGEMENT

I would like to express my gratitude towards my university as well as Gokboru Tech Pvt. Ltd. for providing me the golden opportunity to do this wonderful summer training regarding AWS Cloud computing, which also helped me in doing a lot of homework and learning. As a result, I came to know about so many new things. So, I am thankful to them.

Moreover, I would like to thank my friends who helped me a lot whenever I got stuck in some problem related to my course. I am thankful to have such a good support of them as they always have my back whenever I need.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

Deepest thanks to our Trainer **Muhammad Samir Akhtar (Mentor) and Yash Raj (Trainer for AWS)** for their guidance, monitoring, constant encouragement and correcting various assignments of ours with attention and care. They have taken pain to go through the project and training sessions and make necessary corrections as when needed and we are very grateful for that.

# Summer Training Certificate by Gokboru Tech

## Table of Contents

## INTRODUCTION

Cloud computing is a technology that allows users to access computing resources like servers, storage, databases, networking, software, and analytics over the internet, or “the cloud.” Instead of owning and maintaining physical hardware and software, businesses and individuals can use these resources on-demand from cloud service providers. This approach offers flexibility, scalability, and cost-efficiency. Users can scale up or down depending on their needs, pay only for what they use, and avoid the complexities of managing physical infrastructure.

## TECHNOLOGY LEARNT

**It had 16 units which was further divided into chapters and then topics so during my whole 6-week course I learned the following:**

**1st Week:**

1. **Introduction to Cloud Computing:**

Cloud computing is a technology that provides access to computing resources such as servers, storage, and software over the internet. Instead of relying on personal computers or local servers, these resources are utilized remotely. Services are available as needed, with the ability to scale up or down, and payment is made based on usage. This approach offers a flexible and cost-effective solution for managing and processing data.

2. **Definition, Characteristics, Benefits:**

***Definition:*** Cloud computing refers to the delivery of computing services—including servers, storage, databases, networking, software, and analytics—over the internet ("the cloud"). These services enable users to access and manage resources without the need for direct active management by the user.

***Characteristics:***

*On-Demand Self-Service:* Computing resources can be provisioned automatically, as needed, without requiring human intervention.
*Broad Network Access:* Services are accessible over the internet from various devices, such as computers, tablets, and smartphones.
Resource Pooling: Computing resources are pooled to serve multiple users, with resources dynamically assigned and reassigned based on demand.
Rapid Elasticity: Resources can be scaled up or down quickly to meet changing demands,

appearing unlimited to users.

Measured Service: Resource usage is monitored and reported, allowing for pay-as-you-go billing based on actual consumption.

***Benefits:***

*Cost Efficiency:* Reduces the need for significant capital investment in hardware and software, as resources are provided on a pay-per-use basis.

*Scalability:* Easily scales resources up or down to handle varying workloads, ensuring that users only pay for what they use.

*Accessibility:* Enables access to applications and data from anywhere with an internet connection, facilitating remote work and collaboration.

*Reliability:* Offers enhanced reliability through data backup, disaster recovery, and redundancy, often exceeding the capabilities of on-premises solutions.

*Flexibility:* Provides the flexibility to choose from a wide range of services and customize them to fit specific business needs.

**3. Cloud Service Models:**

There are 3 types of Cloud Service Models:

***i.*** *Infrastructure as a Service (IaaS):* Provides virtualized computing resources like servers, storage, and networking over the cloud, allowing users to manage operating systems and applications.

***ii.*** *Platform as a Service (PaaS):* Offers a cloud-based platform for developing, testing, and deploying applications, without managing the underlying infrastructure.

***iii.*** *Software as a Service (SaaS):* Delivers software applications over the internet, accessible via a web browser, without requiring installation or management by the user.

4. **Cloud Deployment Models:**

5. There are 4 types of deployment model:

   - *Private:* Exclusively used by single organizations, offering greater control and security. Example: VMware vSphere
   - *Public:* Resources are owned and managed by a third-party provider, accessible to multiple users. Example: Amazon Web Services (AWS)
   - *Hybrid:* Combines public and private clouds, enabling data and application portability between them. Example: Microsoft Azure
   - *Community:* Shared infrastructure among organizations with common goals, ensuring collaboration and resource sharing. Example: OpenNebula for Research and Education.

6. **Introduction to Linux:**

- *Linux Operating System Basics and History of Linux.*
- *Basic Linux Commands, Navigation and file management system.*
- *Networking and Internet tools and networking with linux.*
- *Functions and variables, SSH basics.*

**2nd Week:**

7. **Getting Started With AWS:**

- *Creating an AWS account.*
- *Navigating through AWS console.*
- *Overview of AWS services and platforms.*
- *AWS compute services (EC2, Lambda, ECS)*
- *AWS Storage services (S3, EBS)*

- *Understanding security groups and data stream pipelines.*
- *Understanding lambda, triggers, load balancing and auto-scaling.*

**3rd Week:**

## 8. AWS Security:

- *AWS Identity and Access Management (IAM):* It is a framework used to ensure that the right individuals have the appropriate access to technology resources. In cloud computing, IAM is a crucial security feature that helps manage users, permissions and roles within an organization.
- *IAM Users and Group*:

  **IAM Users:**

  In AWS, an IAM (Identity and Access Management) user is an entity that represents a person or service that interacts with AWS. Each IAM user has unique credentials, such as a username and password, and can have permissions assigned to control what actions they can perform within AWS.

  **IAM Groups:**

  An IAM group is a collection of IAM users that share common permissions. By adding users to a group, administrators can manage permissions for multiple users at once, simplifying access management. Groups are typically used to apply consistent policies and access rights to users with similar roles.

- *Multi-Factor Authentication (MFA):* additional layer of security used in AWS to enhance the protection of user accounts and sensitive resources. With MFA enabled, users are required to provide not just their password but also a second form of authentication, such as a one-time code generated by a hardware or software token, when accessing AWS services. This helps ensure that even if a password is compromised, unauthorized access is still prevented.
- *Best Practices for encryption.*
- *AWS Resource Monitoring and Logging*: essential practices for managing and maintaining the performance, availability, and security of your AWS infrastructure.

✓ ***Monitoring:*** AWS offers services like Amazon CloudWatch to track metrics, set alarms, and visualize logs for various AWS resources, helping detect performance issues and respond to operational changes.

- ✓ ***Logging:*** AWS provides logging services such as AWS CloudTrail and Amazon CloudWatch Logs to capture detailed event logs, track user activities, and ensure compliance by keeping an audit trail of all actions taken on your AWS resources.

## 4th Week:

**9. Networking on AWS:** Networking on AWS involves configuring and managing virtual networks that enable the communication between different AWS resources and with external networks.

- *Amazon VPC (Virtual Private Cloud):* Allows you to create isolated networks within AWS, where you can launch resources like EC2 instances. It provides control over IP addresses, subnets, route tables, and gateways.
- *Security Groups and NACLs:* Used to control inbound and outbound traffic to and from resources within a VPC, offering a layer of security.
- *Elastic Load Balancing (ELB):* Distributes incoming traffic across multiple targets, such as EC2 instances, in one or more Availability Zones to ensure high availability.
- *AWS Direct Connect:* Establishes a dedicated network connection from your premises to AWS, providing more consistent network performance.
- **Route 53:** AWS's scalable DNS and domain name management service, used to route end-user requests to AWS-hosted applications.

**10. AWS Elastic Load Balancing (ELB):** AWS Elastic Load Balancing (ELB) is a service that automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses, across one or more Availability Zones. This helps ensure the high availability and fault tolerance of applications by spreading the load, preventing any single resource from becoming a bottleneck.

- Key features include:
  - ✓ *Automatic Scaling:* ELB automatically adjusts the load balancing capacity to handle incoming traffic as it changes over time.
  - ✓ *Health Checks:* ELB performs regular health checks on registered targets and only routes traffic to healthy instances.
  - ✓ *Security:* ELB supports secure applications by allowing the use of HTTPS and SSL/TLS.
  - ✓ Types of Load Balancers:
  - ✓ *Application Load Balancer (ALB):* Best suited for HTTP/HTTPS traffic, operating at the application layer (Layer 7).
  - ✓ *Network Load Balancer (NLB):* Designed for ultra-high performance and low latency, operating at the transport layer (Layer 4).
  - ✓ *Gateway Load Balancer:* Distributes traffic across multiple virtual appliances while scaling them automatically.
  - ✓ Use Case: ELB is essential for applications that require high availability and redundancy, ensuring continuous operation even if individual servers fail.

**5th Week:**

**11. Containerization and Orchestration on AWS:**

**Containerization** is a lightweight form of virtualization that involves packaging an application and its dependencies into a container. This ensures the application runs consistently across different computing environments. Containers are isolated from each other and the underlying infrastructure, making them highly portable and efficient.

**Orchestration** involves managing and automating the deployment, scaling, and operation of containerized applications. On AWS, several services support containerization and orchestration, enabling developers to run and manage containers effectively.

***Key AWS Services for Containerization and Orchestration: -***

*Amazon Elastic Container Service (ECS):*

*Overview:* ECS is a fully managed container orchestration service that allows you to run, stop, and manage Docker containers on a cluster of Amazon EC2 instances.

*Features:*

*Task Definitions:* Define how containers should be run, including the Docker images, CPU, memory requirements, and network settings.

*Cluster Management:* Automatically manages the cluster of EC2 instances, ensuring optimal resource utilization.

*Integration with AWS Services*: Seamlessly integrates with other AWS services like IAM, CloudWatch, and VPC for enhanced security and monitoring.

*Amazon Elastic Kubernetes Service (EKS):*

*Overview:* EKS is a fully managed Kubernetes service that makes it easy to run Kubernetes on AWS without needing to install, operate, and maintain your Kubernetes control plane.

*Features:*

*Kubernetes Compatibility*: EKS runs upstream Kubernetes, ensuring compatibility with existing Kubernetes applications.

*Managed Control Plane:* AWS manages the Kubernetes control plane, ensuring high availability and security.

*Scalability:* Automatically scales the Kubernetes clusters based on demand.

*AWS Fargate:*

*Overview*: AWS Fargate is a serverless compute engine for containers that works with both ECS and EKS. It allows you to run containers without managing the underlying EC2 instances.

*Features:*
*Serverless Operation*: Automatically provisions and scales the infrastructure required to run your containers, eliminating the need to manage servers.

*Pay-as-You-Go:* Only pay for the compute resources used by your containers, with no upfront costs or commitments.

*Security:* Provides enhanced security through isolated runtime environments for each container.

*Amazon ECR (Elastic Container Registry):*
*Overview*: ECR is a fully managed Docker container registry that makes it easy to store, manage, and deploy Docker container images.

*Features:*

*Image Storage:* Securely store container images in a scalable repository.
Integration with CI/CD: Integrates with AWS CodePipeline, Jenkins, and other CI/CD tools for streamlined development and deployment workflows.

*Security:* Offers encryption, access control, and integration with IAM for secure image management.

*Use Cases:*

*Microservices Architecture:* Easily deploy and manage microservices using containers, with orchestration ensuring seamless scaling and availability.

*DevOps Pipelines:* Integrate containerization into CI/CD pipelines for faster development and deployment cycles.

*Hybrid Cloud Environments:* Run containers across on-premises and cloud environments, with Kubernetes providing a consistent orchestration layer.

*Benefits:*
*Portability:* Containers can run consistently across different environments, making it easier to develop and deploy applications.

*Efficiency:* Containers are lightweight and consume fewer resources compared to traditional virtual machines.

*Scalability:* Orchestration tools automatically scale containerized applications based on demand, ensuring optimal performance.

**6th Week:**

## 12. Multi-Cloud Models and Providers:

Multi-Cloud is an IT strategy that involves utilizing multiple cloud computing services from different providers. This approach allows organizations to leverage the unique strengths of each provider, avoid vendor lock-in, and enhance redundancy for critical systems.

### Overview of Multi-Cloud Benefits

***Vendor Independence:***

- *Flexibility:* Utilizing multiple cloud providers prevents dependency on a single vendor, offering greater freedom in pricing, service offerings, and technology choices.
- *Negotiation Power:* Multi-cloud strategies provide leverage during negotiations, as organizations are not committed to a single provider.

***Enhanced Resilience and Redundancy:***

- *High Availability*: Workloads distributed across multiple cloud providers ensure continuous availability, mitigating the risk of a single provider's service failure affecting overall application performance.
- *Disaster Recovery*: Multi-cloud environments offer a robust solution for disaster recovery, ensuring business continuity in the event of a major outage with one provider.

***Best-of-Breed Services:***

- *Optimized Performance*: Different cloud providers specialize in various areas (e.g., AI/ML, analytics, networking), allowing organizations to select the best services for specific needs.
- *Innovation*: Leveraging unique offerings from multiple providers helps organizations stay ahead of technological advancements.

***Regulatory Compliance:***

- *Data Sovereignty:* Multi-cloud strategies enable compliance with regional data residency regulations by selecting providers that meet specific compliance standards.

## 13. Multi-Cloud Deployment Models:

***Poly Cloud:***

- *Definition:* The use of multiple cloud providers, each chosen for its strengths in specific areas or services.
- *Example:* Utilizing AWS for infrastructure, Google Cloud for machine learning, and Azure for enterprise integrations.

***Hybrid multi-cloud:***

- *Definition:* Integration of multiple public cloud providers with on-premises data centers or private clouds.
- *Example:* Combining AWS for general workloads, Azure for compliance-sensitive applications, and an on-premises data center for legacy systems.

***Distributed Cloud:***

- *Definition:* Cloud services distributed across different locations (e.g., on-premises, edge, public cloud), managed as a single entity.
- *Example:* Using Google Cloud Anthos to manage workloads across on-premises environments and multiple clouds from a unified platform.

## 14. Key Multi-Cloud Providers:

***Amazon Web Services (AWS):***

- *Strengths:* Leader in infrastructure services, global reach, and diverse service offerings.
- *Use Case:* Ideal for scalable infrastructure, storage, and cloud-native applications.

***Microsoft Azure:***

- Strengths: Strong enterprise solutions, hybrid cloud capabilities, and integration with Microsoft products.
- Use Case: Suitable for organizations with existing Microsoft environments or those requiring compliance and security features.

***Google Cloud Platform (GCP):***

- Strengths: Advanced data analytics, AI/ML services, and networking capabilities.
- Use Case: Ideal for data-intensive applications, machine learning, and big data analytics.

***IBM Cloud:***

- Strengths: Enterprise-grade solutions, AI integration (Watson), and hybrid cloud support.
- Use Case: Best for large enterprises with complex infrastructure needs and AI/ML capabilities.

***Oracle Cloud:***

- Strengths: Database services, ERP solutions, and integration with Oracle software.
- Use Case: Preferred by organizations running Oracle databases or applications.

***Vmware Cloud:***

- Strengths: Consistent infrastructure across clouds and on-premises with Vmware environments.
- Use Case: Ideal for organizations with significant Vmware investments.

**15. Multi-Cloud Challenges:**

***Operational Complexity:***

- Managing multiple cloud environments requires advanced skills and tools for consistent security, compliance, and performance.

***Cost Management:***

- Monitoring and optimizing costs across different cloud providers can be challenging, necessitating sophisticated tools.

***Data Integration:***

- Ensuring data consistency across multiple cloud environments, particularly for large-scale or real-time data, can be difficult.

***Security and Compliance:***

- Consistent security policies and compliance across different providers can be complex due to varying security standards.

**16. Tools and Best Practices for Multi-Cloud Management**

***Multi-Cloud Management Platforms:***

- Tools such as HashiCorp Terraform and Google Anthos provide unified management across multiple cloud environments, simplifying operations and governance.

***Cost Optimization Tools:***

- Solutions like CloudHealth by Vmware and AWS Cost Explorer assist in monitoring and optimizing cloud spending across providers.

***Security Management:***

- Implementing unified security frameworks like AWS Security Hub or Azure Security Center ensures consistent security policies across environments.

***Automation and Orchestration:***

- Tools such as Ansible, Puppet, and Kubernetes support efficient deployment and management of applications across multiple clouds.

## Reasons for Choosing AWS Cloud Computing

Amazon Web Services (AWS) is a leading cloud computing platform that has become a preferred choice for businesses of all sizes. Here are several detailed reasons why AWS stands out as a top choice for cloud computing:

**2. Broad Range of Services**

AWS provides an extensive array of services that cover virtually every aspect of cloud computing. These include:

- **Compute Services**: EC2 instances, Lambda (serverless computing), and ECS (container services).
- **Storage Solutions**: S3 for object storage, EBS for block storage, and Glacier for archival storage.
- **Databases**: Managed relational databases with RDS, NoSQL databases like DynamoDB, and fully managed data warehouse services with Redshift.
- **Networking**: VPC (Virtual Private Cloud), Route 53 for DNS services, and Direct Connect for dedicated network connections.

The wide range of services allows businesses to choose exactly what they need, and the modularity of these services ensures that organizations only pay for what they use.

**3. Global Presence and Availability**

AWS operates in multiple geographic regions around the world, each with multiple availability zones. This global presence ensures that applications can be deployed in multiple regions to achieve low latency, comply with local data regulations, and enhance disaster recovery capabilities. AWS's network infrastructure is designed for reliability, with built-in redundancy across regions and availability zones, making it a reliable choice for mission-critical applications.

**4. Security and Compliance**

Security is a top priority for AWS. The platform offers robust security features such as:

- **Encryption**: AWS provides encryption in transit and at rest across its services.
- **Identity and Access Management (IAM)**: Granular control over who can access what resources.
- **Monitoring and Logging**: Services like CloudWatch and CloudTrail help monitor and audit actions and events within AWS.

Additionally, AWS complies with a wide array of global security standards and certifications, such as ISO 27001, SOC 1/2/3, GDPR, and HIPAA. This makes AWS a strong choice for industries with stringent regulatory requirements.

**5. Cost Efficiency**

AWS operates on a pay-as-you-go pricing model, which means organizations only pay for the resources they actually use. This pricing structure is ideal for businesses of all sizes, as it allows them to scale their expenses with their growth. AWS also offers various pricing models such as Reserved Instances and Spot Instances, which can further reduce costs for predictable or flexible

workloads. Additionally, AWS provides cost management tools like the AWS Pricing Calculator, AWS Budgets, and Cost Explorer to help organizations monitor and optimize their spending.

**6. Innovation and Ecosystem**

AWS is known for its rapid pace of innovation, frequently releasing new services and features that enable businesses to stay ahead of technological trends. For example:

- **AI and Machine Learning**: Services like Amazon SageMaker for building and training machine learning models.
- **IoT**: AWS IoT Core for managing connected devices.
- **Blockchain**: Amazon Managed Blockchain for creating and managing blockchain networks.

AWS also has a vast ecosystem of partners and third-party integrations available through the AWS Marketplace, providing additional tools and services to enhance the AWS environment.

**7. Developer and Enterprise Support**

AWS offers a wide range of support options to meet the needs of developers and enterprises alike.

- **Developer Tools**: Services like AWS CodePipeline, CodeBuild, and CodeDeploy help automate and streamline the software development lifecycle.
- **Enterprise Support**: AWS provides 24/7 support through various support plans, including technical account managers, architecture support, and business-critical system monitoring.
- **Training and Certification**: AWS offers extensive training resources, certification programs, and a well-established community that helps users build their cloud skills and stay updated with best practices.

**8. Interoperability and Integration**

AWS supports a wide range of programming languages, operating systems, databases, and other third-party applications, making it easy to integrate with existing IT environments. This interoperability allows businesses to migrate to the cloud without having to re-engineer their applications entirely, which is particularly advantageous for organizations with complex IT infrastructures.

**9. Customer Success Stories**

Many of the world's largest companies, including Netflix, Airbnb, and Coca-Cola, have successfully adopted AWS for their cloud computing needs. These customer success stories provide tangible evidence of AWS's capability to support large-scale, highly demanding applications.

**10. Continuous Improvement and Updates**

AWS continuously invests in its platform, regularly adding new features, updating existing services, and improving overall performance. This commitment to continuous improvement

ensures that AWS customers always have access to the latest technologies and can stay competitive in their industries.

**Project:** "Email Forwarding System Using AWS SES, Lambda, and IAM Roles"

## Introduction:

Email communication is a fundamental aspect of modern life, facilitating seamless conversations between individuals. This project, titled "Email Forwarding System Using AWS SES, Lambda, and IAM Roles," focuses on developing a simple yet efficient solution for forwarding emails from one address to another in a one-to-one conversation scenario.

In this project, AWS Simple Email Service (SES) is utilized to receive emails, which are then processed by an AWS Lambda function. The Lambda function, acting as the core logic of the system, forwards the received email to a specified recipient. AWS Identity and Access Management (IAM) roles are employed to ensure that the system operates securely, with precise control over the permissions granted to the Lambda function and SES.

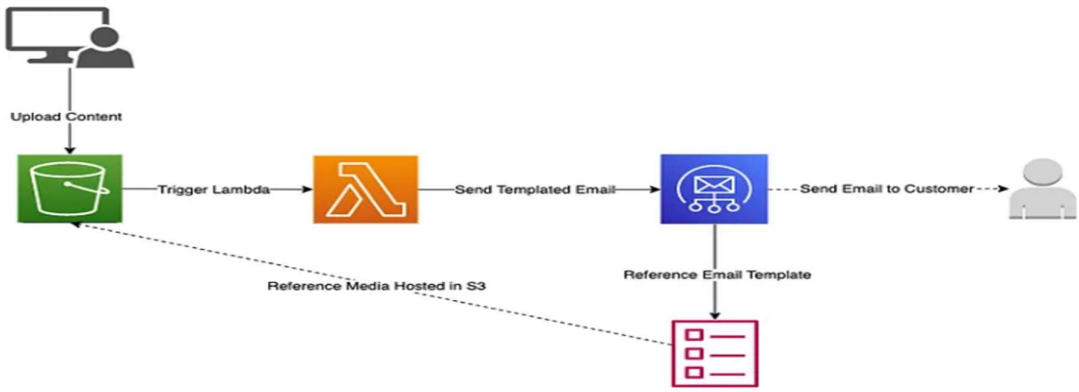
This one-to-one email forwarding solution is ideal for scenarios where an individual wants to automatically redirect incoming emails from one account to another, such as forwarding work emails to a personal account or vice versa. The use of AWS services allows for a serverless, cost-effective, and highly available solution that requires minimal maintenance.

By completing this project, you'll gain practical experience with AWS SES, Lambda, and IAM roles, and learn how to integrate these services to create a focused, real-world application for personal email management.

**Steps For creating My project:**

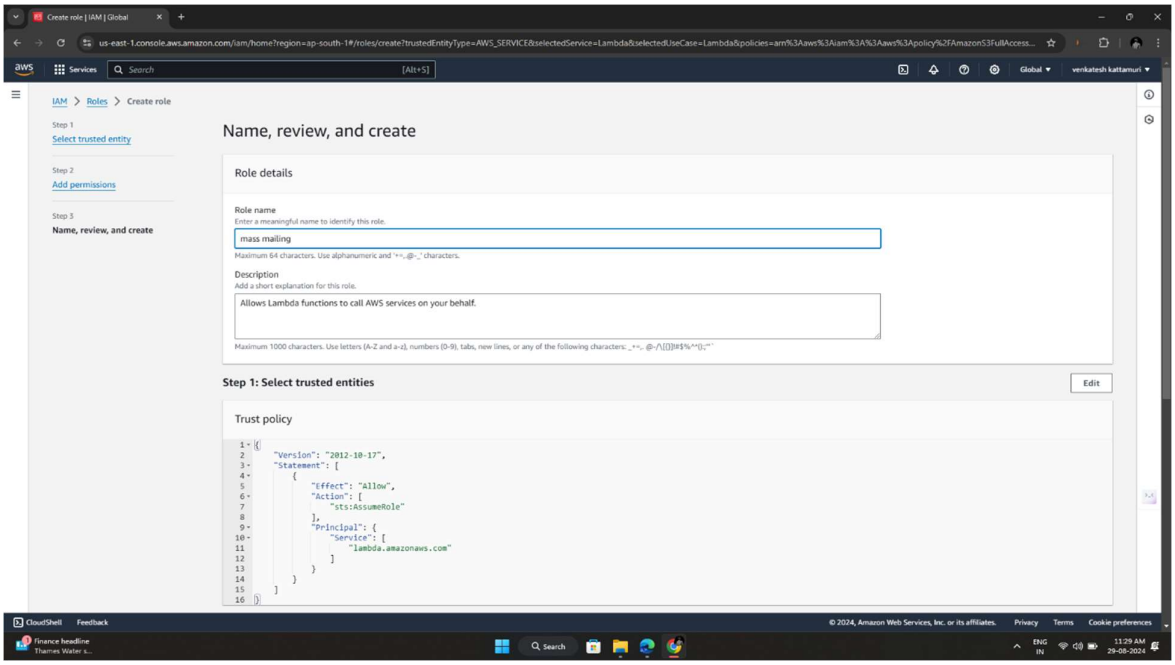
*Step 1:* First an AWS an account is created and then logged in through using root user.

*Step 2:* Then A flow Chart is created for creation and understanding for my project.

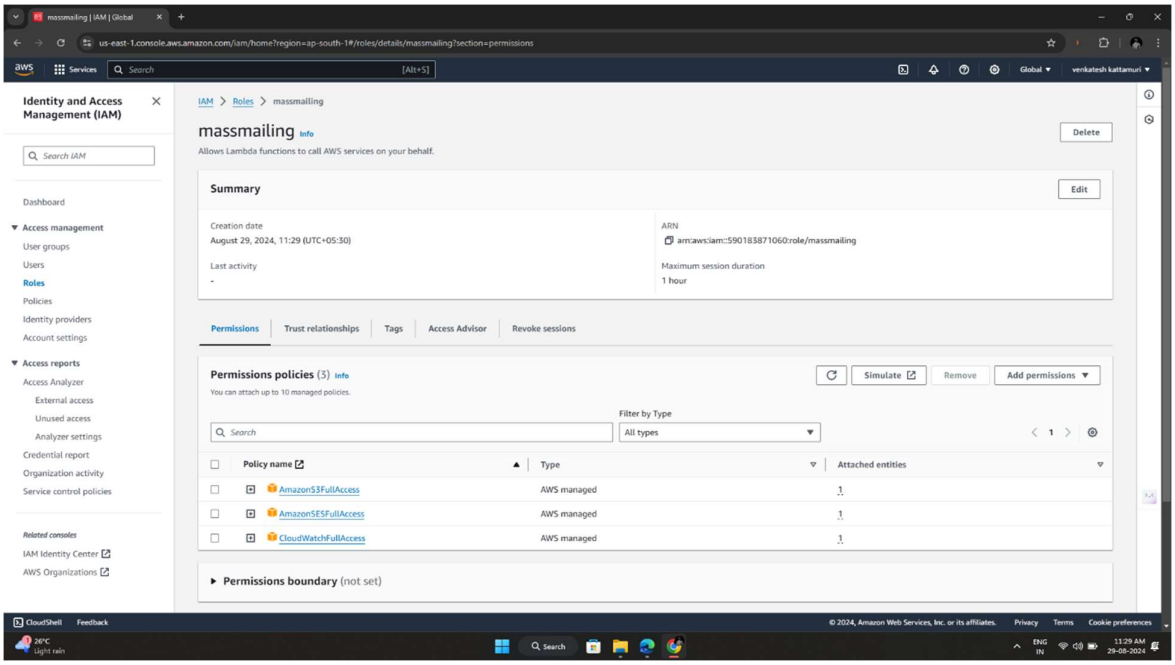



*Step 3:* So, for the above project an role is create by the name of mass mailing which includes the following permissions policies:

- **AmazonS3FullAccess**
- **CloudWatchFullAccess**
- **AmazonSESFullacces**

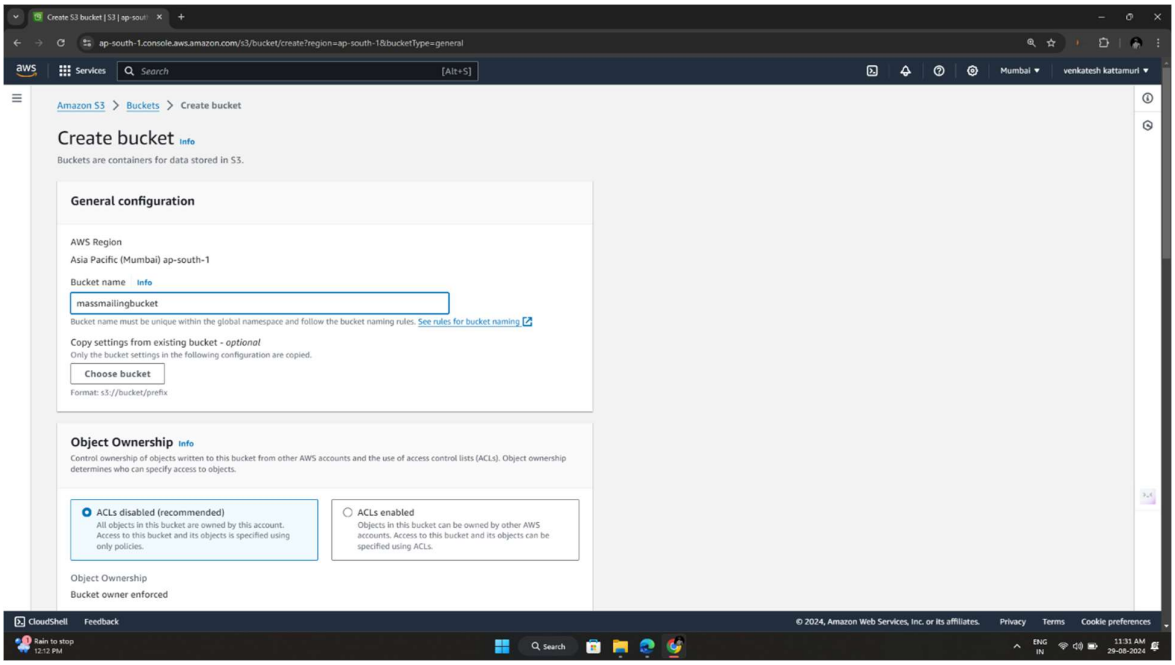



This is IAM roles image where mass mailing is made.

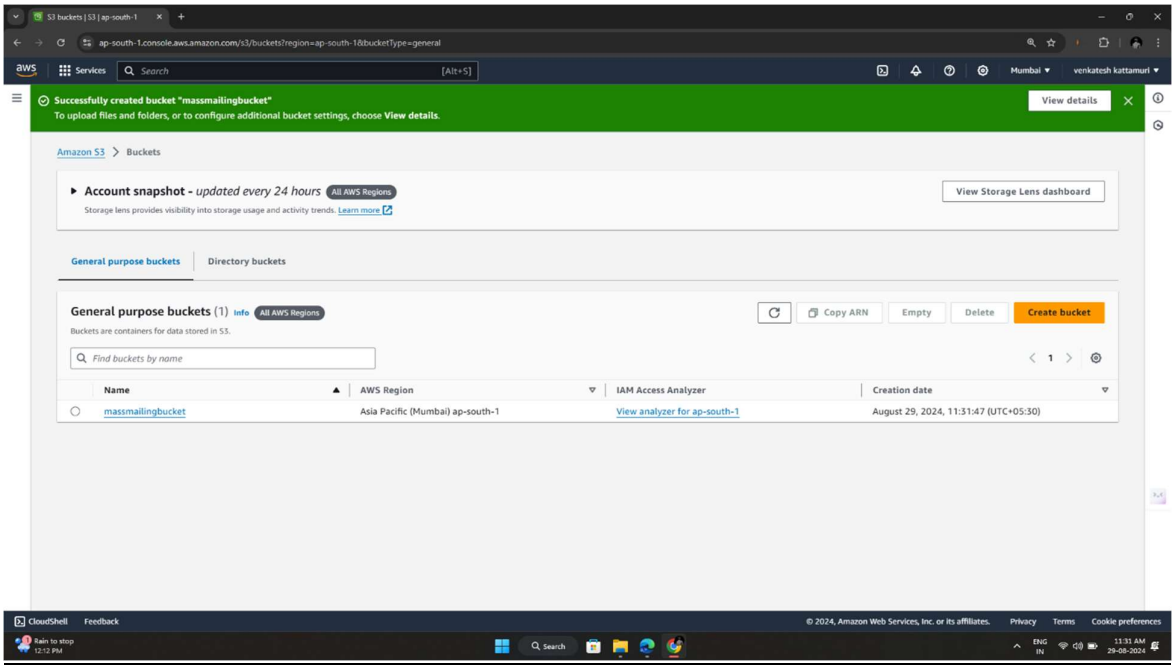



Here are the permission Policies that I used for my mass mailing role.

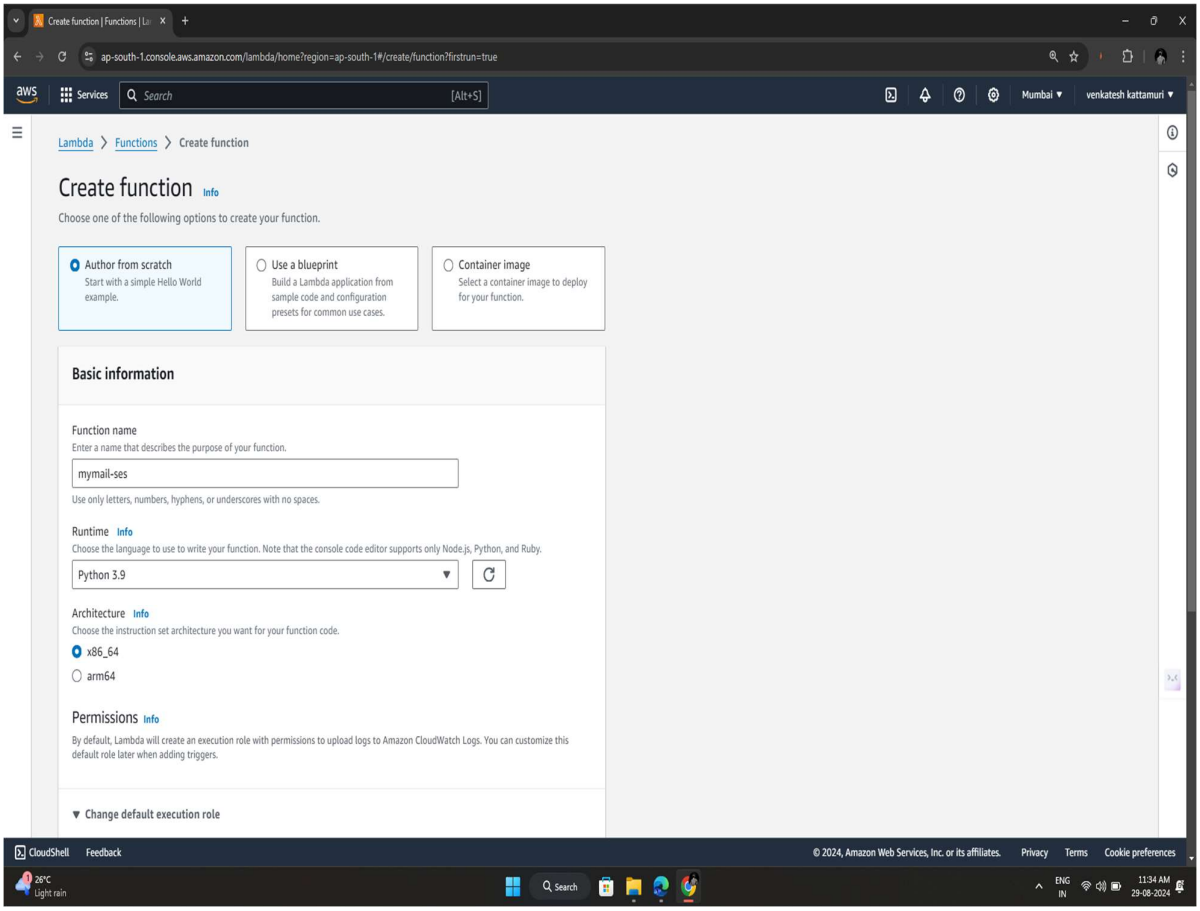
An S3 bucket is created with name **massmailingbucket** where all type of objects i.e. images, videos, text files and other type of files can be uploaded.



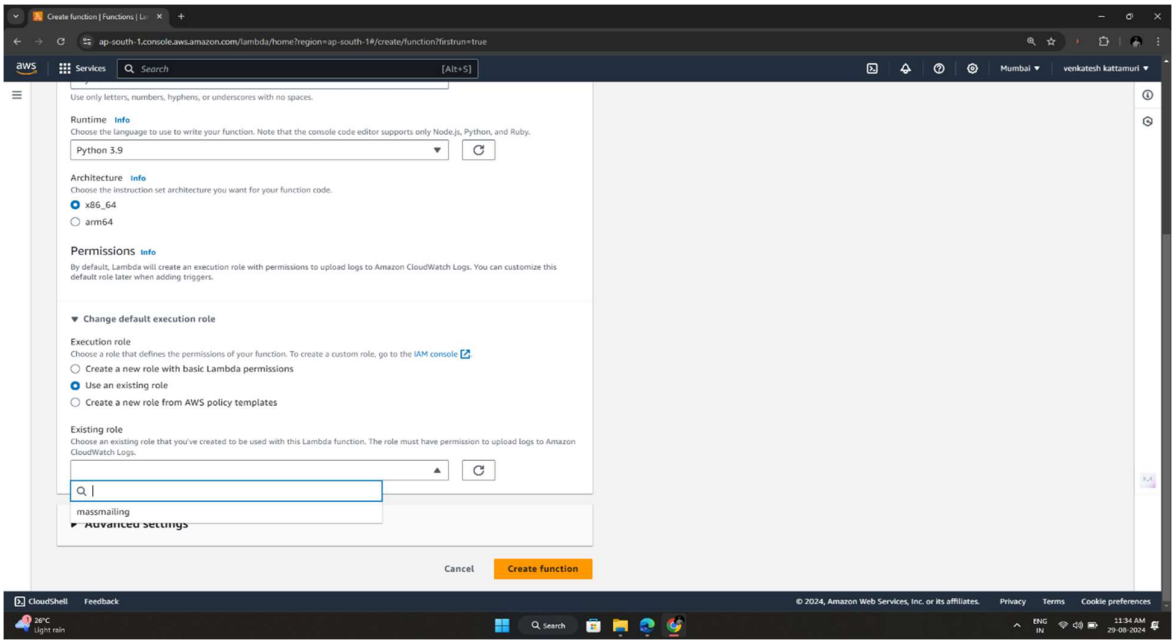

Here is my S3 bucket with name **information-storing-device.**



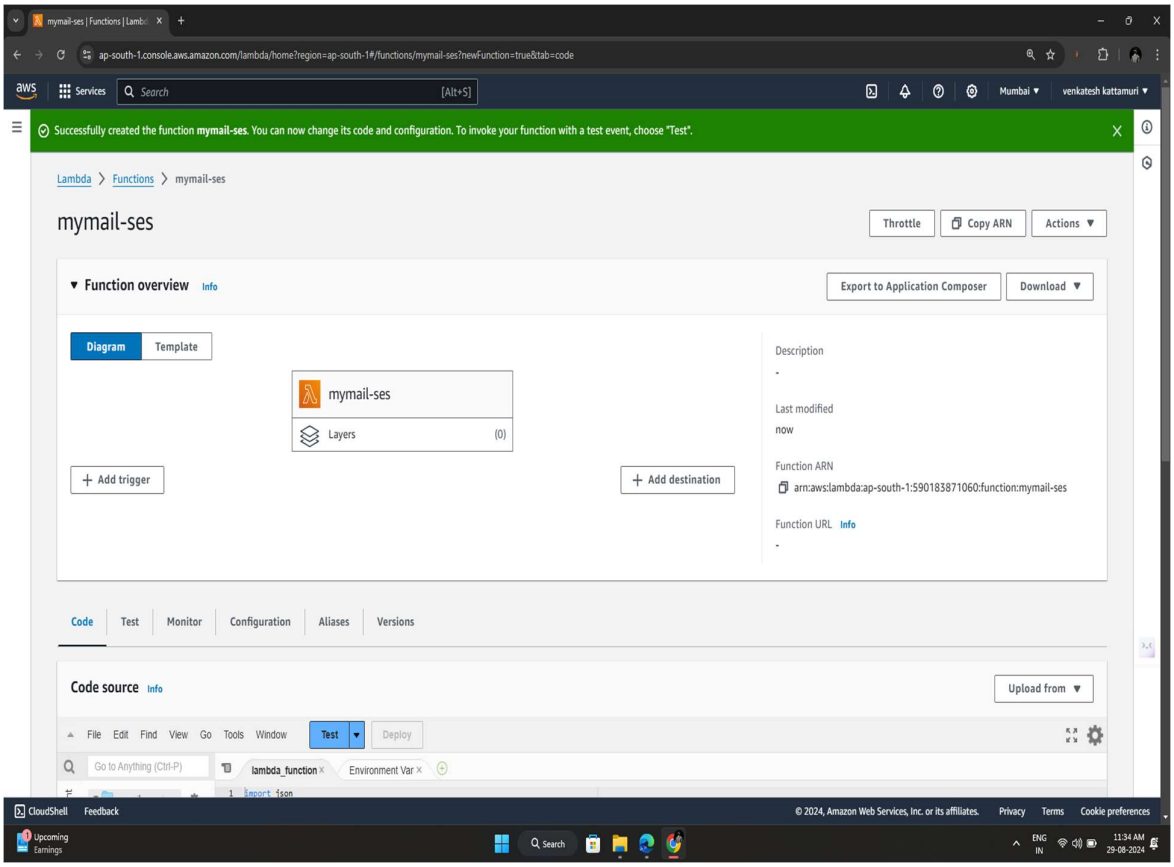
## Create a lambda funtion



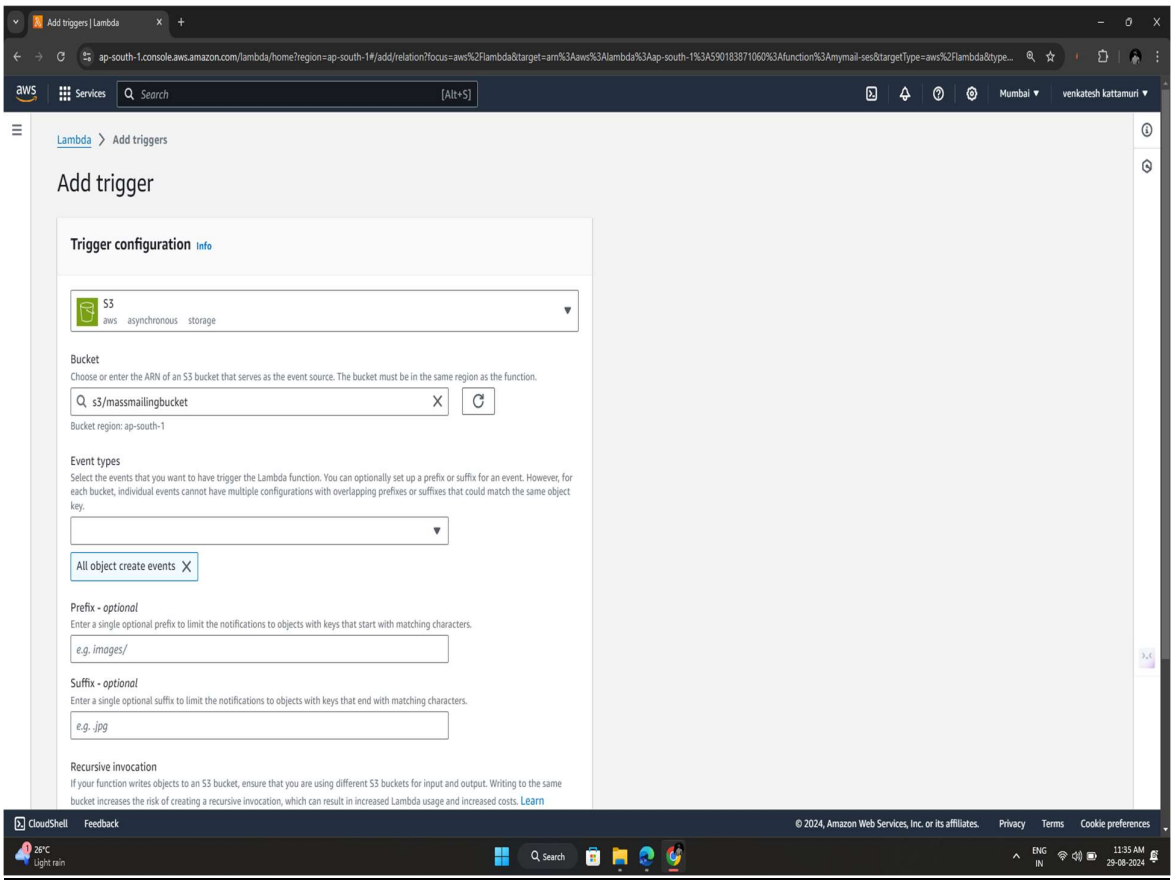

I created a lambda function named as **mymail-ses**




I had selected the existing role which i created earlier

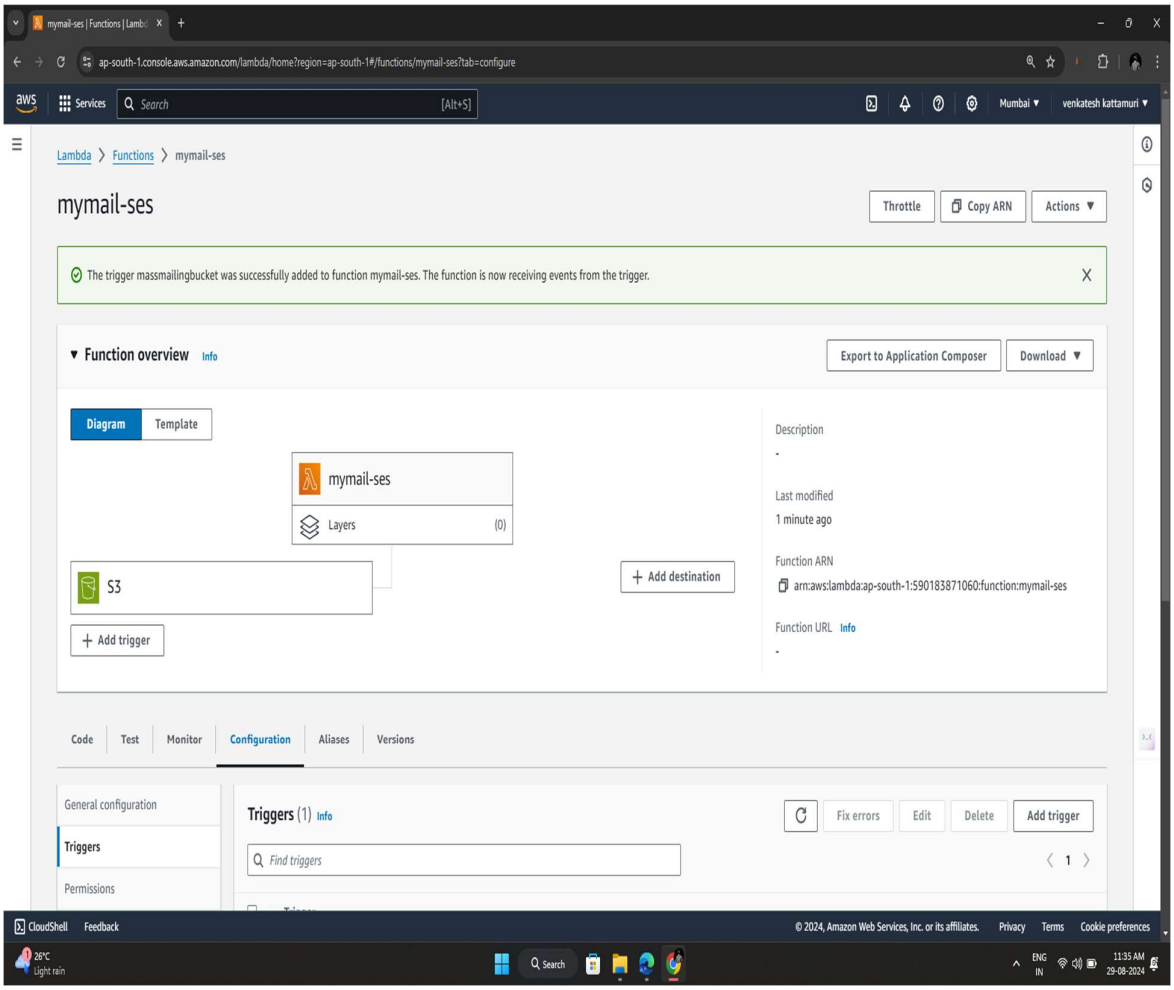



**Lambda function is created**




Add a trigger to the lambda function

Choose S3 in that and select your bucket over there.


mymail-ses | Functions | Lambda
ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions/mymail-ses?tab=configure
Services
Search
[Alt+S]
Mumbai
venkatesh kattamuri
Lambda
Functions
mymail-ses
mymail-ses
Throttle
Copy ARN
Actions
The trigger massmailingbucket was successfully added to function mymail-ses. The function is now receiving events from the trigger.
Function overview
Info
Export to Application Composer
Download
Diagram
Template
mymail-ses
Layers
(0)
S3
Add trigger
Add destination
Description
-
Last modified
1 minute ago
Function ARN
arn:aws:lambda:ap-south-1:590183871060:function:mymail-ses
Function URL
Info
-
Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Triggers (1)
Info
Fix errors
Edit
Delete
Add trigger
Find triggers
1
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
26°C
Light rain
Search
ENG
IN
11:35 AM
29-08-2024

Amazon Simple Email Service (SES) is a cloud-based email sending and receiving service provided by Amazon Web Services (AWS). It is designed to be a scalable and cost-effective solution for sending and receiving emails, making it ideal for businesses of all sizes. Here's a brief overview:

Key Features of Amazon SES

**1. Email Sending:**

Transactional Emails: SES is commonly used for sending transactional emails, such as order confirmations, password resets, and notifications.

Marketing Emails: Businesses can use SES to send bulk marketing emails, newsletters, and promotional content.

High Deliverability: SES is built on the same infrastructure used by Amazon.com for its email services, ensuring high deliverability rates and compliance with anti-spam regulations.

**2. Email Receiving:**

- SES can also be configured to receive emails, which allows users to process incoming emails, store them in S3, or trigger actions via Lambda functions.

**3. Cost-Effective:**

- SES is a pay-as-you-go service, meaning users only pay for the number of emails sent and received. This pricing model makes it highly cost-effective for both small-scale and large-scale email operations.

**4. Security:**

- SES integrates with AWS Identity and Access Management (IAM) to provide fine-grained control over who can send emails and access email-related resources.

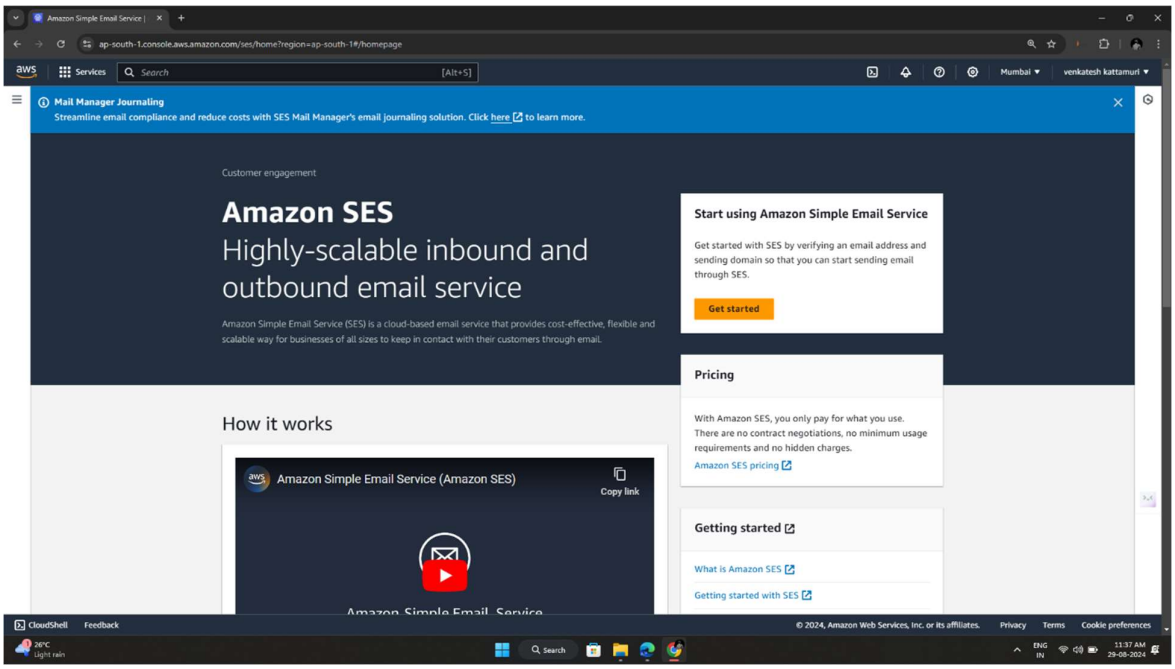
- It also supports domain authentication mechanisms like DKIM and SPF, which help prevent email spoofing and improve deliverability.

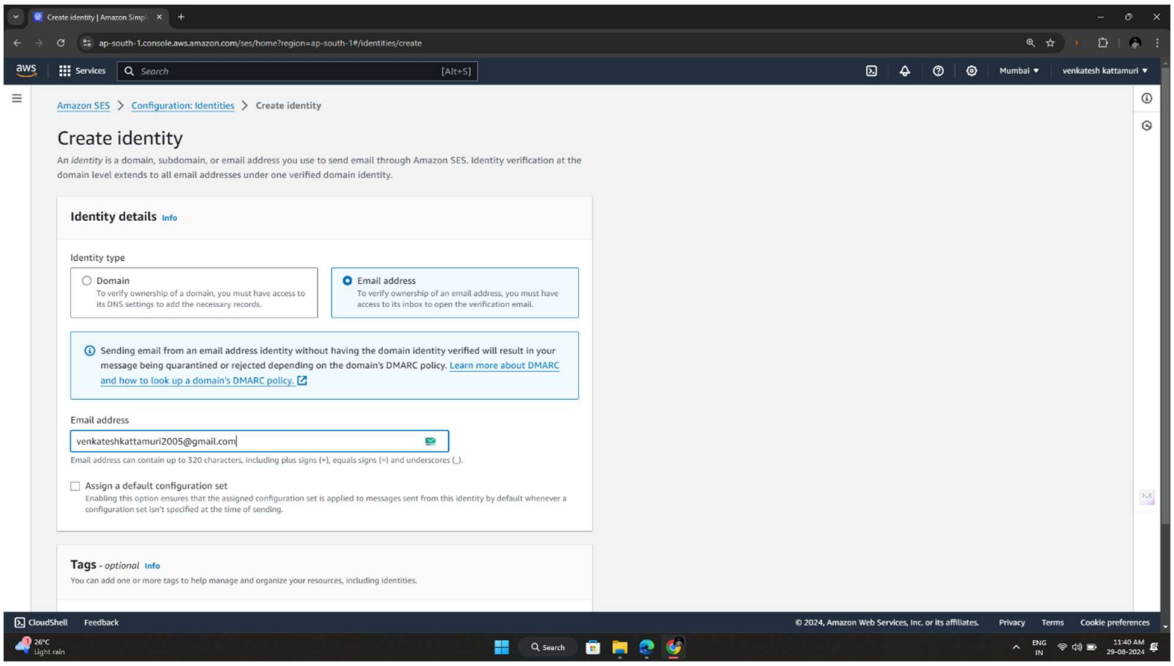
**5. Scalability:**

- SES is designed to handle a high volume of emails, allowing businesses to scale their email operations as needed without worrying about infrastructure management.

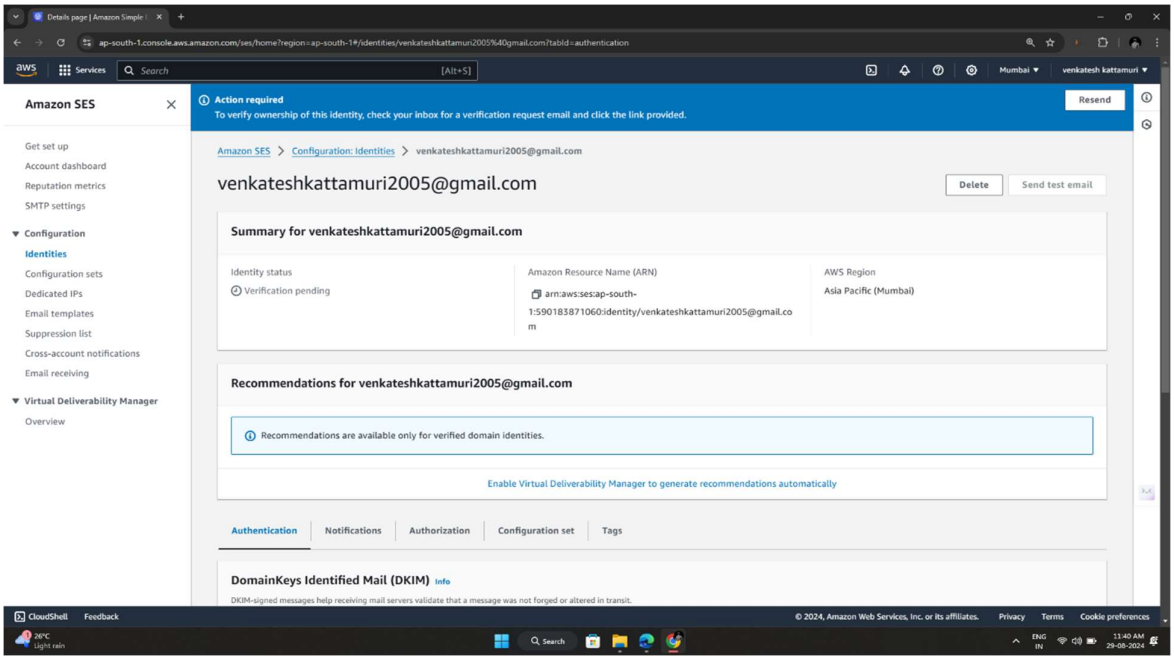
**6. Customizable Email Sending:**

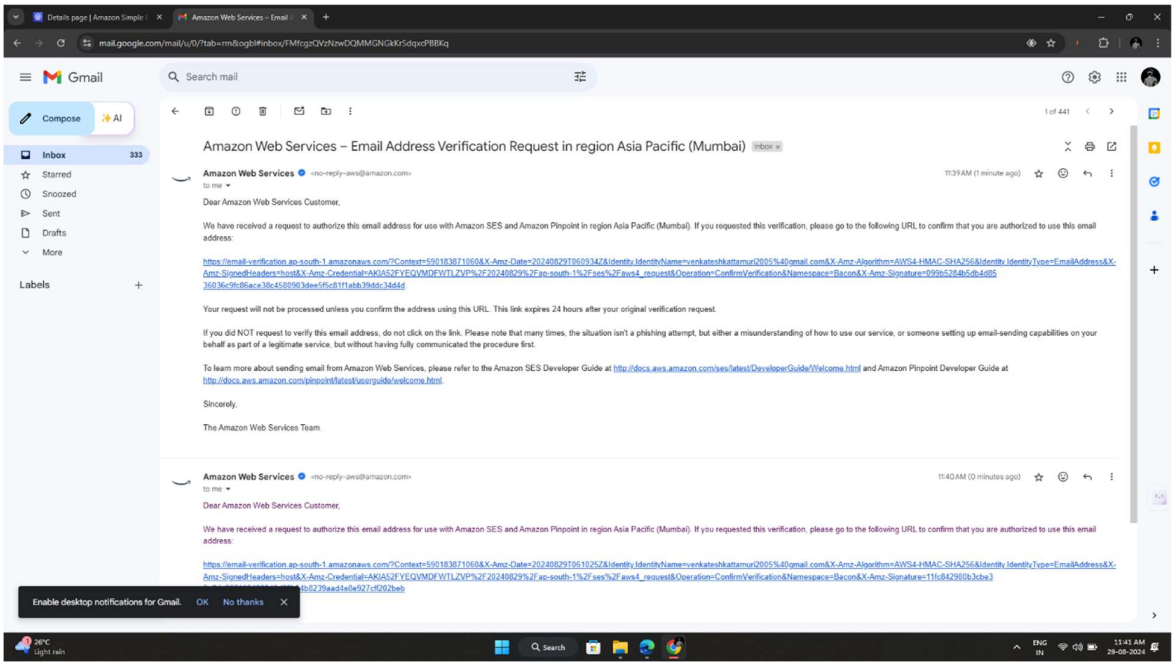
- Users can customize email headers, content, and tracking options. SES supports multiple email formats, including plain text, HTML, and attachments.



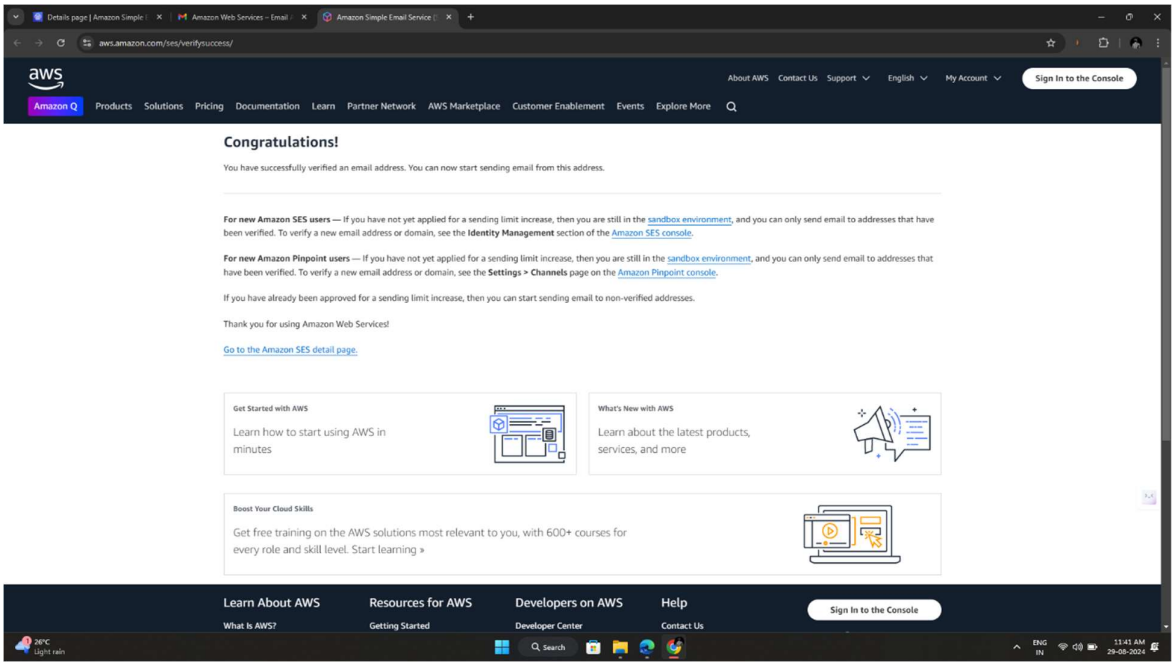


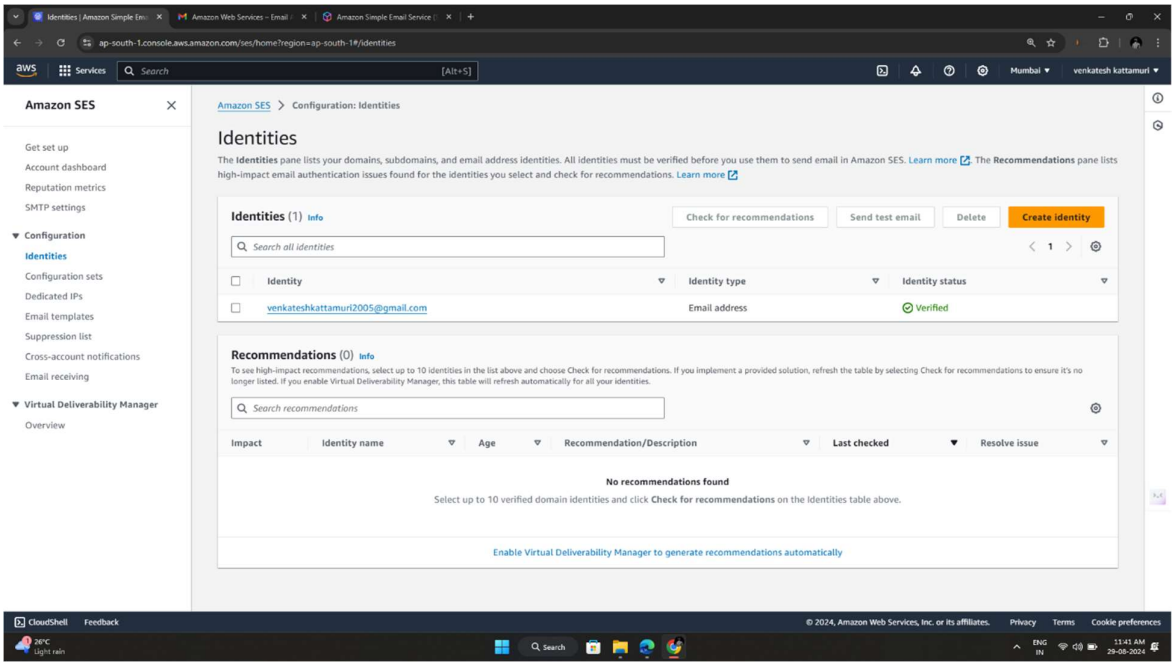

In AMAZON SES go to create identity and select email address and create it.



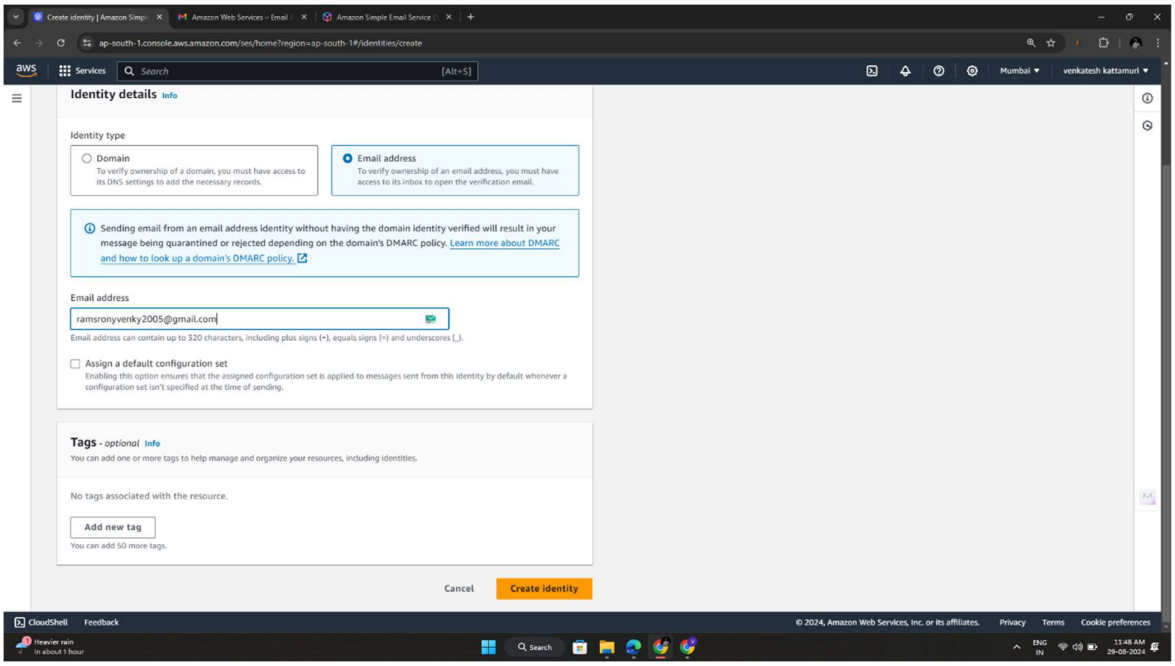



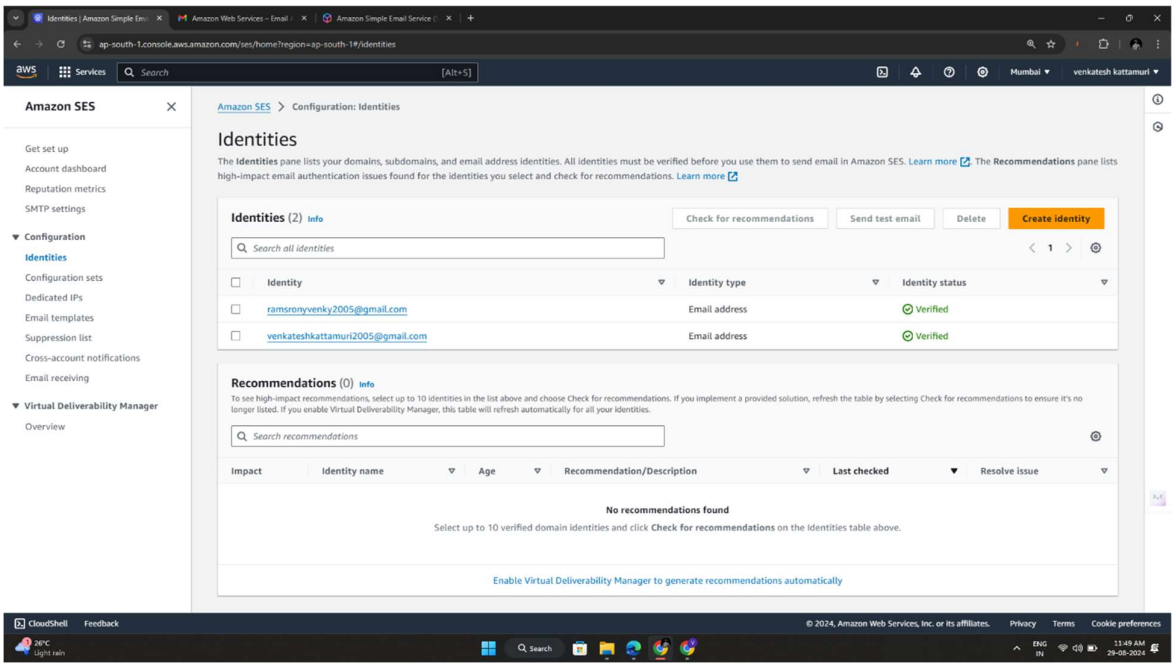

And get confirmation mail to your mail id



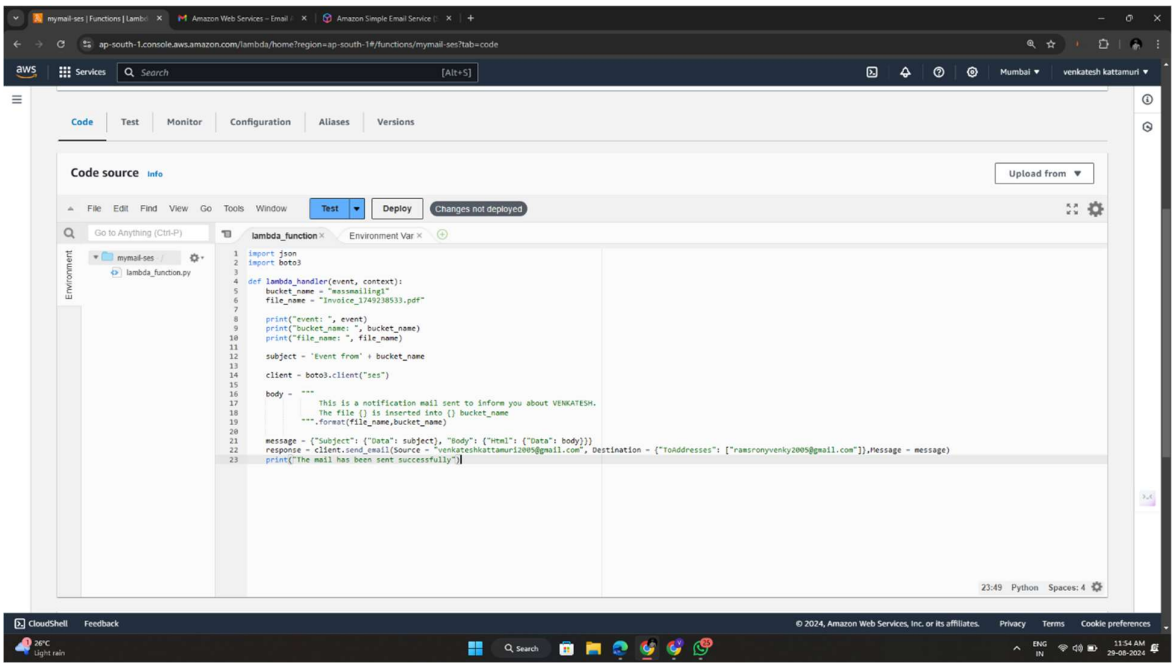




**After confirmation the identity is verified and create another identity**




**Two identities are created and verified successfully.**

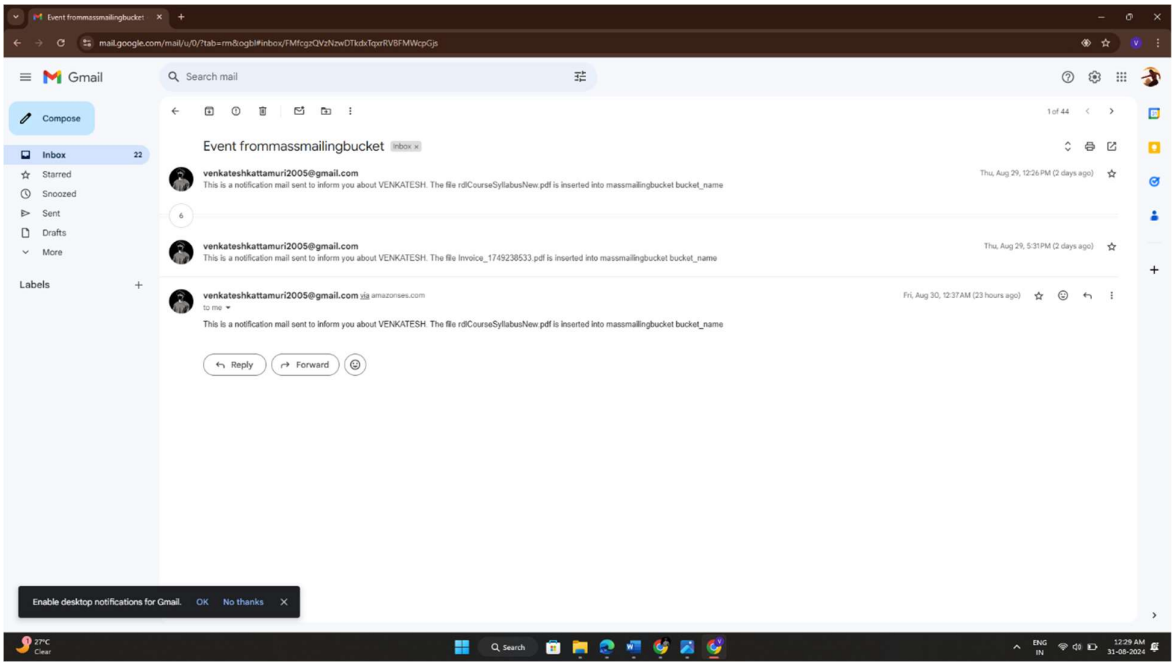



“Now, in the Lambda function, write the code for receiving emails and sending them to the specified recipient.”

**"Conduct testing of the code and proceed with its deployment."**

**CODE:**

```
import json

import boto3


def lambda_handler(event, context):

    bucket_name = "massmailingbucket"

    file_name = "rdlCourseSyllabusNew.pdf"


    print("event: ", event)

    print("bucket_name: ", bucket_name)

    print("file_name: ", file_name)


    subject = 'Event from' + bucket_name


    client = boto3.client("ses")


    body =  """

            This is a notification mail sent to inform you about VENKATESH.

            The file {} is inserted into {} bucket_name

          """.format(file_name,bucket_name)


    message = {"Subject": {"Data": subject}, "Body": {"Html": {"Data": body}}}

    response = client.send_email(Source = "venkateshkattamuri2005@gmail.com", Destination =
{"ToAddresses": ["ramsronyvenky2005@gmail.com"]},Message = message)

    print("The mail has been sent successfully")
```




And finally I got the email from sender side.

# Conclusion

In this project, we successfully designed and implemented an email forwarding system using AWS Simple Email Service (SES), AWS Lambda, Amazon S3, and IAM Roles. The integration of these AWS services has enabled the creation of a highly scalable and efficient email forwarding solution.

Key Achievements:

1. AWS SES Integration: We configured SES to handle incoming emails, setting up necessary rules for processing and forwarding them.

2. AWS Lambda Function Deployment: A Lambda function was developed to process and forward emails based on predefined criteria, automating the email handling process.

3. Amazon S3 Integration: Emails were stored in Amazon S3 buckets for archival and retrieval purposes, ensuring that a secure and scalable storage solution is in place.

4. IAM Role Management: Appropriate IAM roles and permissions were established to secure access, ensuring that SES, Lambda, and S3 services operate within their intended scope while maintaining a high level of security.

The project demonstrates a robust and efficient approach to automating email management processes using a combination of serverless and storage technologies. The implemented system is now fully operational and ready for deployment in a production environment, offering scalability, reliability, and cost-effectiveness.

## Bibliography

**1.** Gokboru Tech for AWS learning basics and getting experience.

**2**. AWS documentation and manuals for learning how to use their tools.

**3**. Project images are the screenshot of the project.

**4.** Code was written and helped by Trainer Mr. Yash Raj.