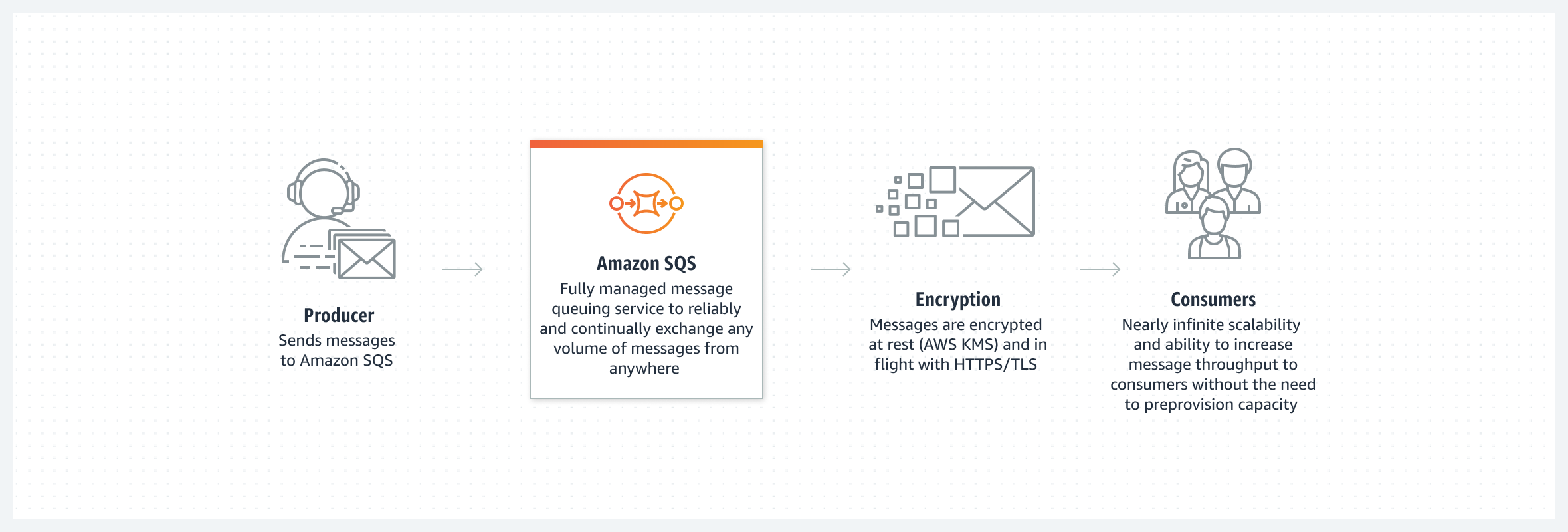
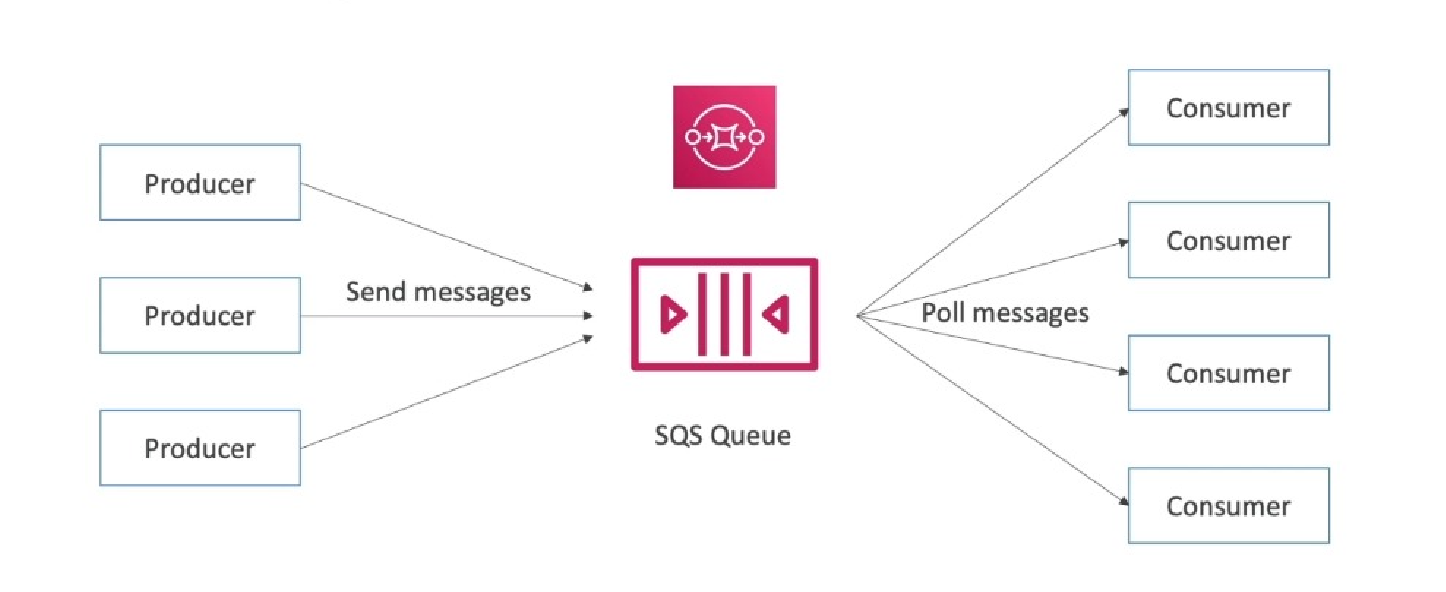
**Amazon SQS**

Amazon Simple Queue Service (SQS) lets you send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.

Amazon SQS 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 where a queue is a temporary repository for messages that are awaiting processing.

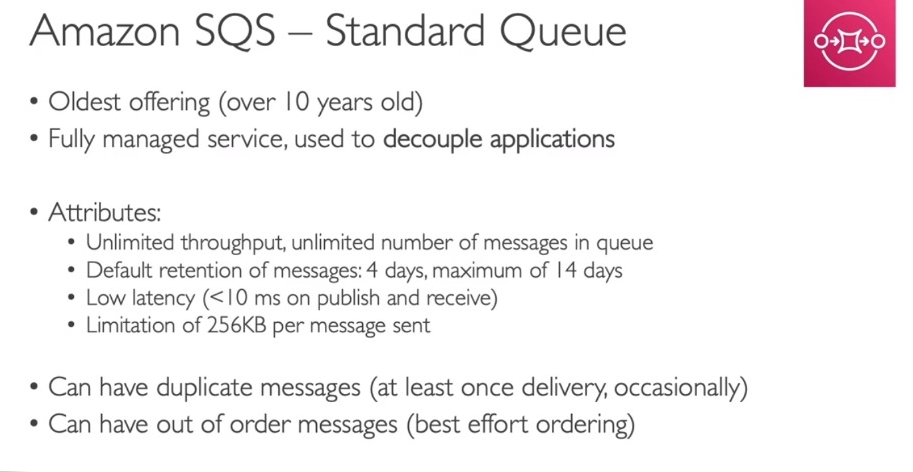
With the help of SQS, you can send, store and receive messages between software components at any volume without losing messages.

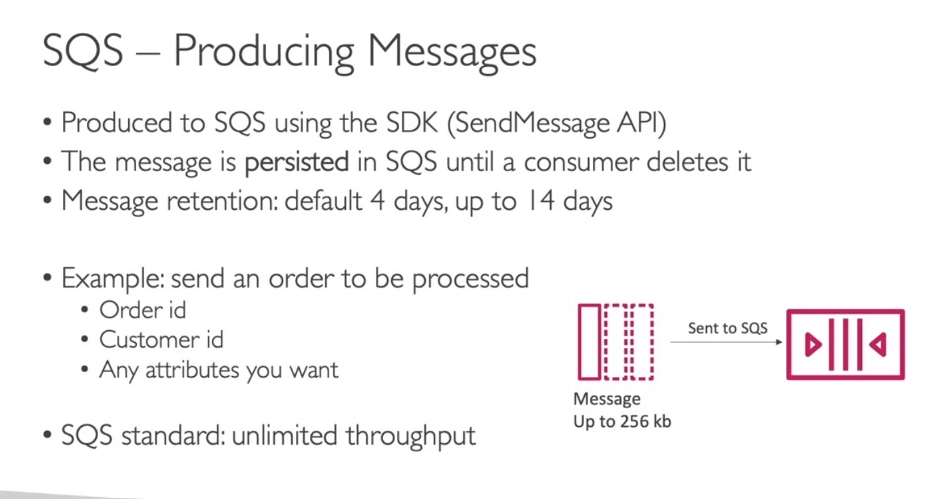


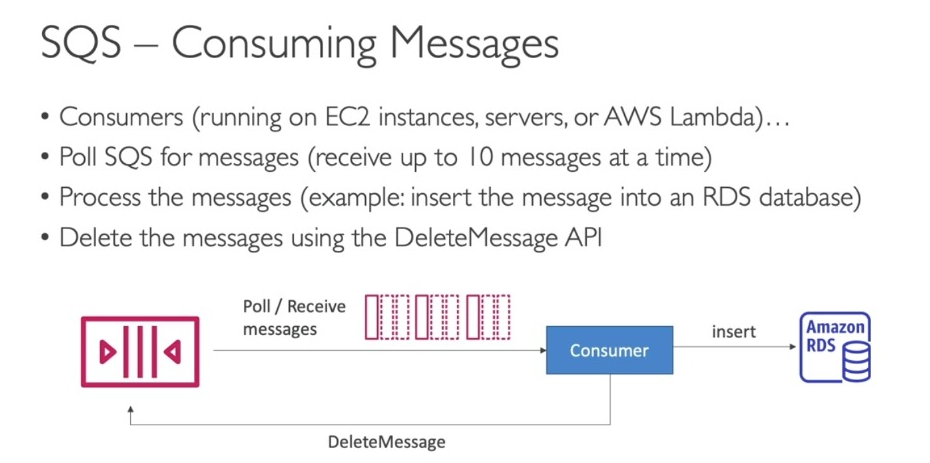


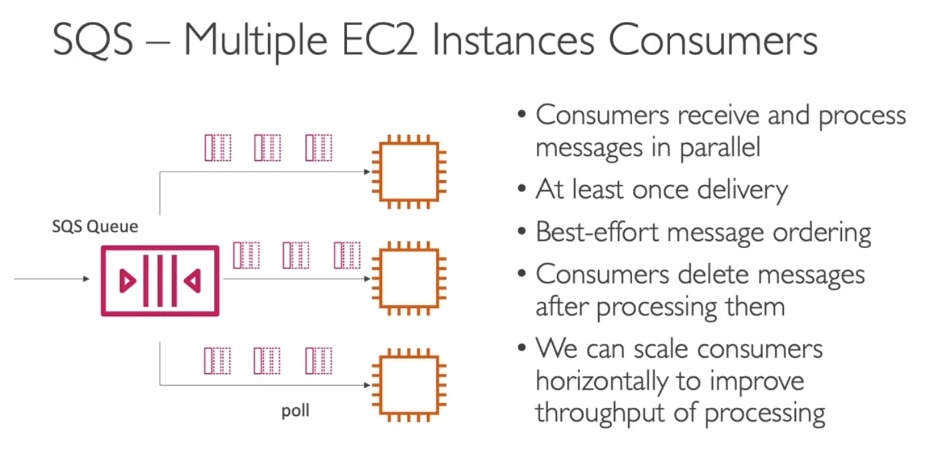
**Amazon SQS Standard queues**

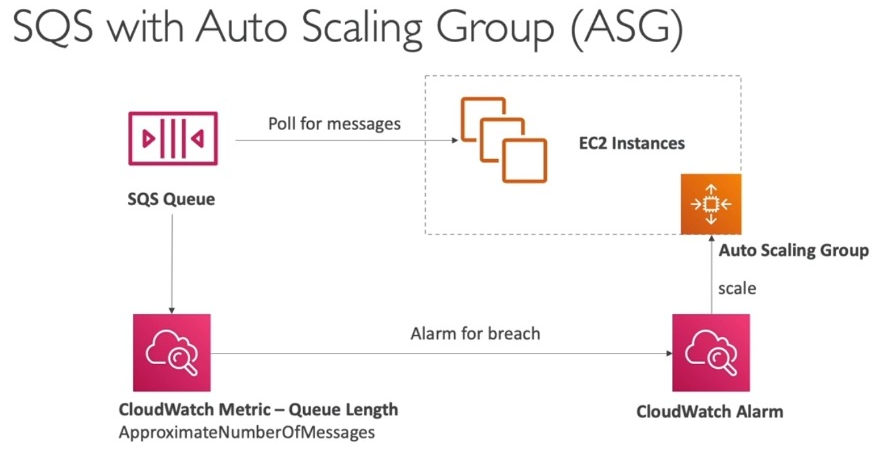
Amazon SQS offers *standard* as the default queue type. Standard queues support a nearly unlimited number of API calls per second, per API action (SendMessage, ReceiveMessage, or DeleteMessage). Standard queues support at-least-once message delivery. However, occasionally (because of the highly distributed architecture that allows nearly unlimited throughput), more than one copy of a message might be delivered out of order. Standard queues provide best-effort ordering which ensures that messages are generally delivered in the same order as they're sent.

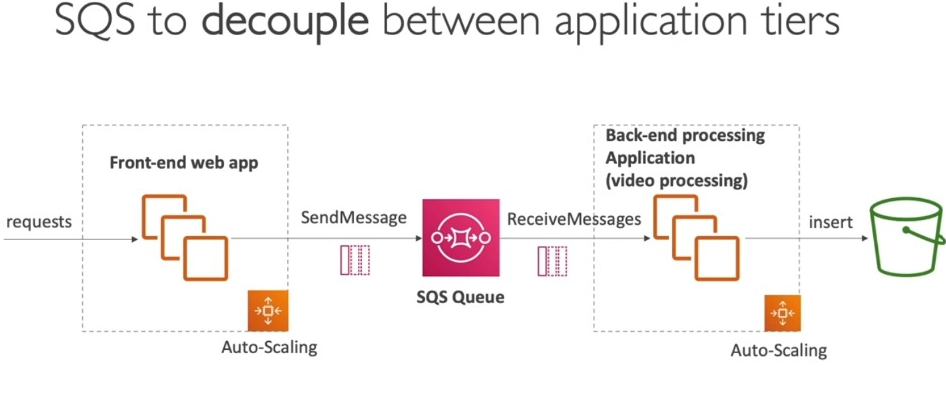
****

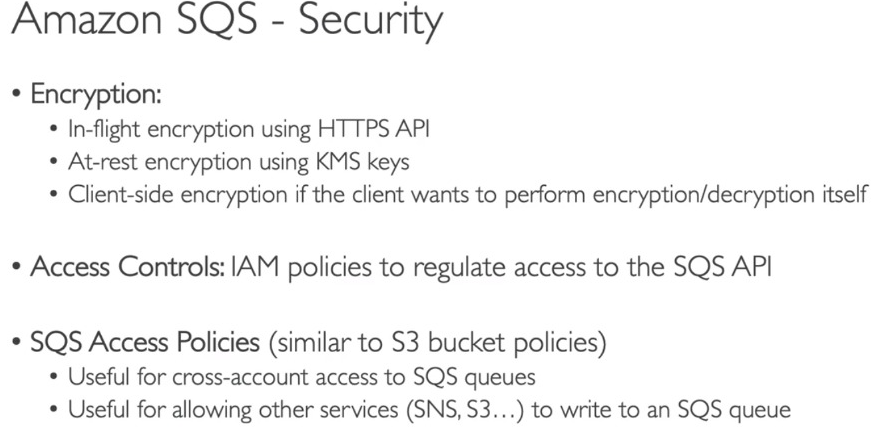
****



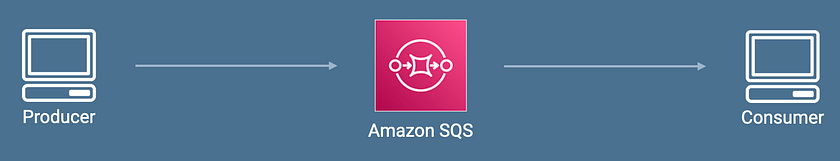








# AWS SQS Visibility Timeout Explained



When the producer sends a message to SQS, it is stored in the queue until consumed by a consumer. When the consumer is ready, it polls SQS for new messages and ultimately receives the message.

Once a message is received by a consumer, SQS doesn’t automatically delete the message. Because there’s no way for SQS to guarantee that the message has been received by the consumer. The message might get lost in transit or the consumer can fail while processing the message.

So the consumer must delete the message from the queue after receiving and processing it.

The consumer might take some time to process the message. While the message is being processed, there’s a chance for an another consumer to receive the same message and process. How does SQS prevents processing a message more than once?

**The Visibility Timeout**

While a consumer is processing a message in the queue, SQS temporary hides the message from other consumers.

This is done by setting a visibility timeout on the message, a period of time during which SQS prevents other consumers from receiving and processing the message.

The default visibility timeout for a message is 30 seconds.

# How it works?

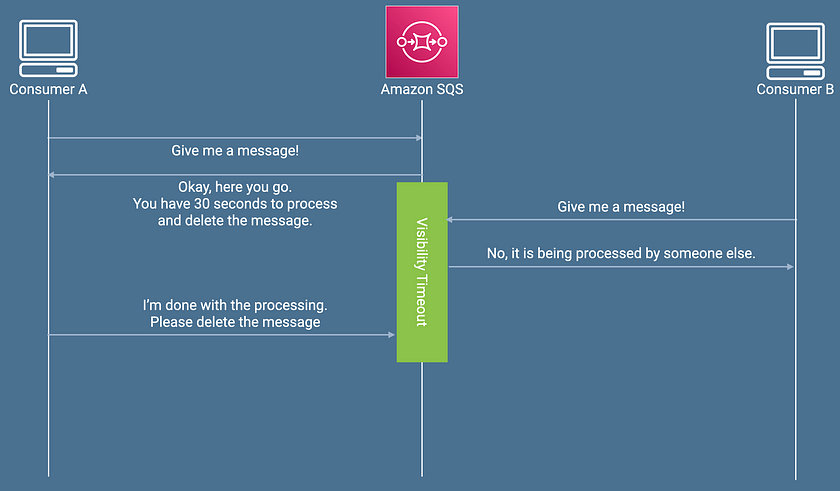
The visibility timeout begins when SQS hands over a message to the consumer. During this time, the consumer has to do two things.

1. It has to complete the processing of the message.

2. Delete the message from the queue.

However, if the consumer fails before deleting the message, the visibility timeout will expire and the message becomes visible to other consumers for receiving.

If a message must be received only once, the consumer should delete it within the duration of the visibility timeout.



**Configuring the Visibility Timeout**

Every SQS queue has the default visibility timeout setting for 30 seconds. The minimum is 0 seconds and the maximum is 12 hours.

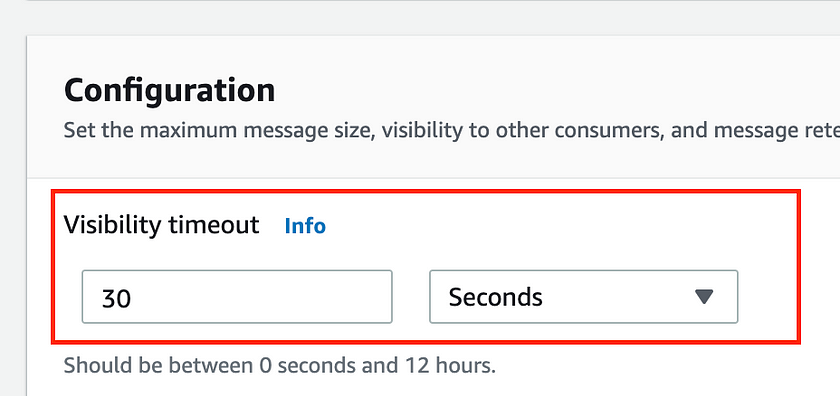
This means your consumer can take up to 12 hours to process and delete a message.

However, you can change this value for the entire queue or per message basis.

**Configure for the entire queue**

Usually, the value of the visibility timeout should be set to the maximum time that it takes your consumer to process and delete a message from the queue. You can perform a series of load tests on the consumer and get a rough idea about how long will it take.

If you know this value beforehand, you can set it when creating the queue. In the AWS Console, you can use the ***Visibility timeout***field to set the visibility timeout for all the messages in the queue.



**Configure per message basis**

There can be situations where you don’t know how long it will take to process a message. For example, when the consumer is in the middle of processing a message, it may need additional time to complete the processing. Thus, the default visibility timeout set for the entire queue will be insufficient.

In a situation like that, SQS allows you to set a special visibility timeout for the received messages without changing the overall queue timeout.

The consumer can call the ***ChangeMessageVisibility*** operation of the SQS API with a new value to shorten or extend the visibility timeout for a message.

For example, you can specify the initial visibility timeout (say, 2 minutes) and then — as long as your consumer still works on the message — keep extending the visibility timeout by 2 minutes every minute.

**Best practices when setting the Visibility Timeout**

Setting the visibility timeout depends on how long it takes your consumer to process and delete a message.

If you set it to a higher value, your consumer has to wait for a relatively long time to attempt to process the message again if the previous processing attempt fails.

Conversely, if you set to a low value, a duplicate message is received by another consumer while the original consumer is still working on the message.

Thus, the best strategy is to either

1. Set the timeout to the maximum value that it takes the consumer to process and delete a message.

0r

2. If you don’t know this value beforehand, set the timeout to a baseline value and keep on extending it until your consumer is done with the message.

**Amazon SQS – Long Polling**

