

Experiment 7

Creating a lambda function in AWS to email daily reports

Name : JAYESH S CHAUDHARI

Reg No : RA2011028010094

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

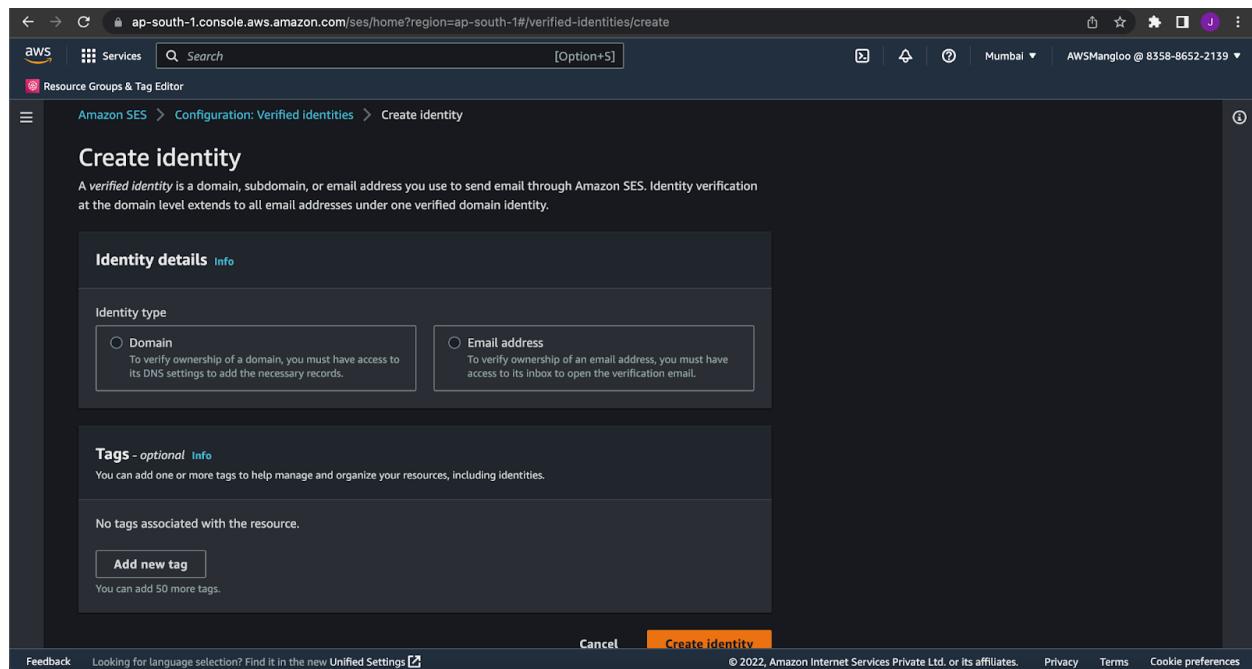
Pre-requisites : AWS Console, Amazon SES, Amazon Lambda, Amazon CloudWatch.

Procedure :

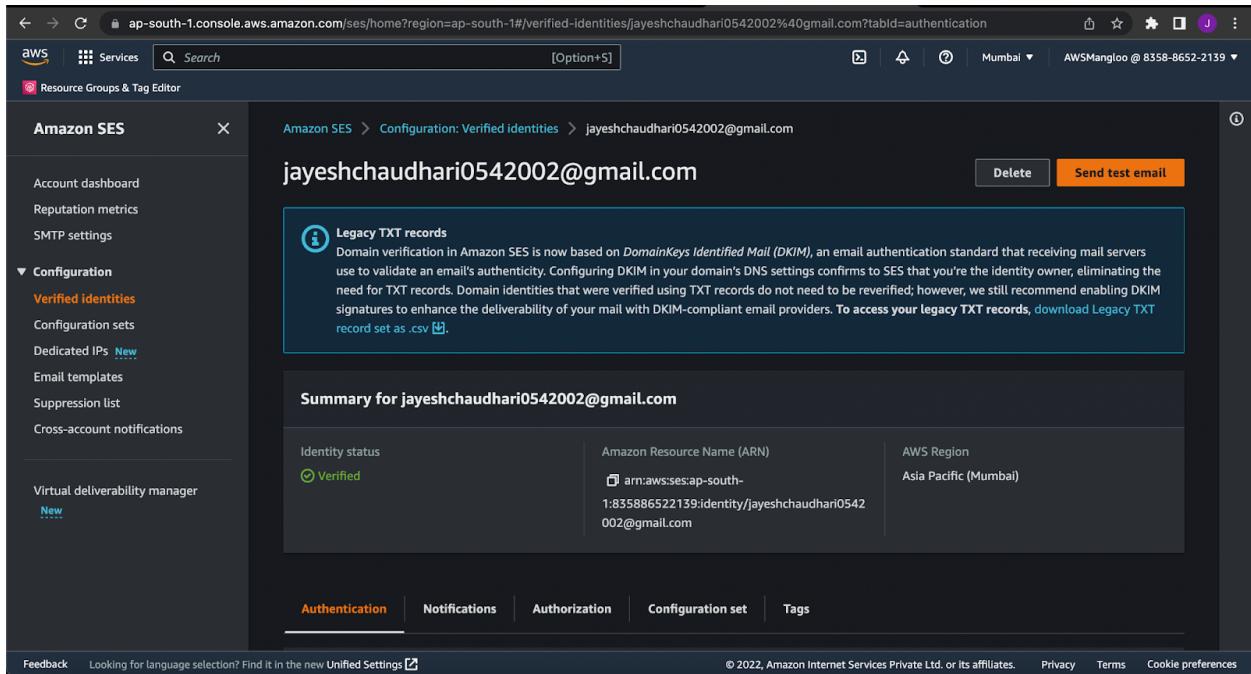
We are going automate sending email to a person or a group of people. AWS **Cloudwatch** is used to setup a schedule to trigger AWS **Lambda** function and then its going to use AWS **SES (Simple Email Service)** to send out emails to people.

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 received an email from aws to tell you to verify that.



The screenshot shows the Amazon SES console with the URL ap-south-1.console.aws.amazon.com/ses/home?region=ap-south-1#/verified-identities/jayeshchaudhari0542002%40gmail.com?tabId=authentication. The left sidebar shows the 'Verified identities' section under 'Configuration'. The main page displays the identity `jayeshchaudhari0542002@gmail.com` with a status of 'Verified'. A summary table provides details like ARN and AWS Region. Navigation tabs include Authentication, Notifications, Authorization, Configuration set, and Tags.

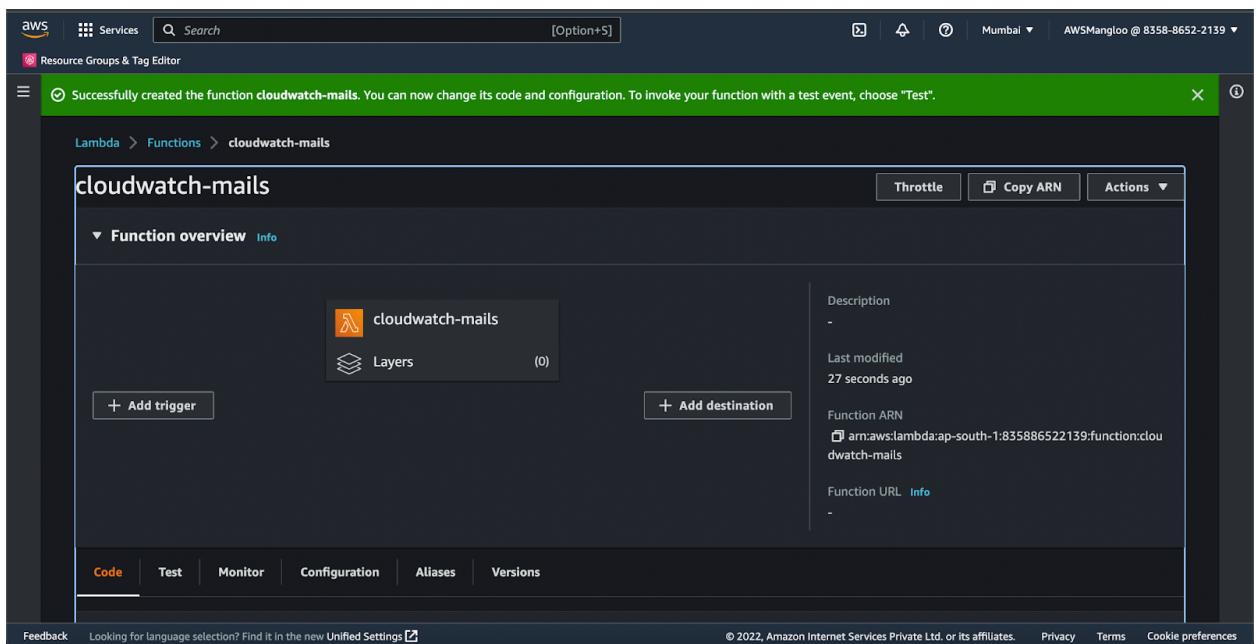
3. Create two identities (email address).

One for sending emails and another for receiving.

4. Create an IAM role.

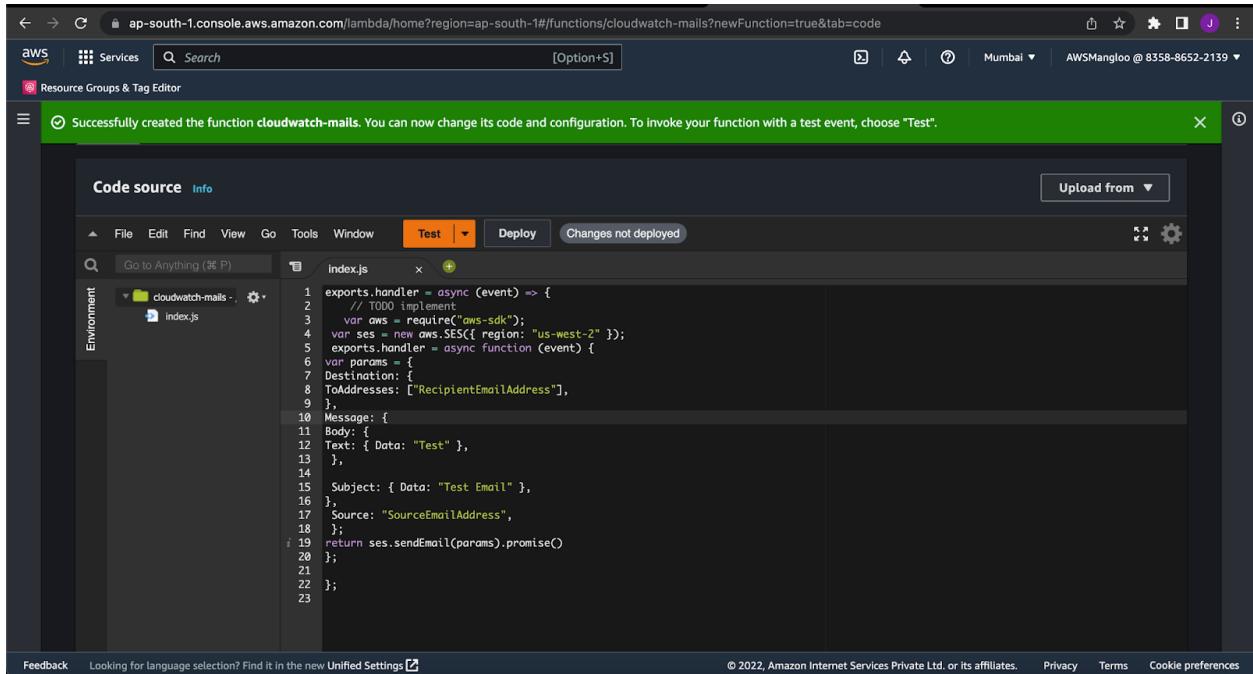
Give Use case as lambda and give full access to cloudwatch, SES. 5. Go to Lambda Service, create a lambda function.

Give name, runtime as NodeJS, execution role as created IAM role previously.



The screenshot shows the AWS Lambda console with the URL <https://ap-south-1.console.aws.amazon.com/lambda/functions/cloudwatch-mails?tab=Code>. A green success message states 'Successfully created the function cloudwatch-mails. You can now change its code and configuration. To invoke your function with a test event, choose "Test".' The function overview shows the name 'cloudwatch-mails', runtime 'Node.js', and a description. It also lists 'Layers' (0), 'Throttle', 'Copy ARN', and 'Actions'. The 'Code' tab is selected, showing the function's code editor. Other tabs include Test, Monitor, Configuration, Aliases, and Versions.

Use this template for the code:



```
1 exports.handler = async (event) => {
2     // TODO implement
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         params.Source = "SourceEmailAddress";
17         return ses.sendEmail(params).promise()
18     };
19 };
20 };
21 };
22 };
23 }
```

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

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 * * ? *

Define rule detail

Rule detail

Name: rule7

Description - optional: Enter description

Event bus: default

Rule type: Schedule

Schedule pattern

A schedule that runs at a regular rate, such as every 10 minutes.

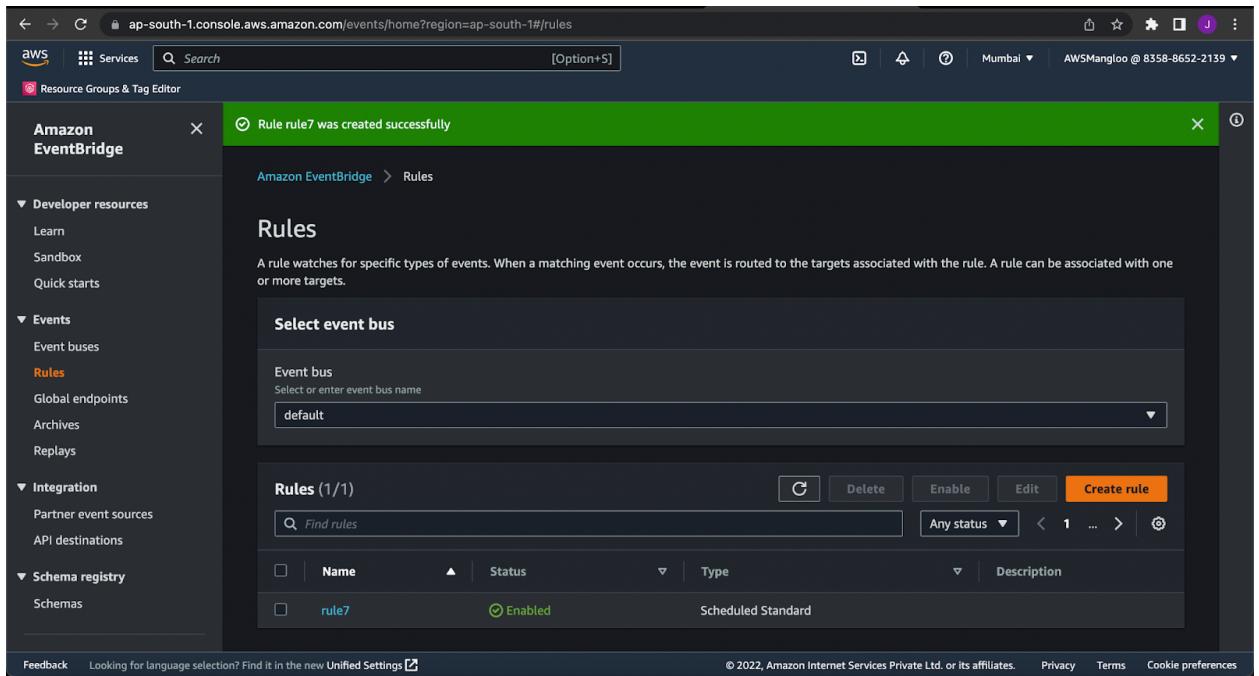
Cron expression: cron(15 9 * * ? *)

Local time zone: Nov 14, 2022, 02:45 PM GMT+5:30

Next 10 trigger date(s): Nov 14, 2022, 02:45 PM GMT+5:30; Nov 15, 2022, 02:45 PM GMT+5:30; Nov 16, 2022, 02:45 PM GMT+5:30; Nov 17, 2022, 02:45 PM GMT+5:30; Nov 18, 2022, 02:45 PM GMT+5:30; Nov 19, 2022, 02:45 PM GMT+5:30; Nov 20, 2022, 02:45 PM GMT+5:30; Nov 21, 2022, 02:45 PM GMT+5:30; Nov 22, 2022, 02:45 PM GMT+5:30; Nov 23, 2022, 02:45 PM GMT+5:30

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 .



The screenshot shows the AWS Lambda service interface. On the left, there's a sidebar with 'Amazon EventBridge' selected. The main area displays a success message: 'Rule rule7 was created successfully'. Below this, it says 'Amazon EventBridge > Rules'. A sub-section titled 'Select event bus' shows 'Event bus' set to 'default'. The 'Rules' section shows one rule named 'rule7' with status 'Enabled' and type 'Scheduled Standard'. The bottom of the screen includes standard AWS navigation links like Feedback, Unified Settings, and Copyright information.

Result:

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