Creating a lambda function in AWS to email daily reports

Name: Raj Sharma   **Reg. No.:** RA2011028010055

**Aim :** Automate Sending Emails at a Specific Time with AWS Lambda, CloudWatch and SES

**Pre-requisites :** AWS Console, Amazon SES, Amazon Lambda, Amazon CloudWatch.

Procedure :

We are going automate sending email to a person or a group of people. AWS **Cloudwatch** is used to setup a schedule to trigger AWS **Lambda** function and then its going to use AWS **SES** (**Simple Email Service**) to send out emails to people.

Steps:

1. Go to AWS SES (Simple email service), click on "Create Identity".
   Use email address as a type and type the email address.
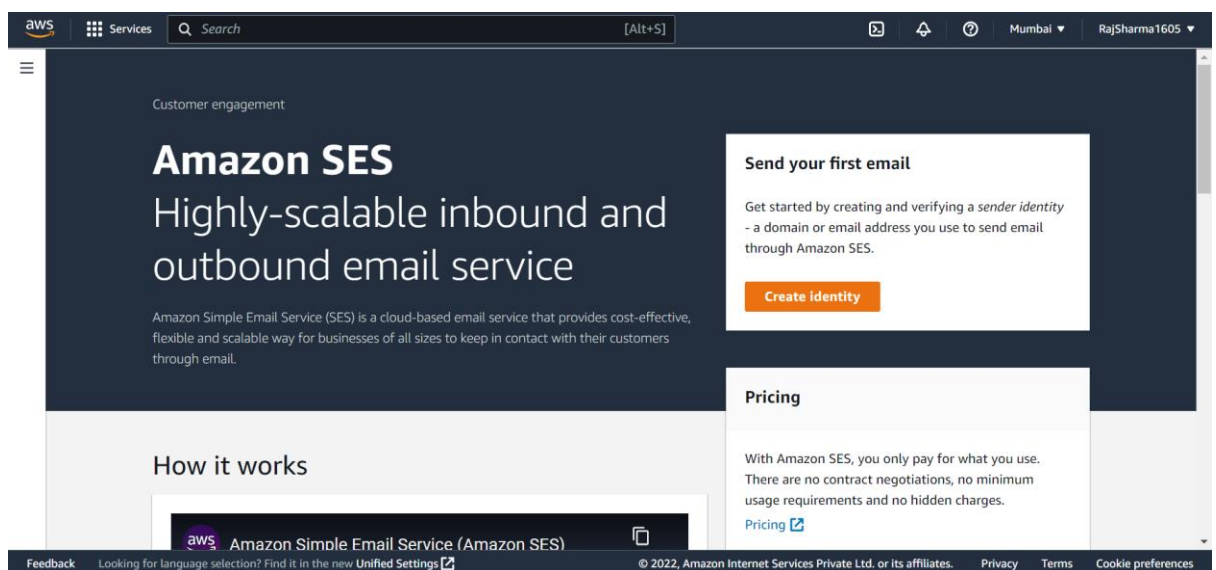
2. Verify the email address that reviewed an email from aws to tell you to verify that.
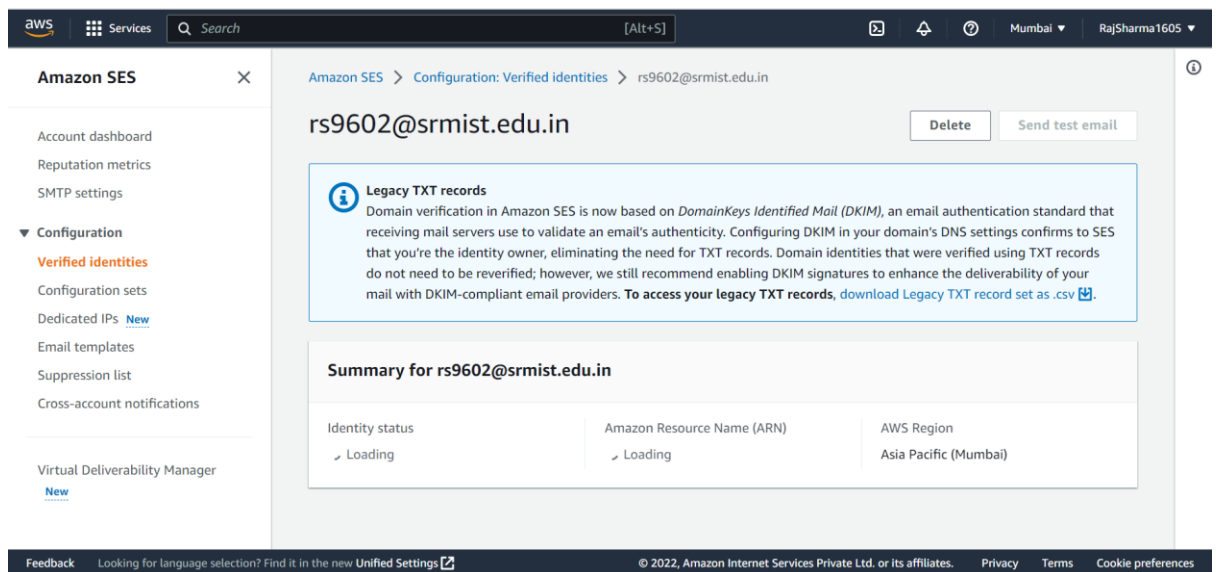
3. Create two identities (email address).
   One for sending emails and another for receiving.
4. Create an IAM role.
   Give Use case as lambda and give full access to cloudwatch, SES.
5. Go to Lambda Service, create a lambda function.
   Give name, runtime as NodeJS, execution role as created IAM role
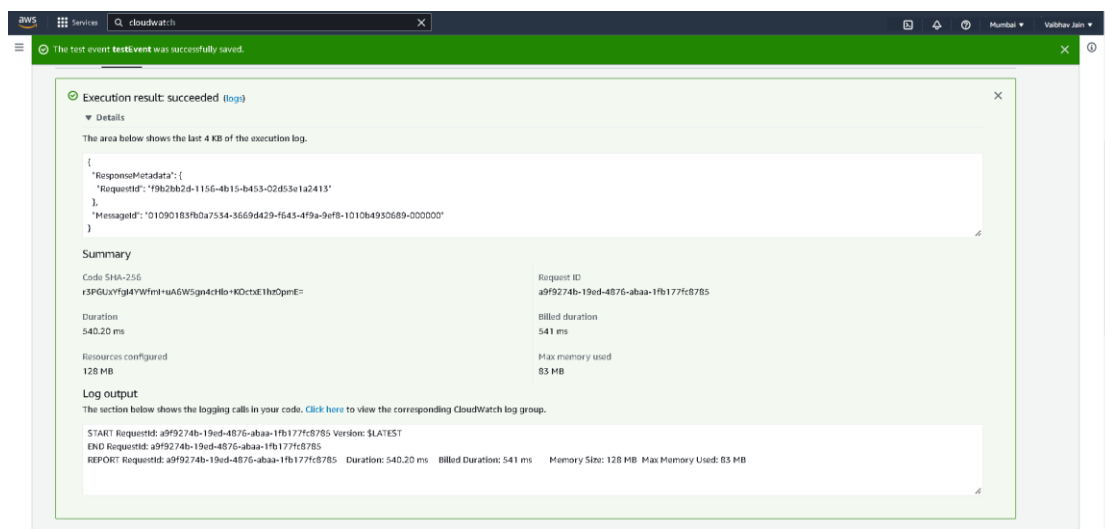   previously.
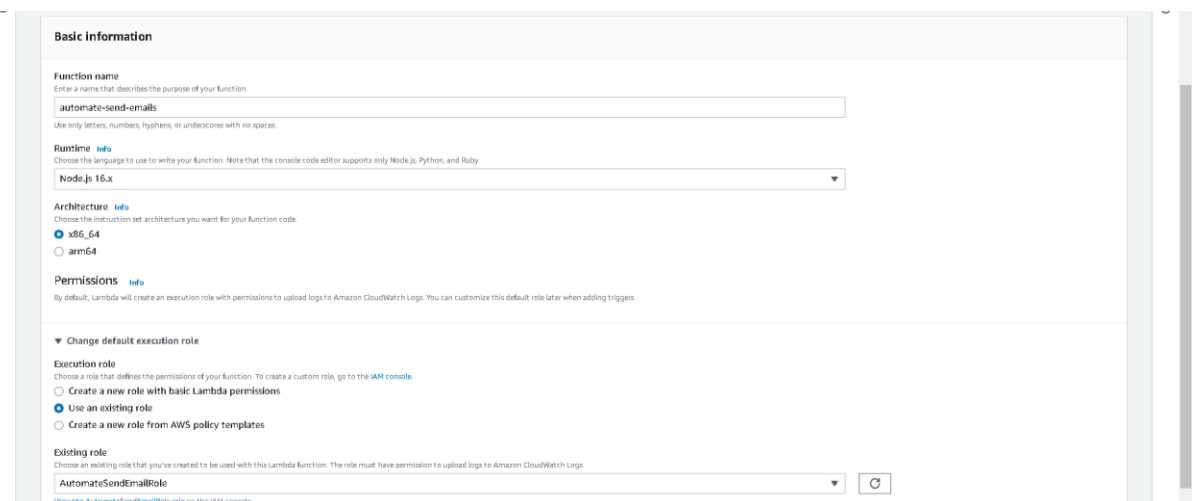
6. Use this template for the code:
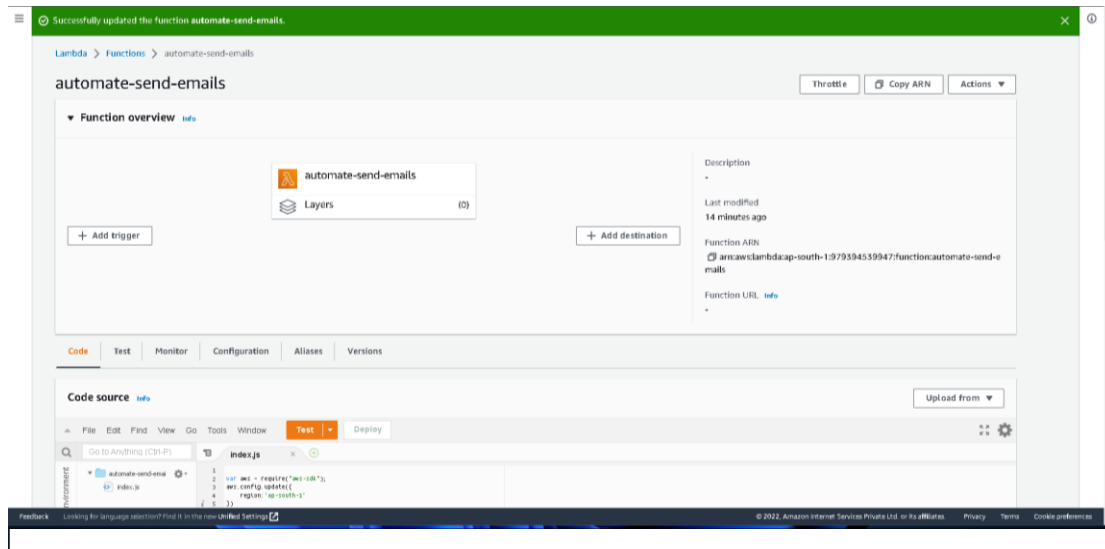
```
1)
2)  var aws = require("aws-sdk");
3)  var ses = new aws.SES({ region: "us-west-2" });
4)  exports.handler = async function (event) {
5)    var params = {
6)      Destination: {
7)        ToAddresses: ["RecipientEmailAddress"],
8)      },
9)      Message: {
10)        Body: {
11)          Text: { Data: "Test" },
12)        },
13)
14)        Subject: { Data: "Test Email" },
15)      },
16)      Source: "SourceEmailAddress",
17)    };
18)
19)    return ses.sendEmail(params).promise()
20)  };
```

7. Click on Deploy and then TEST, you wil receive the message in your mentioned emails.

⊘ Execution result: succeeded (logs)　　　　　　　　　　　　　　　　　　　　　　　　　　×

▼ Details

The area below shows the last 4 KB of the execution log.

```
{
  "ResponseMetadata": {
    "RequestId": "f9b2bb2d-1156-4b15-b453-02d53e1a2413"
  },
  "MessageId": "01090183fb0a7534-3669d429-f643-4f9a-9ef8-1010b4930689-000000"
}
```

Summary

Code SHA-256
r3PGUxYfgI4YWfmI+uA6W5gn4cHlo+KOctxE1hzOpmE=

Request ID
a9f9274b-19ed-4876-abaa-1fb177fc8785

Duration
540.20 ms

Billed duration
541 ms

Resources configured
128 MB

Max memory used
83 MB

Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: a9f9274b-19ed-4876-abaa-1fb177fc8785 Version: $LATEST
END RequestId: a9f9274b-19ed-4876-abaa-1fb177fc8785
REPORT RequestId: a9f9274b-19ed-4876-abaa-1fb177fc8785  Duration: 540.20 ms  Billed Duration: 541 ms    Memory Size: 128 MB  Max Memory Used: 83 MB
```

8. For scheduled daily report, go to AWS Cloudwatch , navigate to rule section (now called as eventBridge).

9. Create rule- give name, ruletype- schedule, use cron expression for
   schedule pattern .
   For e.g. :   15 19 * * ? *
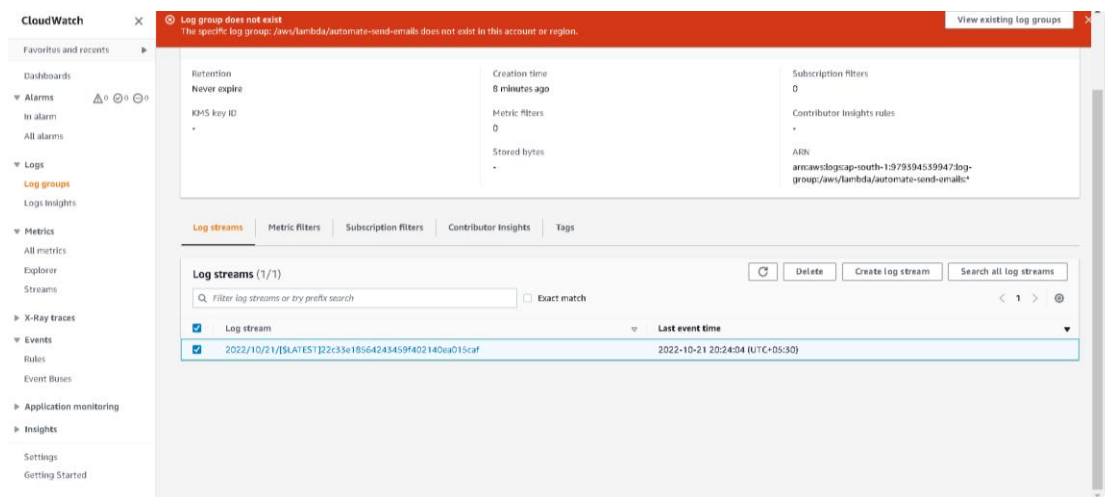
10. Select Targets as lambda function, and use the above defined function.
11. Go to monitoring in Lambda service, click on View logs in cloudWatch and check your mail inbox .
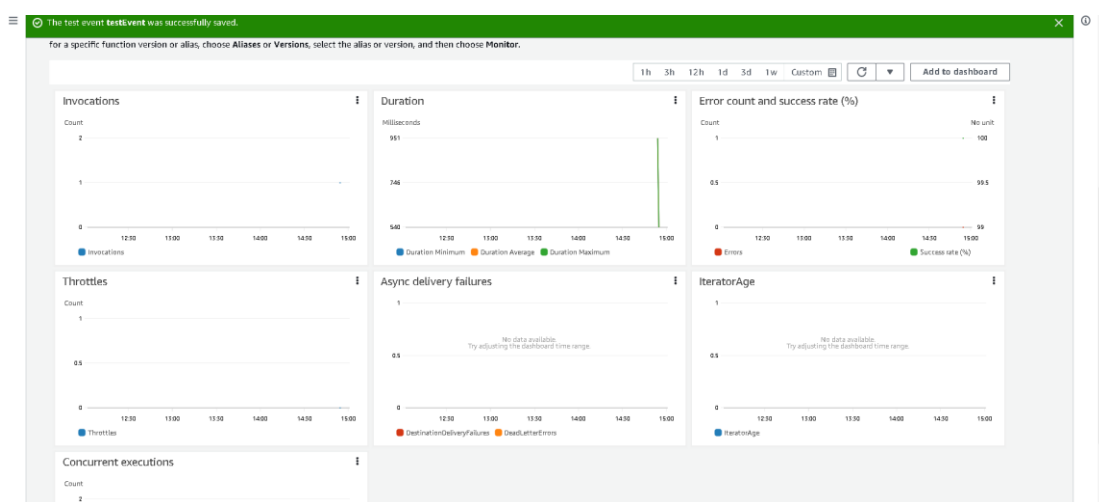
Result:

Hence, the lambda function is created and also implemented using SES, CloudWatch to schedule daily reports.

Result:

Hence, the lambda function is created and also implemented using SES, CloudWatch to schedule daily reports.