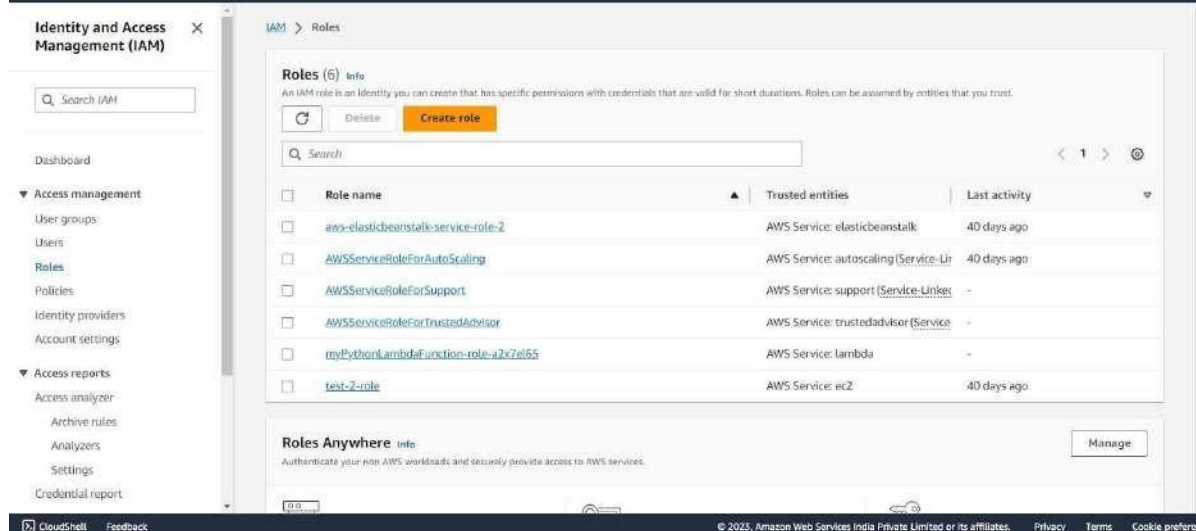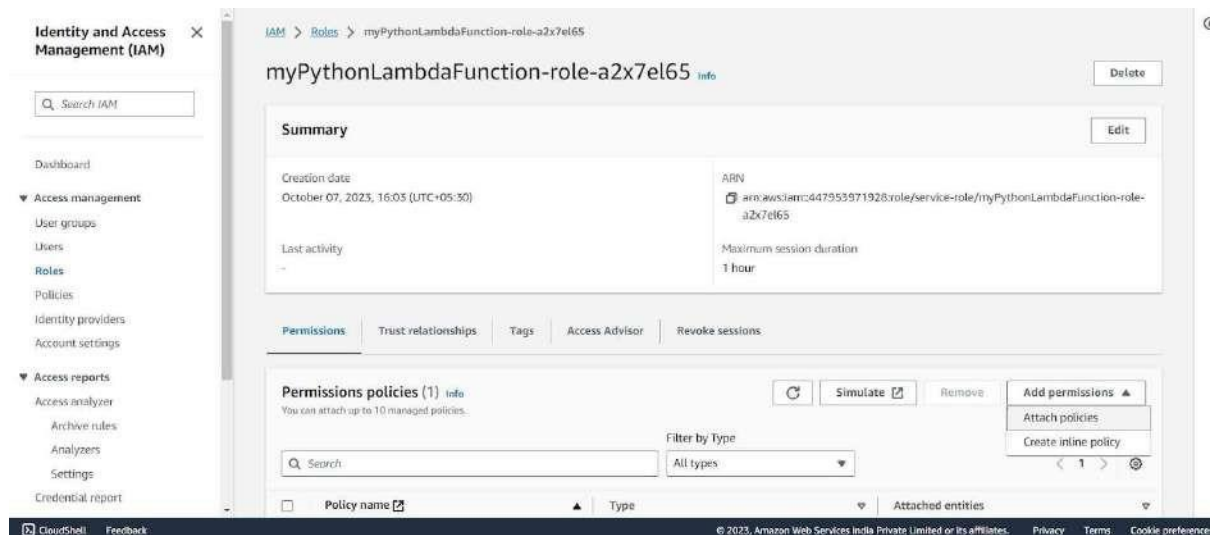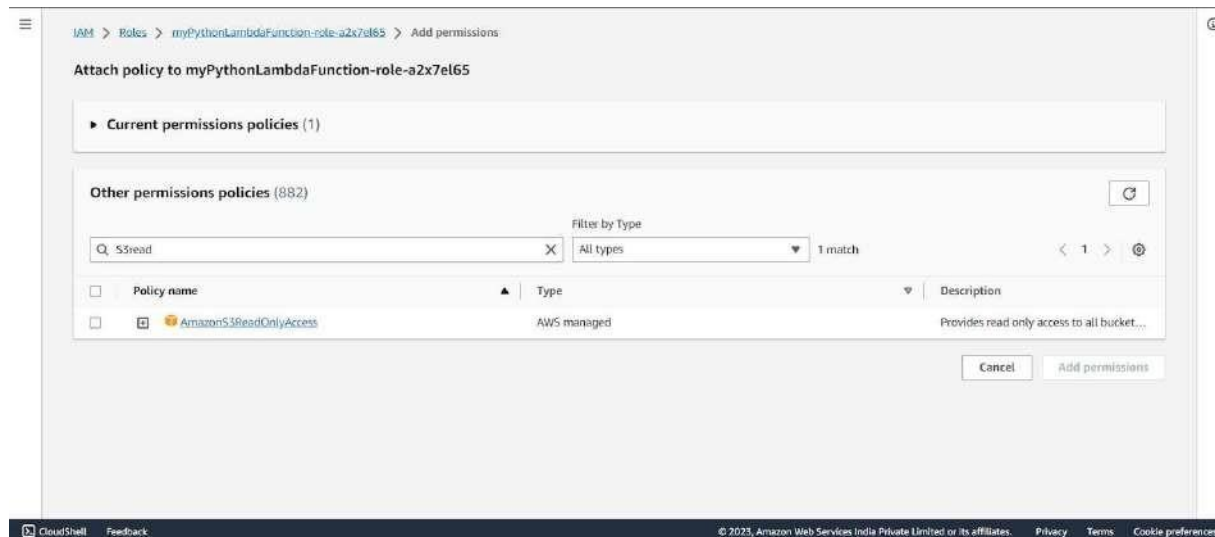# Adv. DevOps Exp. 12

Soham Satpute
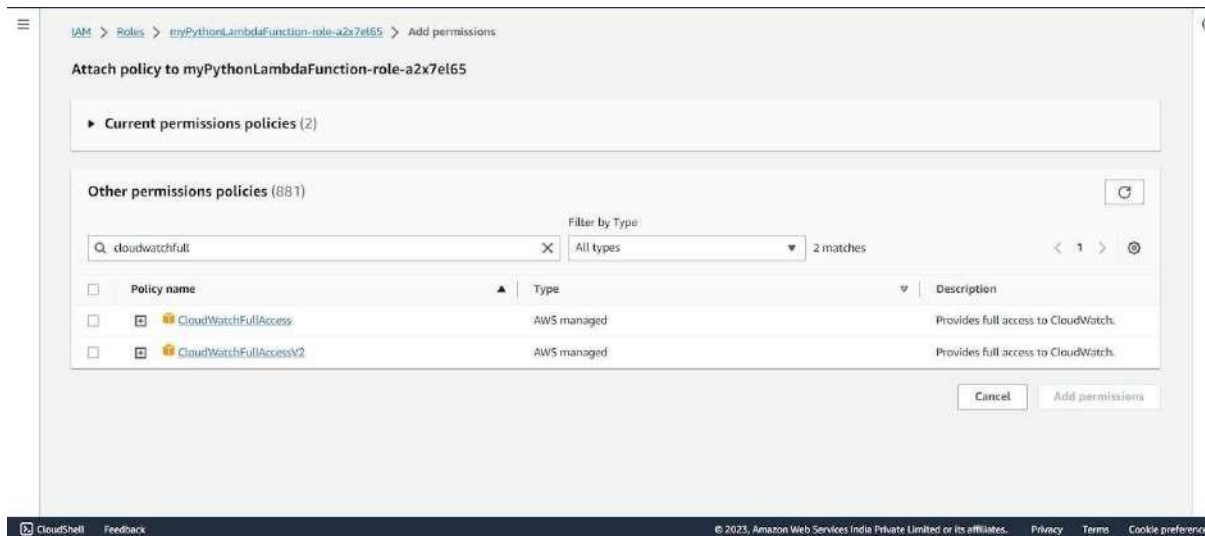
D15A - 52

Step 1: Open the IAM (user)

Step 2: Under Attach Policies, add S3-ReadOnly and CloudWatchFull permissions to this role.

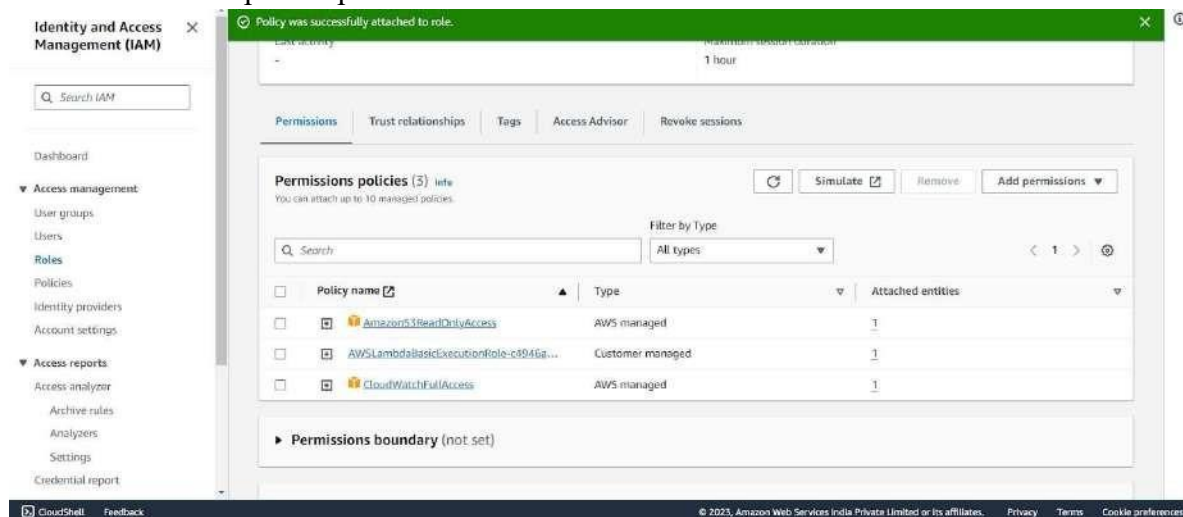

S3-ReadOnly

CloudWatchFull

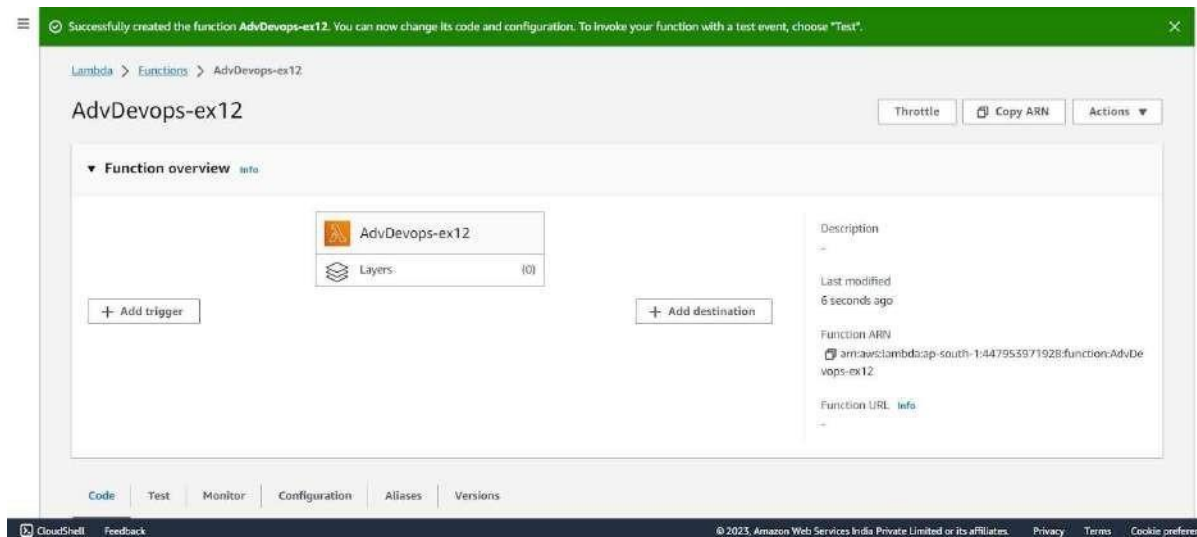After successful attachment of policy you will see something like this you will be able to see the updated policies.
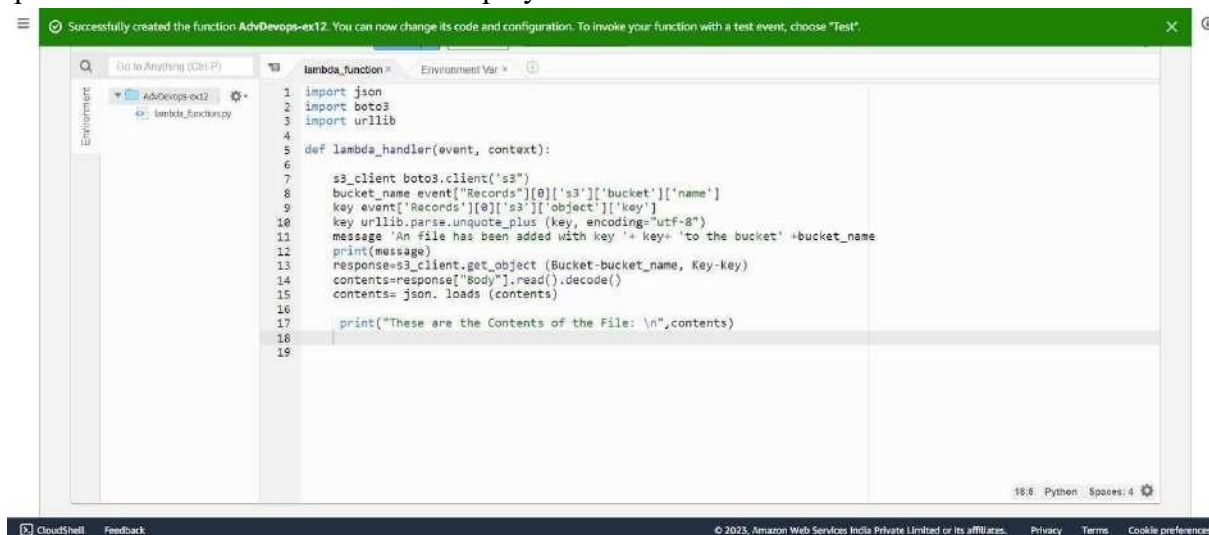
Step 3: Open up AWS Lambda and create a new Python function.

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

Step 4: The function is up and running.

Lambda > Functions > AdvDevops-ex12

AdvDevops-ex12

Throttle   Copy ARN   Actions ▼

▼ Function overview info

AdvDevops-ex12

Layers   (0)

+ Add trigger

+ Add destination

Description
-

Last modified
6 seconds ago

Function ARN
arn:aws:lambda:ap-south-1:447953971928:function:AdvDevops-ex12
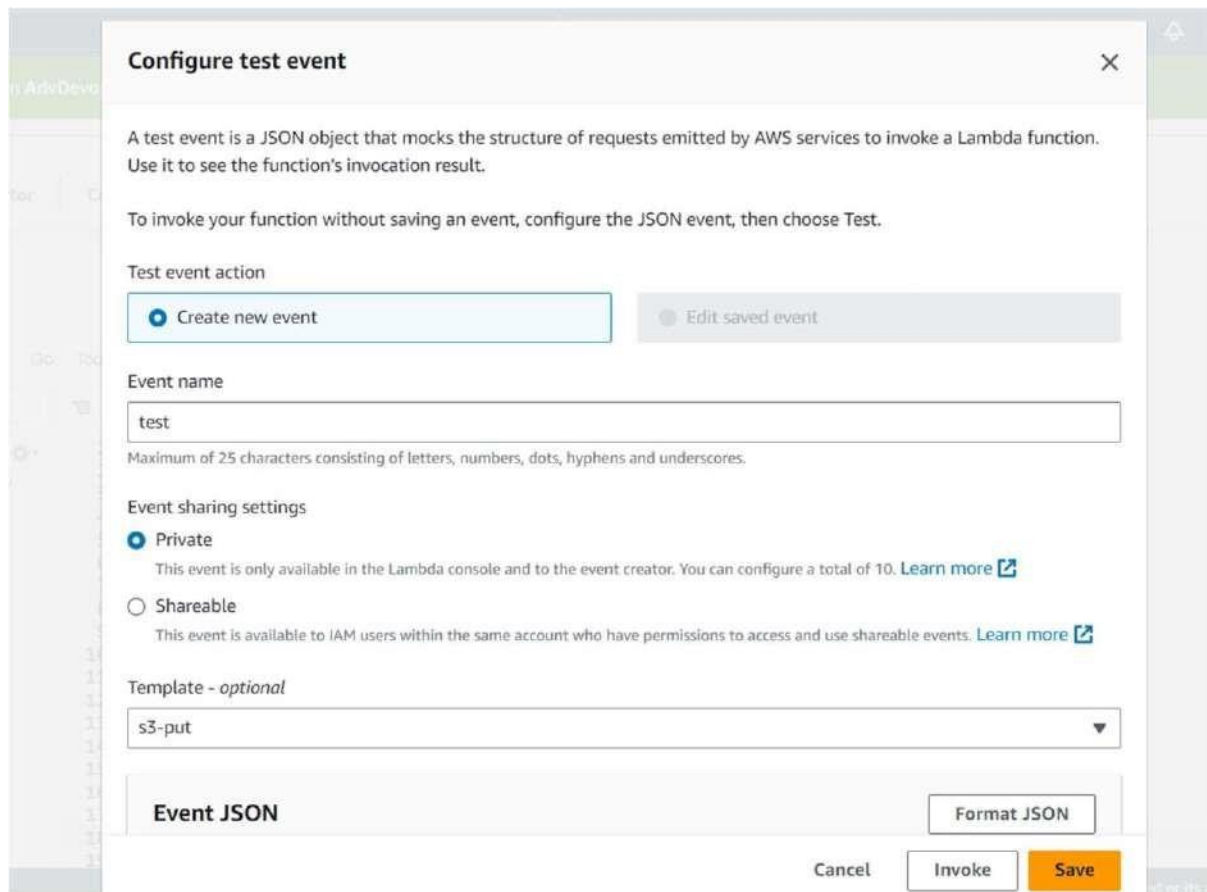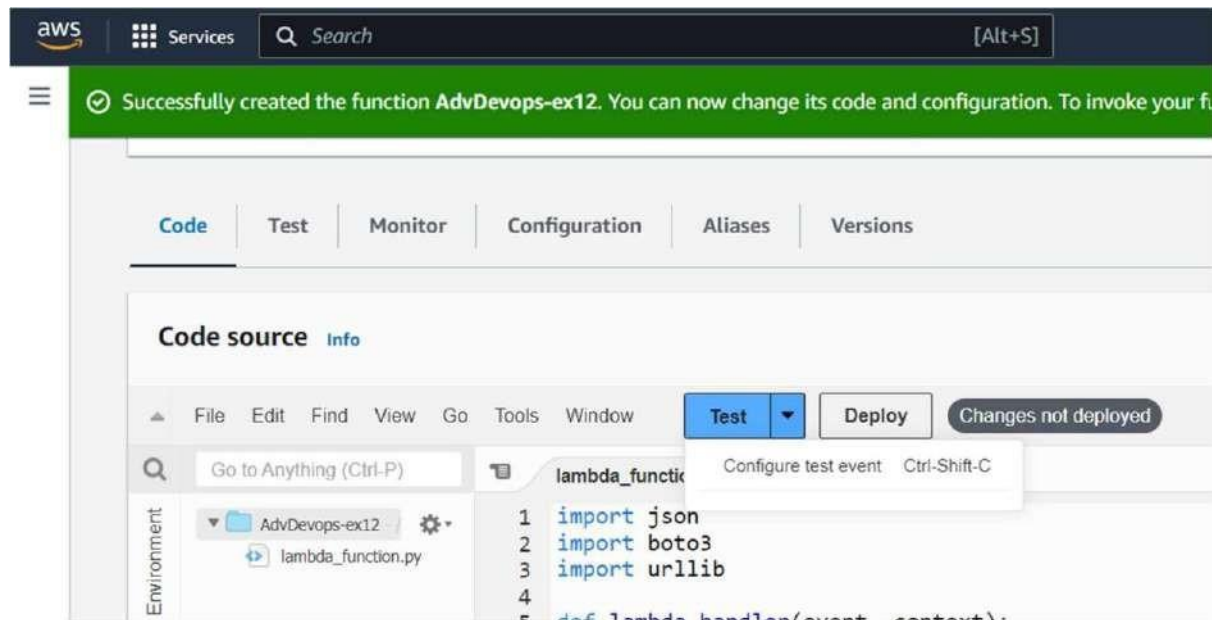
Function URL info
-

Code   Test   Monitor   Configuration   Aliases   Versions

Step 5: 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 and then deploy the code.

```python
import json
import boto3
import urllib

def lambda_handler(event, context):

    s3_client boto3.client('s3')
    bucket_name event["Records"][0]['s3']['bucket']['name']
    key event['Records'][0]['s3']['object']['key']
    key urllib.parse.unquote_plus (key, encoding="utf-8")
    message 'An file has been added with key '+ key+ 'to the bucket' +bucket_name
    print(message)
    response=s3_client.get_object (Bucket-bucket_name, Key-Key)
    contents=response["Body"].read().decode()
    contents= json. loads (contents)

    print("These are the Contents of the File: \n",contents)
```
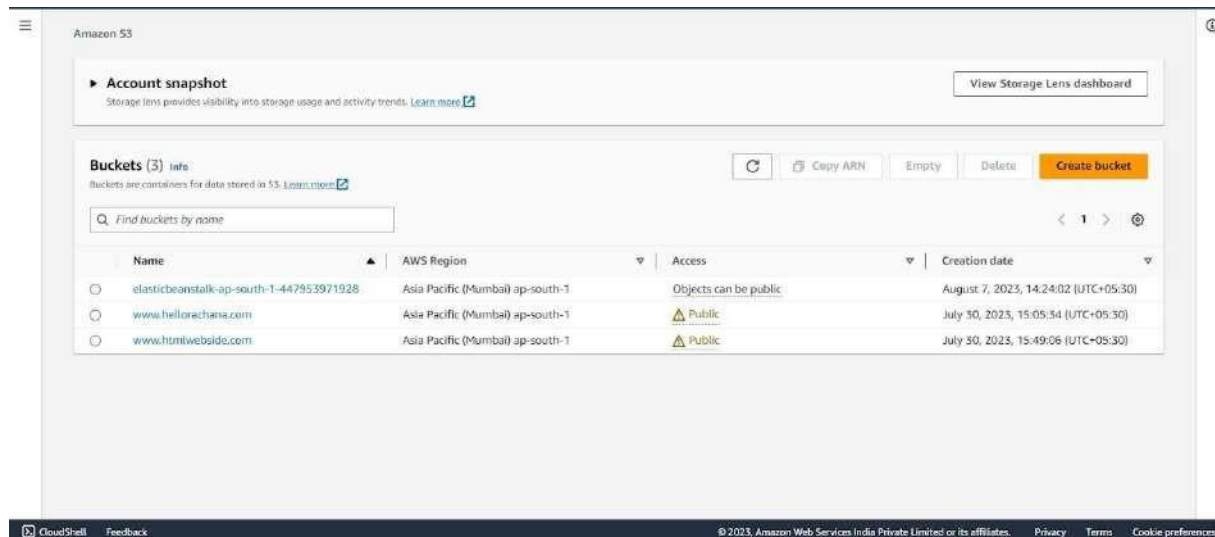
18:8 Python Spaces: 4

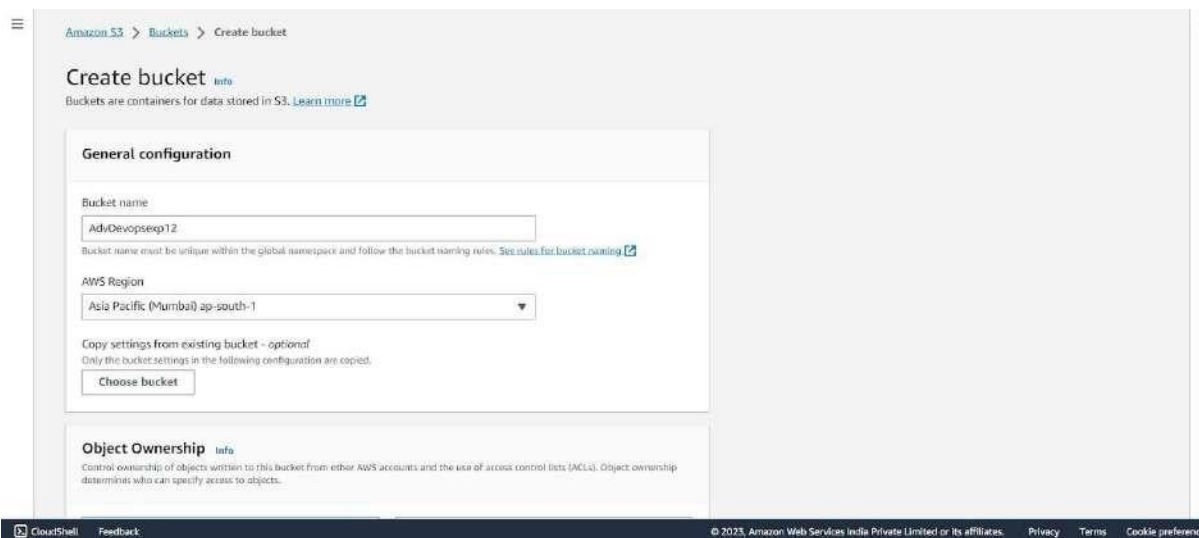Step 6: Click on Test and choose the 'S3 Put' Template.
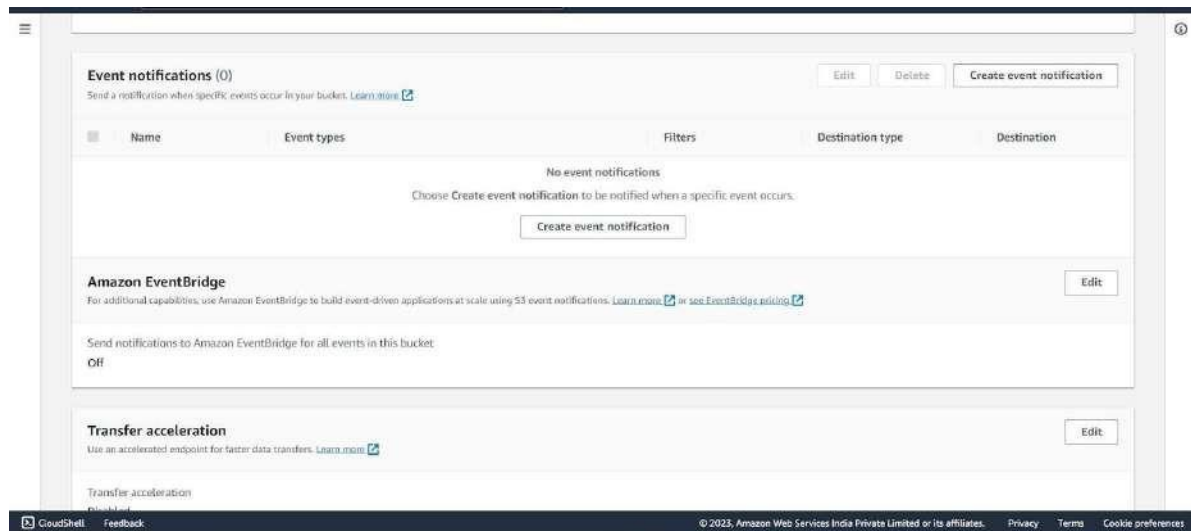
And Save it.

Step 7: Open up the S3 Console and create a new bucket.

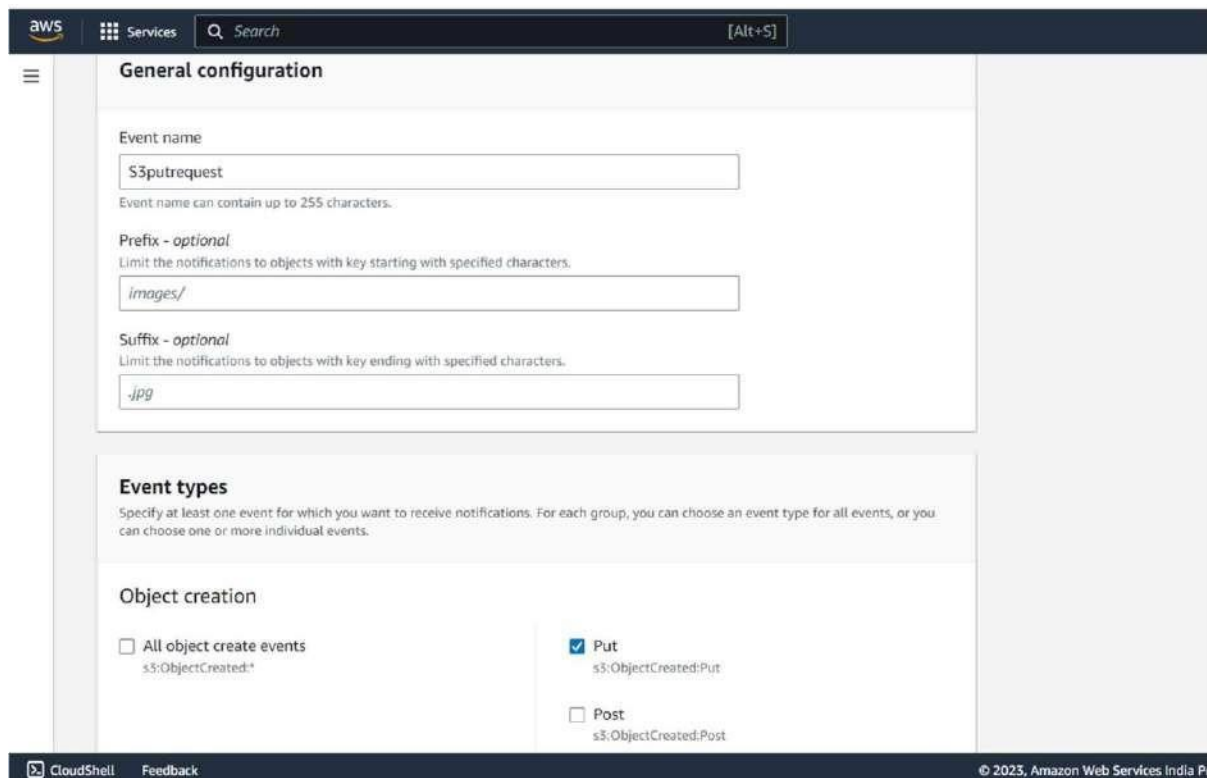Step 8: With all general settings, create the bucket in the same region as the function.

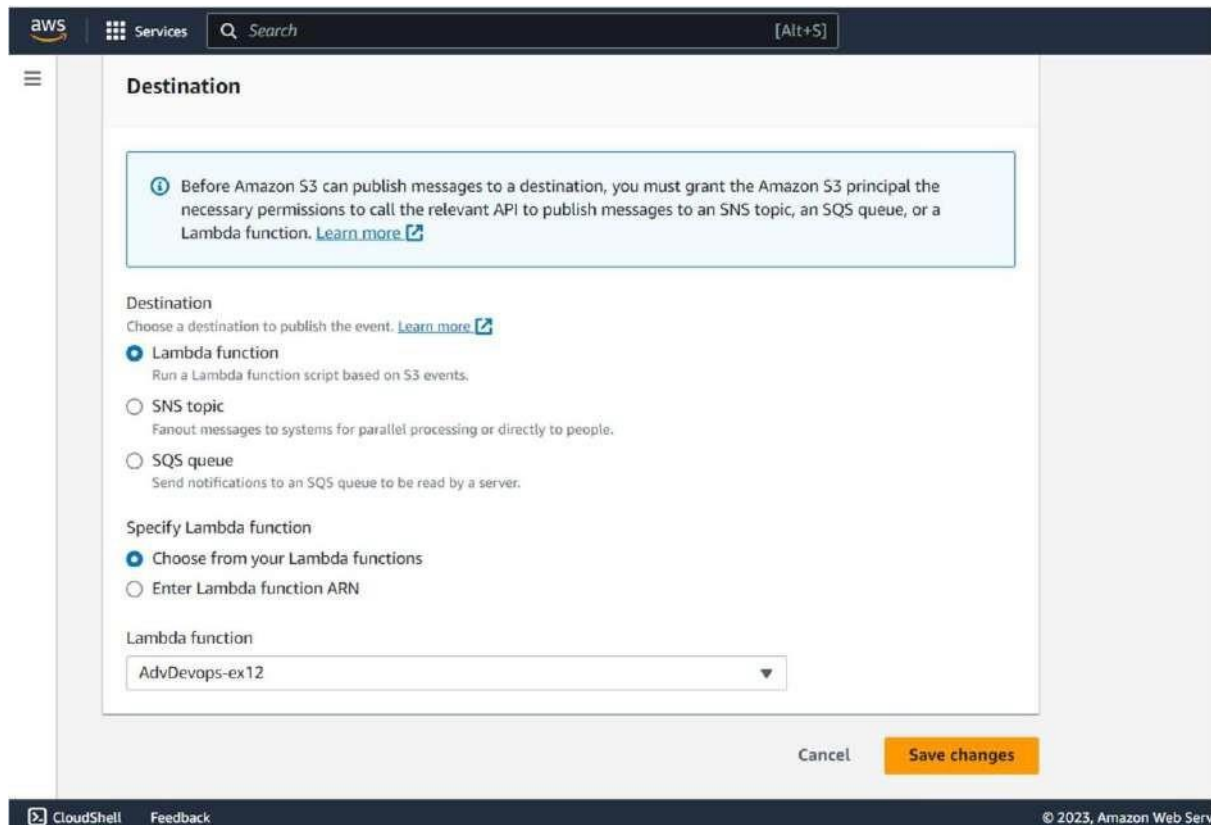Step 9: Click on the created bucket and under properties, look for events.
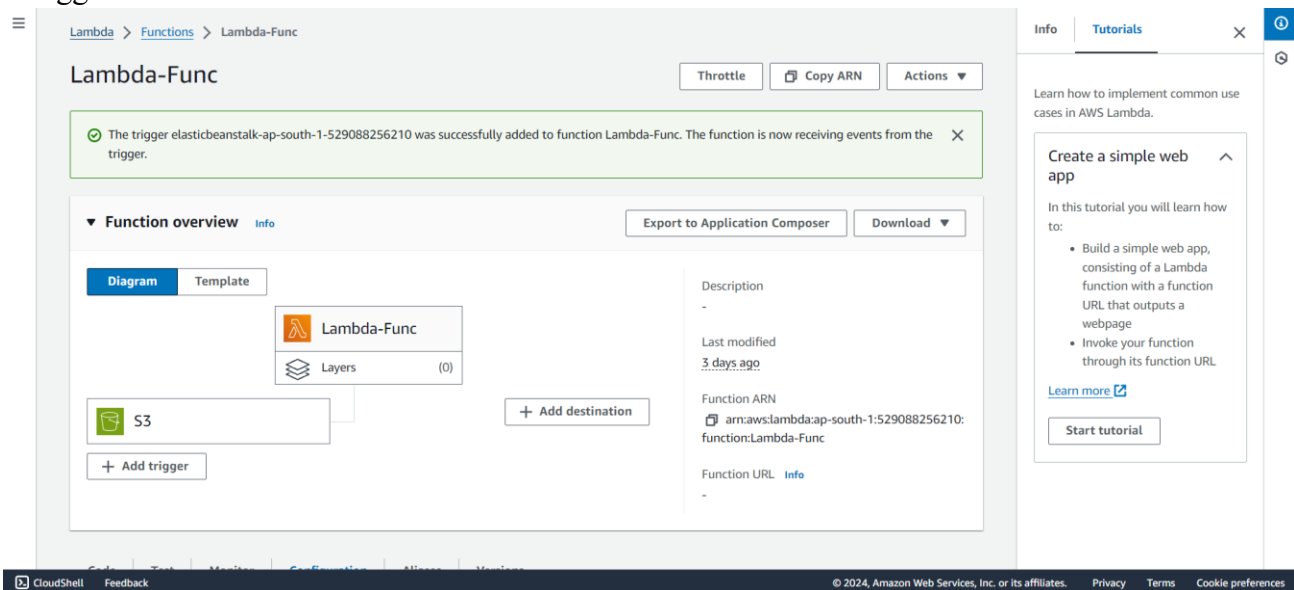
Click on Create Event Notification.

Step 10: Mention an event name and check Put under event types.

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

Step 11: Refresh the Lambda function console and you should be able to see an S3 Trigger in the overview.

Step 12: Now, create a dummy JSON file locally.

Step 13: Go back to your S3 Bucket and click on Add Files to upload a new file.
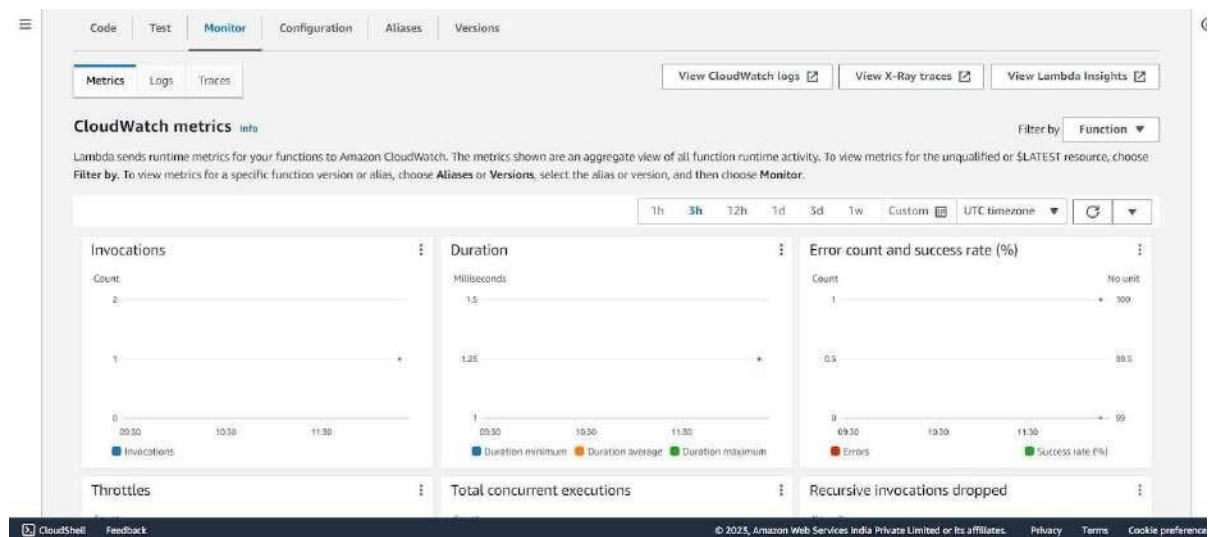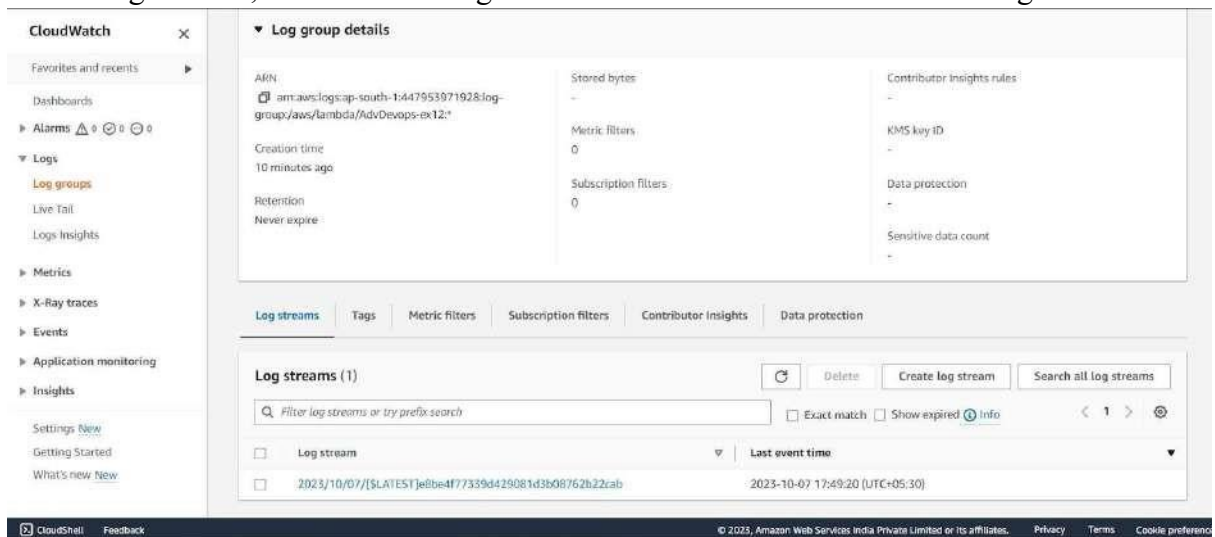Step 14: Select the dummy data file from your computer and click Upload.
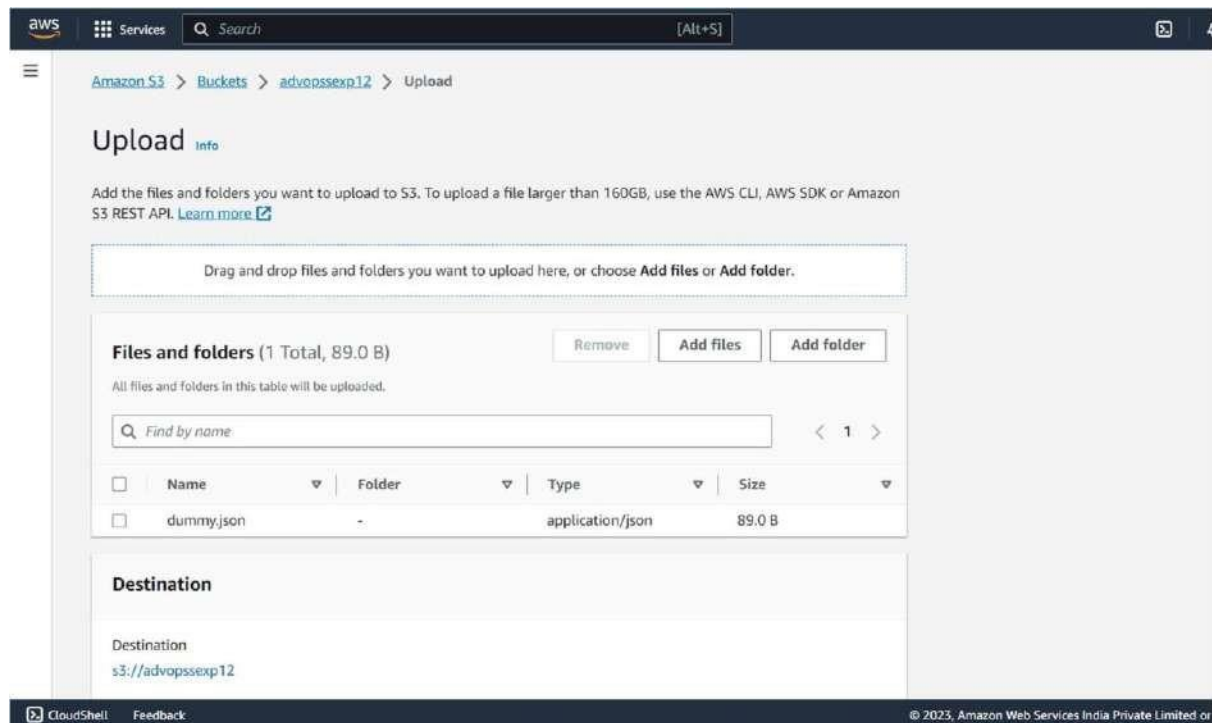
Step 15: After this make the necessary changes in the Test configuration file which we created it previously by replacing the Bucket Name and the ARN of Bucket.

Step 16: Go back to your Lambda function , Refresh it and check the Monitor tab.

Under Log streams, click on View logs in Cloudwatch to check the Function logs.

Step 17: Click on this log Stream that was created to view what was logged by your function.

Conclusion: Thus, we have created a Lambda function which logs "An Image has been added" once you add an object to a specific bucket in S3.