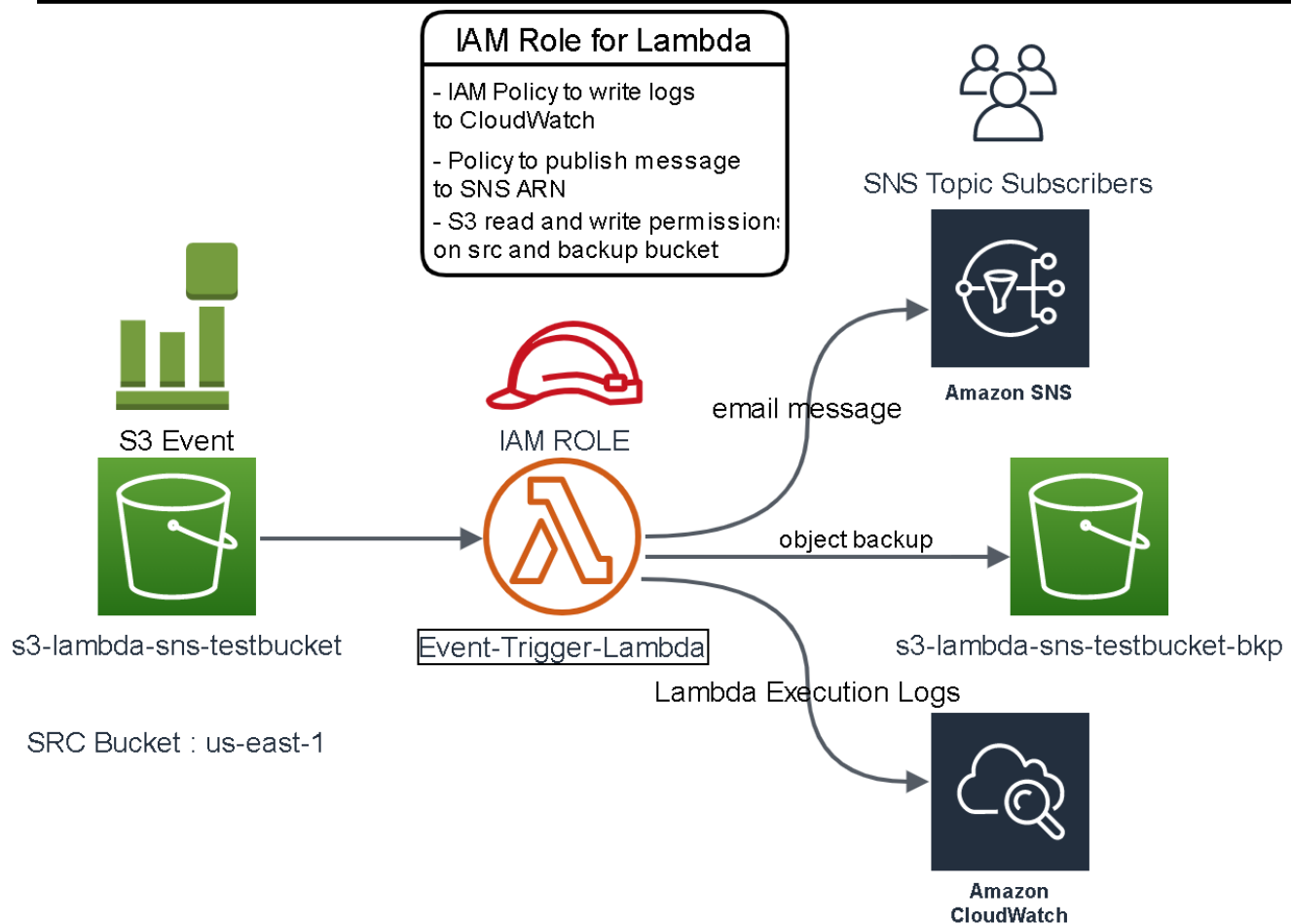


Problem Statement: Create an S3 event Trigger to invoke Lambda Function to trigger SNS email and Copy the Uploaded object into a Backup Bucket. Use Case : To get an email when a file is uploaded in a particular bucket for every file upload with below details : Source Bucket Name , AWS Region Name, File name and file size, Upload IP Address, Event Time

S3 Event Custom Email Notification



Follow Below Steps :

1. Create the Source and Backup Bucket. Navigate to the Source S3 Bucket > Under Properties > Setup a S3 event trigger for PUT Object only on a bucket to trigger the Lambda :
2. Create an SNS topic and add subscribers
3. Write below Lambda Code :
4. Navigate to **Configuration** > choose **Environment variables** > **Edit**. Here, Create two Environment variables that will be used in Python Code

- Under Key enter **BACKUP_BUCKET_NAME** , under Value enter **<BUCKET_NAME_VALUE>**

- Under Key enter **SNS_TOPIC_ARN** , under Value enter **<SNS_TOPIC_ARN_VALUE>**

Make sure to enter Value fields for above Environment Variables as per your AWS Account. Change value of SNS Topic ARN and Backup Bucket Name as per your account.

```
import os,json,boto3

s3 = boto3.client("s3")
sns = boto3.client("sns")

def lambda_handler(event,context):
    print(event)
    source_bucket = event['Records'][0]['s3']['bucket']['name']
    aws_region = event['Records'][0]['awsRegion']
    key_val = event['Records'][0]['s3']['object']['key']
    size_val = event['Records'][0]['s3']['object']['size']/1024
    ipAddress = event['Records'][0]['requestParameters']['sourceIPAddress']
    event_time = event['Records'][0]['eventTime']

    message = "Hi, \n The Event time is : " + event_time + "Hi, \nYou are
receiving this email because you are subscribed to this S3event. \nThe Source
Bucket is: " + source_bucket + "\nThe AWS Region is : " + aws_region + "\nThe
Uploaded Filename is : " + key_val + " having Size : " + str(size_val) + " KB" +
"\nThe Object is upload from IP Address: " + ipAddress
    #Below are the variables for copy_object function parameters
    #Provide below the target bucket name where your object needs to be copied
    backupBucket = os.environ['BACKUP_BUCKET_NAME']
    snsArn = os.environ['SNS_TOPIC_ARN']
    emailSubject = "S3EventTrigger-Notification"
    copy_source = {'Bucket' : source_bucket, 'Key' : key_val}
    sns_response =
sns.publish(TopicArn=snsArn,Message=message,Subject=emailSubject)
    print(sns_response)
    try:
        print("Copying the object from source to destination")
        s3.copy_object(Bucket=backupBucket,Key=key_val, CopySource=copy_source)
    except Exception as e:
        print(e)
        print("Error getting object")
        raise e
```

--

6. Add the below inline policy and add to lambda role, below policy should have only Publish access to specific topic and S3 Permissions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SnsPolicy",
      "Effect": "Allow",
```

```

        "Action": "sns:Publish",
        "Resource": "arn:aws:sns:us-east-1:ACCOUNT_ID:SNSDemoTopic"
    },
    {
        "Sid": "S3Access",
        "Action": "s3:*",
        "Effect": "Allow",
        "Resource": "arn:aws:s3:::*"
    }
]
}

```

--

7. Upload any file to the bucket
8. Verify Lambda Trigger from Cloudwatch Log Group with Lambda Name
[/aws/lambda/<LambdaFunctionName>](#)
9. Verify if the uploaded file is copied to the Backup Bucket.
10. Verify if email notification is trigger from above lambda code.

-
- For details, see Using AWS Lambda with Other Services visit [AWS Lambda with Other Services](#)

Reference:

- [s3.copy_object\(\)](#)
- [sns.publish\(\)](#)

Notes

- Lambda Function Name should be unique across your region and account.

Lambda Function Inside VPC

- Lambda uses your function's permissions to create and manage network interfaces. To connect to a VPC, your function's execution role must have the following permissions:
 - **ec2:CreateNetworkInterface**
 - **ec2:DescribeNetworkInterfaces**
 - **ec2>DeleteNetworkInterface**
- Choose **Configuration > VPC > Under VPC**, choose Edit. Then, do the following:
- For Virtual Private Cloud (VPC), choose your **VPC**.
- For Subnets, select the subnets under the VPC.
- For Security groups, choose a security group
- Once Lambda Function is launched inside the VPC, a ENI will be created inside the subnets selected above, where lambda function will use ENI and all VPC networking routing rules associated.

Scenarios

VPC:

Public Subnet with IGW Route

Public Subnet with IGW Route and VPC Endpoint for S3

Private Subnet with NAT Route and VPC Endpoint for S3

- CloudWatch to store logs ,

Private Subnet with no route:

Private Subnet with not NAT and no IGW

- Interface gateway endpoint for cloudwatch
- VPC Endpoint for S3