Setup for Email Notification for S3 Bucket Operations

Version History

#	Date	Version	Author	Reviewer
1	17/5/2023	1.0.0	P. UMAPOORNIMA	

Overview

The PoC involves setting up a system that sends email notifications whenever specific operations occur within an S3 bucket like add/delete/modifying the objects in your s3 bucket. This is achieved by using AWS Lambda to process S3 events, AWS SES to send the email notifications, and AWS SNS to manage the communication between Lambda and SES.

Services Used

S3
IAM (policy, role)
SES
Lamda
SNS

Procedure

This POC I was tried in three ways

1. S3==Lamda==SNS

2. S3==Lamda==SES
3. S3===SNS

S3===Lamda==SNS

- To configure notifications with **S3, Lambda, and SNS**, follow these steps:
- Open the AWS Management Console and navigate to the S3 service and **create one S3 bucket**, bucket name should be unique (ponna126) or select an existing bucket that you want to configure notifications

- Go to SNS service **Create an SNS topic** to which you will publish notifications & create a topic, select type as a **standard** & give the topic a meaningful name **(snstopic1)** that reflects its purpose.
- After that, **setup SNS Subscription** under SNS Service, click on create subscription and choose the protocol as **"Email"** and enter the email address or choose which protocol you want like mobile, SMS where you want to receive the notifications.
- Confirm the subscription by following the instructions sent to the provided email address & go to your mail id and **confirm the subscription**

- Configure **S3 Event Notifications** and navigate to the S3 service. Select the bucket you created (ponna126) and go to the **"Properties"** tab and scroll down to the **"Events"** section. Click on **"Create event notification"**
- Choose the event(s) that should trigger notifications, such as object creation, deletion, or modification. Select **"SNS topic" as the destination**.
- Choose the SNS topic you created **(snstopic1)** under **specify SNS topic** as the target for the notifications and click on Save the changes
- If you face any errors like "Unknown Error An unexpected error occurred. API response Unable to validate the following destination configurations", Select your **Topic** which you created and **click on Edit** and **Click on Access Policy** remove the previous json policy code and add this code

```
{
 "Version": "2012-10-17",
 "Id": "example-ID",
 "Statement": [
  {
   "Sid": "ExampleSnsTopicPolicy",
   "Effect": "Allow",
   "Principal": {
    "Service": "s3.amazonaws.com"
   },
   "Action": "SNS:Publish",
   "Resource": "arn:aws:sns:us-west-2:800923761879:topic1",
   "Condition": {
    "StringEquals": {
     "aws:SourceAccount": "800923761879"
```

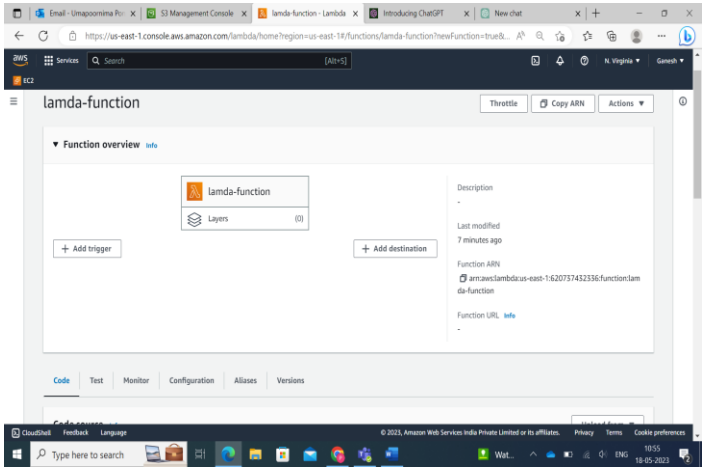
```
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:ponna126"
        }
       }
      }
     ]
    }
```

- Now, **navigate to the Lambda Service** and Click on "Create function"
- Select a suitable runtime for your Lambda function, such as Node.js, Python, or Java and choose the **"Author from scratch"** option. Provide a name and description for the function.
- Set the appropriate permissions, execution role, and resource requirements for the function. Or select **Create a new role with basic Lambda permissions** it will create the role by default with basic lamda permissions under your IAM.
- Click on the Create Lambda function. After that configure lamda function go to that created lamda function and In the Lambda function configuration, click on **"Add trigger".**
- Select **"S3"** as the trigger type. Choose the S3 bucket that you configured.
- Specify the event type(s) that should trigger the Lambda function, such as object created or modified.
- Add the **destination as amazon SNS**
- In the Lambda function code, after processing the S3 event, use the appropriate API to publish a message to the SNS topic.
- The message can contain information about the S3 event, such as the bucket name, object key, operation type, etc.
- The SNS service will automatically deliver the notifications to the subscribed email address.
- Go to your s3 bucket and upload files in your bucket & check your mail-id if you will receive notification.

S3==Lamda==SES

- First, we have to create one s3 bucket
- Go to IAM, **create policy** for your bucket, click on create policy and take policy in **json** format and copy the policy code from the below link & make sure your region will be on **"us-east-1"** and click on create policy, policy is created (lamda-policy)
  https://github.com/gitmurali/aws_snippets/blob/main/lambda/s3_event_lambda_trigger/security_policy.json
- Now you have to **create a role** for that policy, go to role click on create role & select **aws service**, select **use case was lamda**, select lamda under Use cases for other AWS services and click on next, now you will able to see your policy (lamda-policy) select that policy & click on next now you can give role name (lamda-role) & description and click on create role
- Role and policy were created under your IAM check in your Aws console

- Go to AWS Management Console and navigate to the Amazon Simple Email Service and **create Identity** there, configure your email address and confirm the confirmation in your mail & verify your identity status was in **verified state**.
- Create lamda function, take **author from the scratch**, give the function name (lamda-function) select run time as a **python** & select execution role was **use an existing role** and select your existed created role (lamda-role) and click on create function
- Click on that created lamda-function and click on add trigger select **source as a s3** & select your bucket, event type and click on add



- Click on **code,** delete the existing code and **add lamda-trigger code** from below link and change your email address in the source & destination block and click on **deploy**
  https://github.com/gitmurali/aws_snippets/blob/main/lambda/s3_event_lambda_trigger/notify.py
- Navigate to the S3 service. Select the bucket you created, go to the **"Properties"** tab and scroll down to the **"Events"** section.
- Click on **"Create event notification"**. Choose the event(s) that should trigger notifications, such as object creation, deletion, or modification.
- Select **"Lamda function"** as the **destination** and choose your created lamda-function under lamda function and save the changes
- Go to your S3 bucket and upload your files in your bucket and you will receive the notification to the target email id

S3===SNS

- Create one S3 bucket (ponna126)
- Go to the SNS Service, **create SNS topic** select type as a **standard** and create a topic, go to that topic click on **access policy** remove the existed policy and paste new SNS policy from below link, in that policy add your sns topic arn, bucket name & account id and save the changes.
  Granting permissions to publish event notification messages to a destination - Amazon Simple Storage Service (amazonaws.cn)
- **Create subscription** under SNS and configure your email id and confirm the subscription

- Go to your created s3 bucket (ponna126) and go to the **"Properties"** tab and scroll down to the **"Events"** section.
- Click on **"Create event notification"**. Choose the event(s) that should trigger notifications, such as object creation, deletion, or modification.
- Select **SNS topic as a destination** and specify your SNS topic which you are created under SNS topic and save the changes.
- Go to your s3 bucket upload or delete files in your bucket and you will receive notification to the target email id

References

https://www.youtube.com/watch?v=qkSxoSSn1Rc
https://github.com/gitmurali/aws_snippets/blob/main/lambda/s3_event_lambda_trigger/notify.py

https://github.com/gitmurali/aws_snippets/blob/main/lambda/s3_event_lambda_trigger/security_policy.json

Errors Occurred

Error: Unknown Error An unexpected error occurred. API response Unable to validate the following destination configurations.

Reason: S3 is not able to communicate to SNS or SQS. Since the permissions/policies were not defined.

Solution: To grant Amazon S3 permissions to publish messages to the SNS topic, attach an Amazon Identity and Access Management (IAM) policy to the destination SNS topic.
“Replace the following values appeared in the BOLD”

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "Example SNS topic policy",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "SNS-topic-ARN",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:bucket-name"
        },
```

```
            "StringEquals": {
               "aws:SourceAccount": "bucket-owner-account-id"
            }
         }
      }
   ]
}
```