Name - Mikil Lalwani D15B/37

Advance DevOps lab

Experiment 12.

Aim -
To create a lambda function which will log "An image has been added" once you add an object to specific S3 bucket.

Theory -

AWS lambda is a serverless, event-driven compuling platform.

Working -

1) Write code in supported language and upload to AWS lambda.

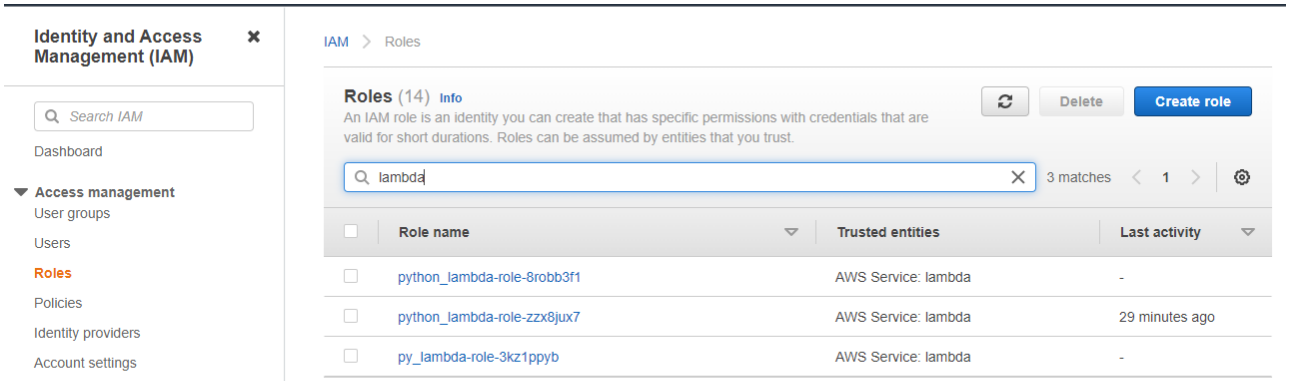
2) Enter event details.

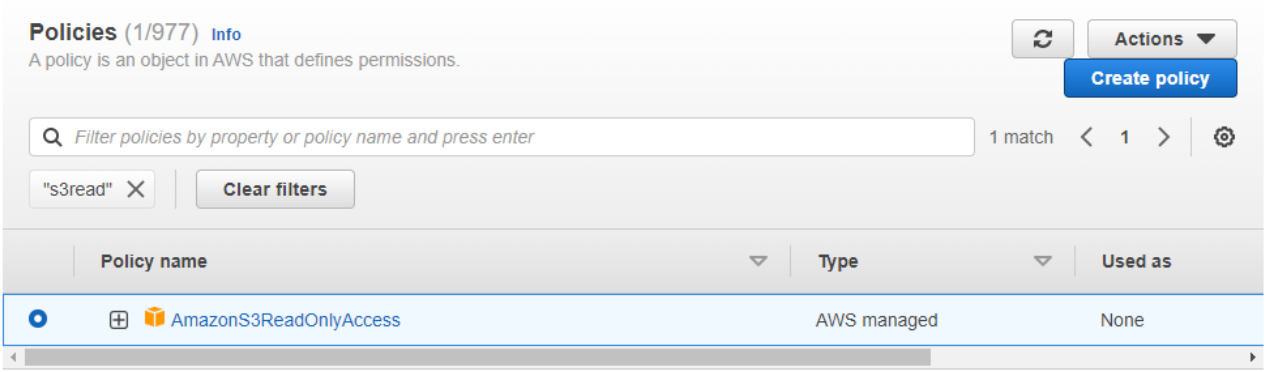
3) Lambda code executes only when certain conditions are met.

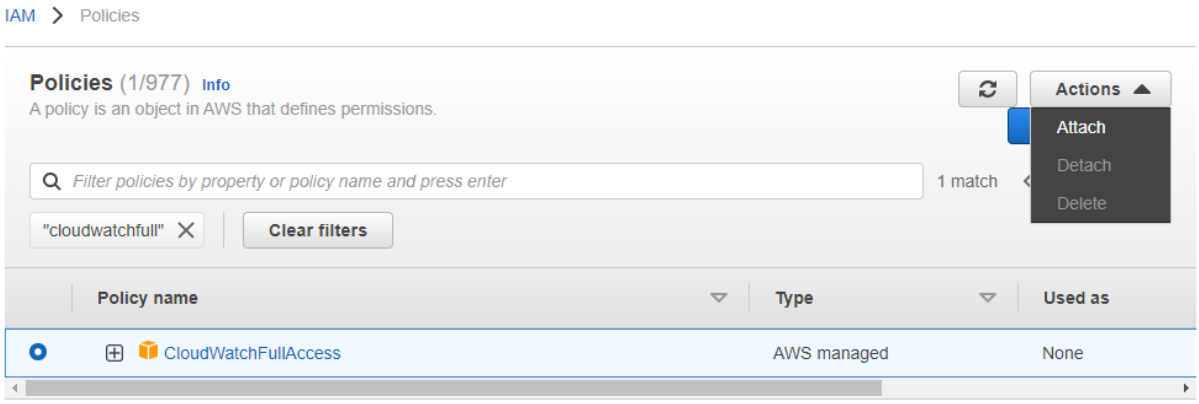
AWS lambda charges users only when executed.
eg Email sending, hosting of website etc.

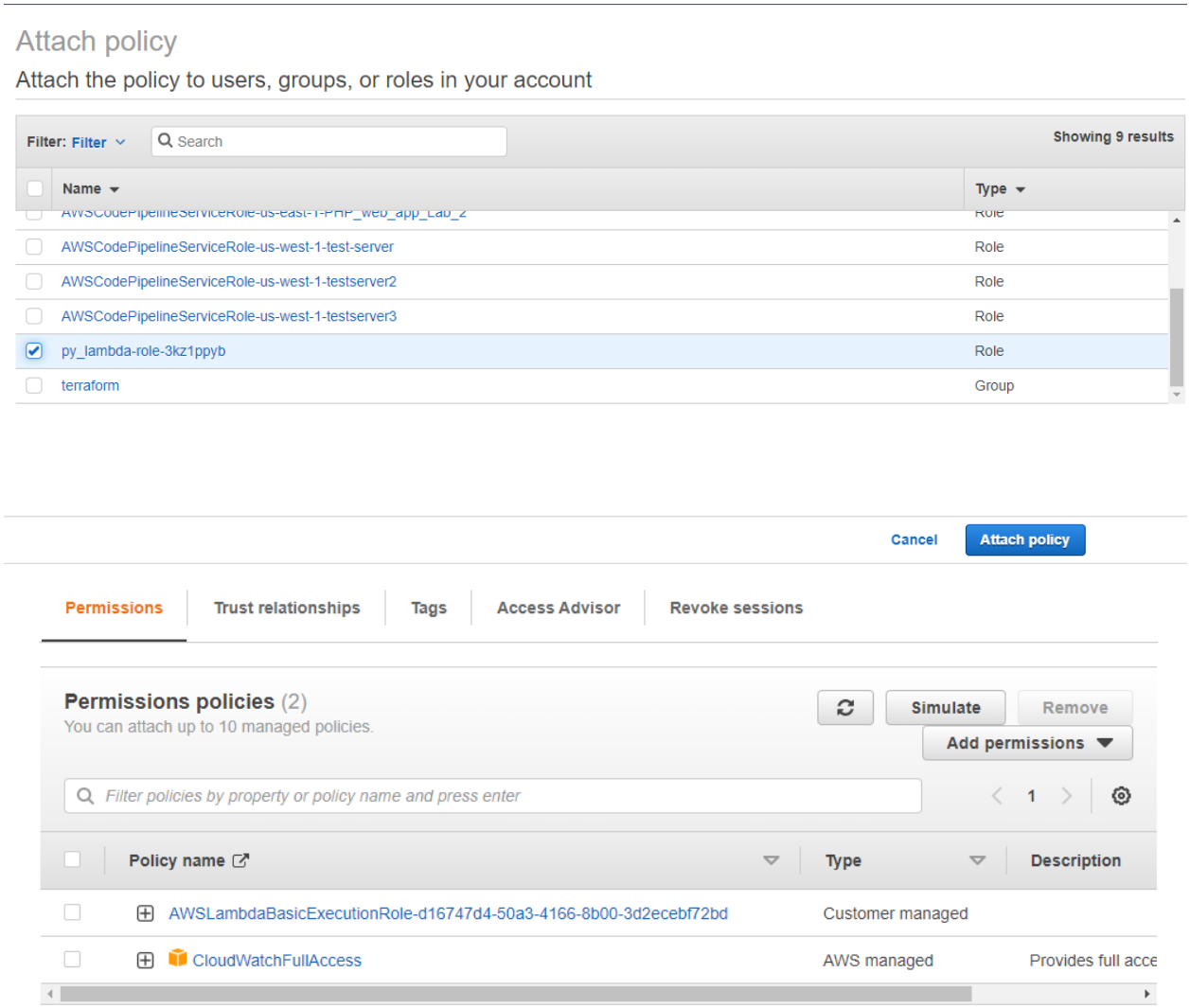
Steps-

1. Open up the IAM Console and under Roles, choose the Role we previously created for the Python Lambda Function.



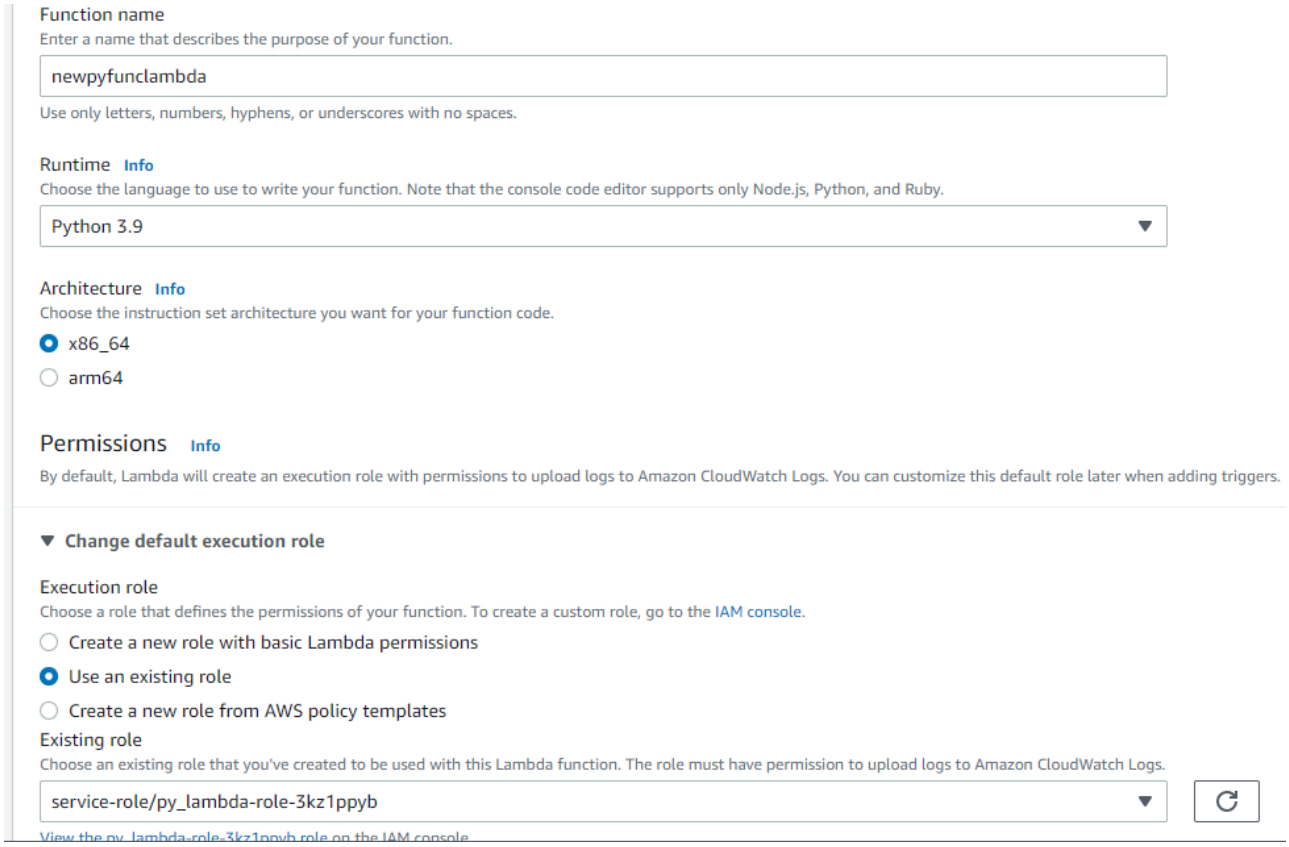






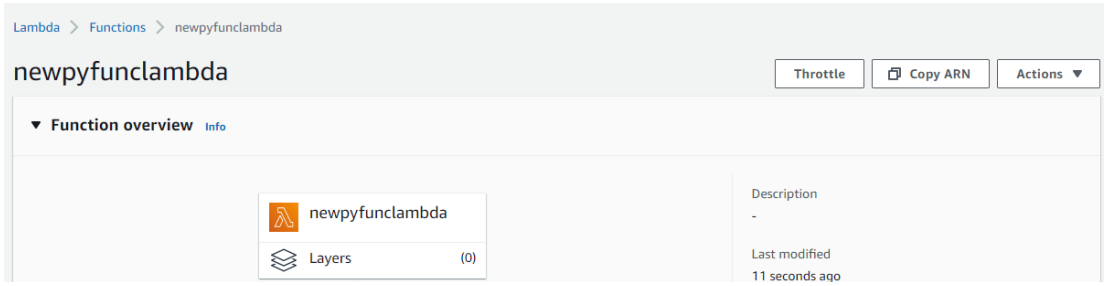
Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.

2. Open up AWS Lambda and create a new Python function

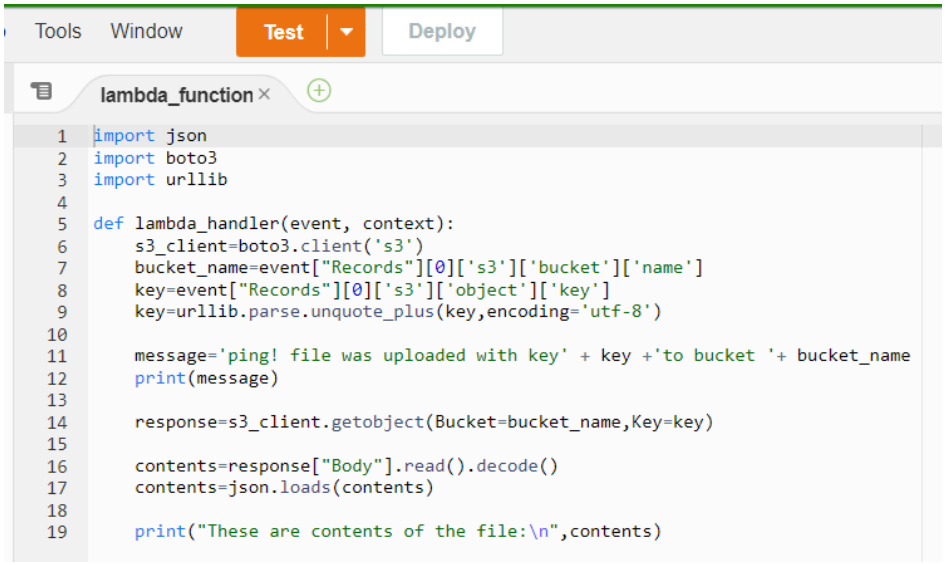


Under Execution Role, choose the existing role, the one which was previously created and to which we just added permissions.

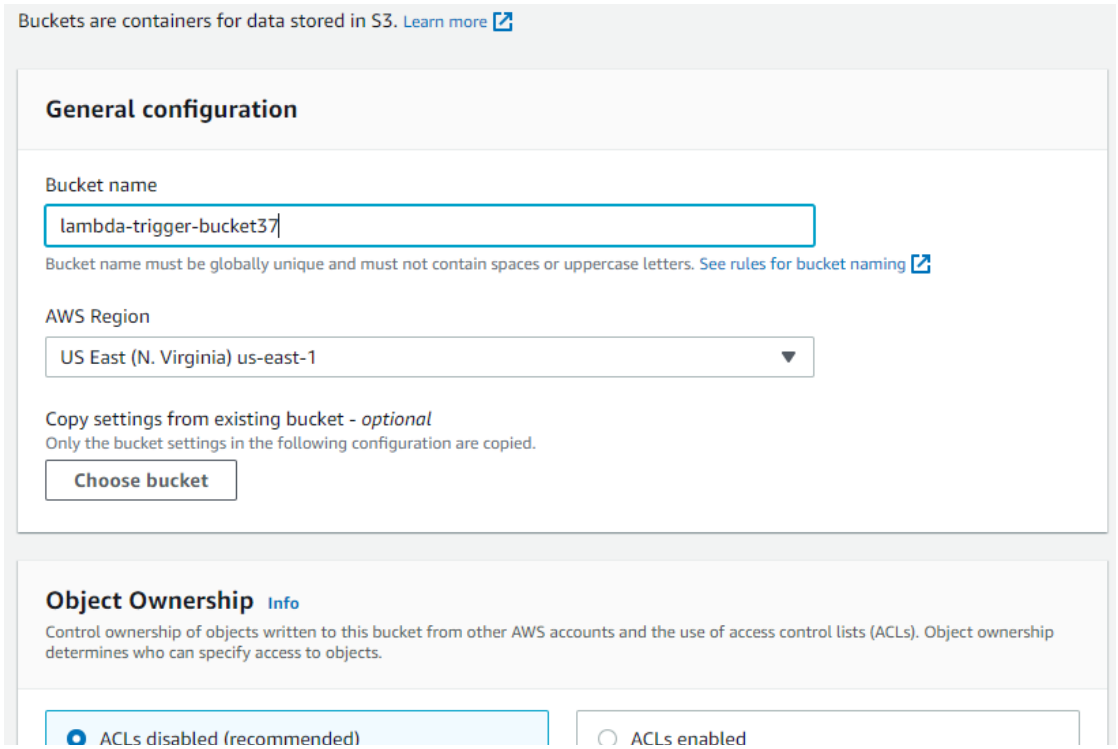
3. The function is up and running.



4. Make the following changes to the function and click on the deploy button. This code basically logs a message and logs the contents of a JSON file which is uploaded to an S3 Bucket.

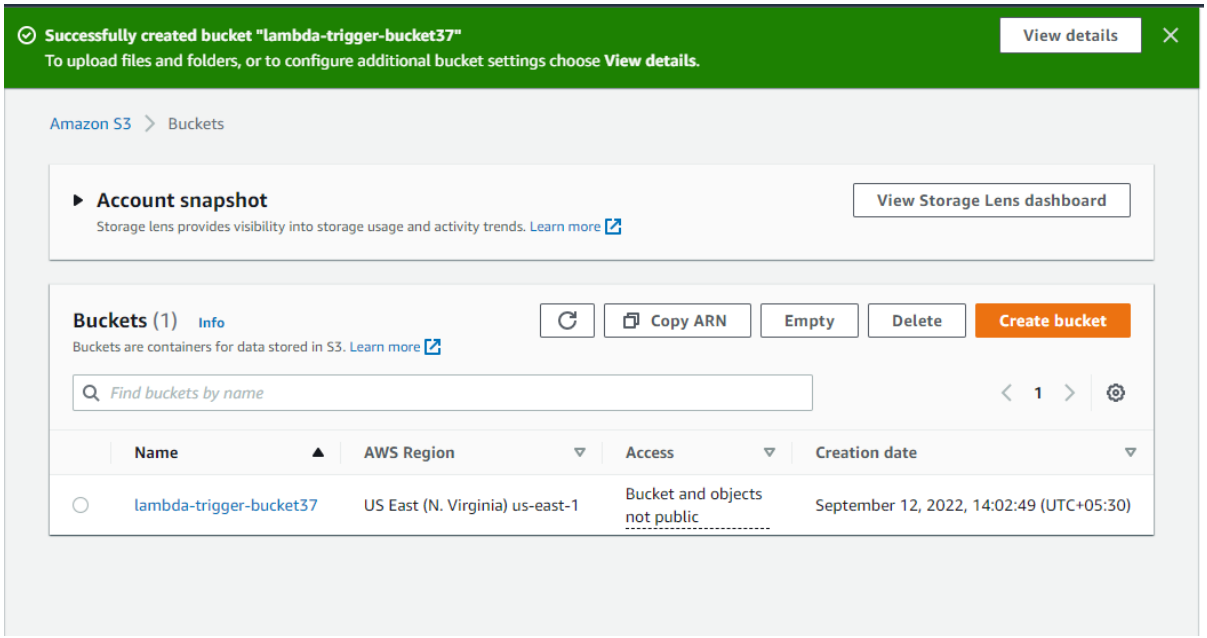


5. Open up the S3 Console and create a new bucket

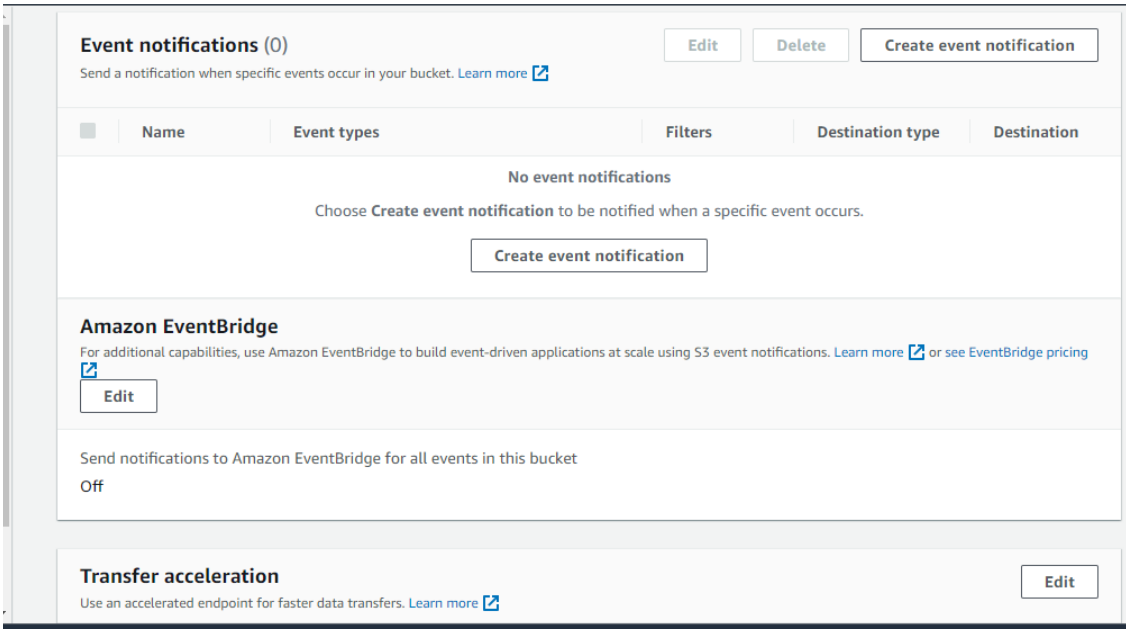


6. With all general settings, create the bucket in the same region as the function.

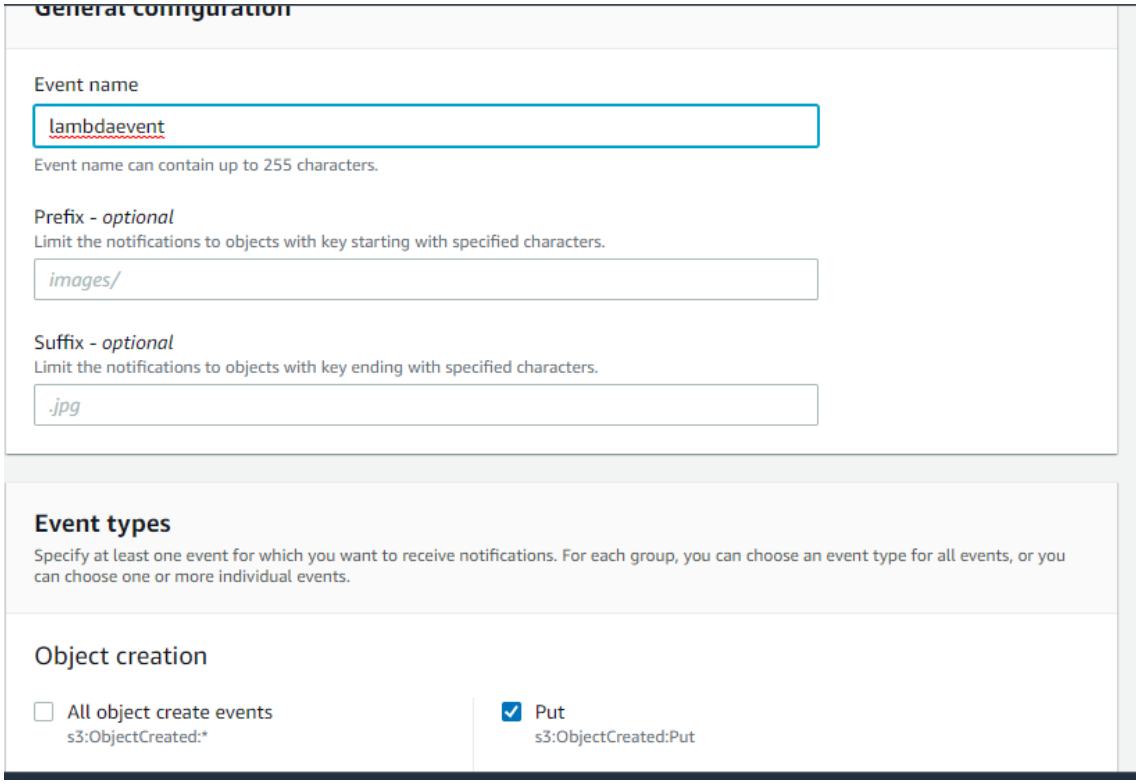
7. Click on the created bucket and under properties, look for events.

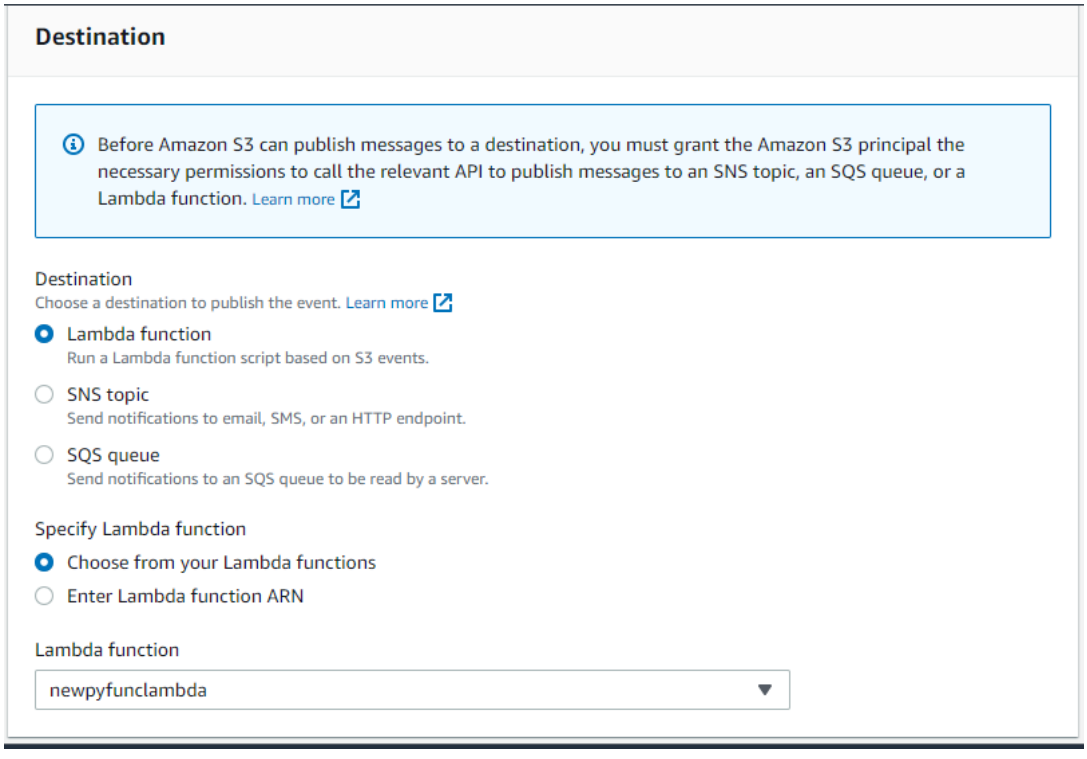


Click on Create Event Notification.



8. Mention an event name and check Put under event types.

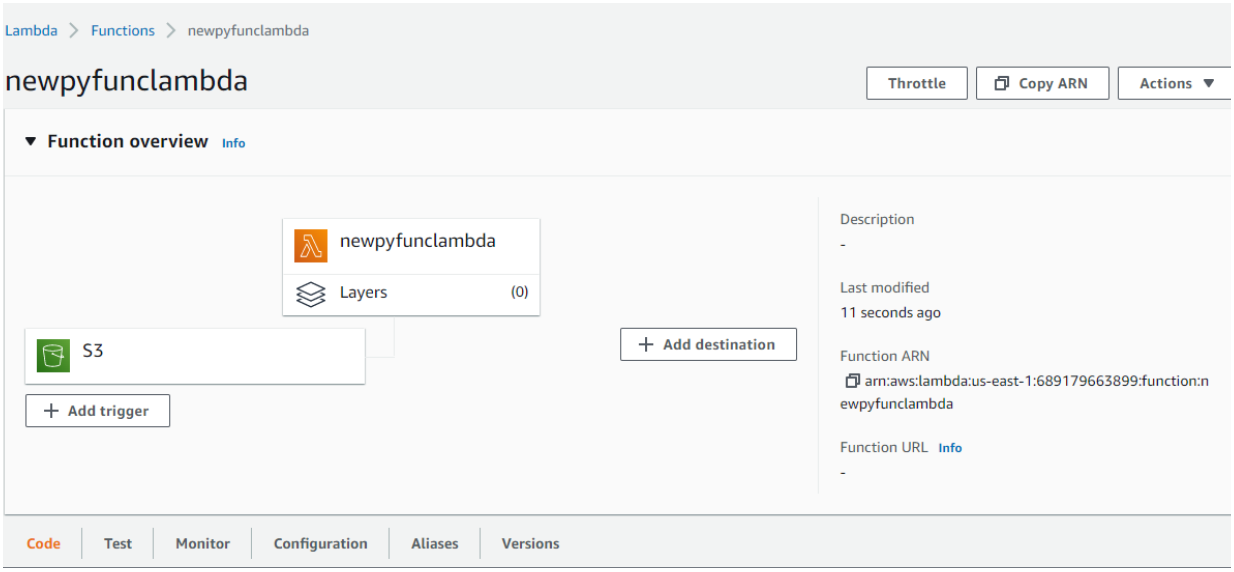




You can optionally choose .json under the suffix since the code only accepts JSON.

Choose the Lambda function as the destination and choose your lambda function and save the changes.

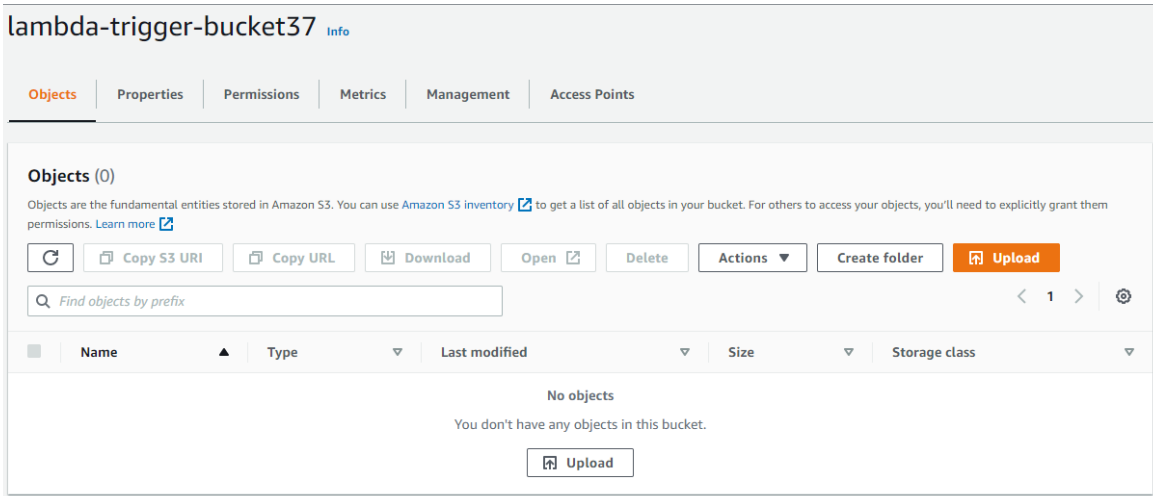
9. Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.



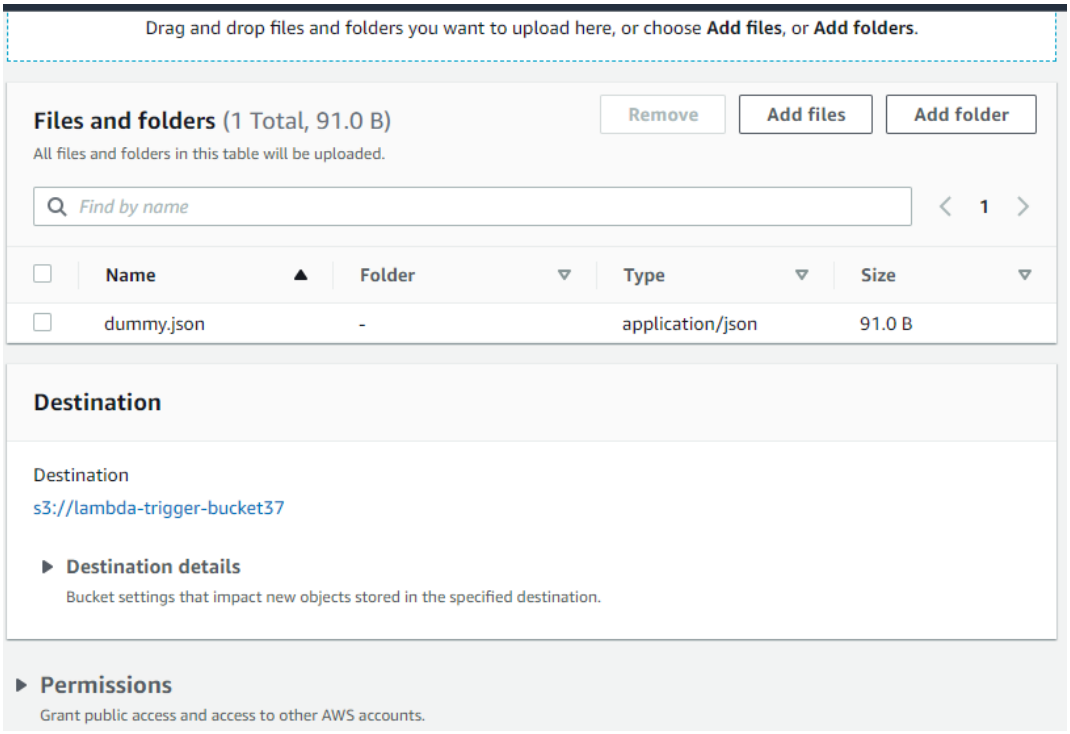
10. Now, create a dummy JSON file locally.
    Dummy.json

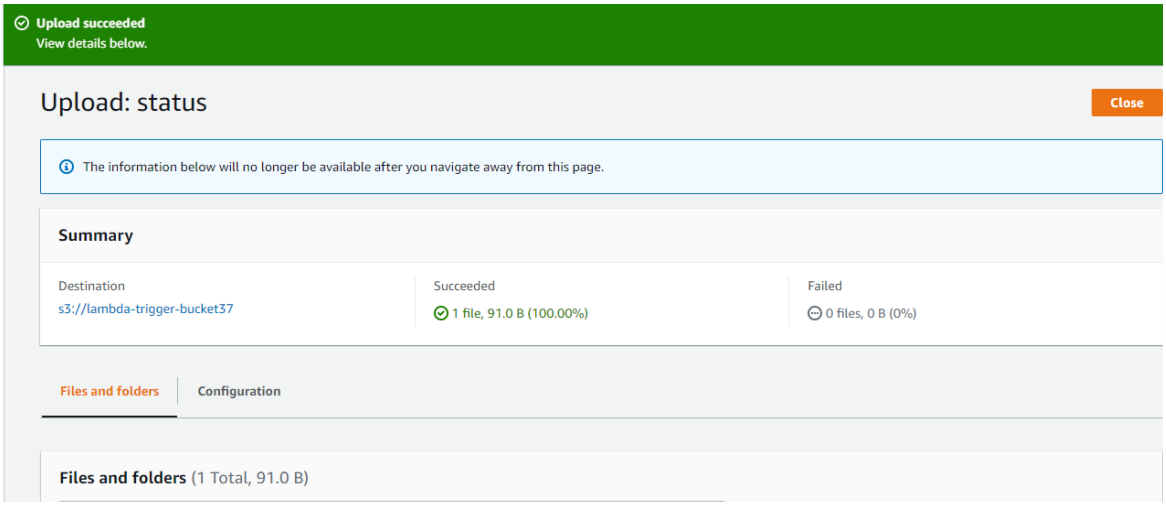
```
{
        "id":49,
        "name":"Kajal Jewani",
        "Designation":"Assistant Professor",
        "Publications":40
}
```

11. Go back to your S3 Bucket and click on Add Files to upload a new file.

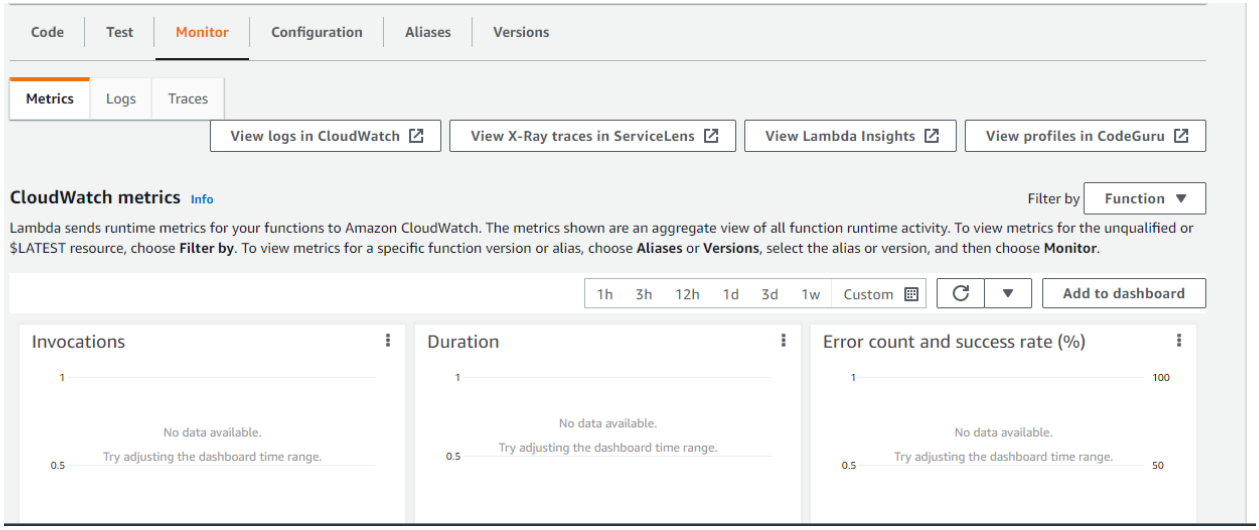


12. Select the dummy data file from your computer and click Upload.

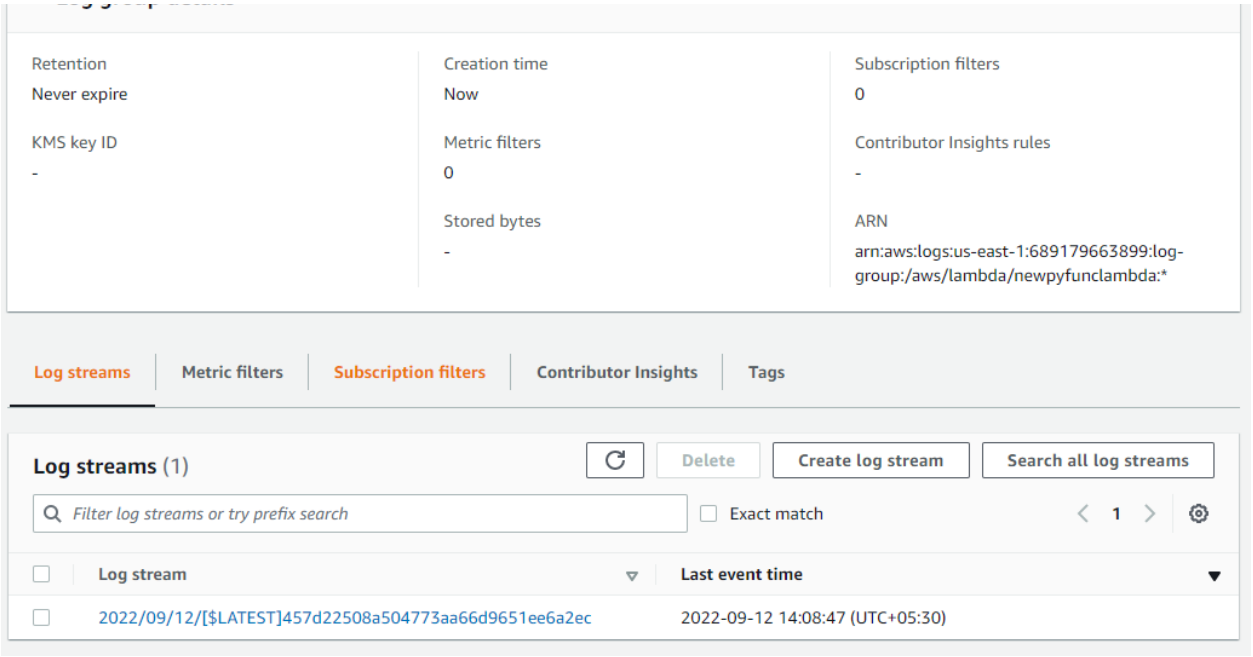




13. Go back to your Lambda function and check the Monitor tab.



Under Metrics, click on View logs in Cloudwatch to check the Function logs



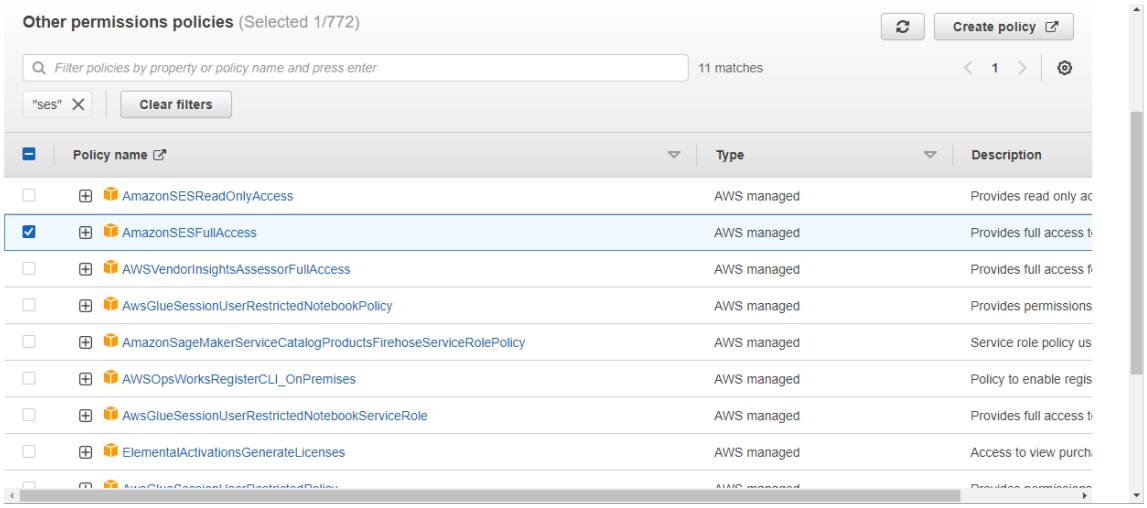
14. Click on this log Stream that was created to view what was logged by your function.

    As you can see, our function logged that a file was uploaded with its file name and the bucket to which it was uploaded. It also mentions the contents inside the file as our function was defined to.
    Hence, we have successfully created a Python function inside AWS Lambda which logs every time an object is uploaded to an S3 Bucket.

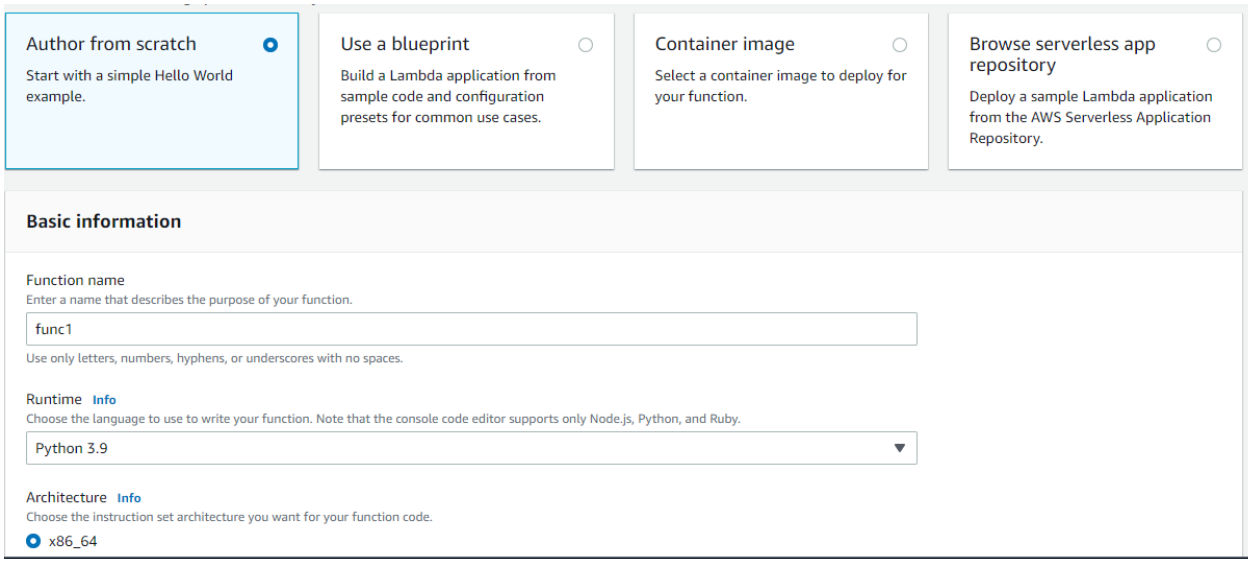
Part 2

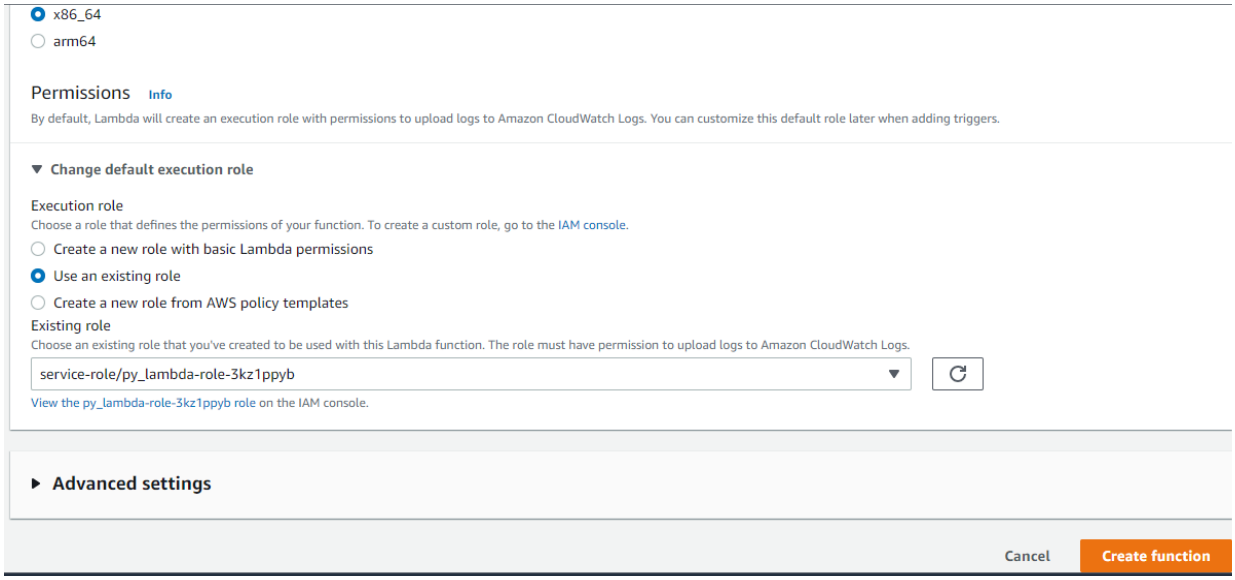
Sending an Email on Bucket additions to Bucket

1. Go to the IAM console and edit the same Lambda Role. This time, add SESFullAccess Permission to the role.

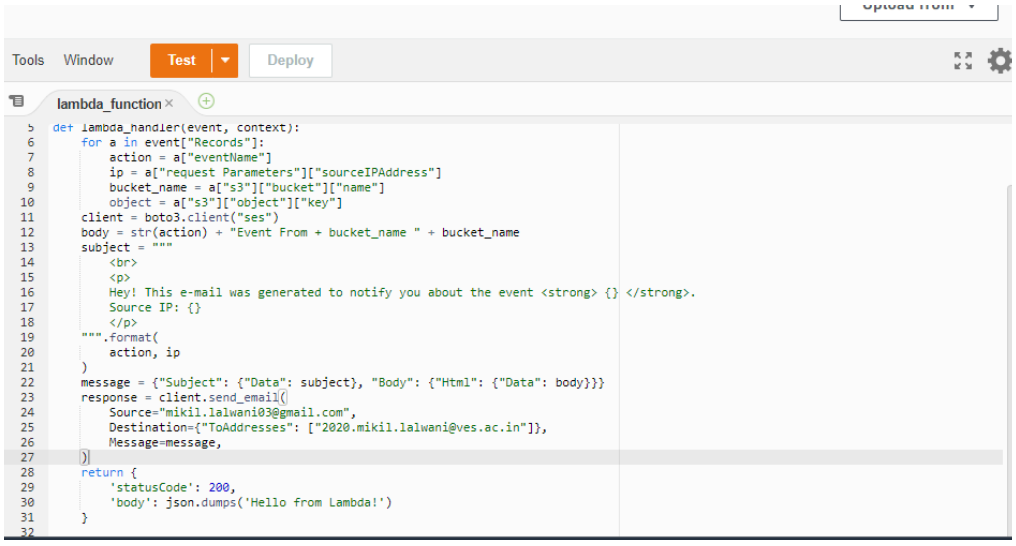


2. Create a new Lambda function in a Python environment. Use the existing role which was previously created.



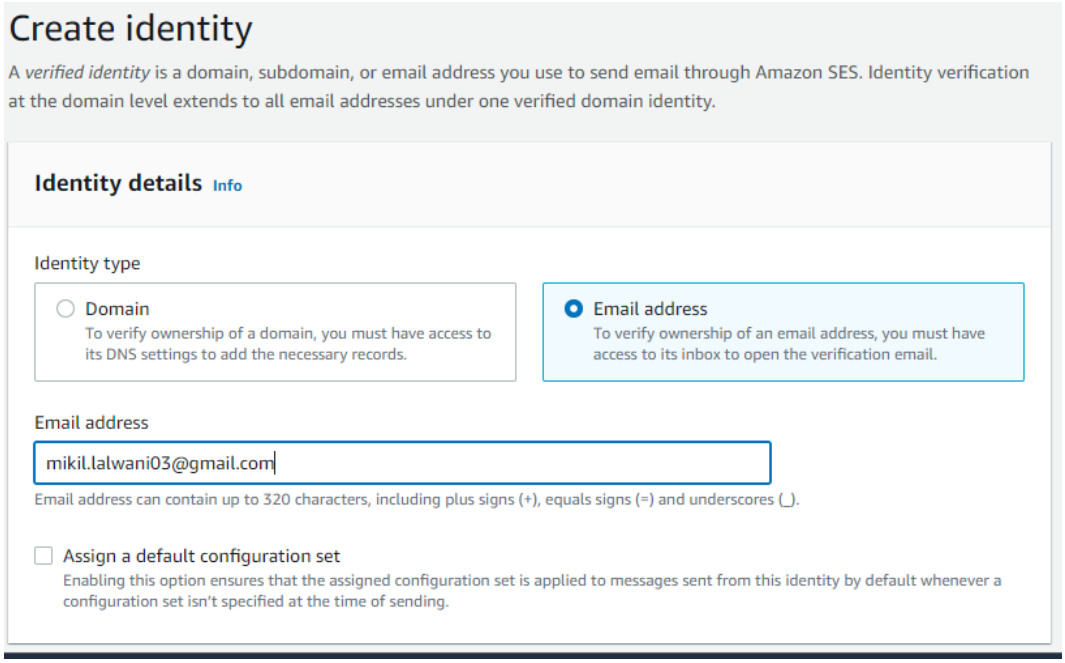


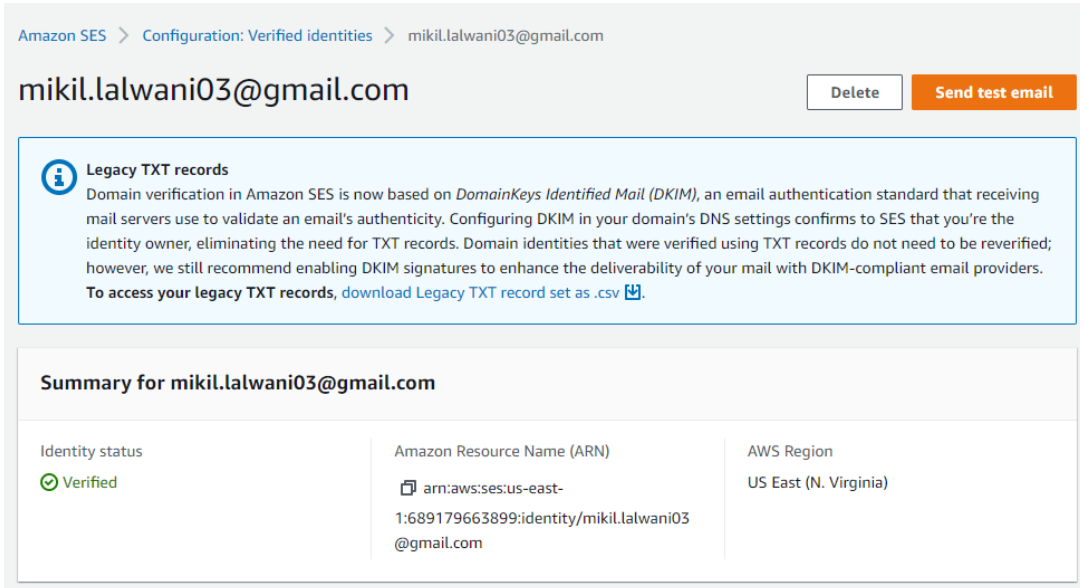
3. In this function, above the default hello-world TODO, add the following code.
   This code is basically to send an email on the creation of an object in the attached S3 Bucket. It sends the bucket name, event, and source IP address. In this code, modify the Source and Destination ToAddresses to your sender and receiver email addresses. Once done, deploy the function.



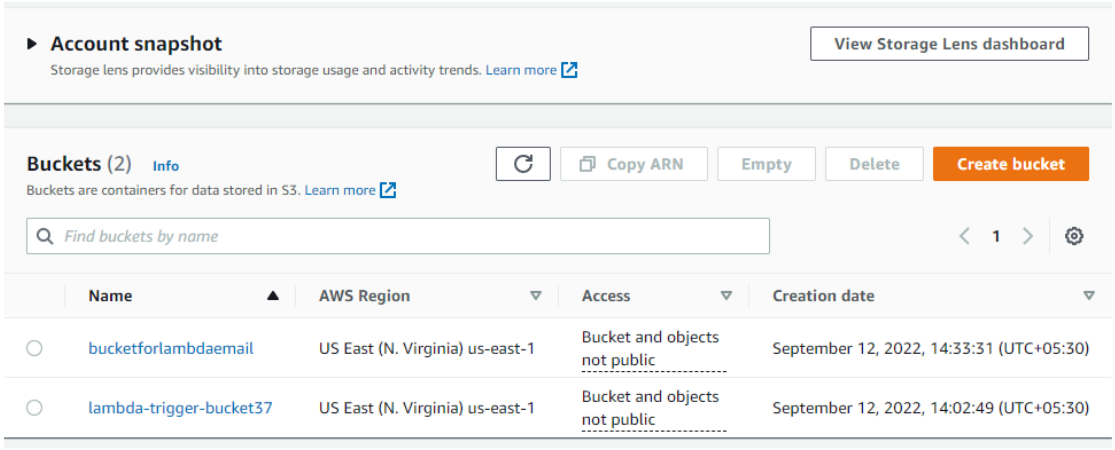
4. Open up the SES Console and click on Manage Email Addresses.

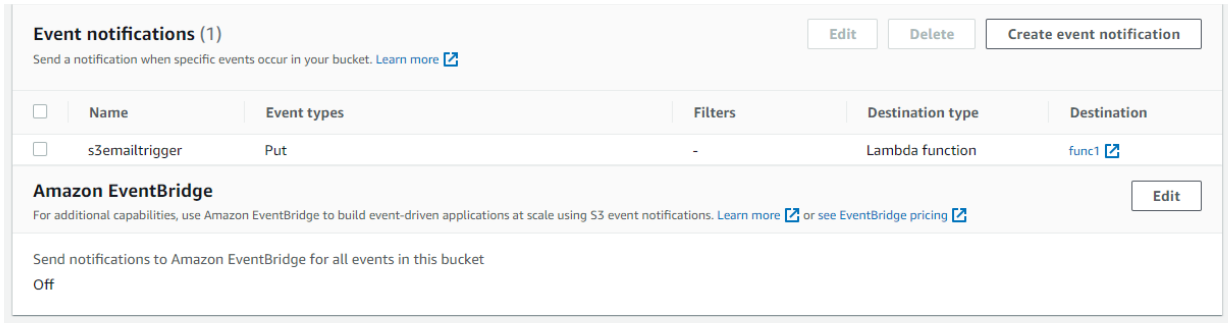
5. Choose Verify Email Address and verify both sender and receiver email addresses.
   Click on the verification links you are sent and verify the emails.



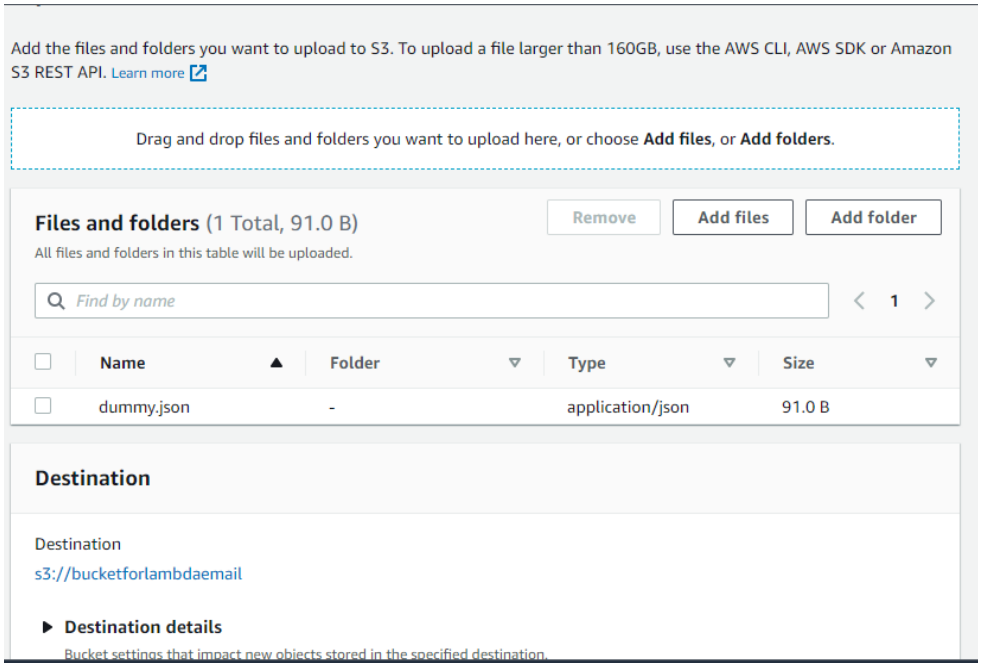


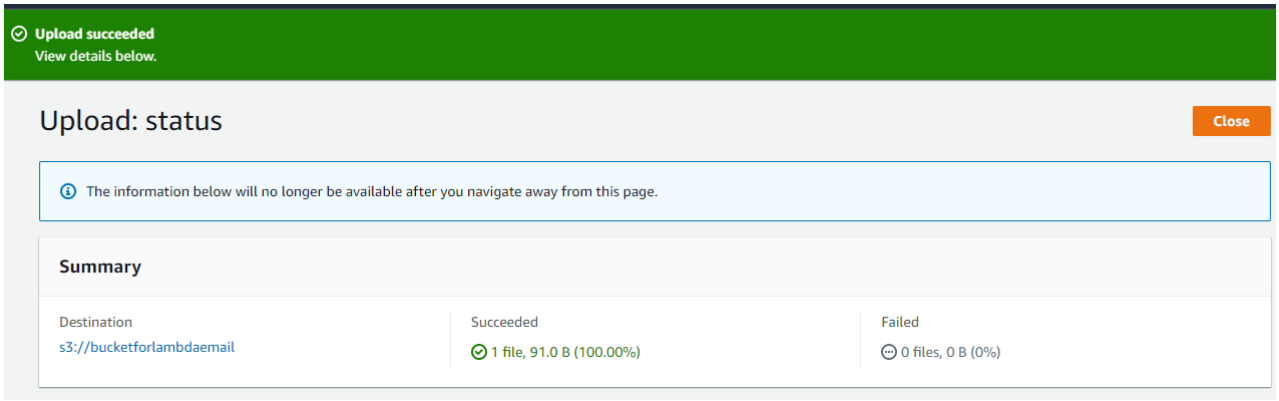
6. Now, open up the S3 Console, create a new bucket as you did previously and add an event notification inside events and attach it to your Lambda function.



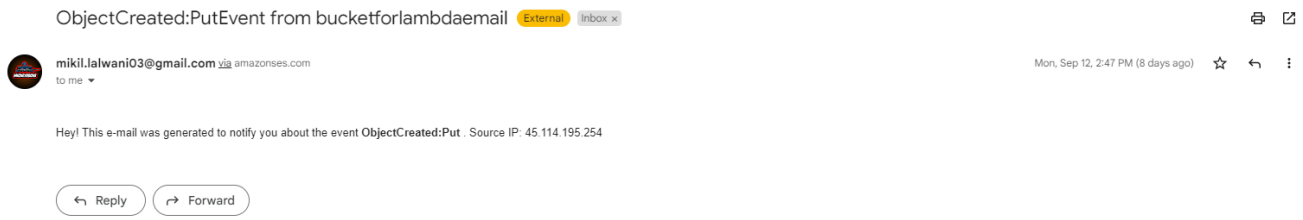


7. Once that's done, upload any file to your S3 Bucket. I'll upload the same dummy JSON file again.





8. Check your ToAddress email. You'll receive an email from the Source Address via Amazon SES.



In this way, we successfully created a function in AWS Lambda that sends an email on uploading an object to an S3 Bucket using Amazon SES.

Recommended Cleanup

Once done with the experiment, it is recommended to delete all resources which have been created and used by us to avoid charges in AWS.
Here is a list of things you may delete:

1. AWS Lambda Function
2. Amazon S3 Storage Bucket
3. Amazon SES Verified Emails
4. AWS Cloudwatch Logs (Optional, won't affect bills)
5. AWS IAM Role (the one which was created for the function, again, won't affect bills)

Conclusion -

In this experiment, we learned how to create an AWS lambda function to log everytime an image is uploaded to an S3 bucket.