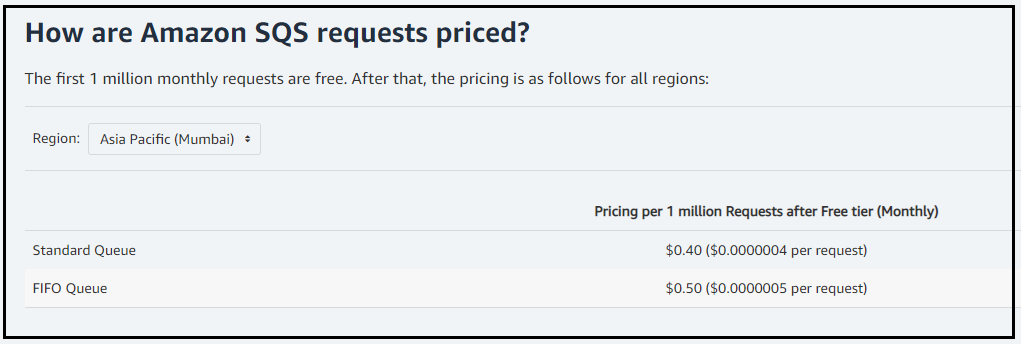
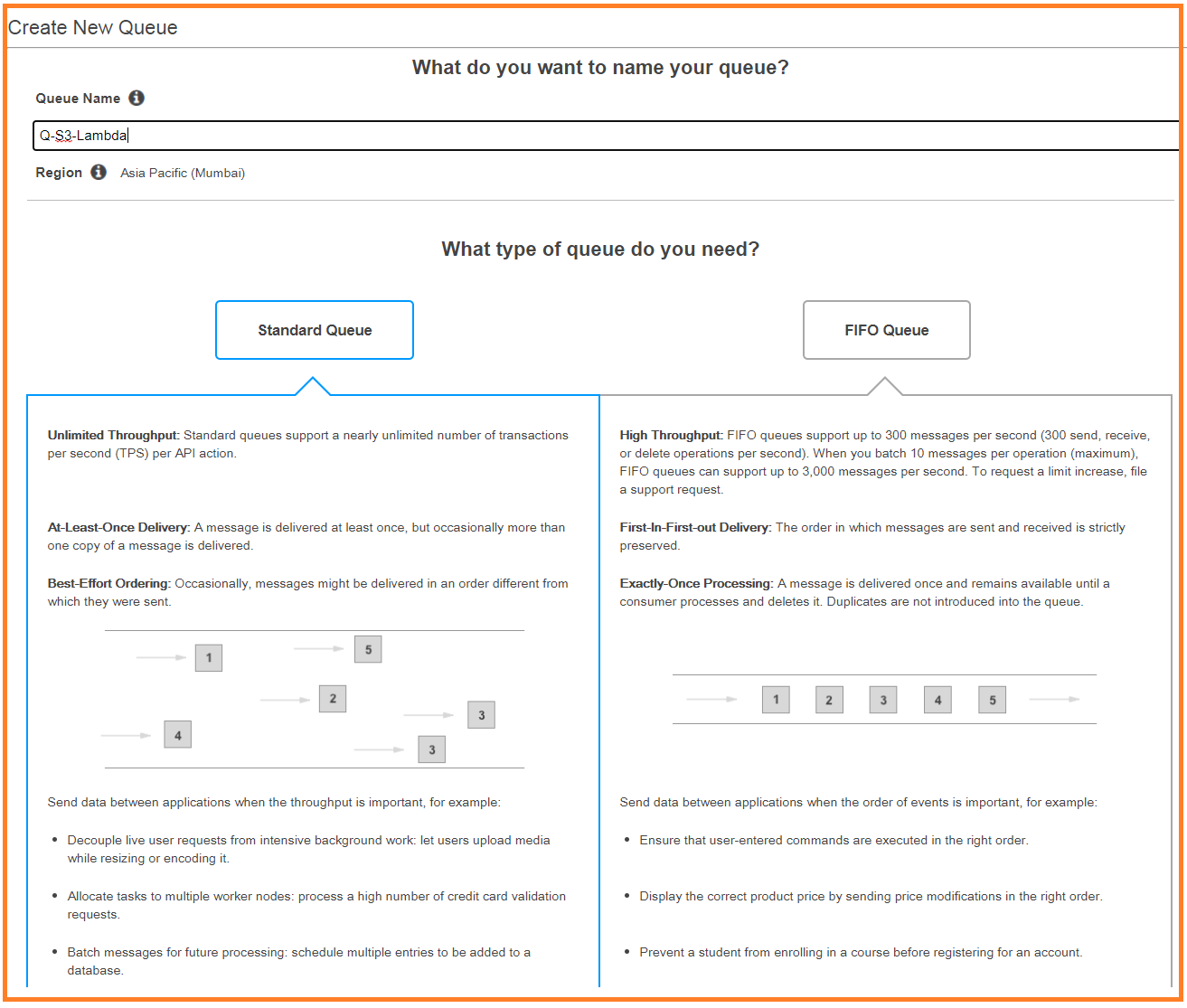
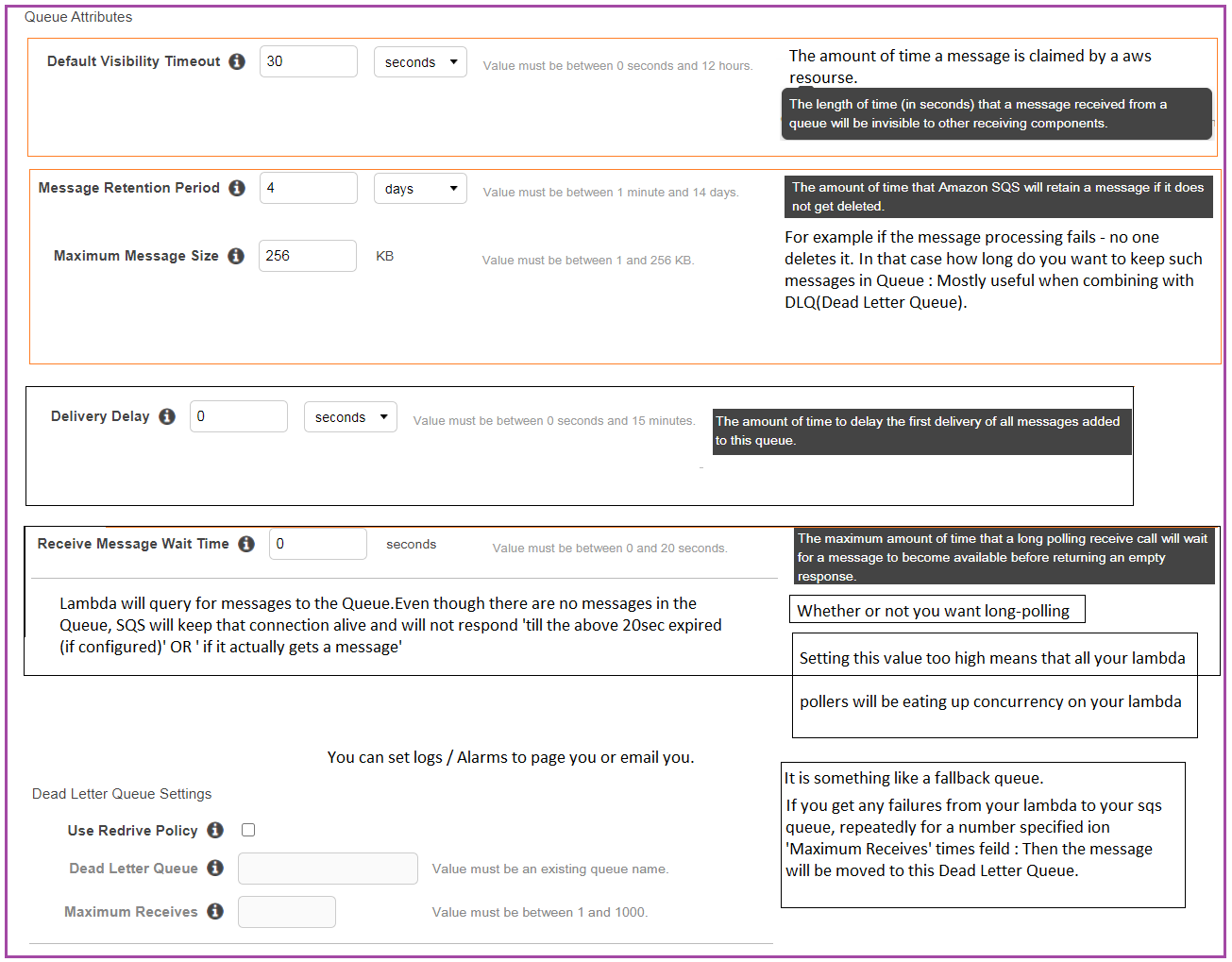
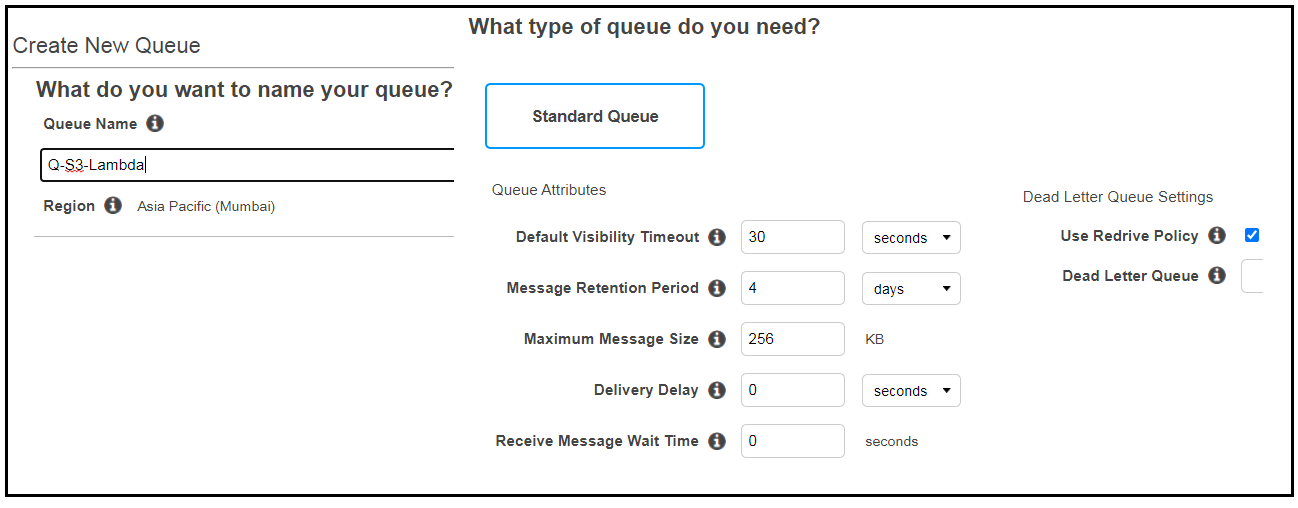
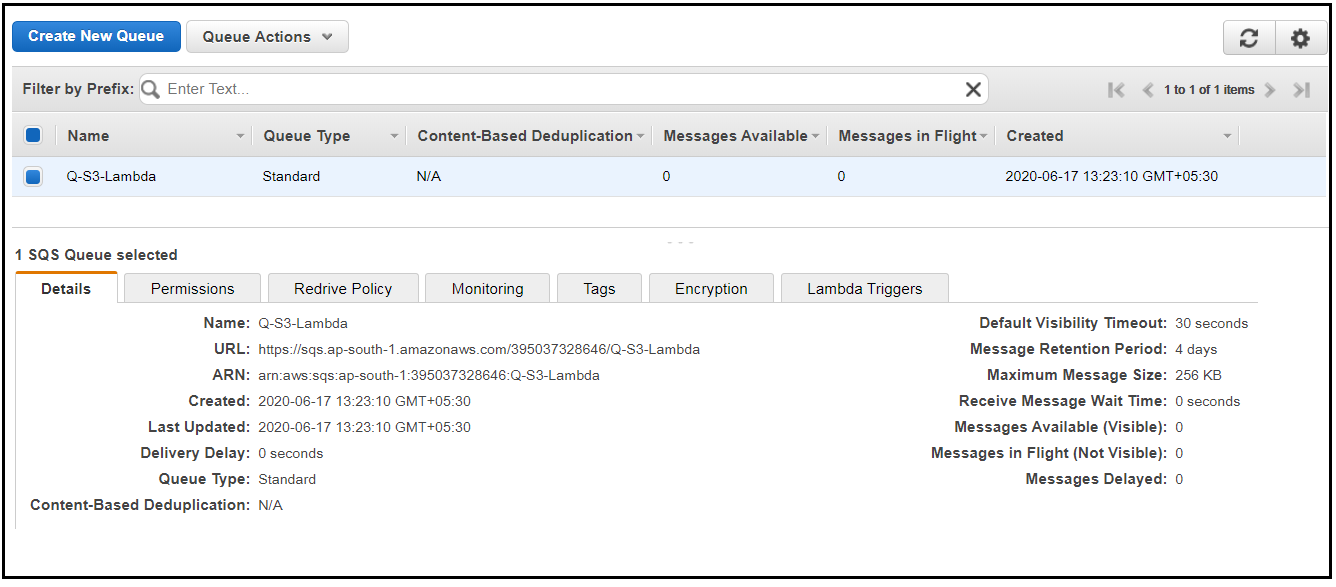
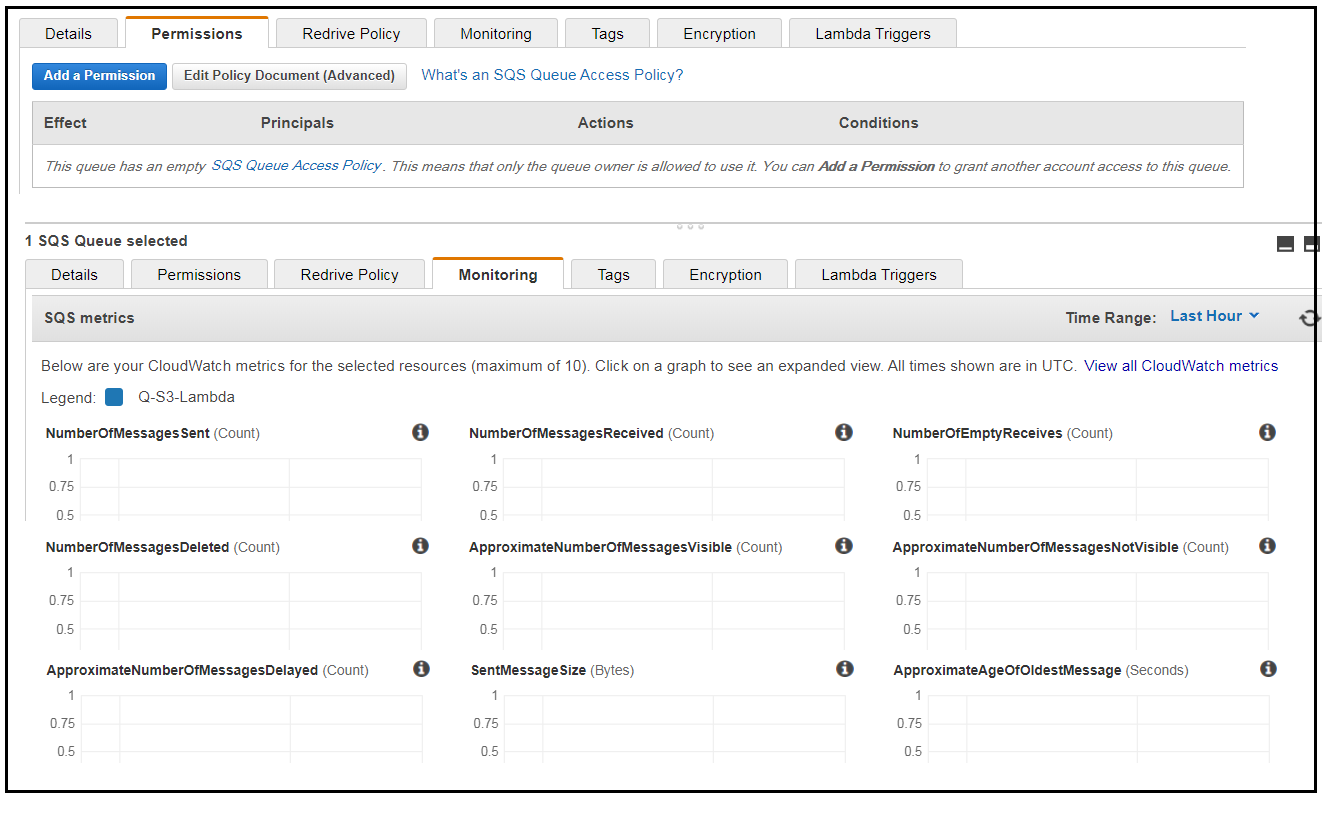
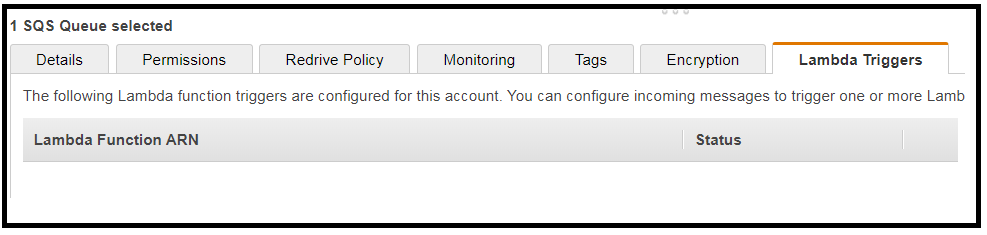
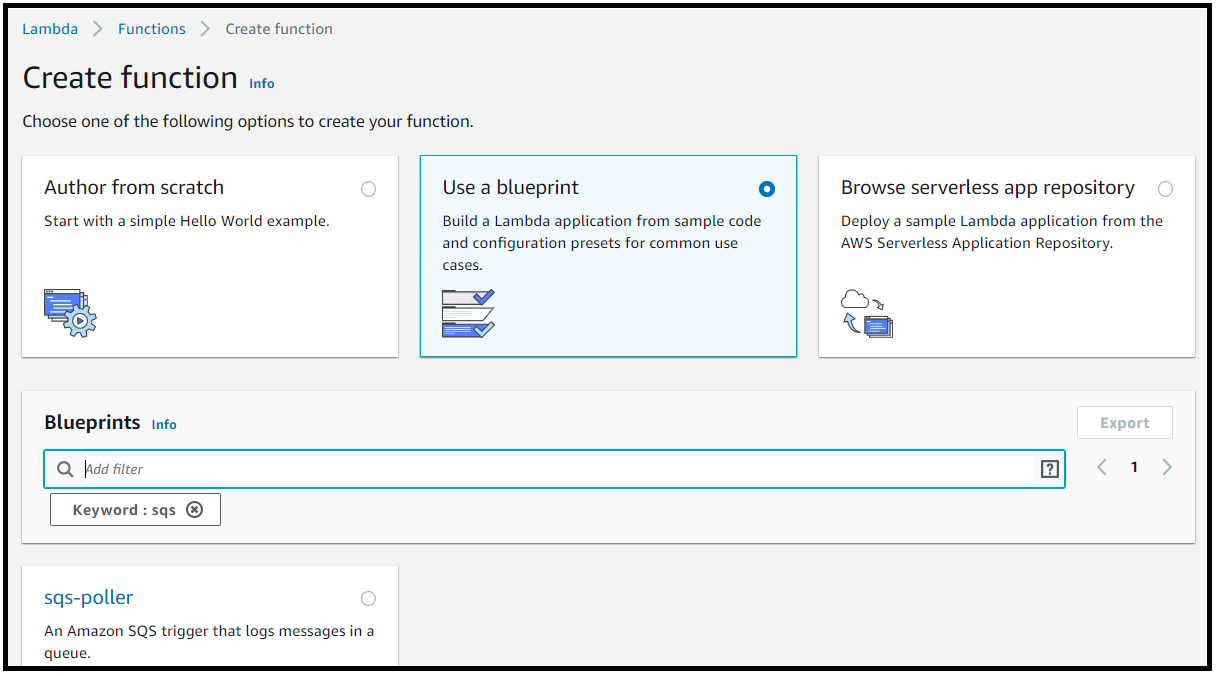
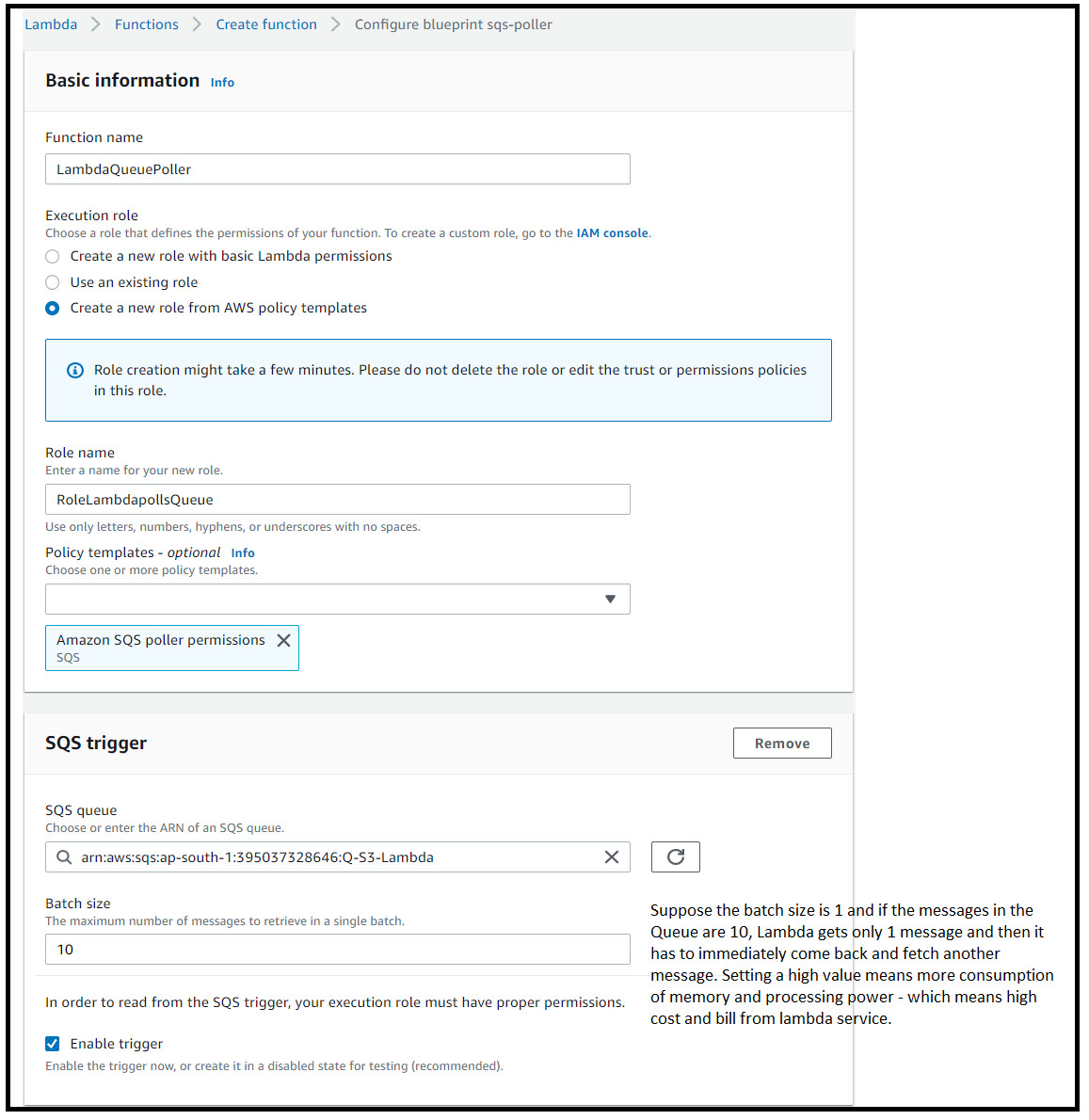
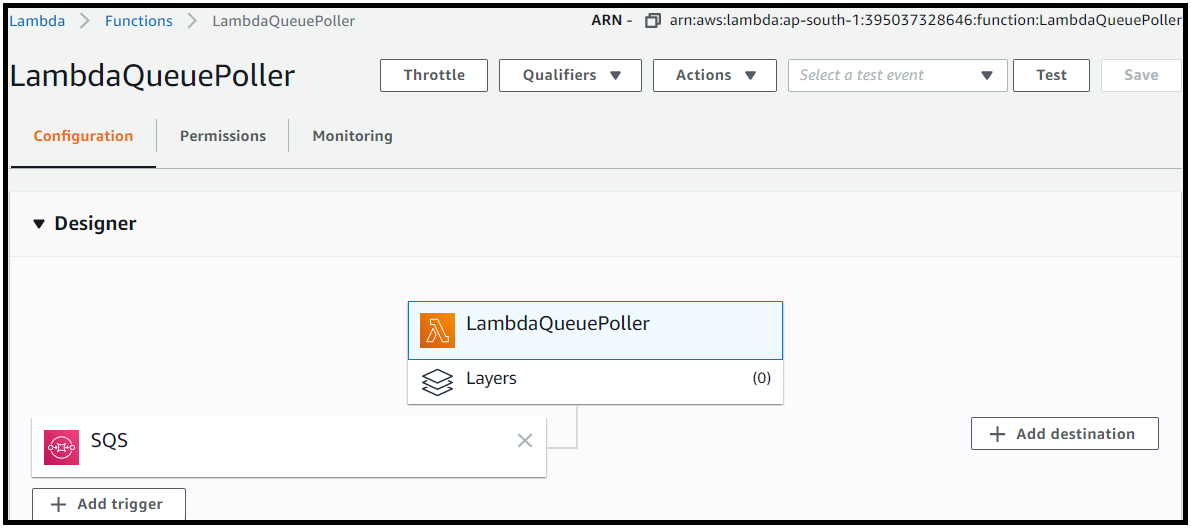
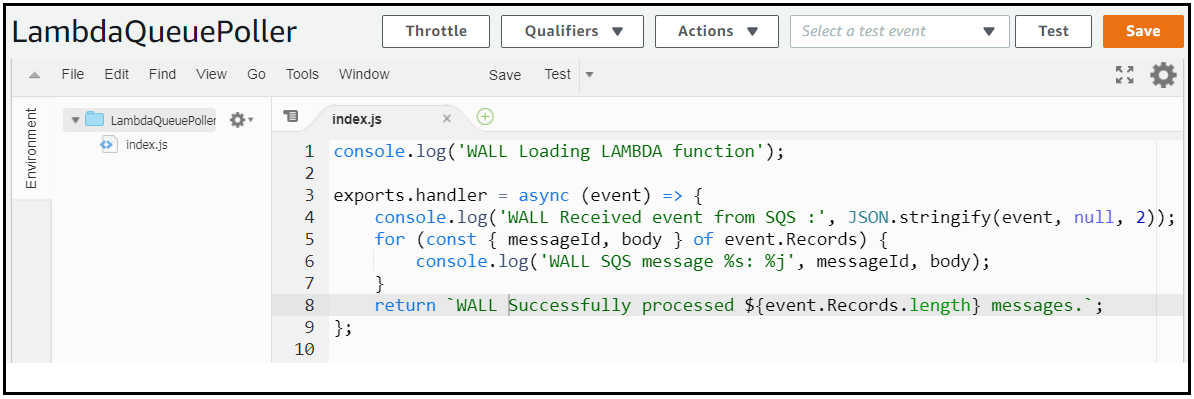
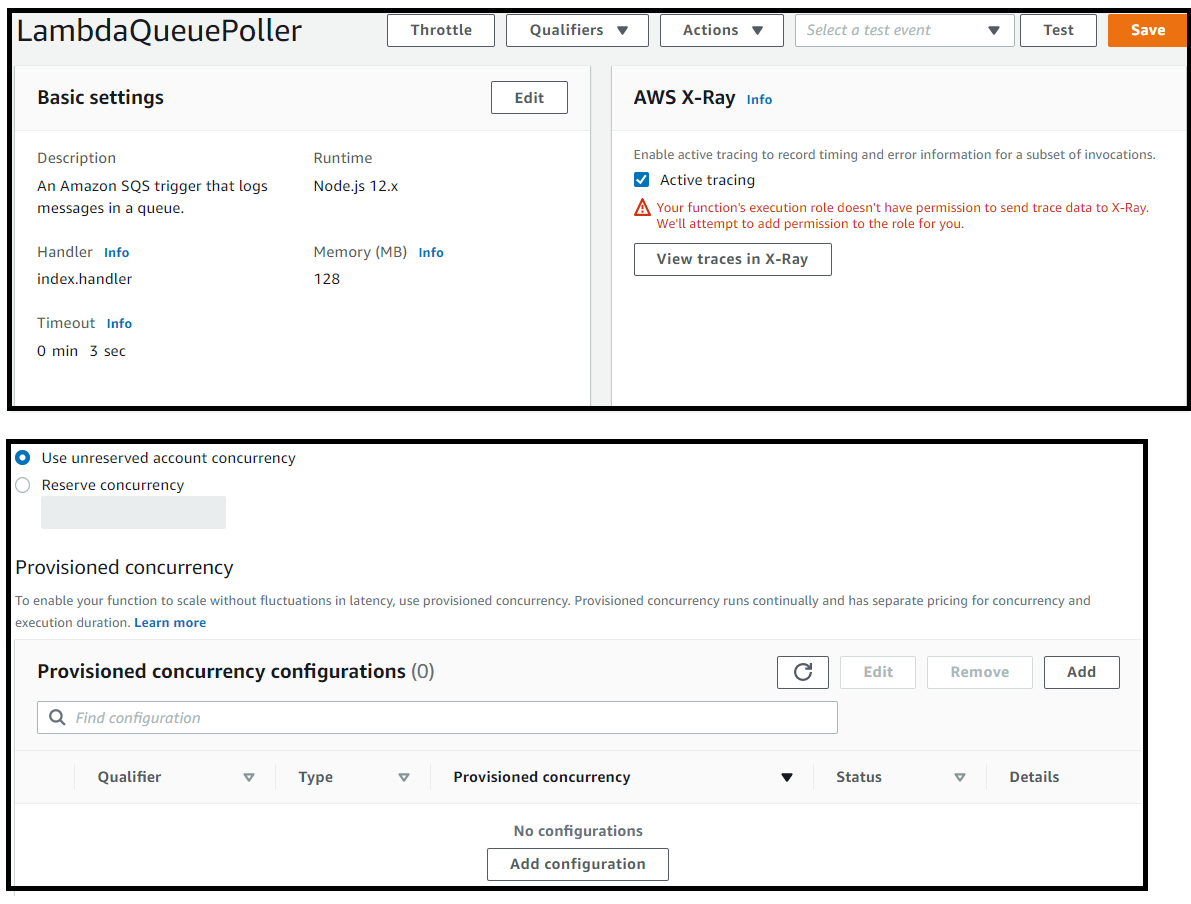
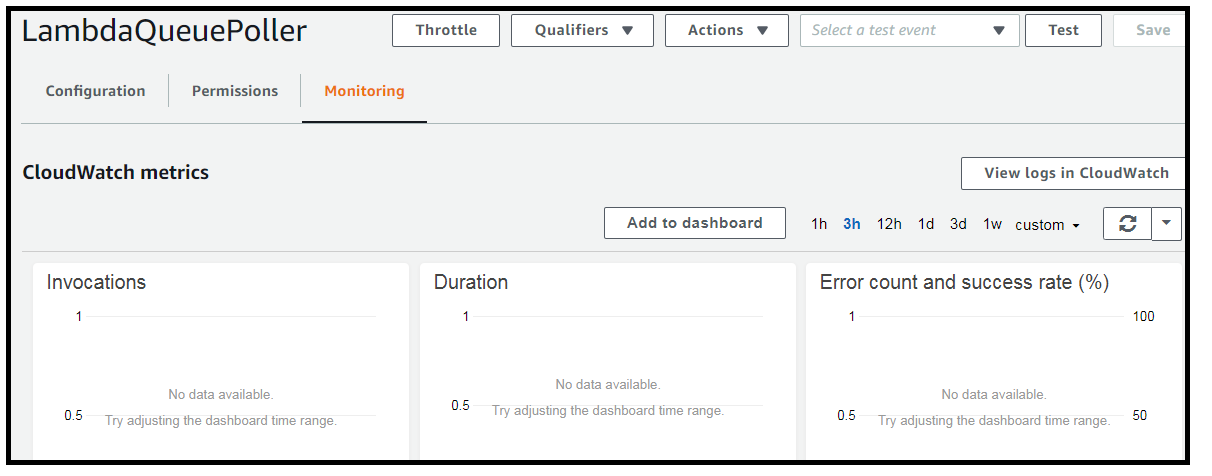
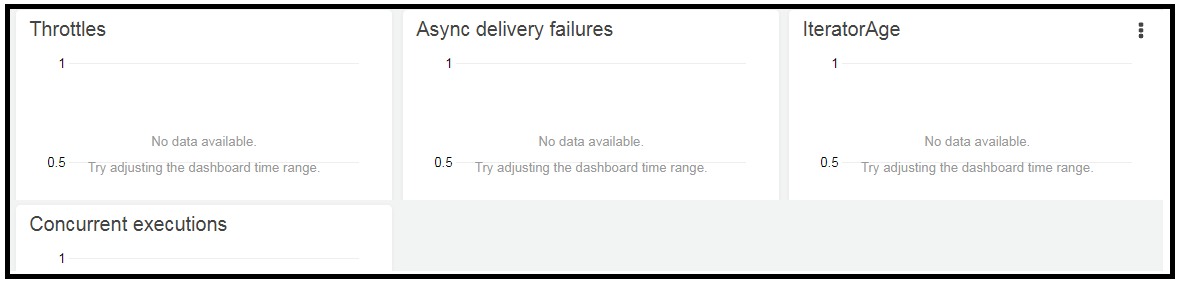
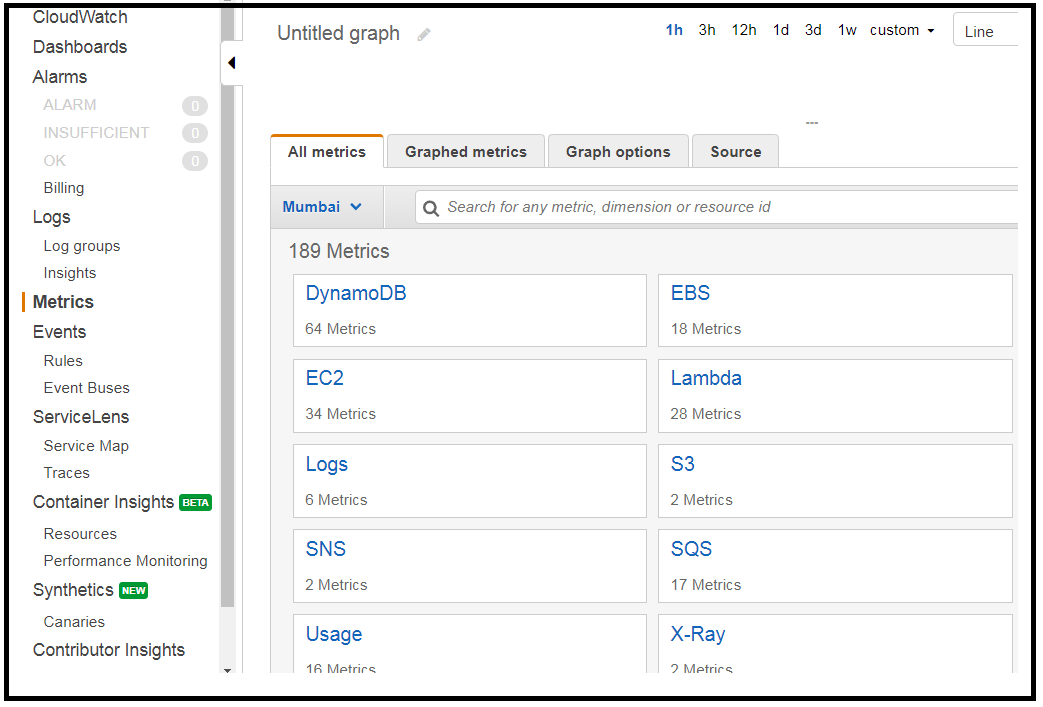
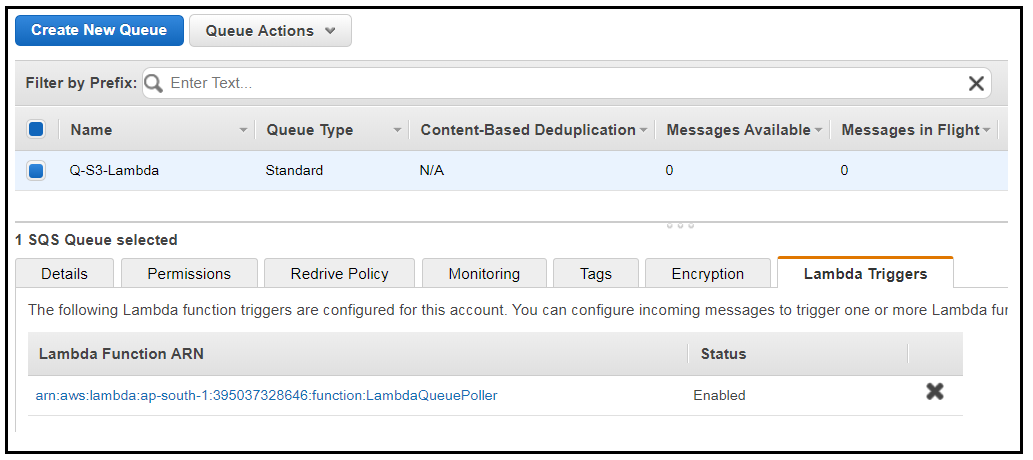
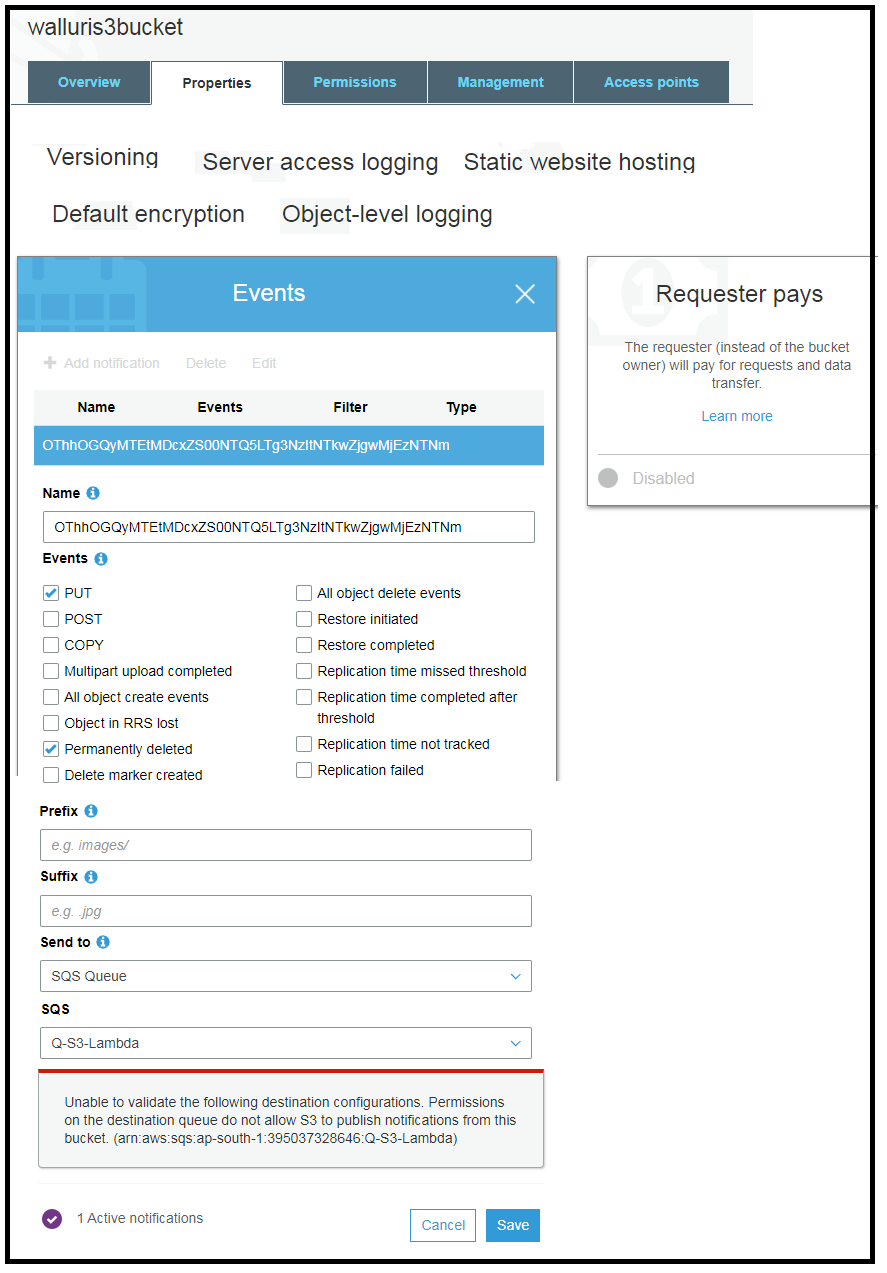
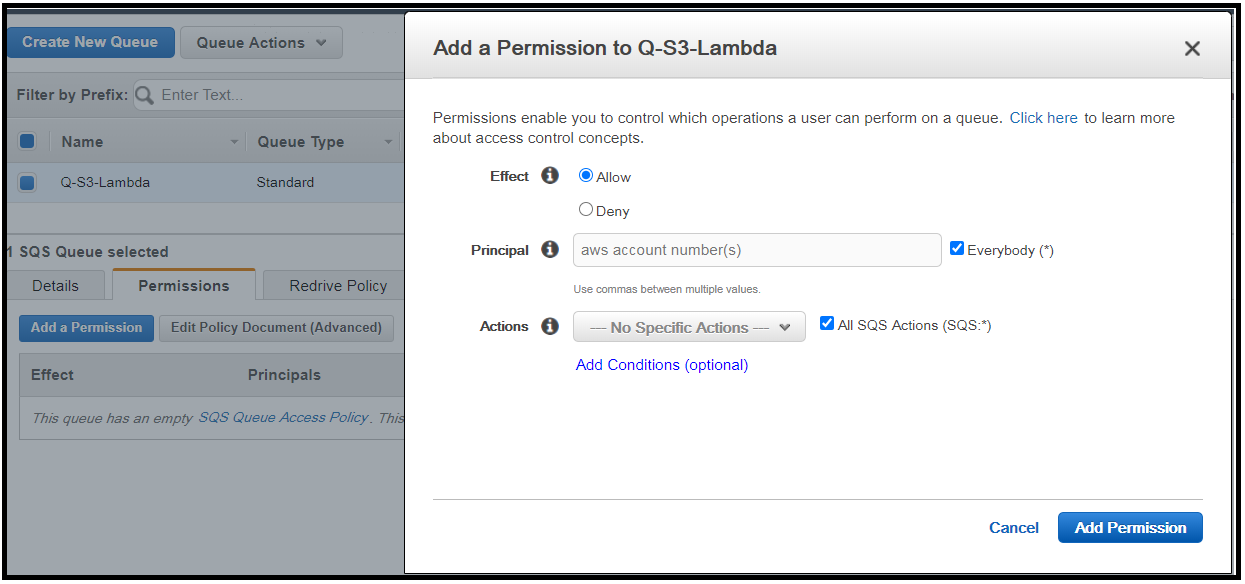
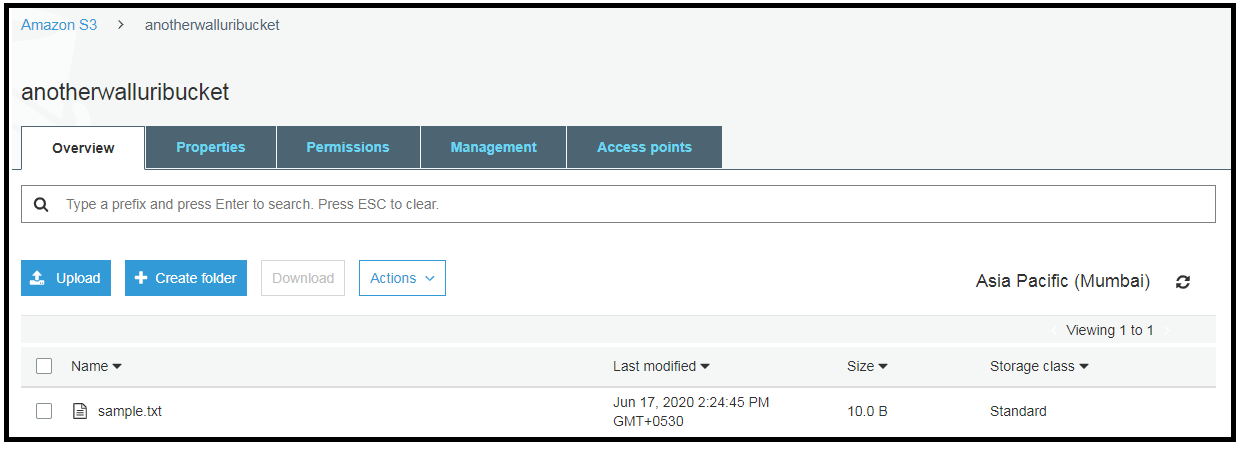
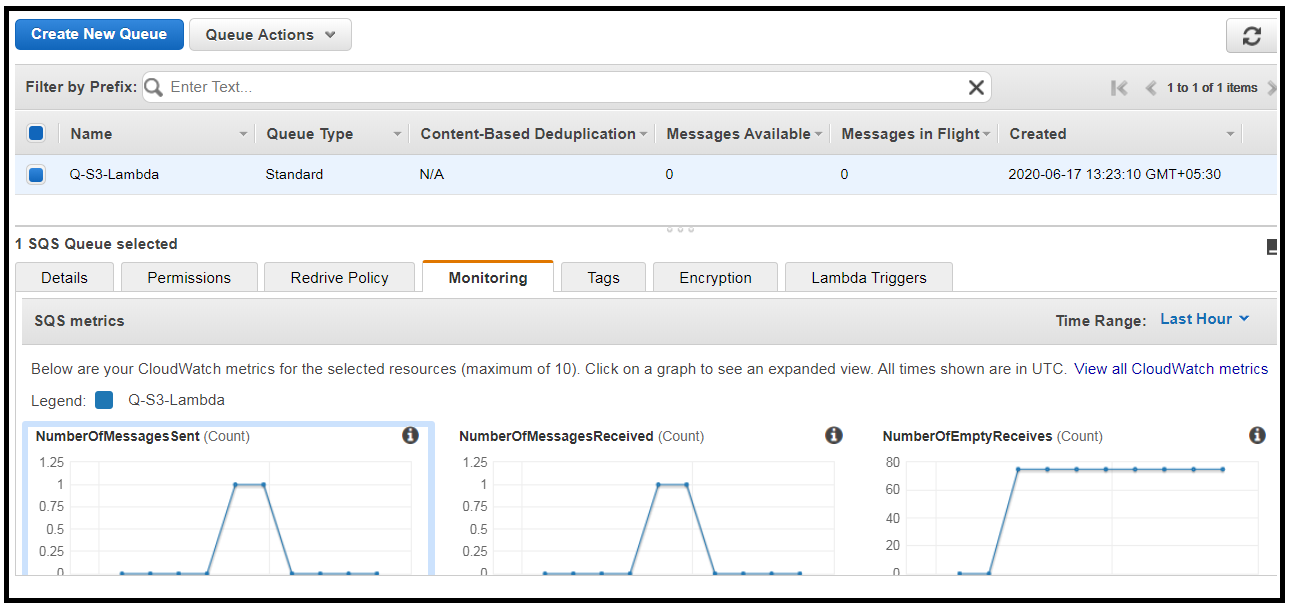
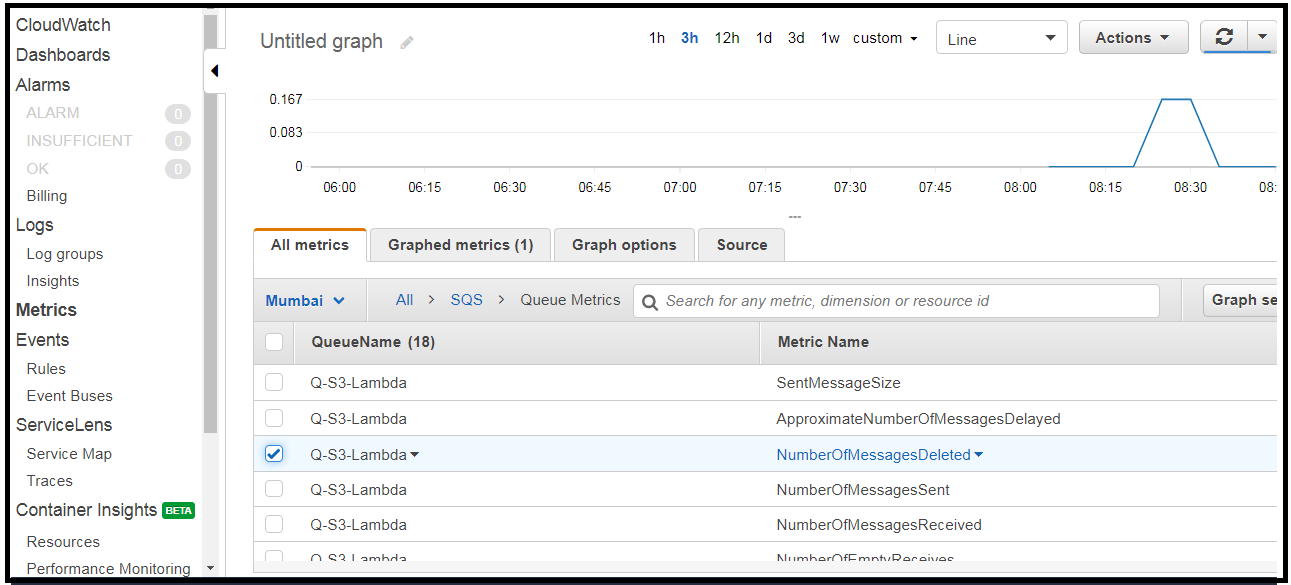
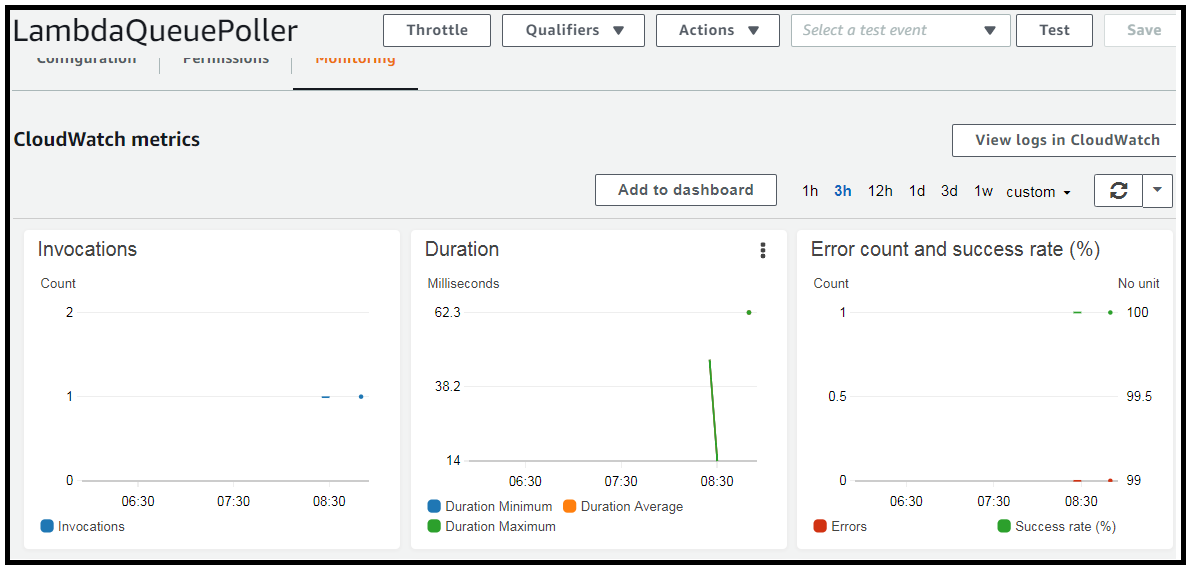
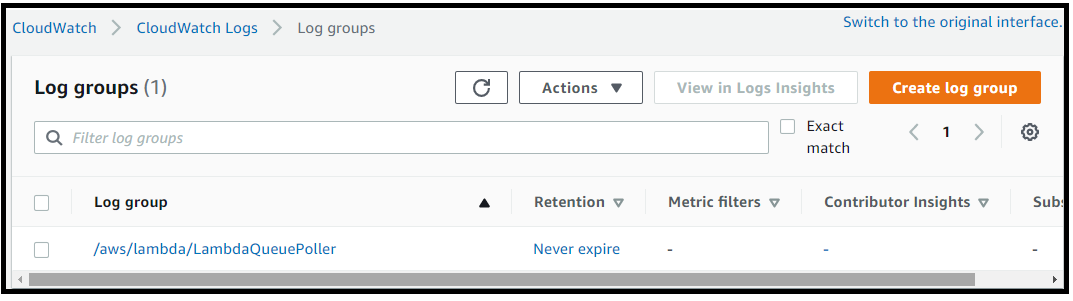
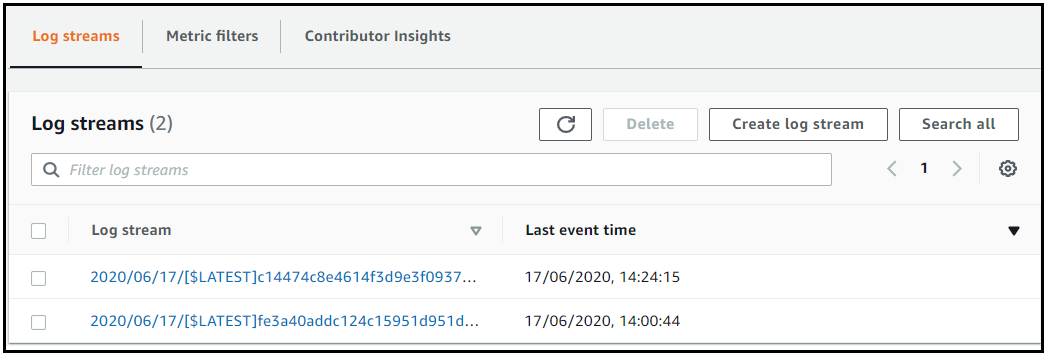
  
  
1. It is the oldest AWS Service.  
  
  
2. What is SQS :   
- It is a web service that gives you access to a message queue, that can used to store messages, While   
waiting for a computer to process them.  
- It is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates, to be consumed by another component.  
- A queue is a temporary repository for messages that are awaiting processing.  
- It is always a pull based system.  
  
Queue : The queue acts as a buffer between the component producing and saving the data, AND the component receiving the data for processing.   
  
Use :  
- Using SQS we can de-couple the components of an application, so they run independently.  
- The queue messages can contain up to 256KB of text in any format.  
- Messages can be kept in the queue from 1 minute to 14 days, with the default retention period being 4 days.  
  
- Visibility timeout (max 12 hours.) : It is the amount of time that the message in invisible in the SQS queue, after a reader picks up the message.   
If the job is processed before the ‘visibility time out’ expires the message will then be deleted from the SQS queue.  
If the job is not processed with in that ‘visibility time out’ period then that message will become again and another reader will process it.   
This could result in the message being delivered twice.  
  
- Any component can retrieve the messages programmatically using the SQS api.  
- The queue resolves issues that arise if the producer is producing work faster than the consumer can process it : For Example we can configure Auto-Scaling groups to monitor the SQS queue And we can start provisioning EC2 instances when a particular limit is reached and VICE-VERSA.  
  
SQS Long Polling : It is a way to retrieve messages from AWS SQS queues. Long polling does not return a response until a message arrives in the message queue OR a long poll times-out. [saves you money]  
SQS Short Polling : This regular short polling returns immediately eve if the message queue being polled is empty. For Ex : Your EC2 instances continuously polls the queue to find out if there is a job, always.  
  
  
3. Types of QUEUES.  
Standard Queues :  
This is the default queue type offered by AWS SQS.  
A Standard queue lets you have *nearly*-*unlimited*  number of transactions per second.  
They guarantee that the messages are delivered at-least once. However occasionally, more than one copy of the message might be delivered out-of-order.  
Standard queues do not have any kind of order to them.  
They provide best-effort ordering which ensures that messages are generally delivered in the same   
order as they are sent.  
  
FIFO Queues :   
The most important feature of this queue type is FIFO delivery + And exactly once-processing.  
The order in which the messages are received and sent is strictly preserved.  
The message is delivered only once and remains available until a consumer processes the message and deletes it.  
Duplicates are not introduced into the queue.  
FIFO Queues are limited to 300 Transactions per second.  
FIFO queues also support Message Groups ?

4. CREATING A QUEUE  
  
  
  
  
5. LAB : Note down what files have been uploaded to S3 using SQS and Lambda and CloudWatch  
Upload a File to S3 and then We have an PUT Event configured on a bucket.  
This Event will be put on a SQS Queue.  
The lambda will always poll the SQS queue and perform duplicate action like logging into the console- which means cloud-watch will have those logs.  
  
5.1 Queue   
Create a SQS Standard-Queue.  
  
  
Queue Console – part 1  
  
  
Queue Console – part 2  
  
  
Queue Console – part 3  
Notice that there are no Lambda-Triggers which will poll this queue.  
  
  
5.2 CREATE A LAMBDA FUNCTION  
Use a blueprint lambda function which is used to poll the SQS Queue.  
  
  
Add the SQS Trigger by giving the SQS name.  
  
  
Lambda Console after creating function.   
  
  
Lambda code + Click Save.  
  
Other Lambda settings.  
  
  
  
  
5.3 Cloud Watch and metrics.

|  |
| --- |
|  |

  
  
5.4   


5.5   
  
  
Grant rights as to who can access SQS Queue  
  
  
  
Queue Console Monitoring.  
  
Cloud watch metrics screenshot.  
  
  
Lambda Console – Monitoring screenshot.  
  
  
CloudWatch – Log Groups (There is only one which is generated by Lambda, but not the SQS queue)  
  
Cloud watch log streams.  


Cloud Watch Events in a Log stream.  
