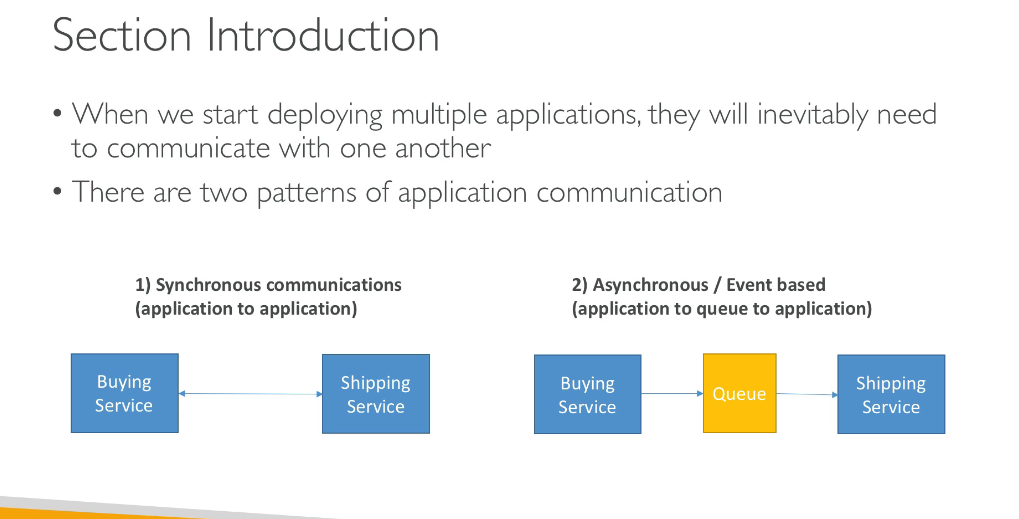
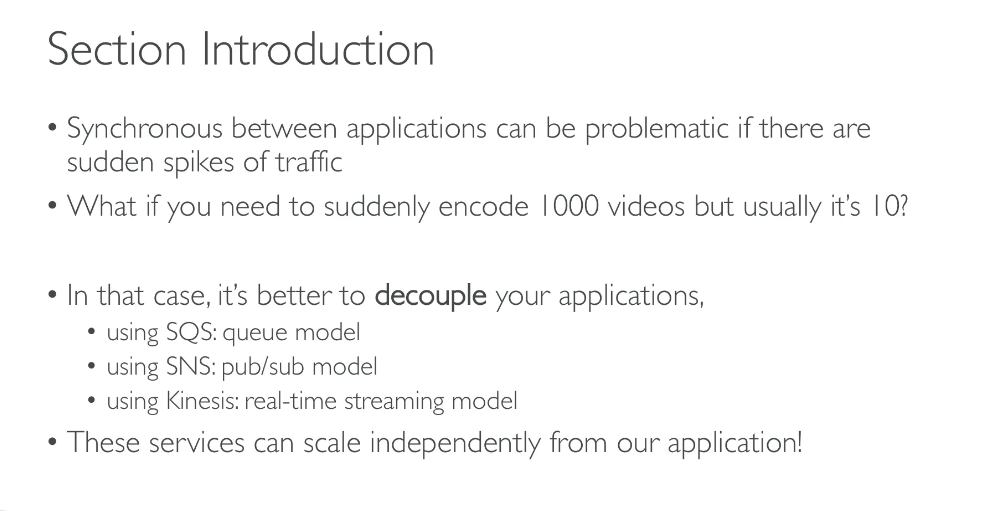
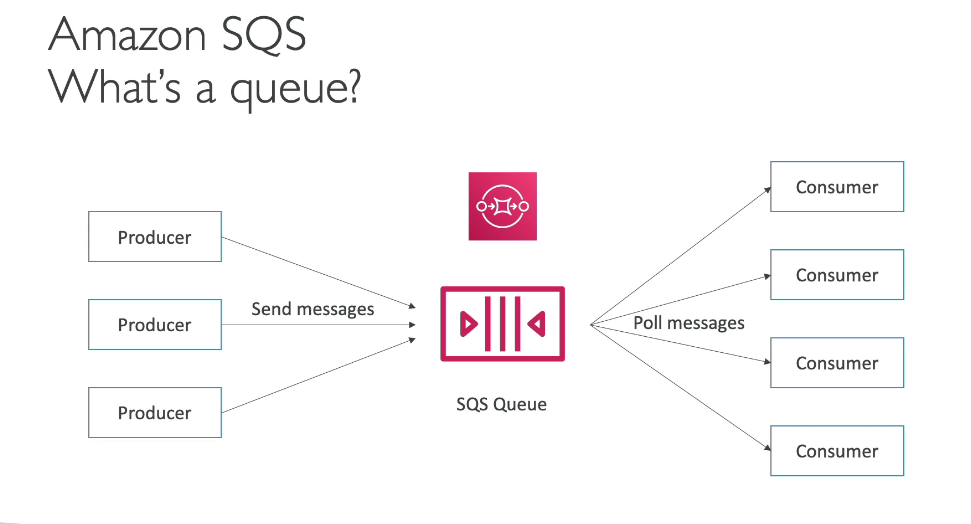
**MESSAGING**

****

****

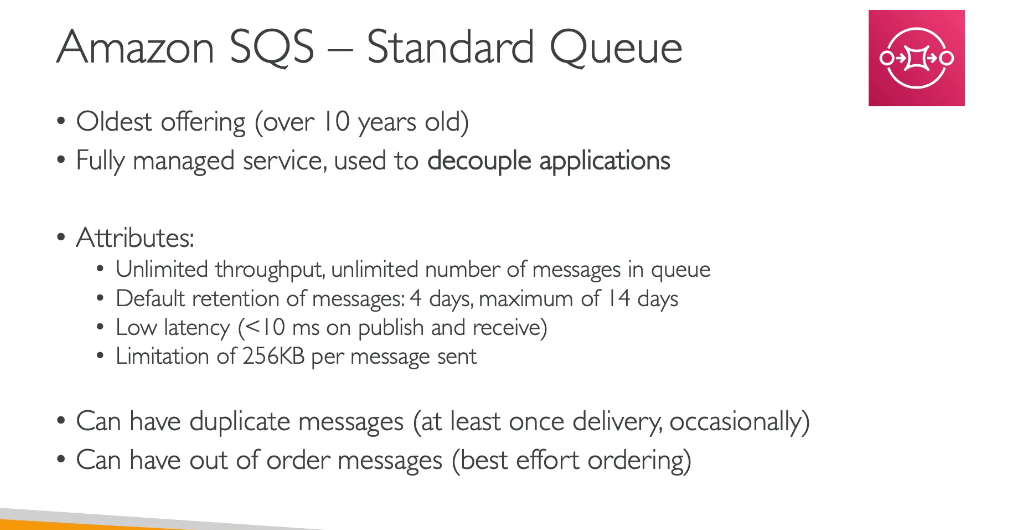
**SQS**

A queue service acts as a buffer between our consumer which can be a buying service and a consumer which canbe a shipping service.So a queue will decople our bith the applications.Our producer will send message to the queue and there can e many producers and our consumer will poll the queue for the messages an dthere can be many consumer consuming messages from th e queue.

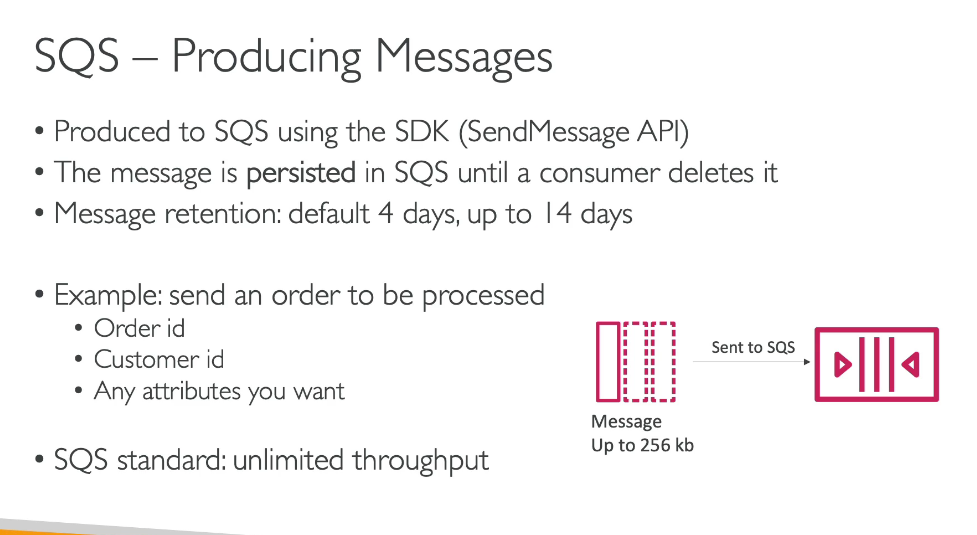


Sqs is the oldest offering of aws and is fully managed by aws.

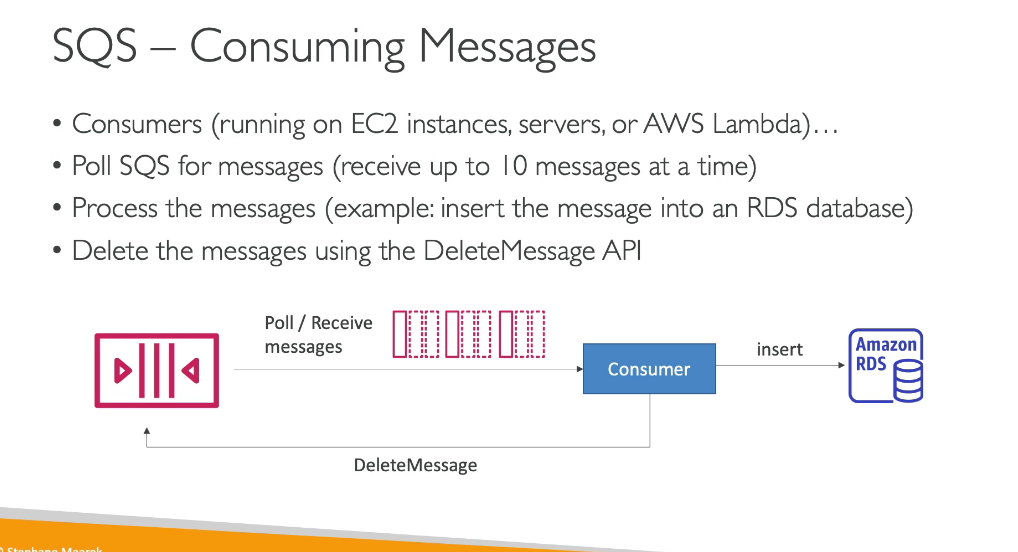
Attributes:



**PRODUCERS**

****

**CONSUMERS**

****

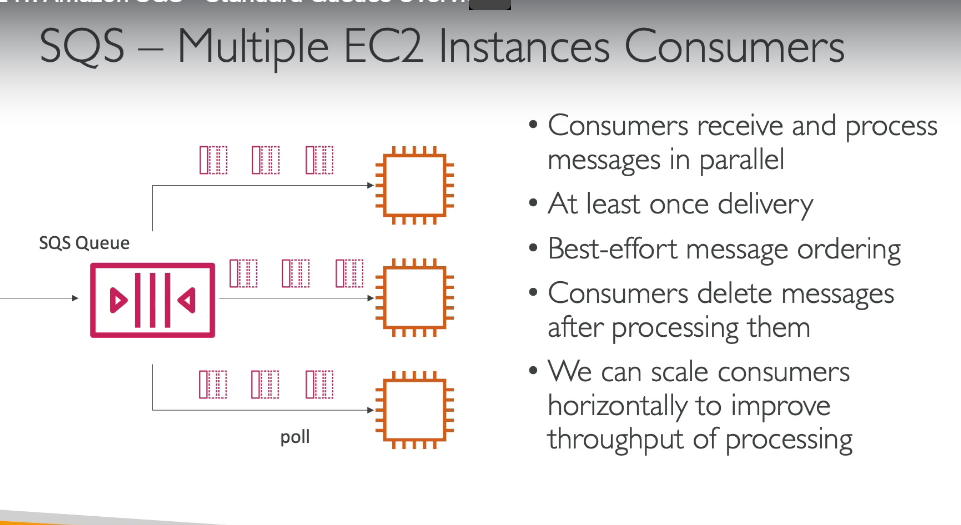
An sqs queue canhave multiple consumers who can receive and process them in parallel.

Here we have three ec2 instances as consumers so each ec2 instance will consume a different set of messages.

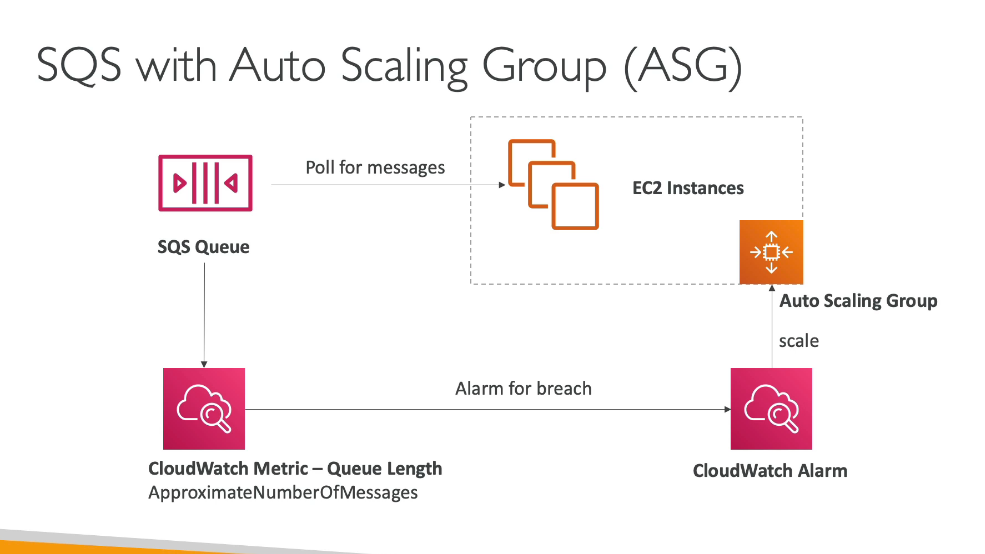
**At least once delivery:**if a message is not processed by one consumer it will be processed by the other consumer.

And thus it is called as best effort message ordering.

Once a consumer has processed a message it needs to delete it from the queue so that other consumer do not process the same msg again.



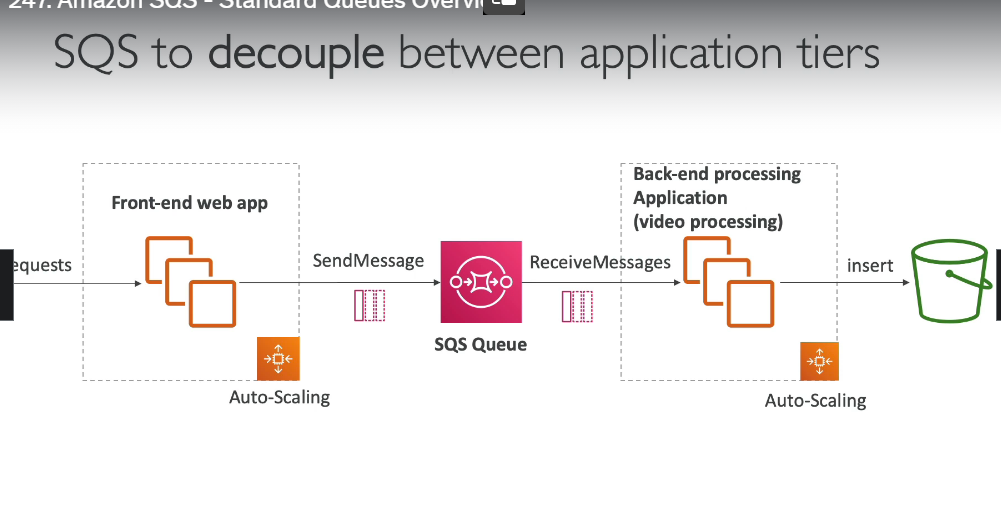
**SQS WITH ASG**

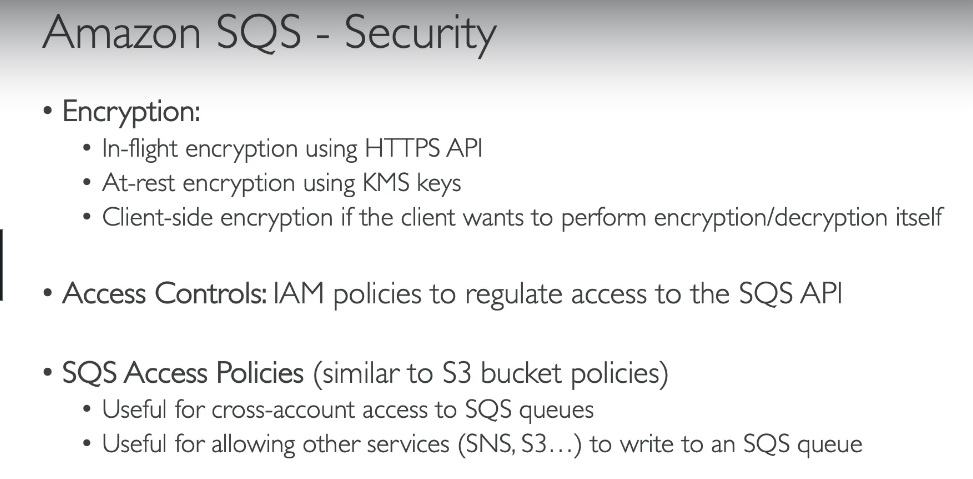
****

We wll setup our consumers in an asg and there will be a queue length metric to decide when the asg has to increase or decrease the no of instances.Suppose when the no of messages increases when there are huge amount of orders then the cloudwatchmetric will sent an alarm of breach to the cloudwatch alarm which in turn will set up the asg to increase the number of ec2 instances.

**Sqs is used to decouple between application tiers.**

Suppose w e have awebsite for processing videos and suppose we have a huge application to do this work.

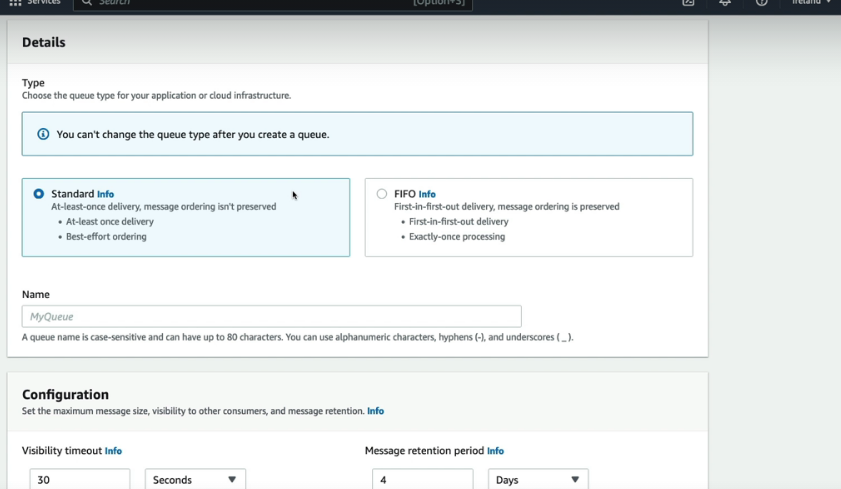




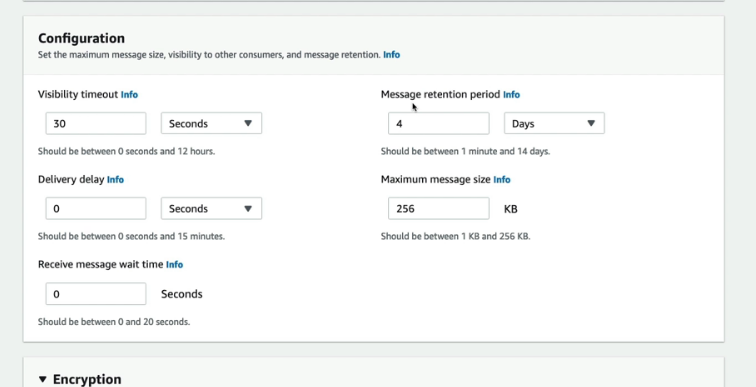
**STANDARD QUEUE HANDS ON**

**1.firts of all we will create a new queue**

2.Next we will sleect the kind of queue that we want to make.May it be a fifo or a standard queue.And we gievthe queue a name according to our need.



Next we need to setup the various configuration settings



5.Next we need to specify how w enat data in our queue to be encrypted.

Graphical user interface, text, application, email

Description automatically generated

6.then we need to specify the access policy in which we tell who can send and read messages from the queue.

Graphical user interface, text, application, email

Description automatically generated

7.Now our queue is createdGraphical user interface, application

Description automatically generated

Then we can check by send and receive button up there on th e screen .

We can edit the queue and edit all the configurations that we had made in the queue.

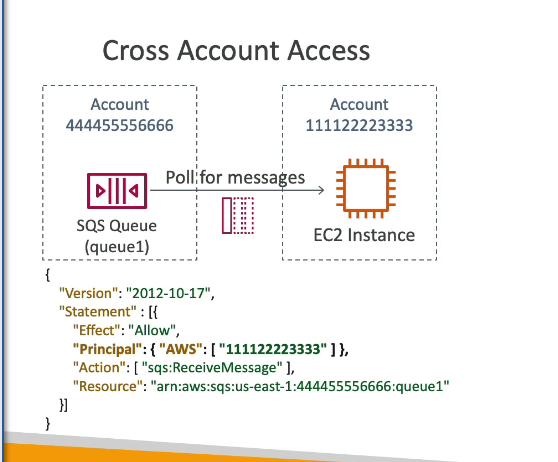
We can  **purge**  the queue in order to delete all the message sin the queue.

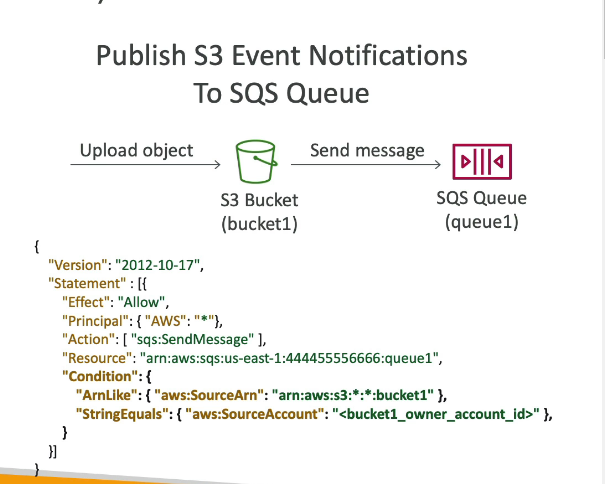
We can check the queue monitoring and in that we can check how longa amessage will remain in a queue and when it can be last accessed.

Also we have the access which tells us the access policy of the queue.

**SQS QUEUE ACCESS POLICY**

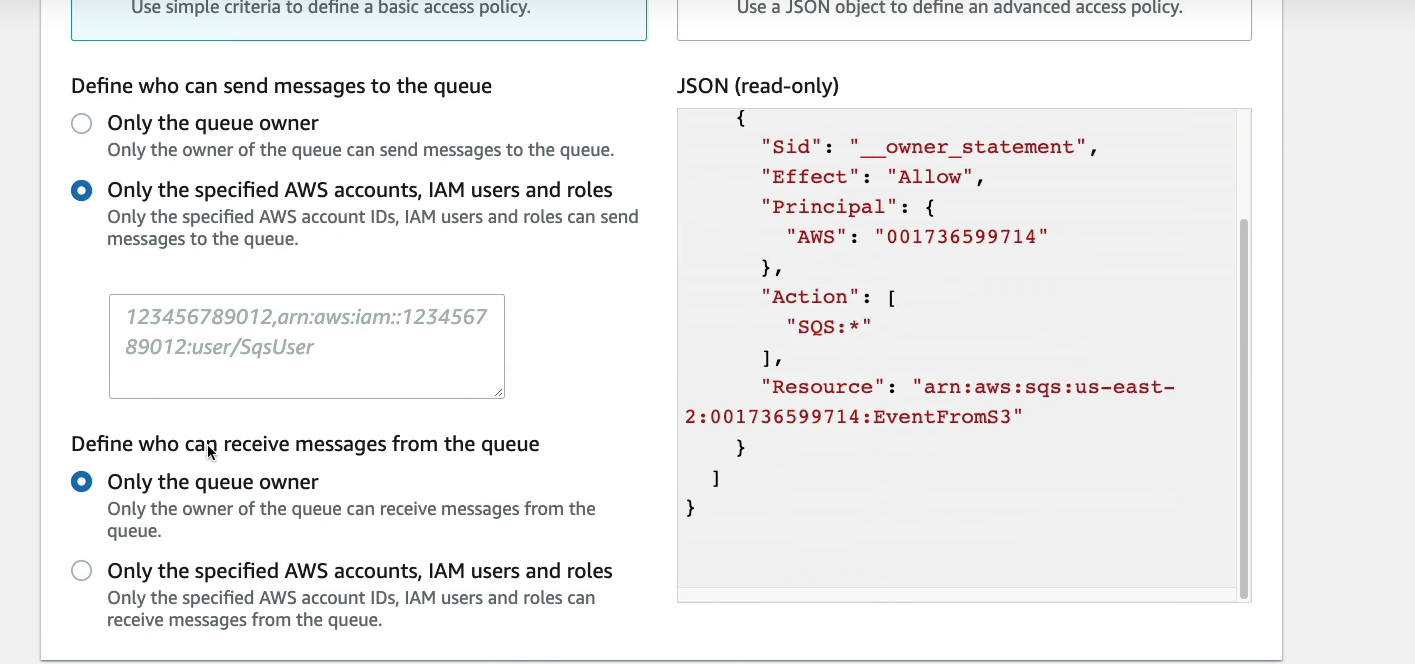
When we want an ec2 instance in another aws account to access our sqs queue in oen account then we need to setup the access policy of our queueu to allow the ec2 instance in another account to access it.



**What is needed to write a message to an sqs from s3 event notification of another account**

In this case we will need to allow our queue to access message from that s3 by specifying the arn and the source account number of the account from which it can allow the access of an event notification.

we create a new queue and in it we specify the access policy as can be seen below:



But in the above an amazon s3 cannot be specified so we go for the advanced policy

Graphical user interface, text, application, email

Description automatically generated

We can write our own rule on who can access our queue.

2.next step is to create an s3 bucket and setup event notifications on it.

When the bucket is created then go to the properties of the bucket an dgo to the event notifications.

Graphical user interface, text, application, email

Description automatically generated

And now create an event notification

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

We choose a destination as to where the event notification be sent to.

Graphical user interface, text, application, email

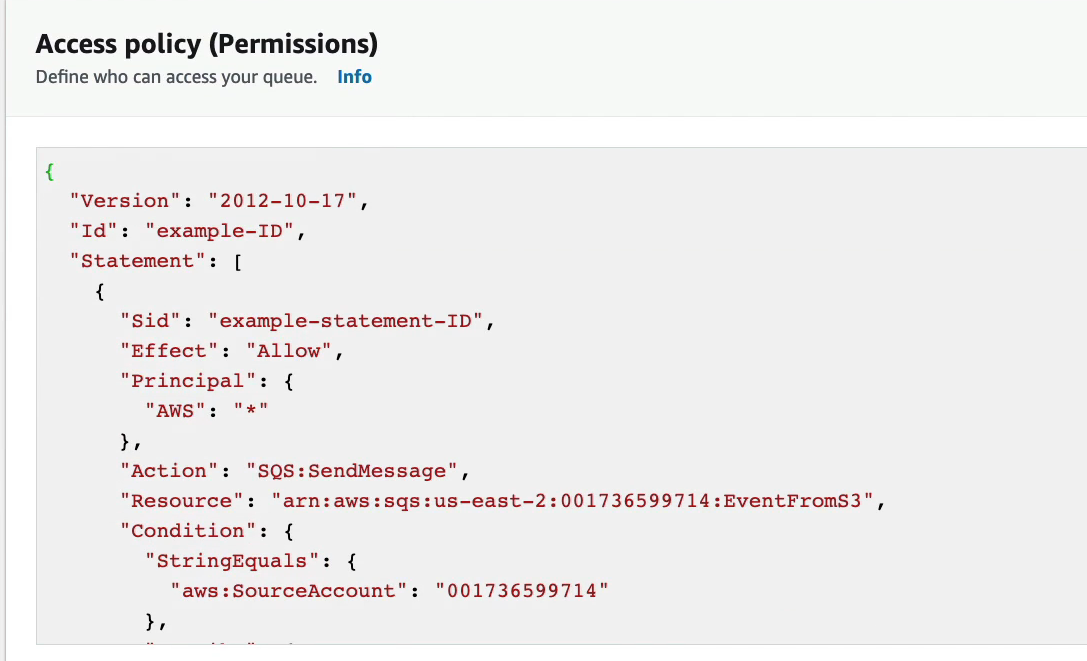
Description automatically generated

We choose our defined queue but we will get an error this time since our sqs queue does not have a property to allow the s3 to send message to it. So now we will set that.

Now we need to allow our s3 bucket to write event notification to sqs so we will find the policy over the internet and specify the arn of our sqs and then can proceed Graphical user interface, text, application, email

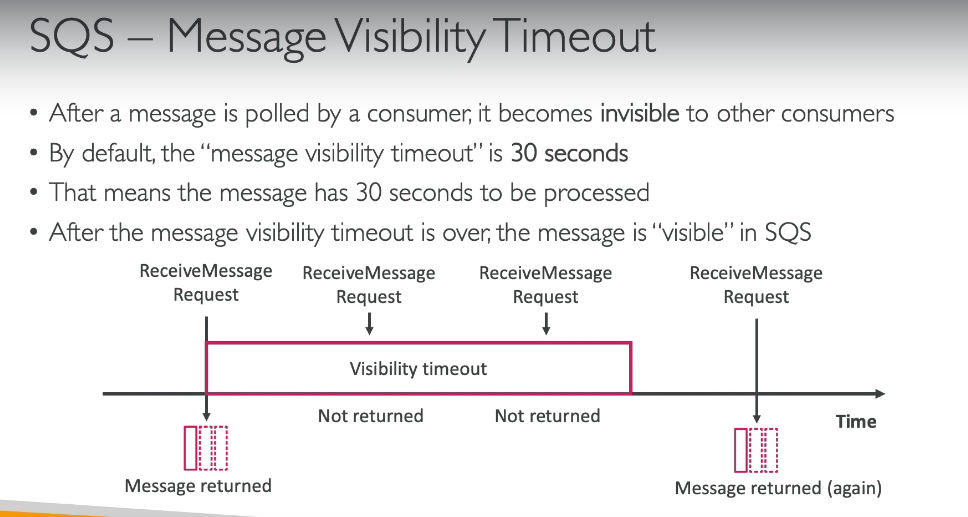
Description automatically generated

Now our s3 will be able to write even tnotifications to the sqs.

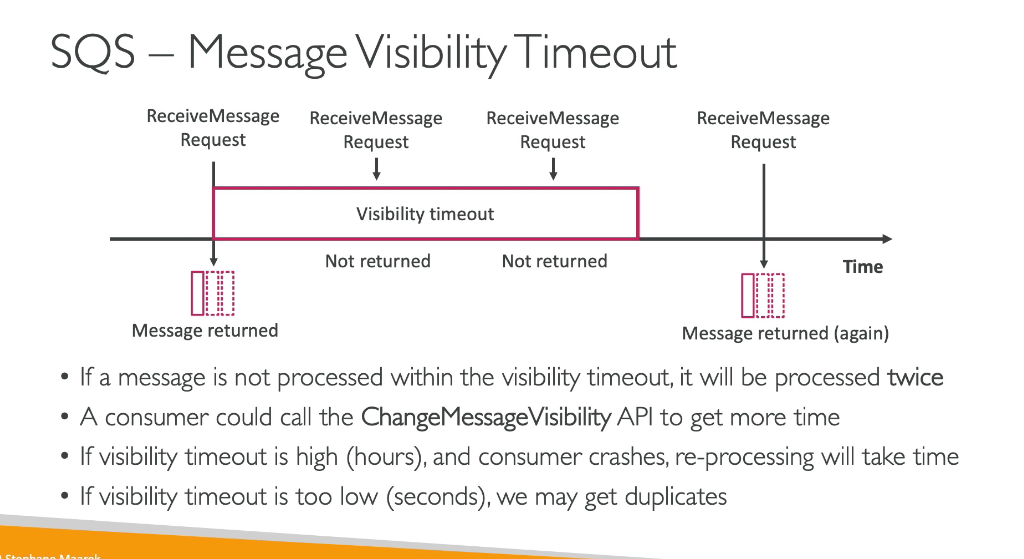


So this way our notoification will be sent to the sqs an dthere it will be read ans a messsge and when it will be read by some consumer after that it will be deleted from the queue.

**MESSAGE VISIBILITY TIMEOUT**

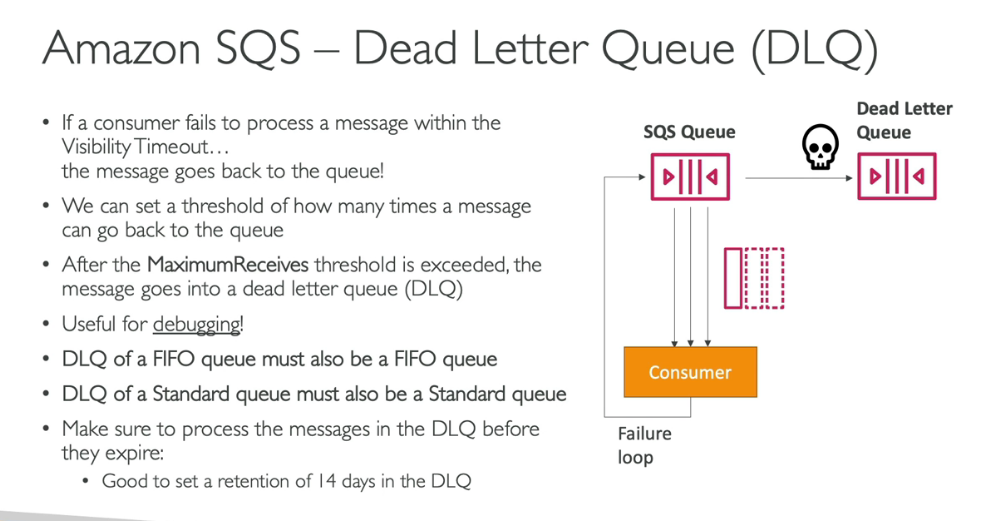
****

**Suppose a consumer is taking longer to process a message so it can acall a change message visibility api In order to increase the invisible time of the message**

****

**DEAD LETTER QUEUE**

If teher is a message which is not processed by a consumer in the visibility y time out then it will be sent back to the quen=ue .But there needs to be a threshold set on the no of times such can happen which is the maximum receives after which the message if not being able to be processed by any cnonsumer will be sent to the dead letter queue for any kind of further processing



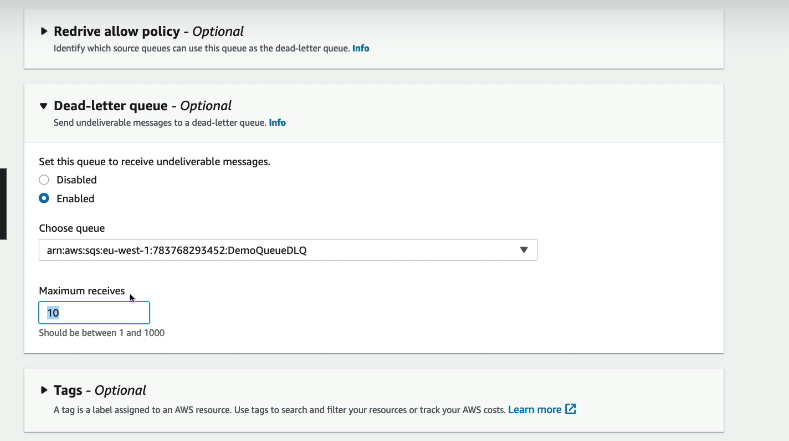
**DEAD LETTER QUEUE REDRIVE TO SOURCE**

When a message reaches a dead letter queue it means it has not been prcessed due to some issue so there will be a manual inspection of these messages and then they will be fixed and then they will be sent back to the standard queue to be processed and fromthere the consumer will take them and process them without even knowing that the message had been gone to the letter queue.

Timeline

Description automatically generated

**SETTING UP A DEAD LETTER QUEUE**

****

We can set the enable a dead letter queue

**DELAY QUEUE**

Graphical user interface, text, application, email

Description automatically generated

Create a queue

Set selivery delay to some time

Graphical user interface, application

Description automatically generated

Rest will be standard.just as the delivery delay has been set we can check that the message will appear after the delay is over.

**DEVELOPER ASSOCUATE LEVEL CONCEPTS**

**Application

Description automatically generated with low confidence**

**SQS EXTENDED CLIENT**

This is used in case when we want to send alarge amount of data to a amessag e queue then the data will be stored in the s3 buckwt and from there it can be retrieved by the consumer and as apart of the message in sqs we can send the metadata whichc is the address of that s3 bucket from where the message can be read by the consumer.

Graphical user interface, text

Description automatically generated

If you are processing a video file then you can store your video file in s3 bucket an d send address in the message.

**MUST KNOW SOMEAPI CALLS OF SQS**

Text

Description automatically generated

**SQS FIFO QUEUE**

First in first out.

Graphical user interface, text, application, email

Description automatically generated

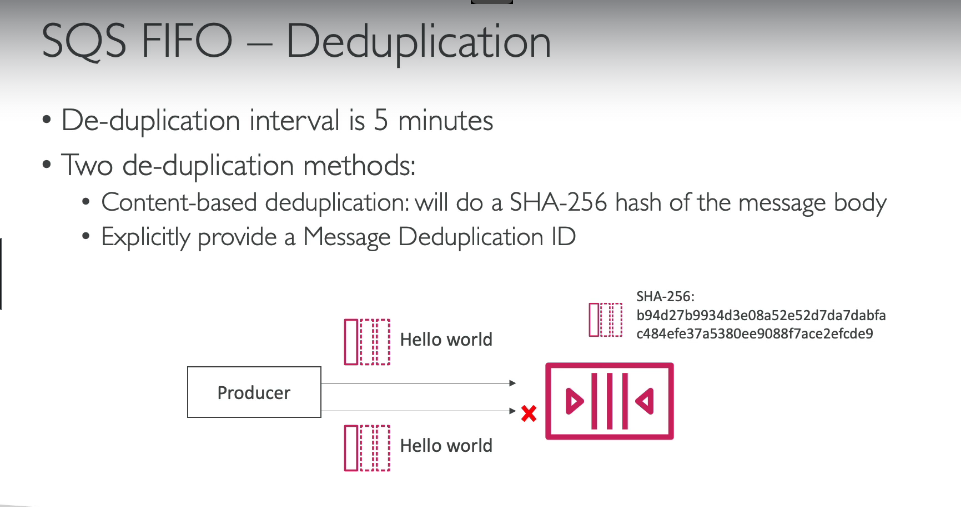
**DEDUPLICATION**

There is a deduplication interval set so that if u send the same message again in the deduplication interval then that message will be refused.

There are two deduplication methods.

1.Content based:in this of every message you send a sha will be calculated of the message body and if the sha of both the bodies comes out to be same then the second message will be refused and only the first one will be processed.

2.Explcit providea a message deduplication id so that message with same id will be discarded.



**MESSAGE GROUPING**

You can group the messages based on the message group id so that a group of emsaages reaches a consumer in order so we can meke groups on the basis of a customer id or anything.

Graphical user interface, text, application

Description automatically generated