

# Banking Bok Using Amazon Lex

## CloudWatch and SES

### Identity and Access Management (IAM)

IAM web service allows us to securely control access to AWS resources. Here we can check which users are authenticated and what permissions they have to use the resources. IAM does not require region selection.

### AWS Lambda

AWS Lambda is one of the serverless services. In these services, we only write the code we want to run and make the necessary resource settings. We do not have to deal with issues such as which server the code is running on, the administration, maintenance and security of the server. Lambda codes run in containers. In this way, it can be easily scaled.

### CloudWatch

CloudWatch; It provides data and actionable insights so you can monitor your applications, respond to system-wide performance changes, optimize resource usage, and have a unified view of operational status. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing a unified view of AWS resources, applications, and services running on AWS and on-premises servers. Using CloudWatch, you can detect anomalous behavior in your environments, set up alarms, visualize

images and metrics side-by-side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

## **AWS EventBridge**

AWS EventBridge is a serverless event bridge that makes it easy to provide real-time access to data changes in AWS services and SaaS applications without writing code. CloudWatch Events builds and extends on Amazon EventBridge. It uses the same service API and endpoint, and the same service infrastructure. Nothing changes for existing CloudWatch Events customers.

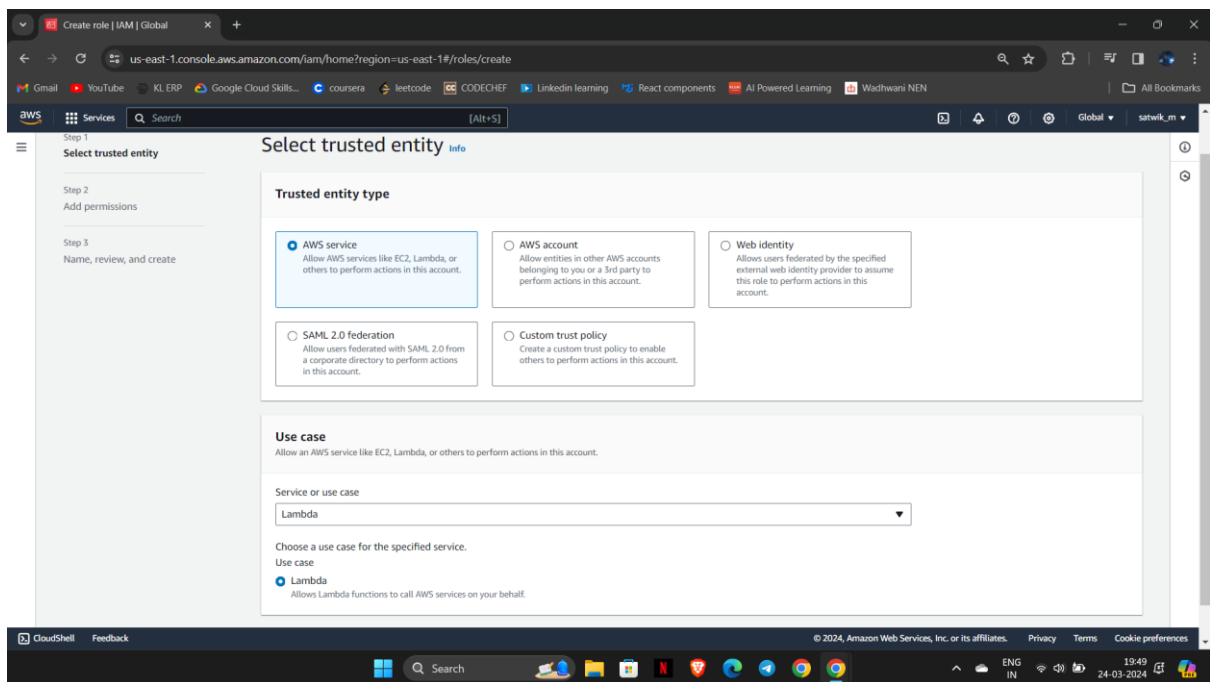
## **AWS Simple E-mail Service**

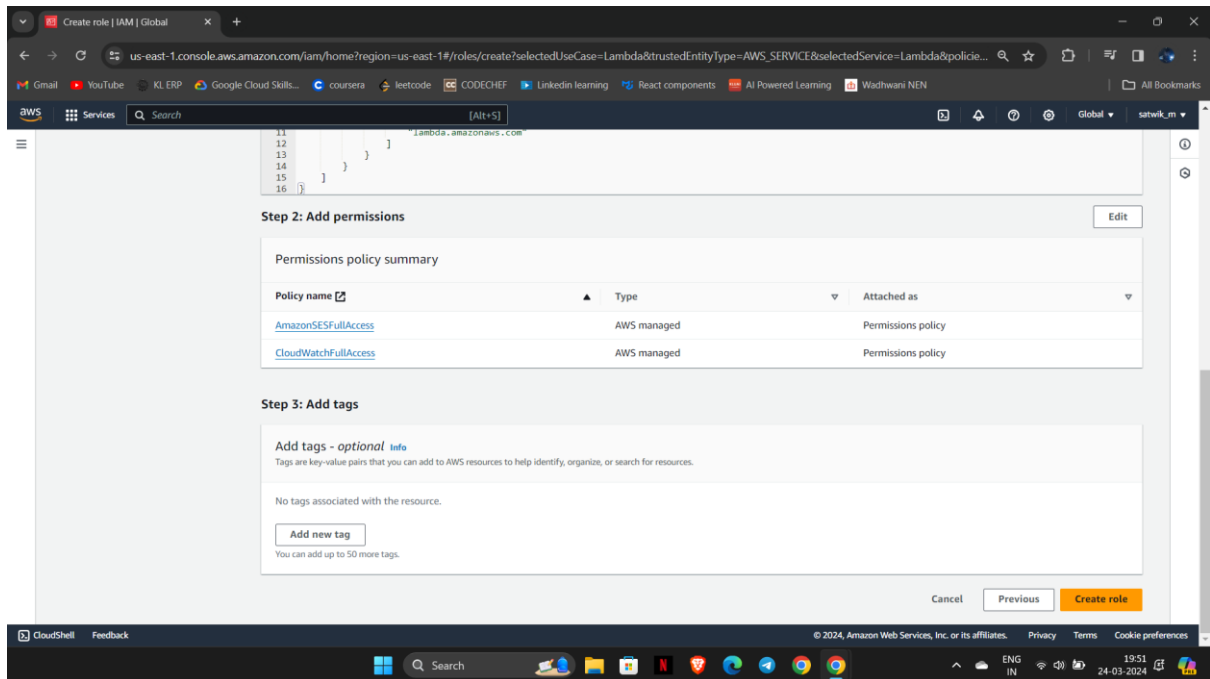
Amazon SES is an easy, reliable, low-cost, pay-as-you-go cloud-based email service designed to help app developers and digital media professionals send notification emails. It is a service that provides solutions for businesses of all sizes that communicate with their customers via email, without dealing with infrastructure problems such as Mail Server management, Network configuration, and most importantly, Spam and IP reputation problems. All new SES accounts start in a sandbox, where you can test the capabilities of Amazon SES. When your account is in the sandbox, you can only send email to your verified sender identities, with a typical limit set to 200 emails per day. A verified sender identity is either an email addresses or domain that you verified ownership of. If you want to send an e-mail to your e-mail list with SES, you must open a request to Amazon to exit the sandbox. In this tutorial, we will perform our operations from the sandbox.

## Steps:

### 1. Setting up IAM roles

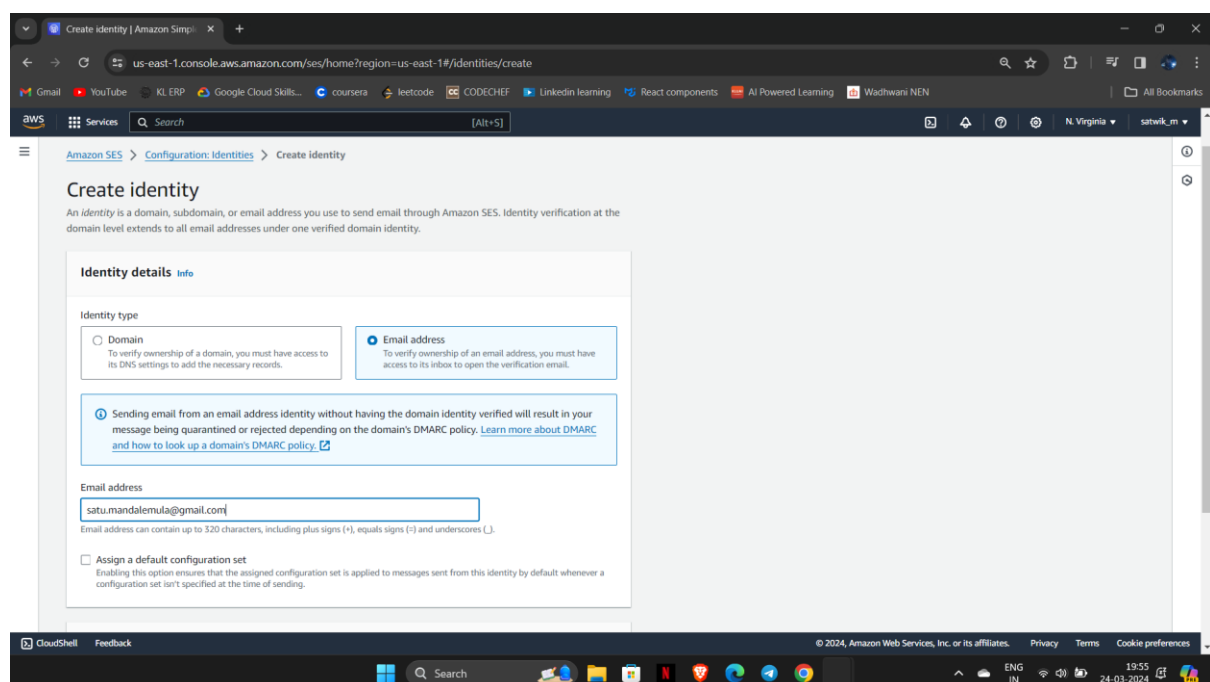
First we should set up IAM roles for security. Open the AWS console and write IAM to searching area. On the screen that opens click roles and create role.





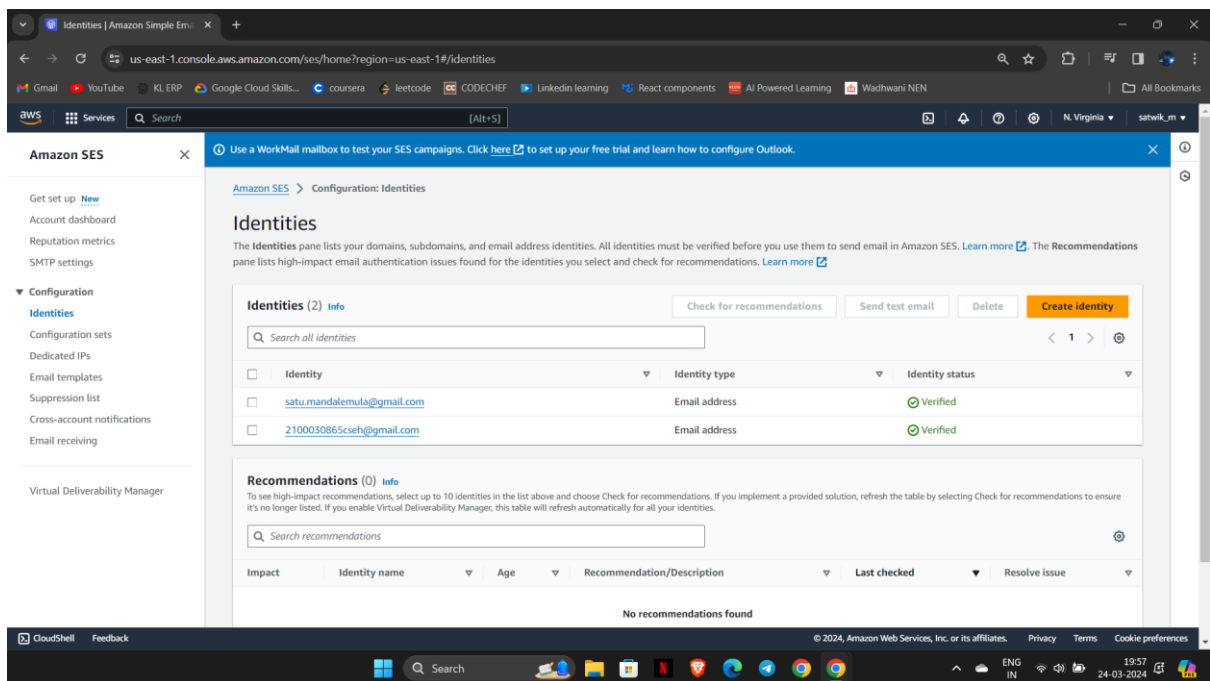
## 2.Amazon SES

Before configuring the other steps, you must ensure that they are all in the same region. My transactions will be in the us-central-1 region.



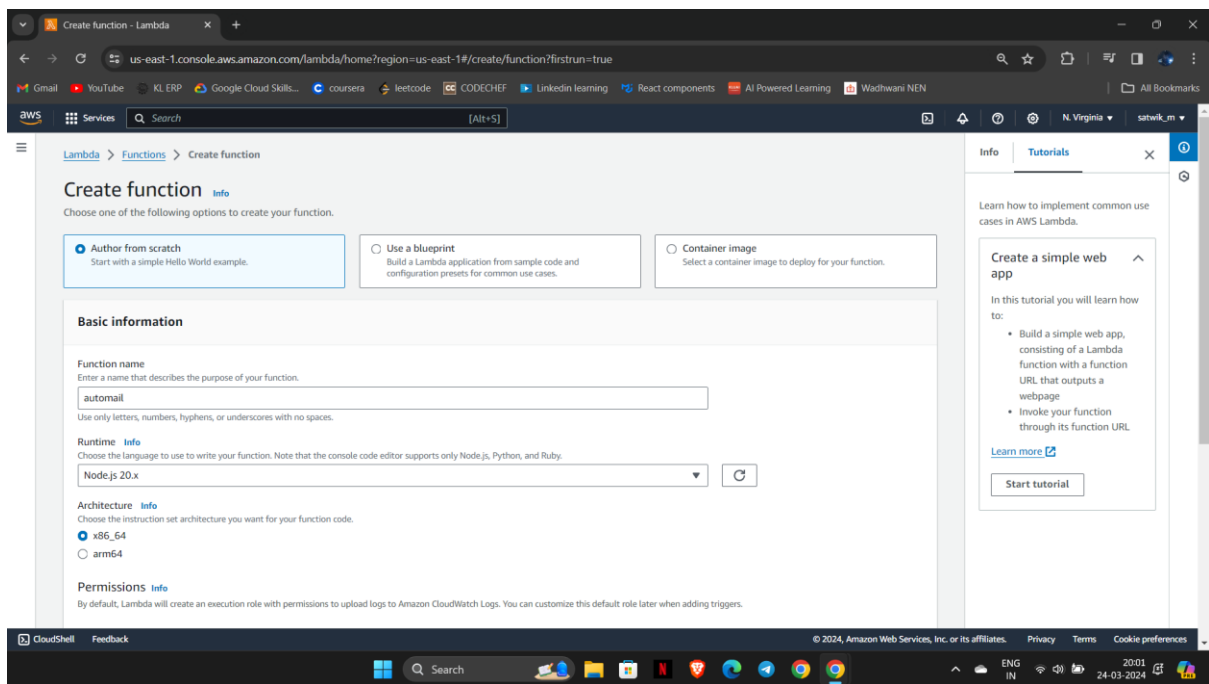
Since we are working in sandbox, we have to do the mail verification process. I have defined one mail. Now press the Create Identity button to define the other.

You will receive a verification mail. After verifying your e-mail address, the status section will be verified.

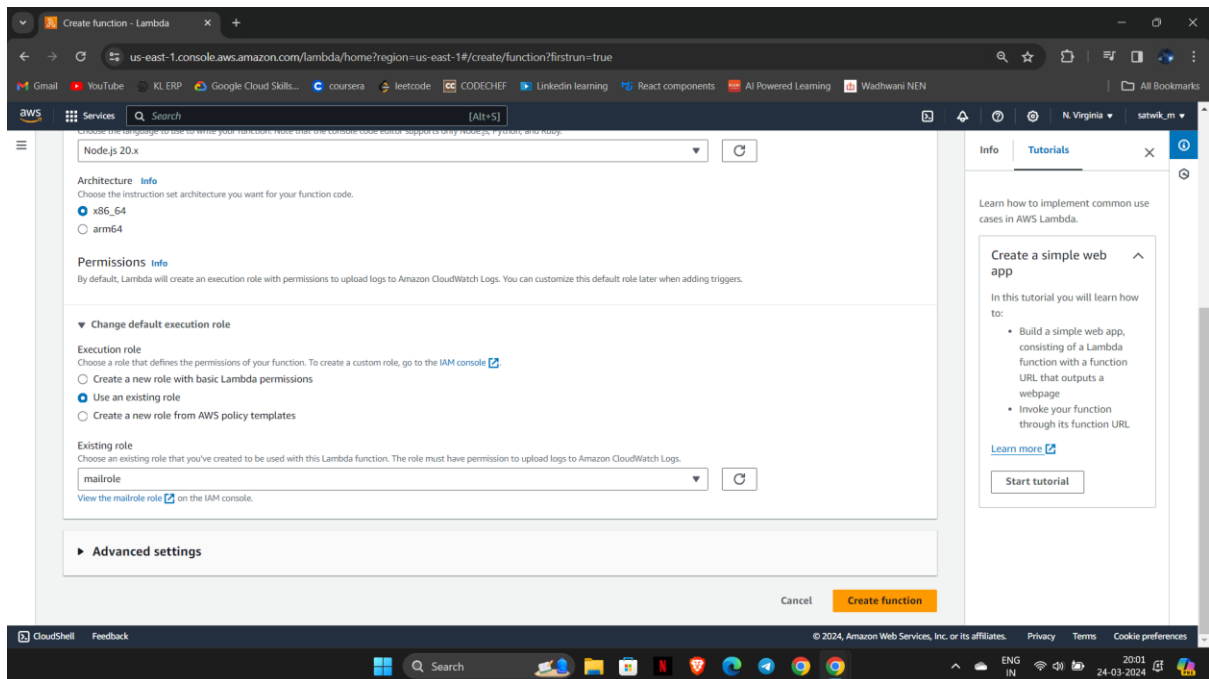


### 3. Writing Lambda Function

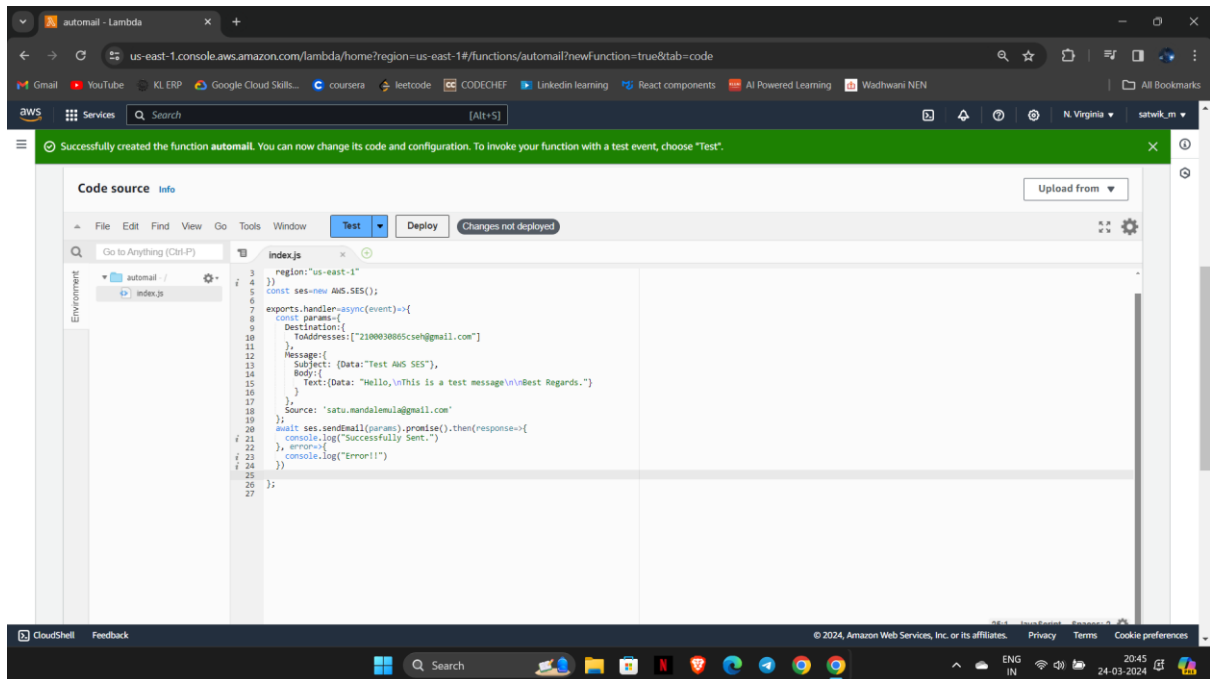
On amazon console write Lamda to search area and click Create Function button. Enter a name that describes the purpose of your function and Choose the language to use to write your function. We will use node.js.



At the bottom of the page, in the Change Default Execution Role section, we click “use an existing role” and select the name of the created role to use the role we just created and we create the Lambda service.



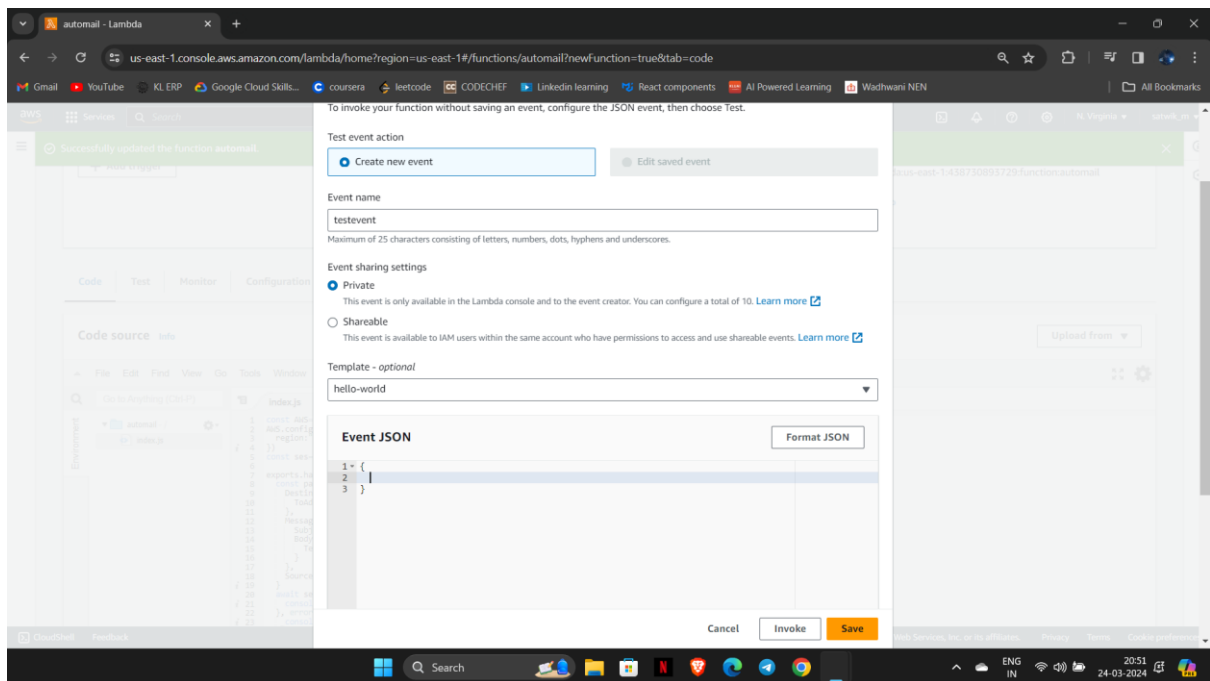
At the bottom of the opening page, there is a code section. We will write the code here then deploy and test it.



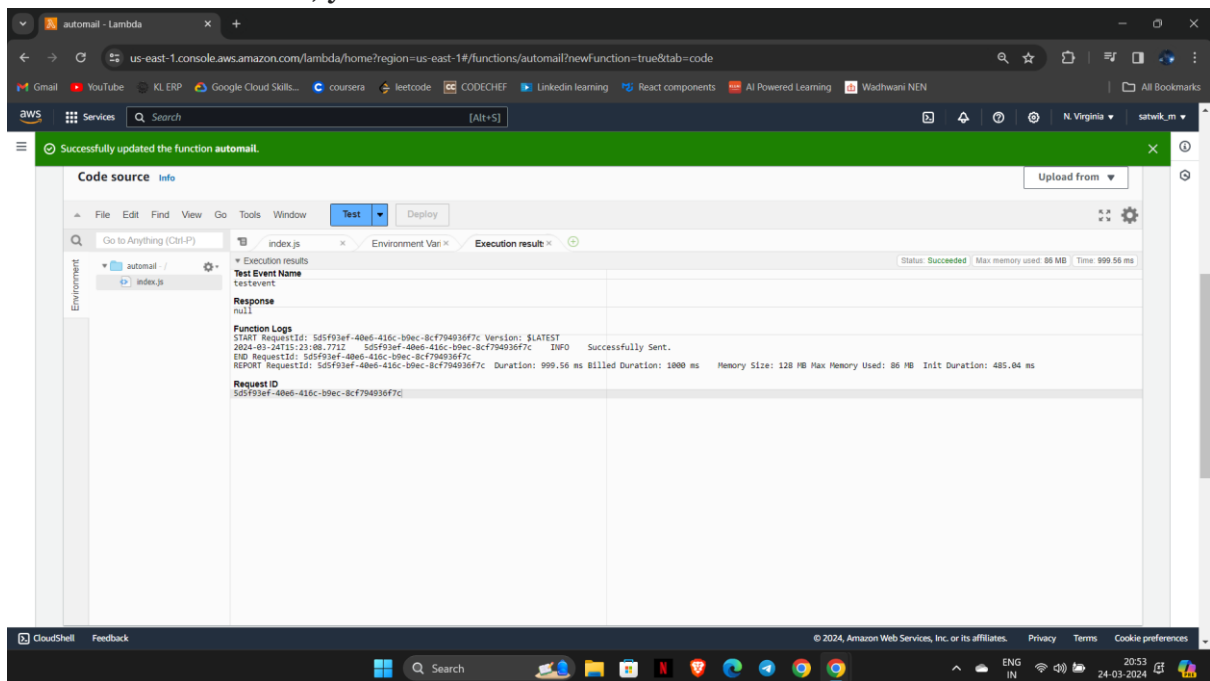
In the code above;

- importing aws sdk
- setting up region (change this part according to your region)
- creating an object from AWS.SES()
- In params variable, we specify mail features. These are destination, message and source. If you want, you can add multiple e-mail address to the “ToAddresses” list.
- Lastly we create an arrow function and we log whether the mail is sent successfully or not.

If you have done these steps click the deploy button to deploy your code. Then click test. On the opening page write a test name, delete sample key values and click Create.



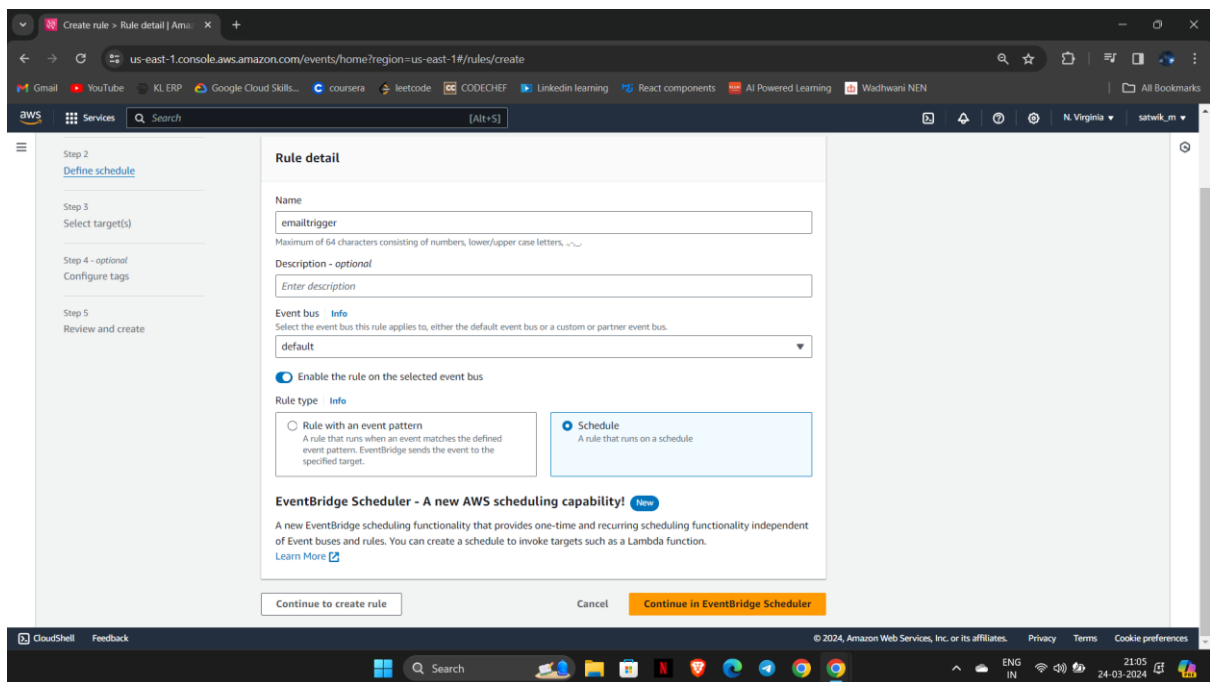
After that click test button again to run your test. You should see the “Successfully sent” message in the execution result page. If you look at destination mailbox, you can see the test mail.





## 4. Triggering with CloudWatch

In amazon console write CloudWatch to search area. On the page that opens, click on the rules under the events section in the menu on the left. Then click the “Go to the Amazon EventBridge” button. In this section we are going to create a rule. Click the “Create rule” button on the opening page. Set up a rule name.



In the Define pattern section, we click on the schedule and select the cron expression.

1. First part is minutes
2. Second part is hours
3. Third part is day-of-month. \* declares every day.
4. Fourth part is month, \* declares every month.

5. Fifth part is day-of-week, ? indicates that it does not matter what day it will be.

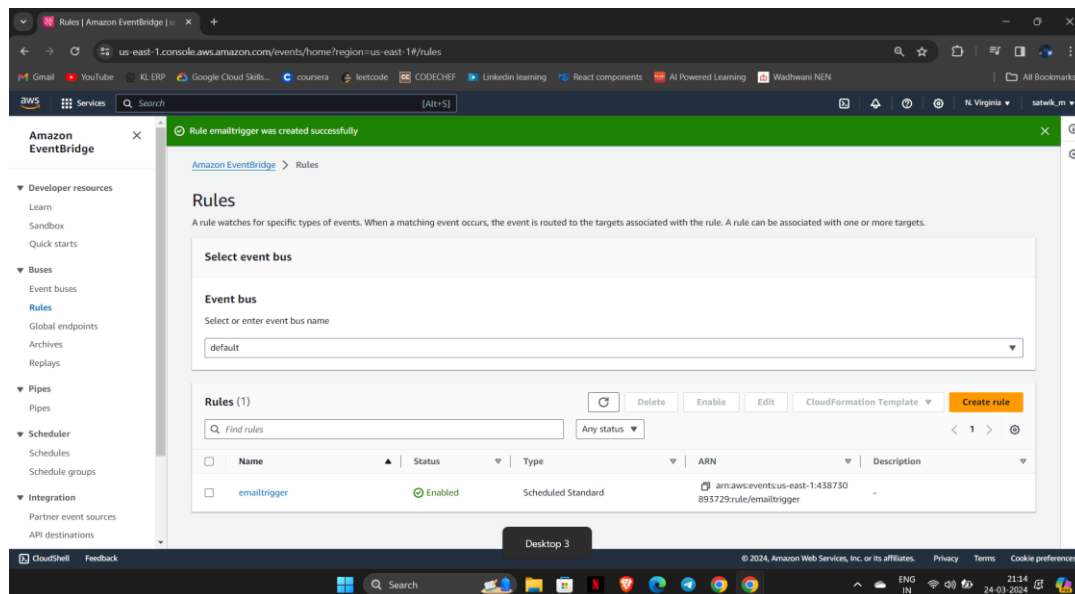
6. Last part is year.

The screenshot shows the 'Define schedule' step in the AWS EventBridge console. The left sidebar lists the steps: Step 1: Define rule detail, Step 2: Define schedule (active), Step 3: Select target(s), Step 4 - optional: Configure tags, and Step 5: Review and create. The main content area is titled 'Define schedule' and includes a 'Schedule pattern' section. It offers two options: 'A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month' (selected) and 'A schedule that runs at a regular rate, such as every 10 minutes'. Below this is a 'Cron expression' field with a dropdown set to 'cron' and a form with fields for Minutes (15), Hours (2), Day of month (\*), Month (\*), Day of week (?), and Year (\*). A 'Next 10 trigger date(s)' section shows a list of dates from Monday, Mar 25, 2024, to Wednesday, Apr 3, 2024, all at 07:45 AM GMT+5:30. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

Finally, we select the Lambda function we created in the select target section and create the rule.

The screenshot shows the 'Select target(s)' step in the AWS EventBridge console. The left sidebar shows the steps: Step 2: Define schedule, Step 3: Select target(s) (active), Step 4 - optional: Configure tags, and Step 5: Review and create. The main content area is titled 'Select target(s)' and includes a 'Permissions' note. Below is the 'Target 1' section. It has a 'Target types' section with three options: 'EventBridge event bus', 'EventBridge API destination', and 'AWS service' (selected). The 'Select a target' dropdown is set to 'Lambda function'. Below this, the 'Function' dropdown is set to 'automail'. There are 'Configure version/alias' and 'Additional settings' sections. At the bottom are 'Add another target', 'Cancel', 'Skip to Review and create', 'Previous', and 'Next' buttons.

Under the created rules, you can see that the rule is running with the enabled mark in the status section.



If you check the destination mailbox when the time you set comes, you can see the mail you sent.

