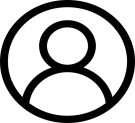
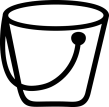
**Event Based Project Using Labda,S3 And SQS**

USER S3 SQS LAMBDA

It is very popular architecture used in serverless data pipeline.Suppose there is a user who is uploading data using **AWS** mangement console or using some **API** to our S3 bucket and from that S3 we are having an event notification which triggered the back end lambda which processes S3 files or may be lambda is triggering some other servises who is processing S3 files.

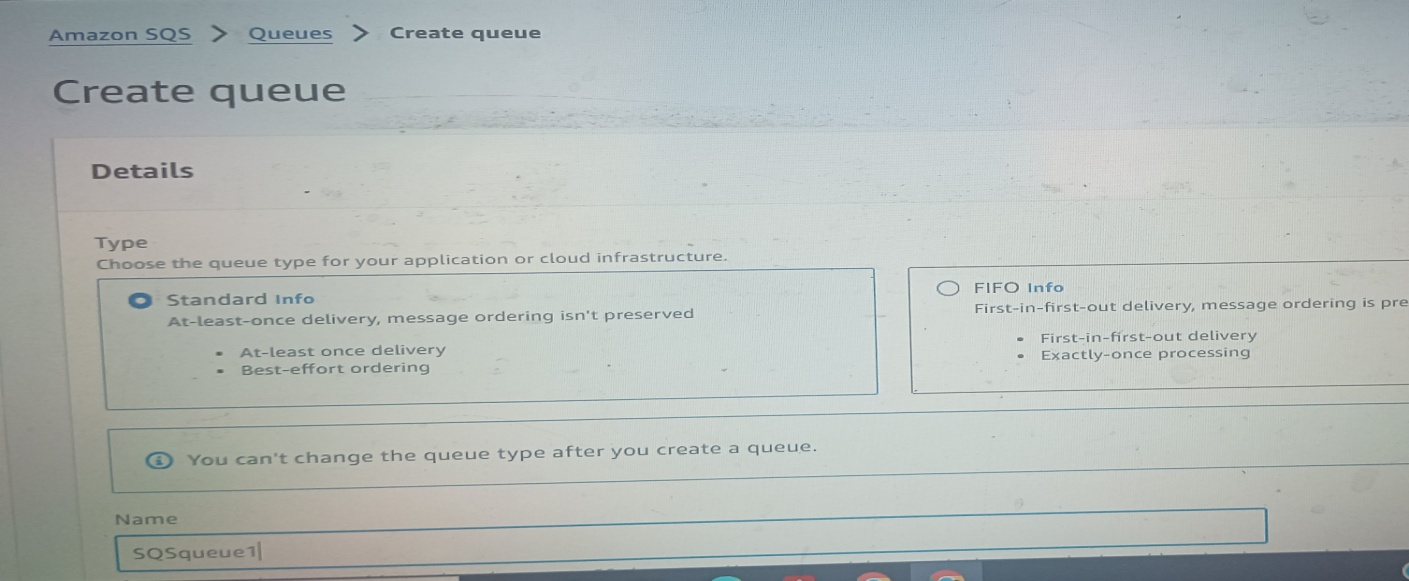
The direct connectivity between S3 event and lambda, then the problem we might face is, suppose in a very rapid speed the events are occuring in S3 then in that same speed lambda need to scale up or lambda will trigger the back end servises. To avoid that we want to accumulate certain no.of S3 events and then we want to trigger lambda. So we put SQS in the middle of lambda and S3. If it in very rapid speed S3 events are coming they will keep on getting accumulated on SQS queue and lambda will read S3 events from SQS queue in batch manner and it will process them.

**# Step 1 Creating SQS Queue**

1.Go to aws management console.

2.Choose SQS create queue Standard name the queue

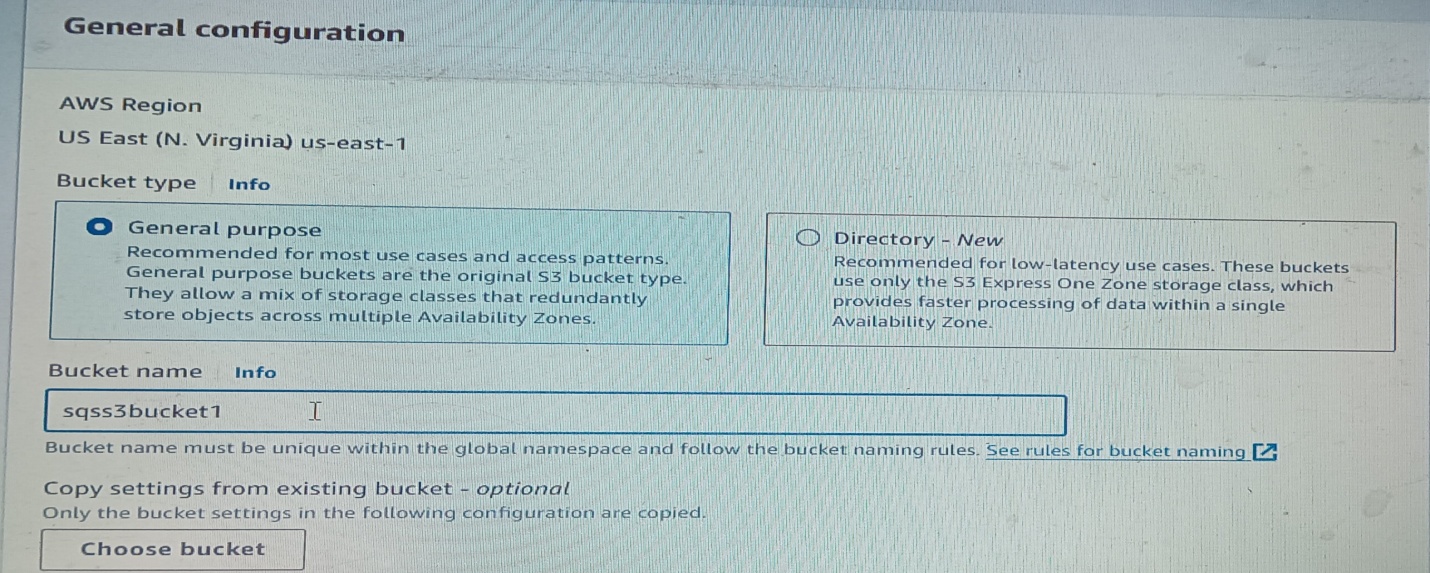
3.Keep everything default create queue.



**#Step 2 Creating S3 Bucket**

1.Go to S3 create bucket

2. Name the bucket. Keep everything default create bucket.



**#Step 3 Connect SQS And S3**

1.SQS queue edit policy generator

2.Type of policy SQS queue policy

3.Add statement Effect select Allow Principal select \*

4. Select “All action”. Then ARN put SQS queue ARN

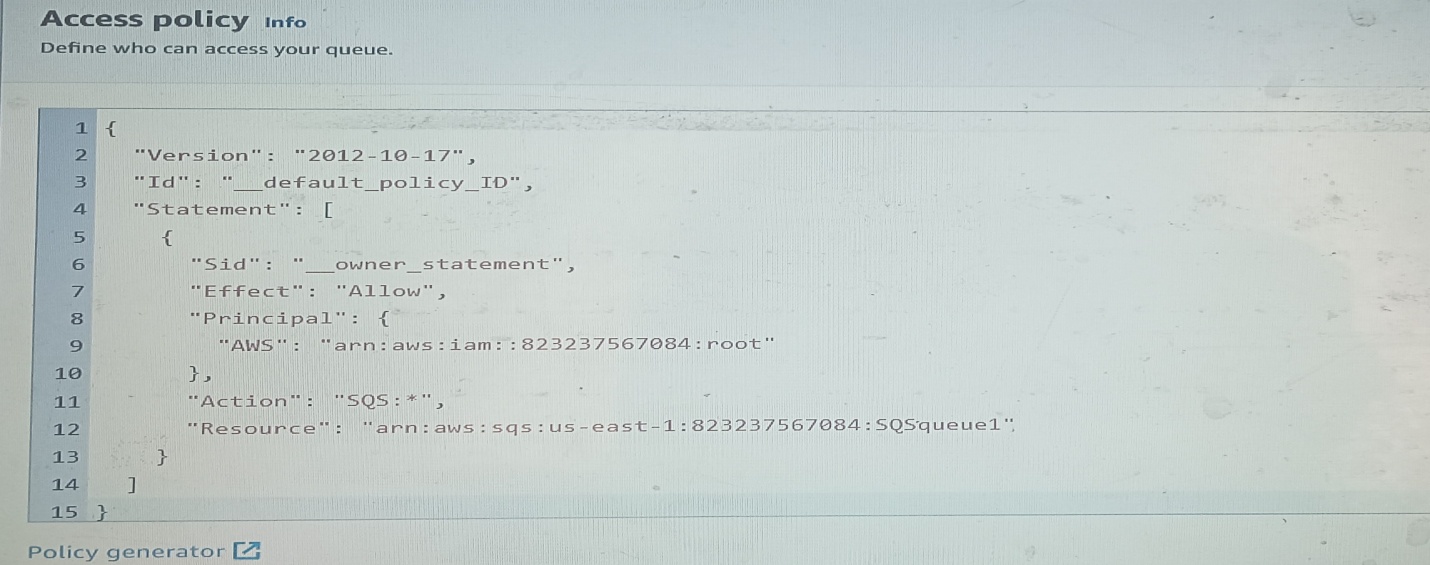
5.Add condition condition ARN equals Key Select “Source ARN” value copy ARN of S3, paste here.

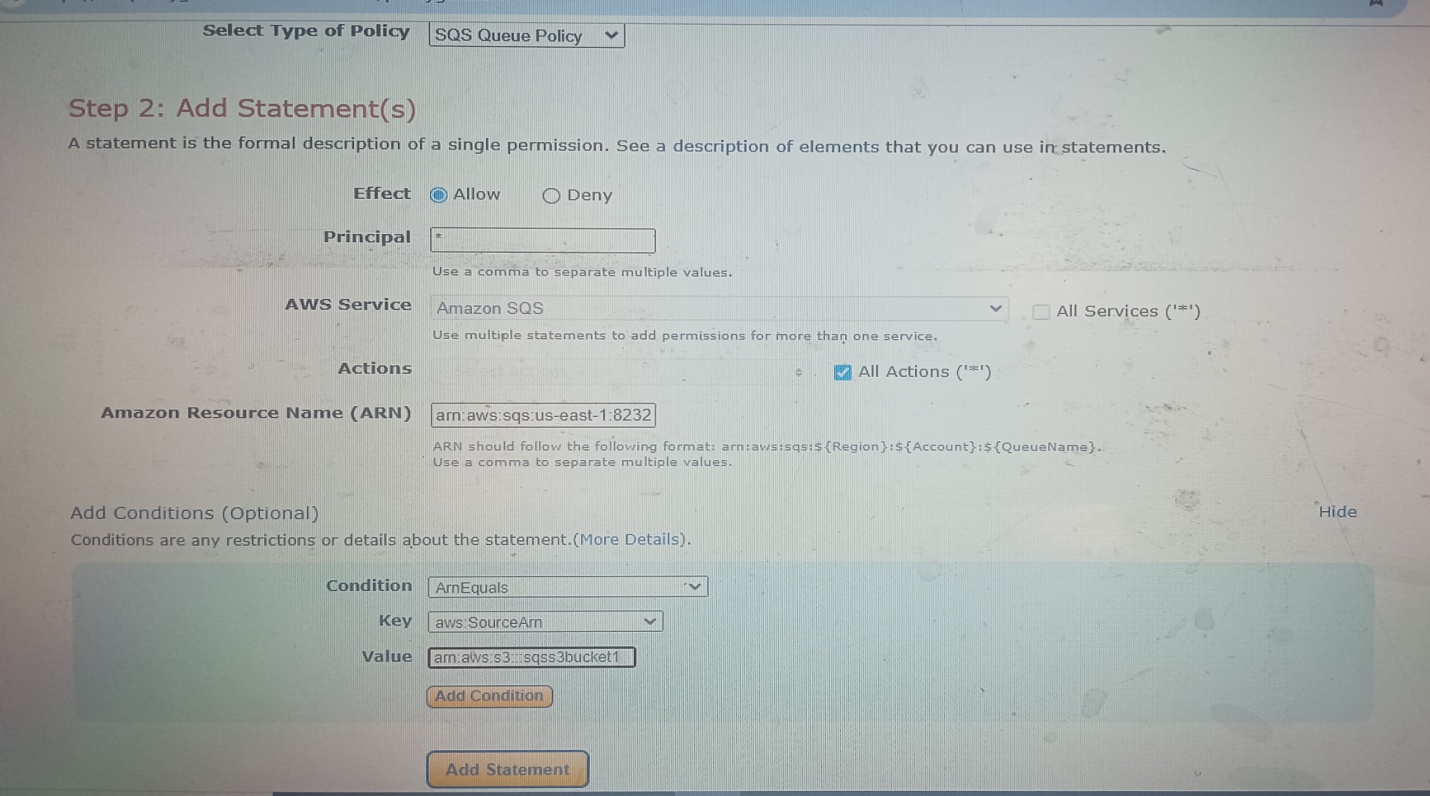
6.Click on “Add condition” then “Add statement” and “Generate policy”.

7.Copy the generated policy, paste over access policy part and save it.

8.Go back to S3 your bucket properties event notification

9.Name the event event type select event type (all object creation) destination SQS queue select SQS queue that we have created click on save changes.





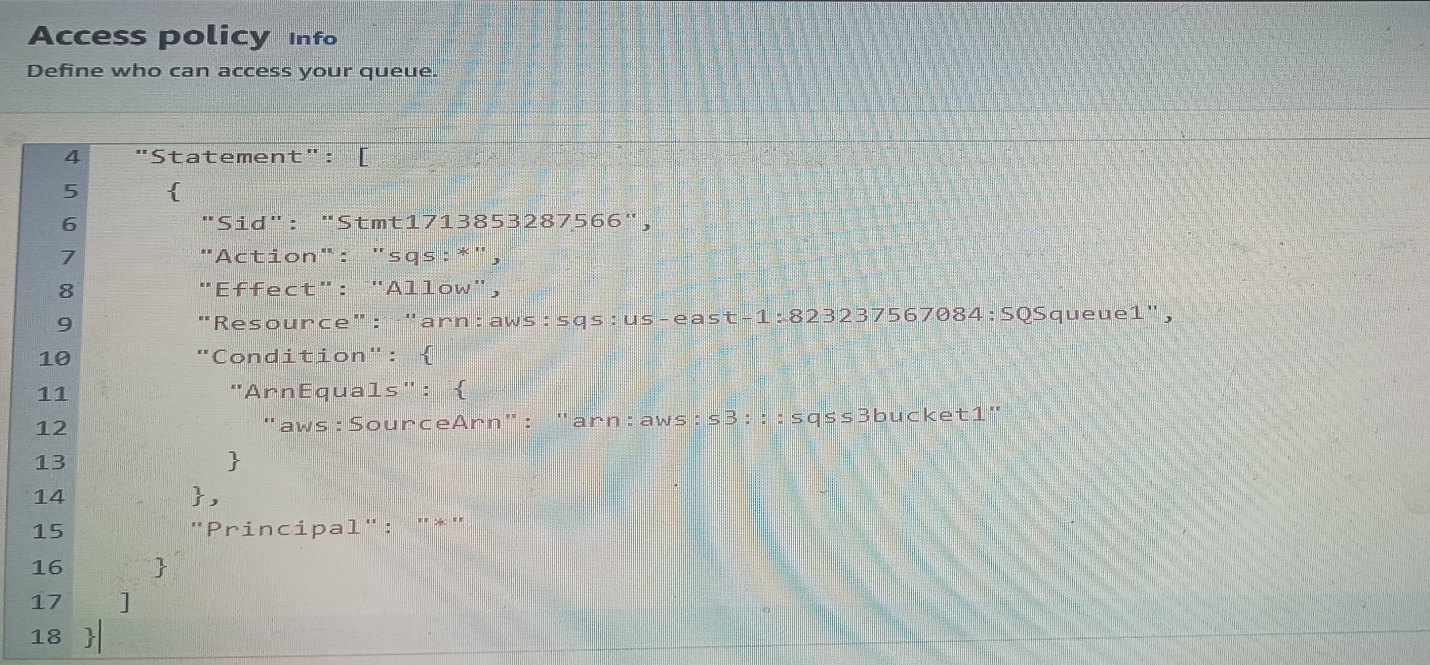
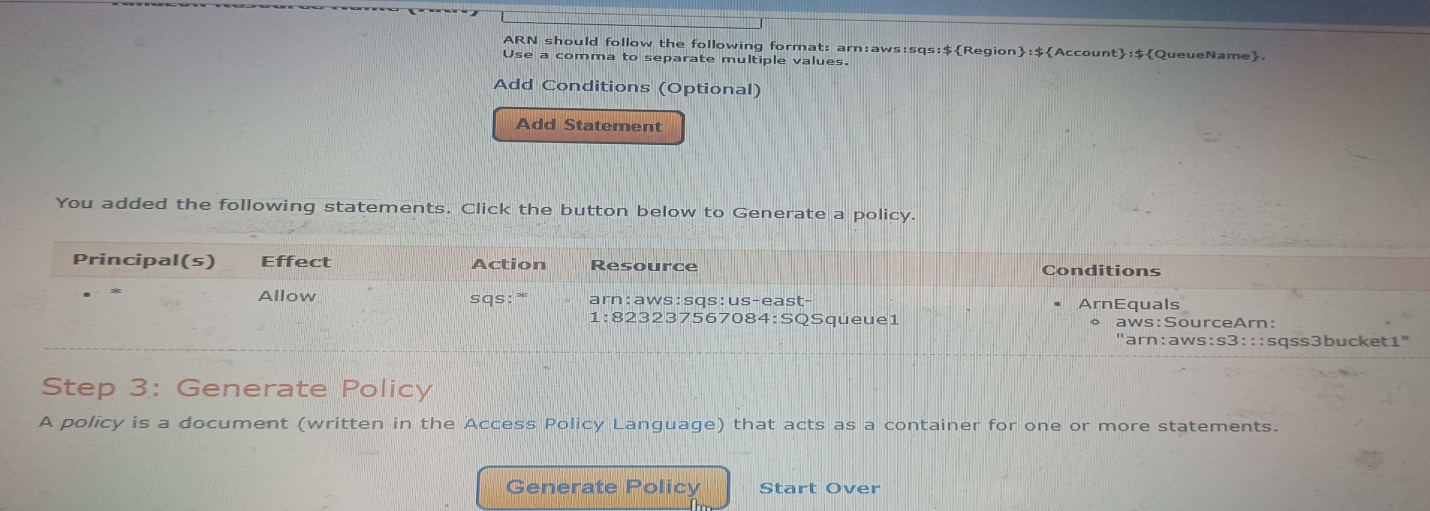
S3 bucket ARN

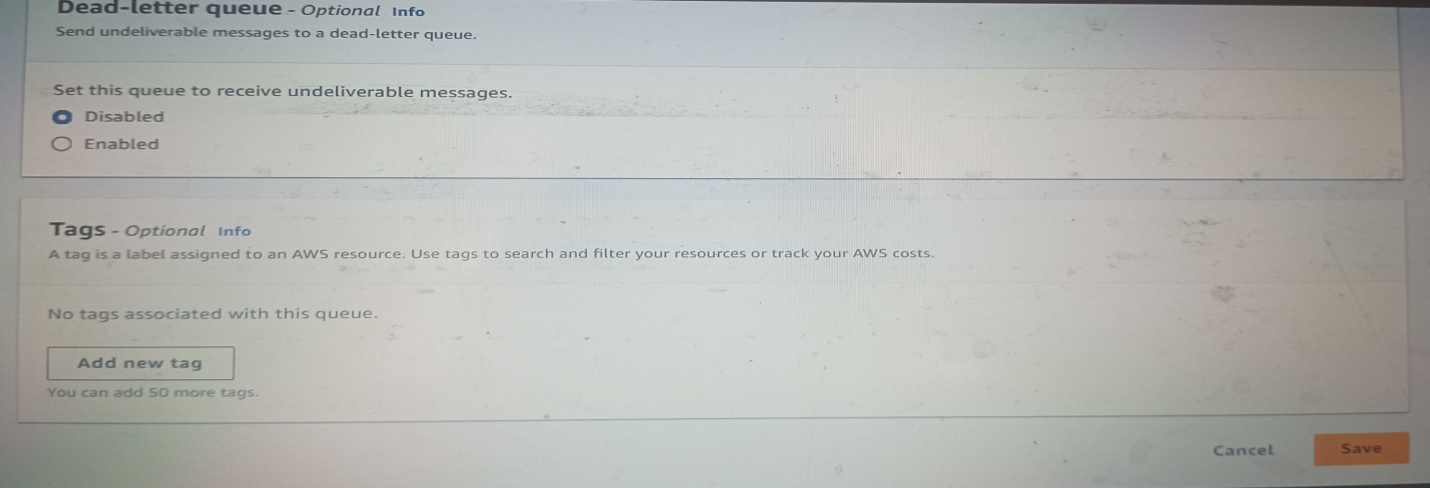
Source ARN

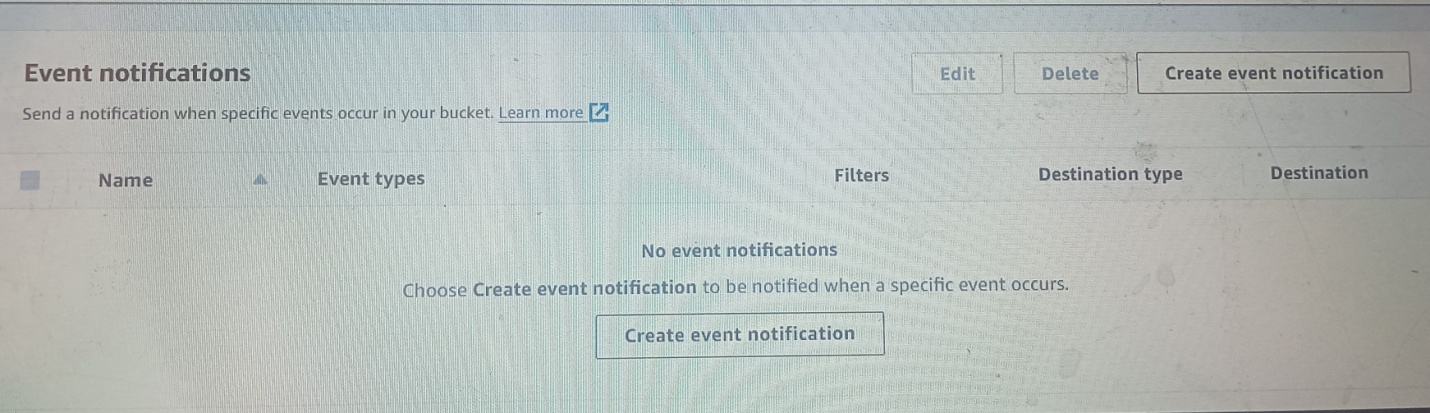
SQS ARN

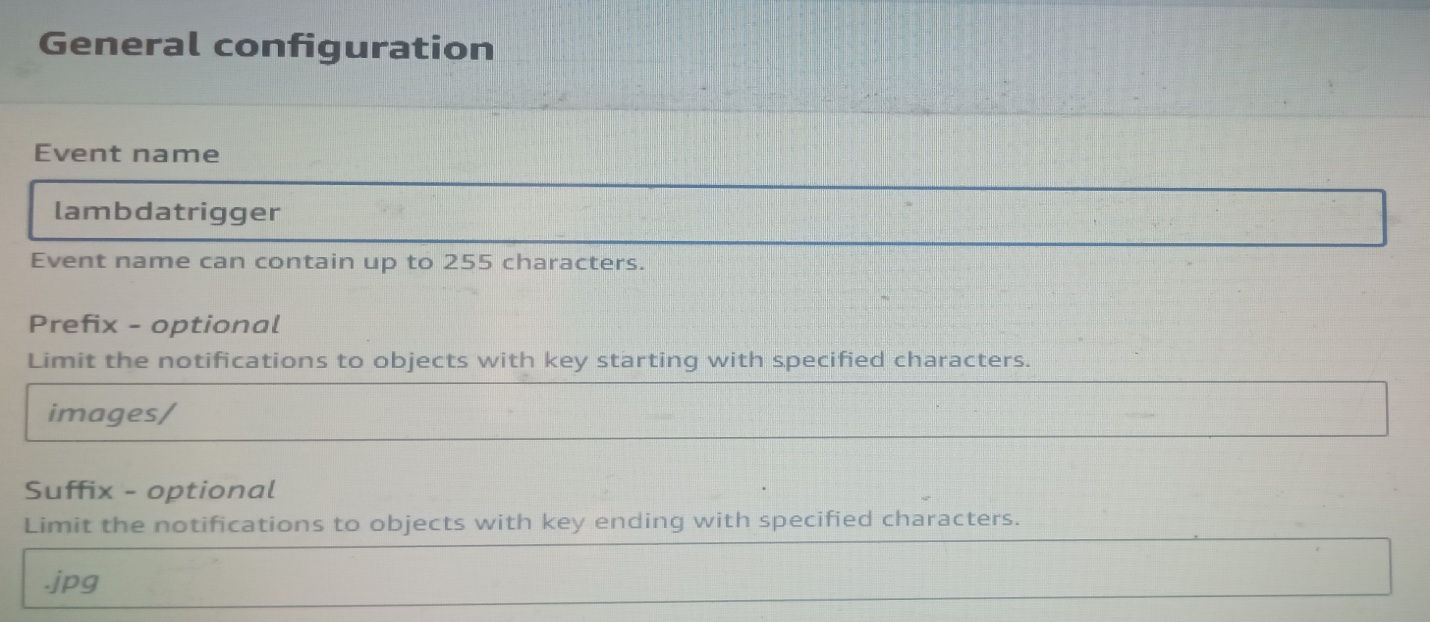
Principal **\***

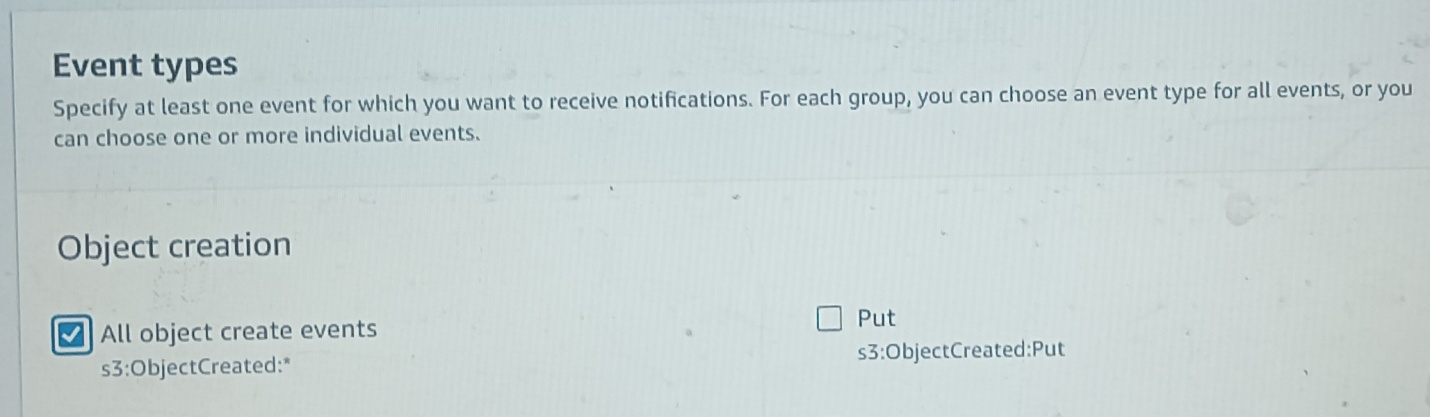
All actions

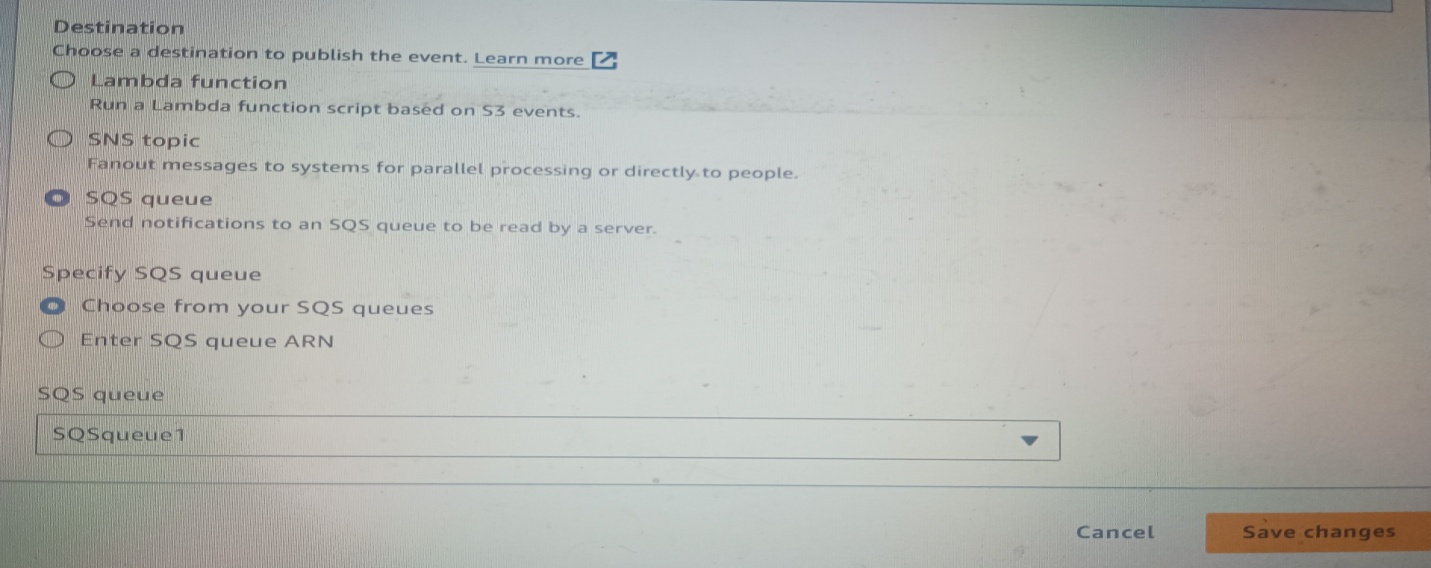












**#Step 4 Ceating Lambda Function**

1.Go to lambda select create function name the function select runtime (python 3.9) create function

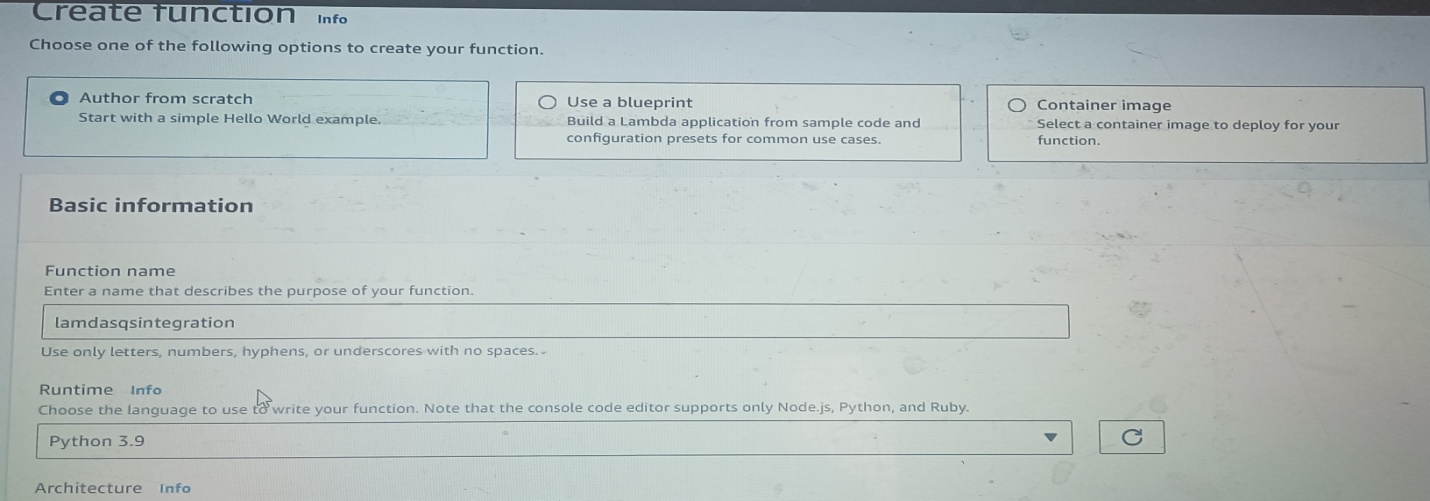
2.Then choose code put your code then deploy.

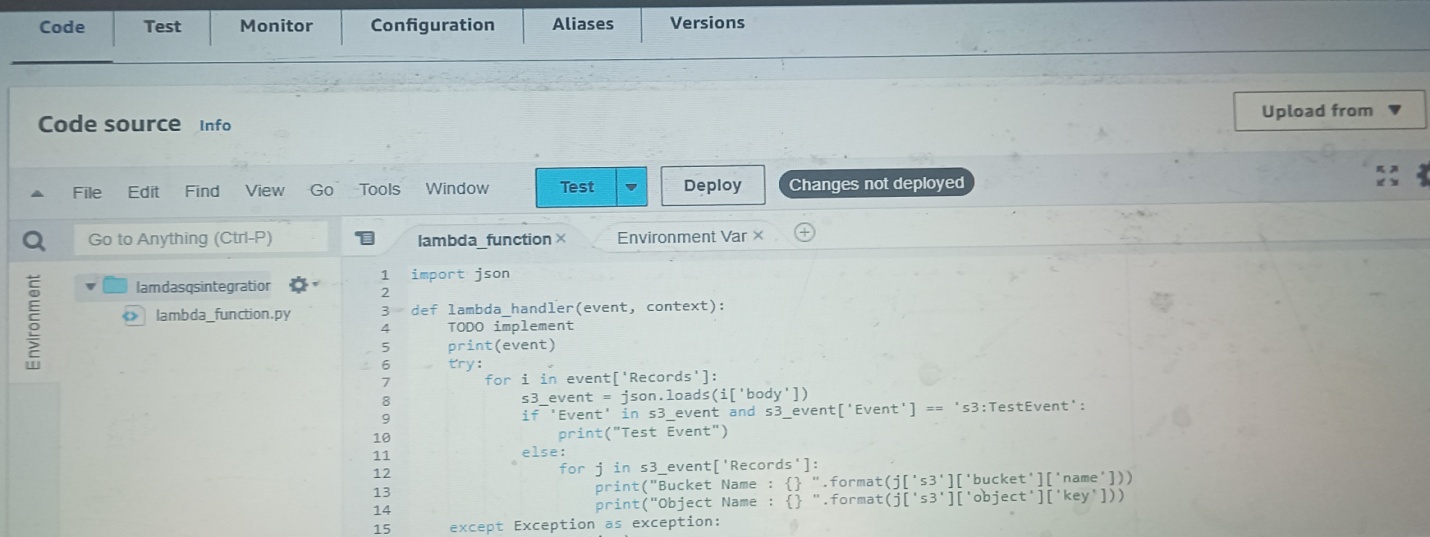
3.Go to configuration permission click on role add permission attach policy give SQS full access save

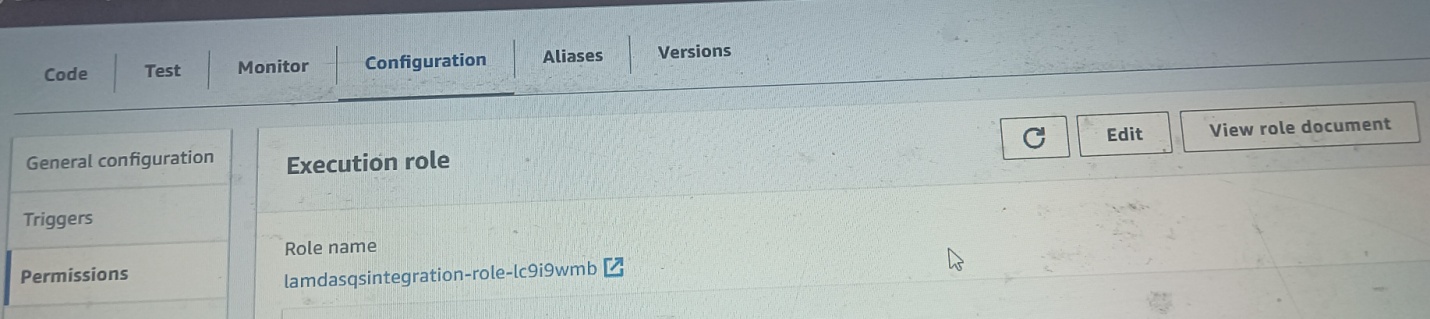
**(**If you check SQS queue, we can see messages(we have not added any files to S3 and we have not triggered lambda). Because whenever we create integration between S3 and SQS, S3 will send a test event. That is the message we can see in queue.

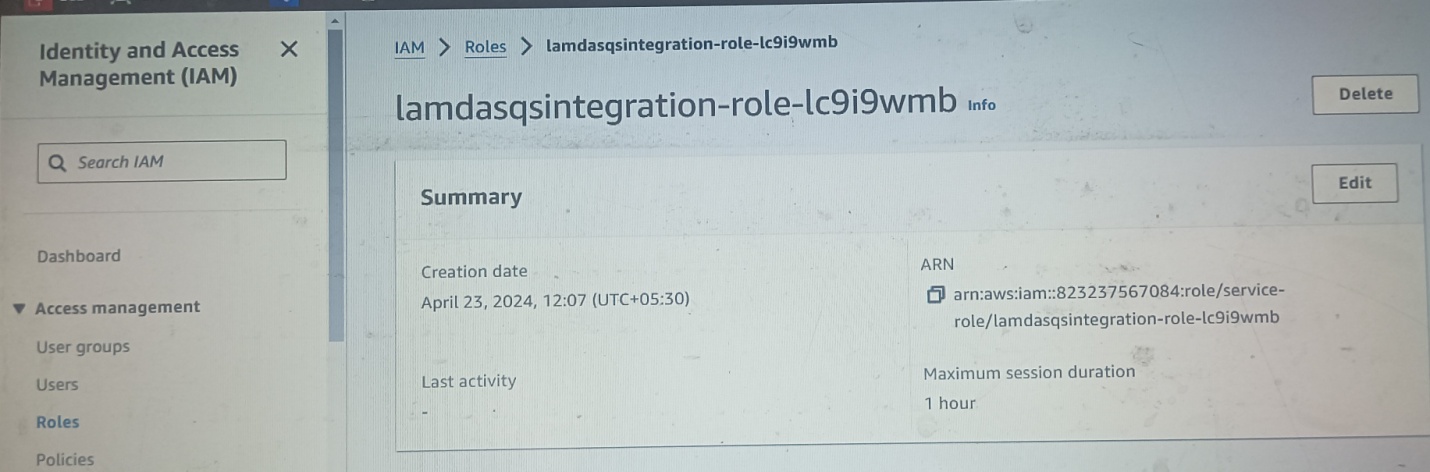
To check that, go to SQS queue Click on “Send and receive messages”

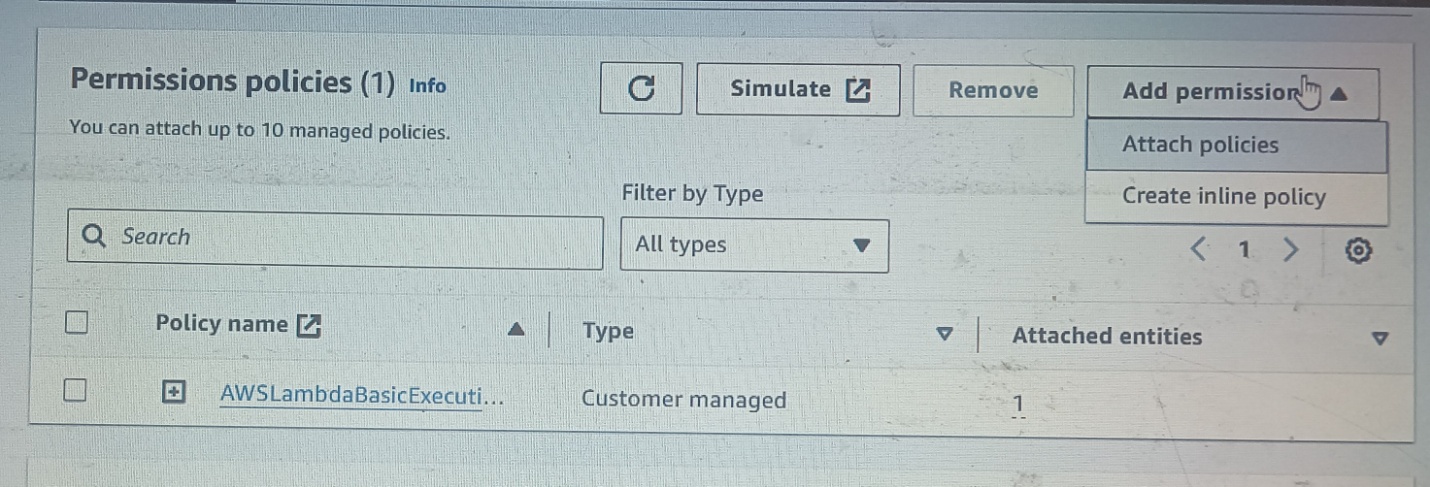
Click on “Poll for messages”. We can see a message . Open that, we can see “Test event message”.**) Refer “Source code” for code.**

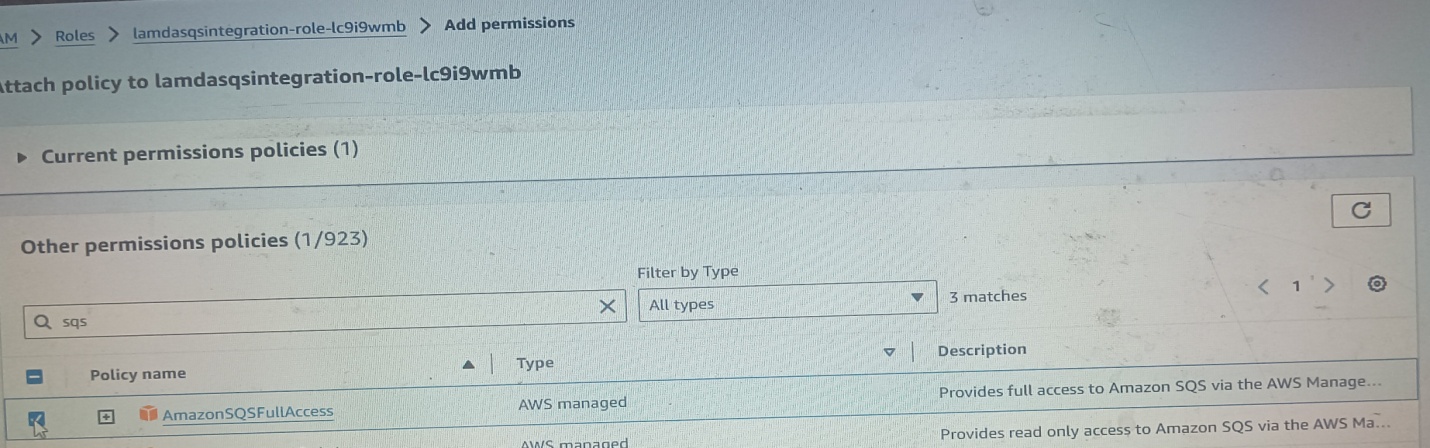




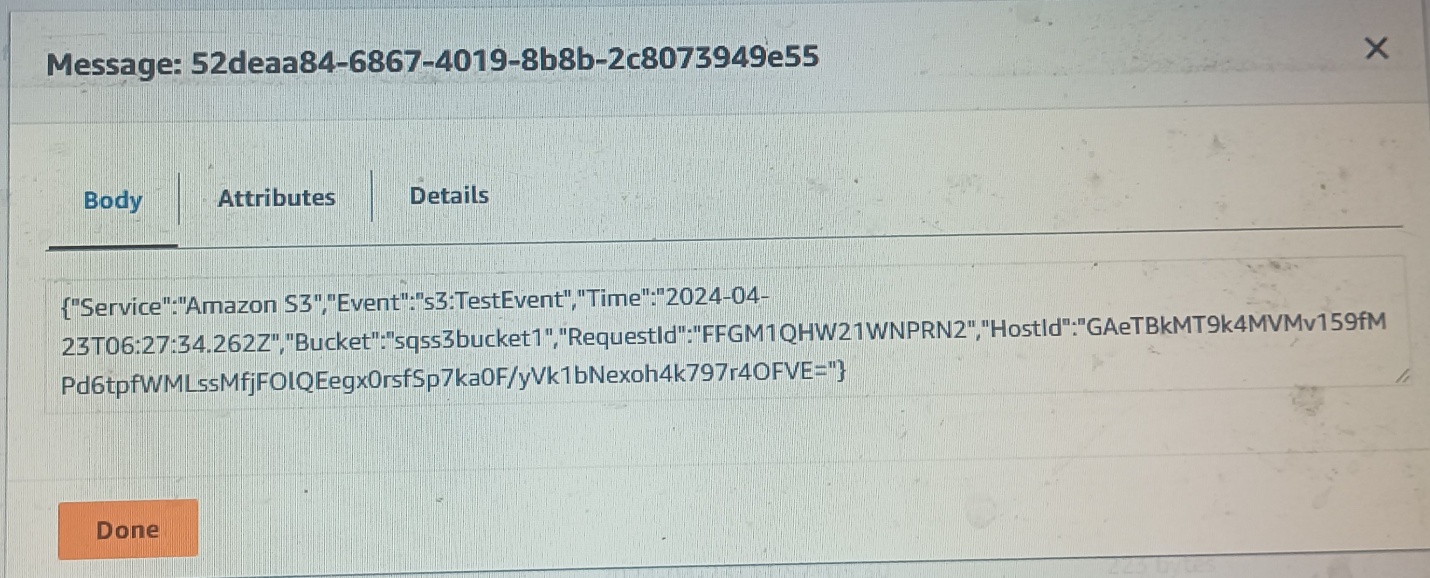








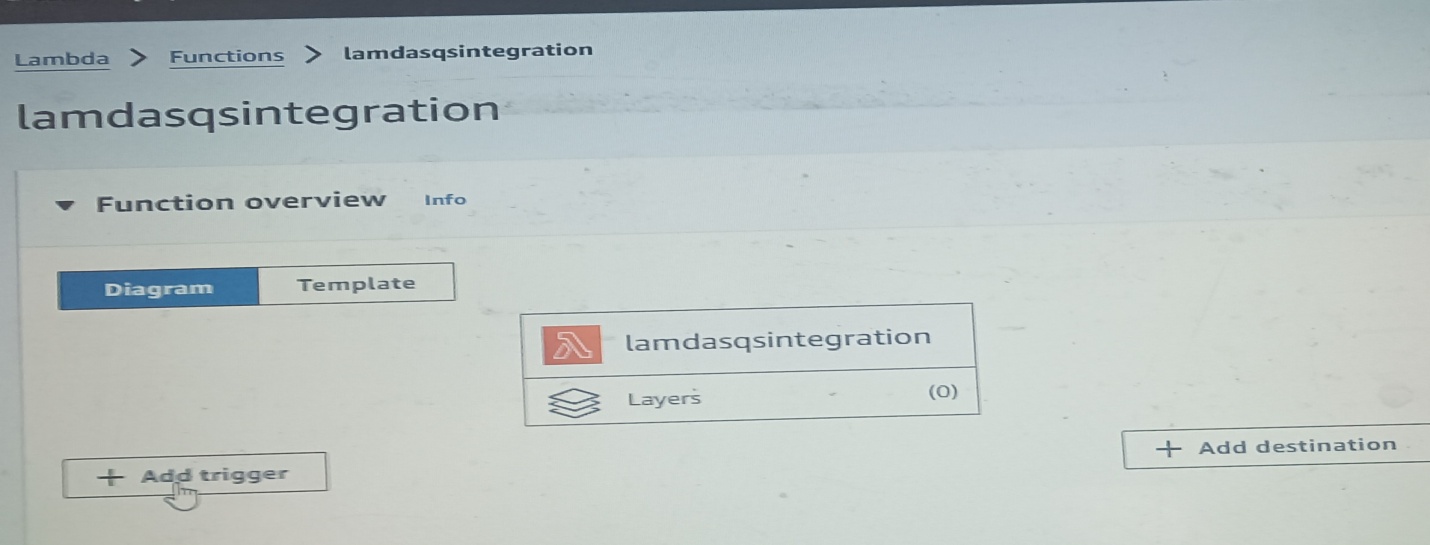
SQS Full Access

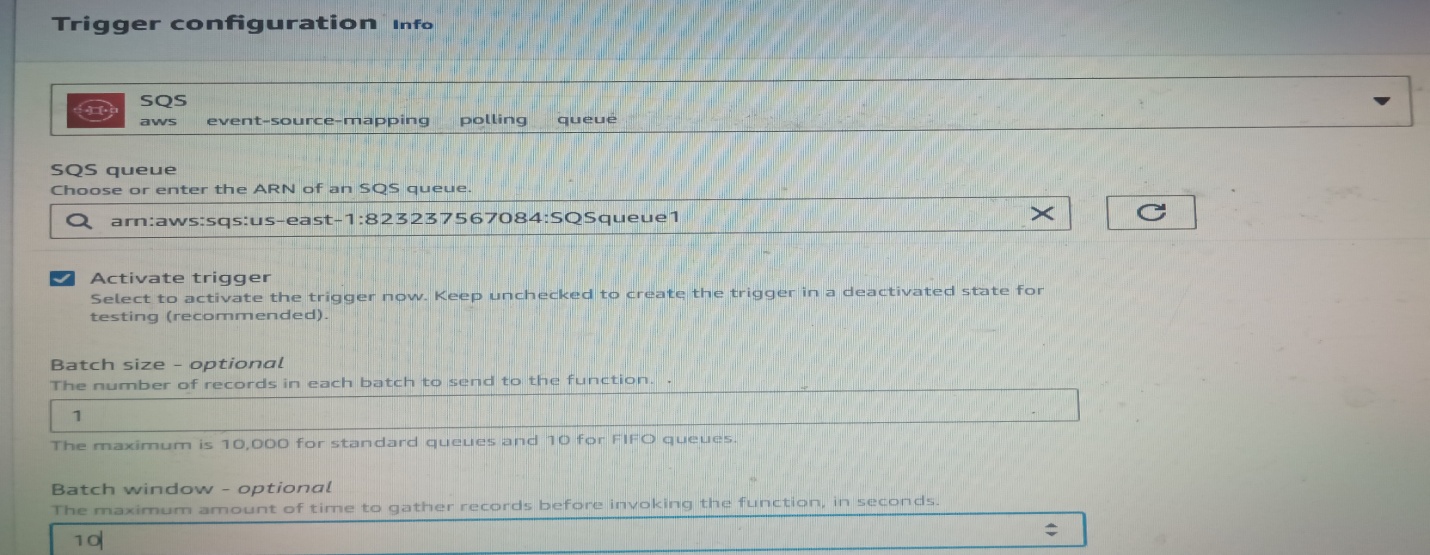


**#Step 5 SQS Trigger For Lambda**

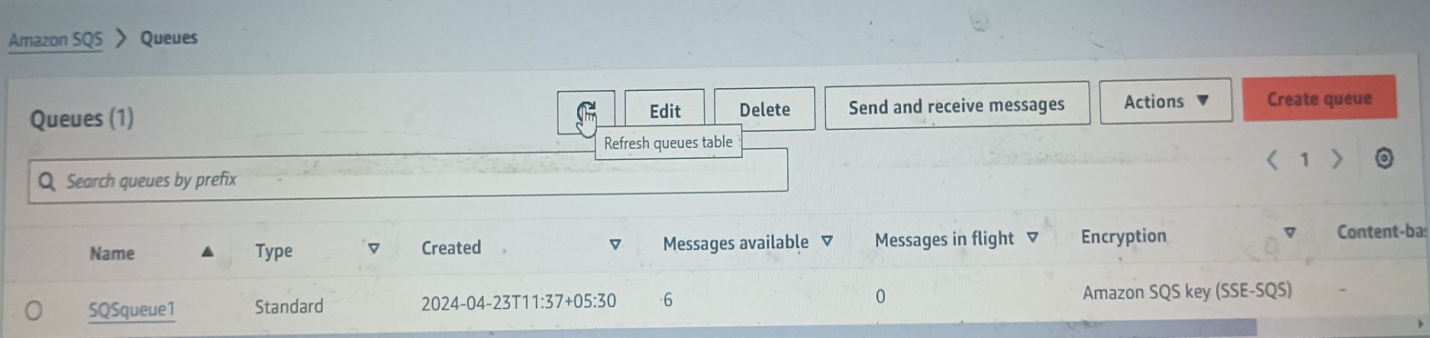
1.Go to Lambda Add trigger choose SQS set batch size (1) batch window(10) add.

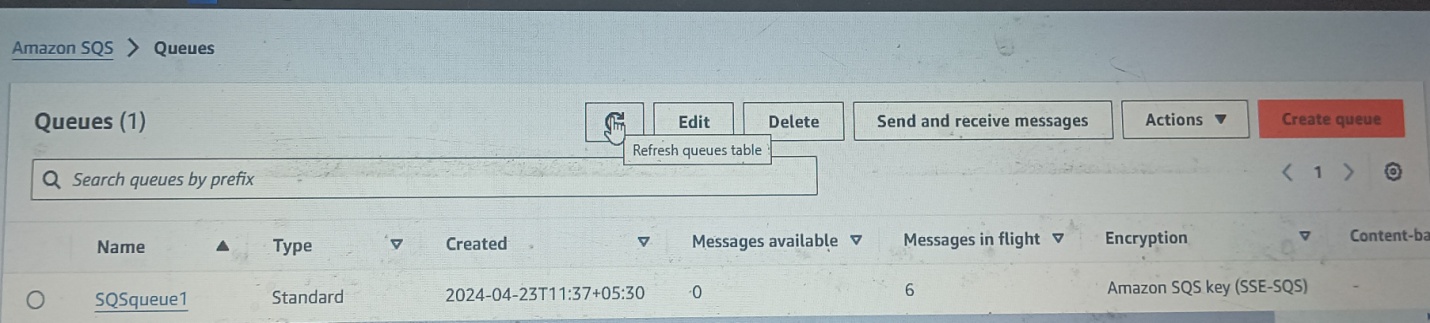
2.Go to S3 and add some files to the bucket that we have created.

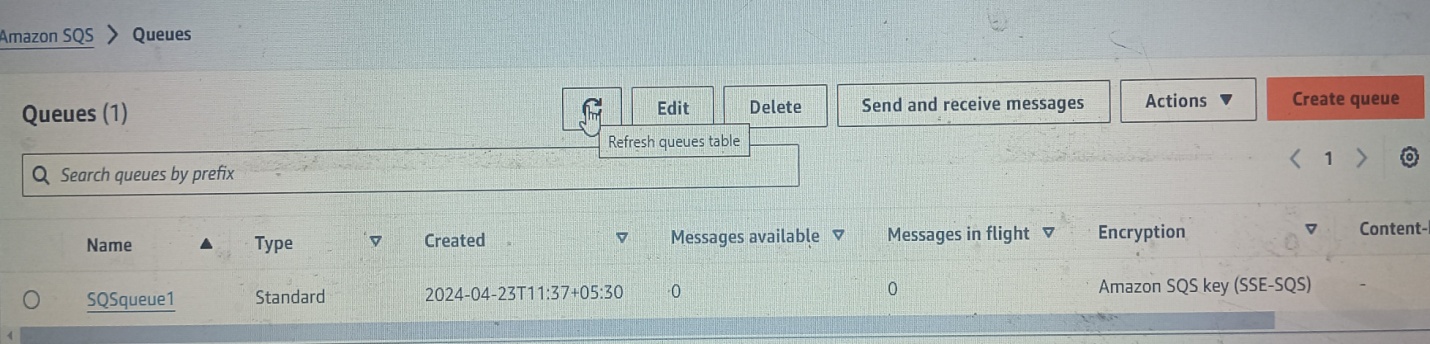




3. Now go to SQS,refresh it. Once the lambda space implication will be set up the message available will be zero. Because lambda will consume all the messages from this queue. (refresh once messages available will be zero and refresh again messages available in flight will also be zero)





**You can monitor through Cloud Watch also. To check cloud watch logs**

Lambda Configuration Monitor View cloudwatch logs Logstream



