

Experiment: 7

Title : Creating a lambda function in AWS to email daily reports

Name: Vaibhav Jain

Reg. No.: RA2011028010103

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

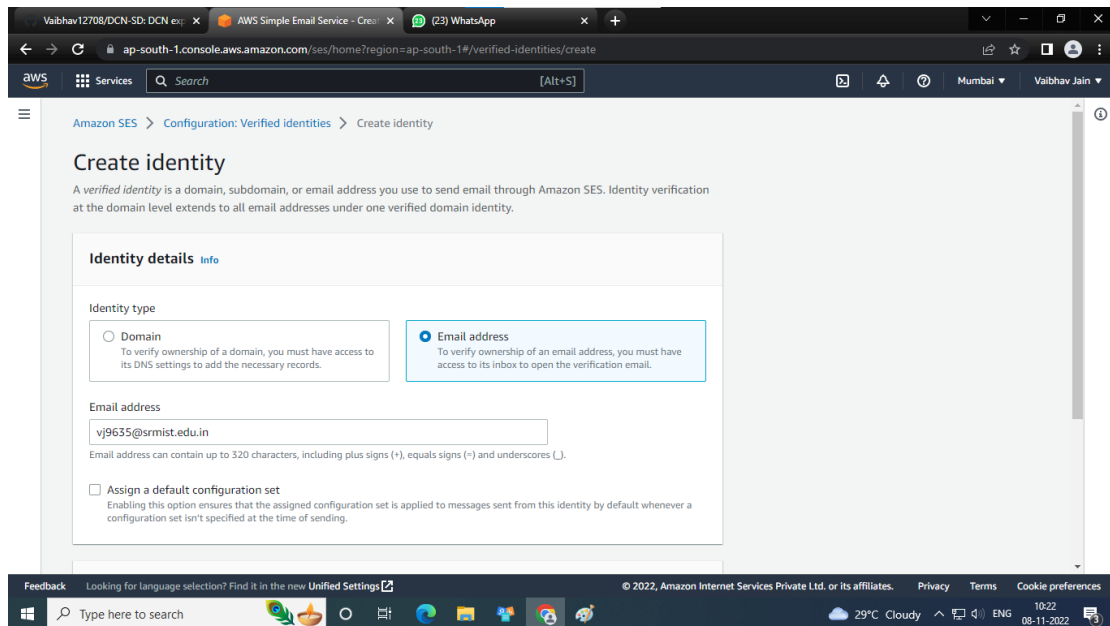
Pre-requisites : AWS Console, Amazon SES, Amazon Lambda, Amazon CloudWatch.

Procedure :

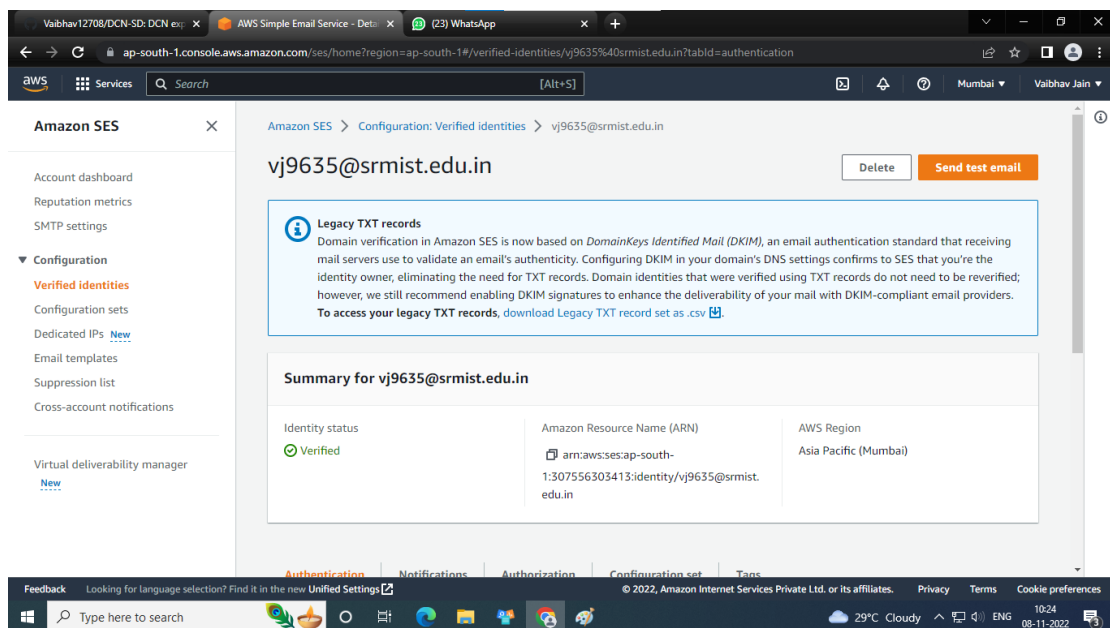
We are going to automate sending email to a person or a group of people. AWS **Cloudwatch** is used to setup a schedule to trigger AWS **Lambda** function and then it's going to use AWS **SES (Simple Email Service)** to send out emails to people.

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.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture info
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

Permissions info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).
☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

[View the AutomateSendEmailRole role on the IAM console.](#)

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: "hello! This is a test message!! FROM -from AWS" },
12      },
13      Subject: { Data: "Test Email" },
14    },
15    Source: "arn:aws:ses:us-west-2:123456789012:email.com",
16  };
17  return ses.sendEmail(params).promise();
18 }
19
20
21
22
23
24
25

```

```

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" },

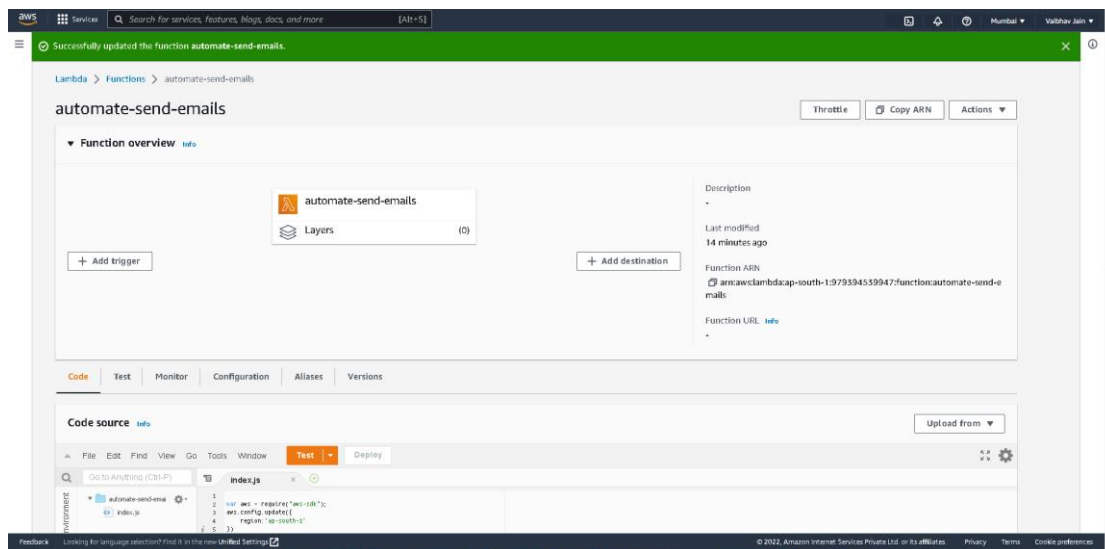
```

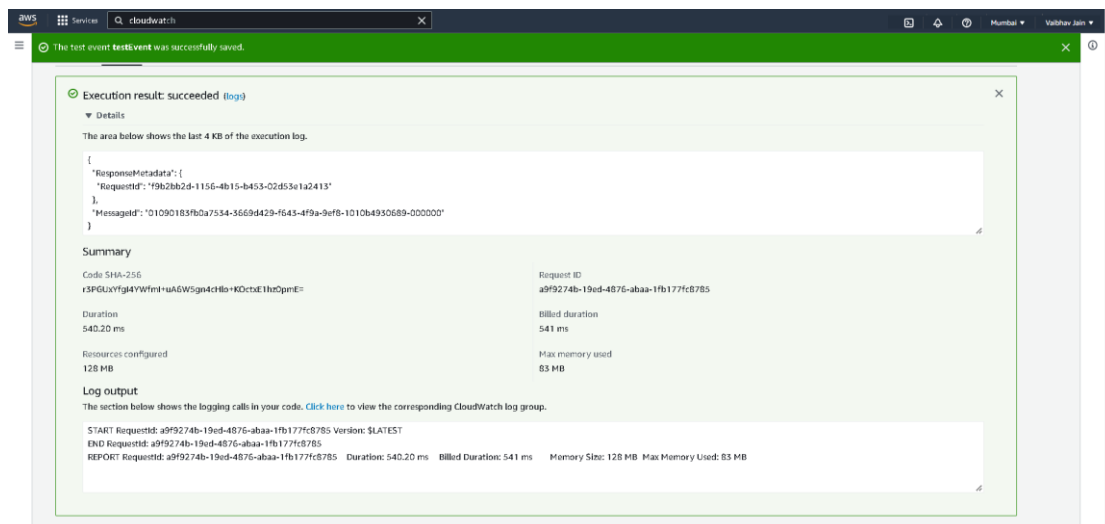
```

12)     },
13)
14)     Subject: { Data: "Test Email" },
15)     },
16)     Source: "SourceEmailAddress",
17) };
18)
19) return ses.sendEmail(params).promise()
20) };

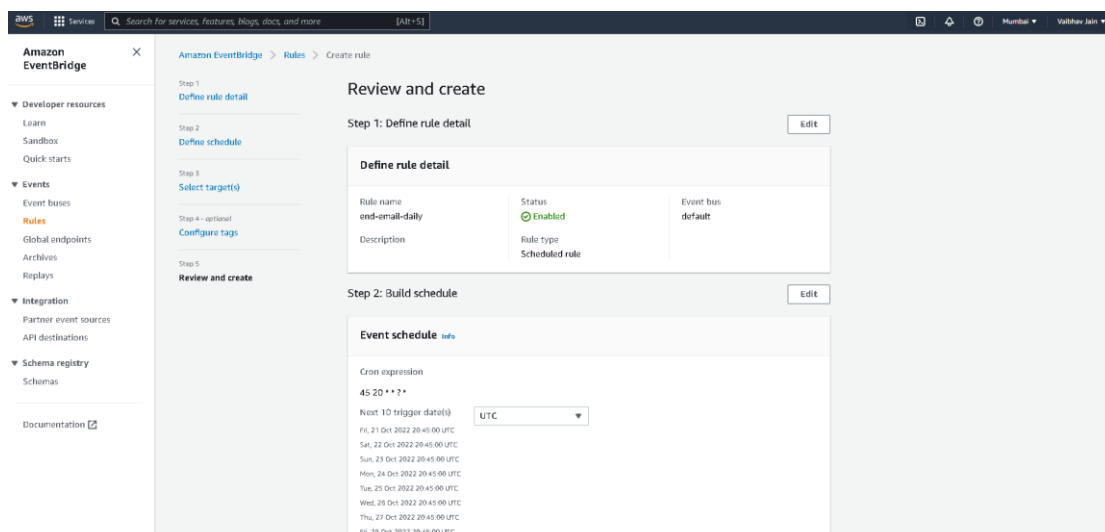
```

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





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



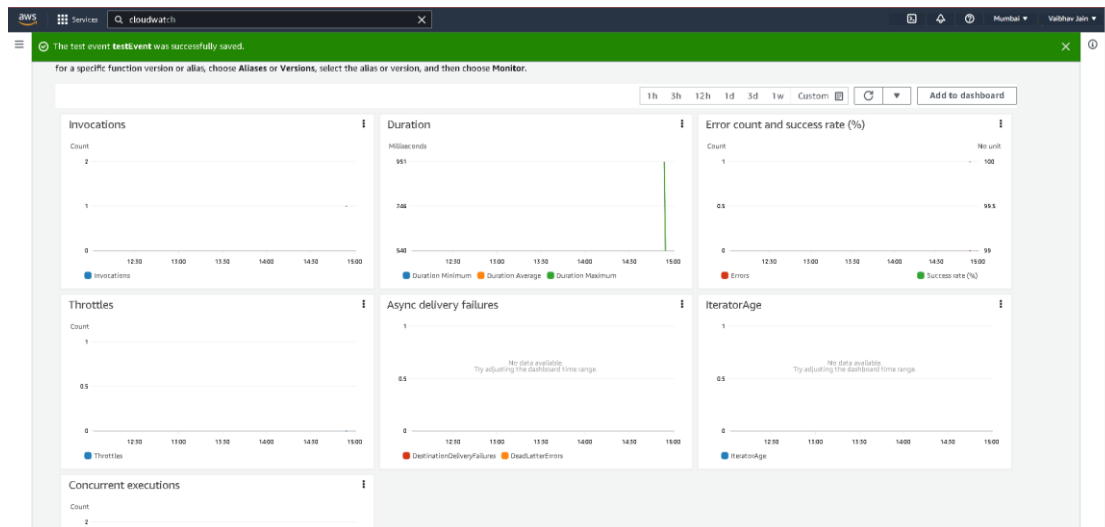
9. Create rule- give name, ruletype- schedule, use cron expression for schedule pattern .

For e.g. : 15 19 * * ? *

The screenshot shows the 'Define schedule' step in the Amazon EventBridge console. The left sidebar contains navigation links for Developer resources, Events, Integration, and Schema registry. The main content area is titled 'Define schedule' and includes a progress bar with five steps: Step 1 (Define rule detail), Step 2 (Define schedule), Step 3 (Select target(s)), Step 4 (optional: Configure tags), and Step 5 (Review and create). The 'Define schedule' section has two radio buttons: '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 menu set to 'cron' and a text input field containing '15 19 * * ? *'. The input field has sub-labels for minutes, hours, day of month, month, day of week, and year. At the bottom, there is a 'Next 10 trigger date(s)' section and a 'Scheduled dates will be generated here upon matching a valid cron expression' note. Navigation buttons at the bottom right include 'Cancel', 'Previous', and 'Next'.

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 .



aws Services Search for services, features, logs, docs, and more [Alt+S]

Mumbai Valbhav Jain

CloudWatch

Log group does not exist
The specific log group: /aws/lambda/automate-send-emails does not exist in this account or region. [View existing log groups](#)

Retention: Never expire
Creation time: 8 minutes ago
Metric filters: 0
Stored bytes: -

Subscription filters: 0
Contributor Insights rules: -
ARN: arn:aws:logs:ap-south-1:979594539947:log-group:/aws/lambda/automate-send-emails

Log streams (1/1)

Filter log streams or try prefix search

Log stream

2022/10/21:[SLATEST]22c53e18564243459f402140ea015caf

Last event time

2022-10-21 20:24:04 (UTC+05:30)

Result:

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