SIMPLE NOTIFICATION SERVICE (SNS)

- SNS is a fast, flexible, fully managed PUSH notification service.
- It is a web service that delivery or sending of messages to subscribing endpoints or clients.
- It allows for sending individual messages of fan-out messages to a large number of recipients or to other distributed AWS services.
- Messages published to an SNS topics will be delivered to the subscriber immediately.
- Inexpensive, pay-as-you-go model with no upfront cost.
- Reliable: at least three copies of the data are store across multiple AZ in same region.
- It is a way of sending messages. When you are using auto scaling, it triggers an SNS service which will email you that "your EC2 instance is growing".

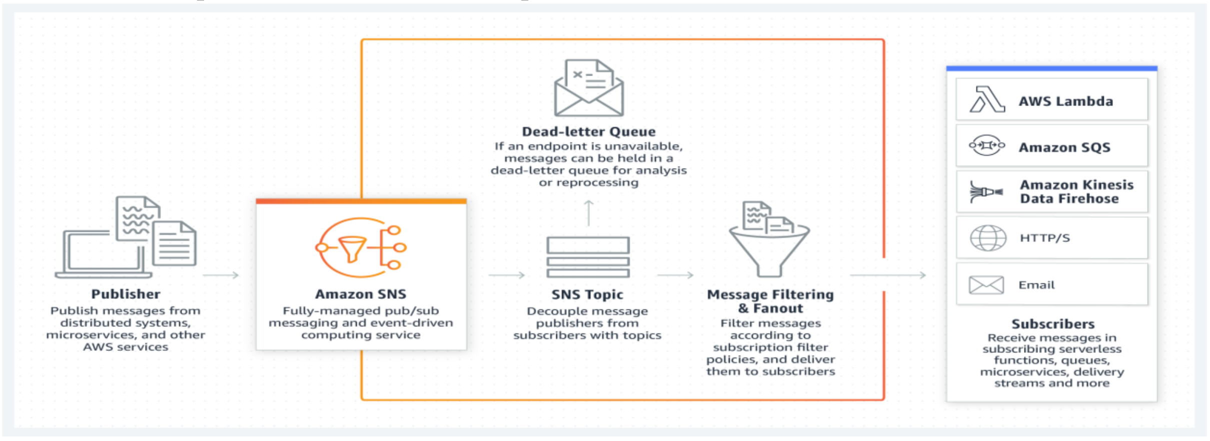
Publisher: publishers are also known as produce and send the message to the SNS which is alogical access point.

Subscriber: subscribers such as web servers, email addresses, Amazon SQS queues, AWS lambda receive the message of notification from the SNS over one of the supported protocols (Amazon SQS, email, lambda, HTTPS, SMS).

SNS Topic

- It is a logical access point and communication channel.
- Each topic has a unique name.
- A topic name is limited to 256 alphanumeric characters.
- The topic name must be unique with in the AWS account.
- Each topic is assigned an AWS ARN once it gets created.
- A topic can support subscribers and notification deliveries over multiple protocols.
- Message/ request published to a single topic can be delivered over multiple protocols as configured when creating each subscriber.
- Delivery format/ transport protocols (endpoints.), SMS, e-mail, email: JSON –for applications, HTTP/ HTTPS, SQS, AWS lambda.
- When using Amazon SNS, you (as the owner) create a topic and control access to it by defining access policies that determine which publishers and subscribers can communicate with the topic.
- Instead of including a specific destination address in each message to topics that they have created or to topics they have permission to publish to.
- Amazon SNS matches the topic to a list of subscribers who have subscribed to that topic and delivers the message to each of these subscribers.
- Each topic has a unique name that identifies the Amazon SNS endpoint for publishers to past message and subscribers to register for notification.
- Subscriber receive all messages published to the topics to which they subscribe, and all subscribers to a topic receive the same message.
  - Amazon Device Messaging
  - Apple Push Notification Service.
  - Google Cloud Messaging
  - Windows Push Notification Service, Baidu Cloud Push for Android

- SNS topic can have subscribers from any supported push notification platforms as well as any other endpoint type such as SMS or Email.
- When you publish a notification to a topic SNS will send identical copies of that, message to each endpoint subscribed to the topic.

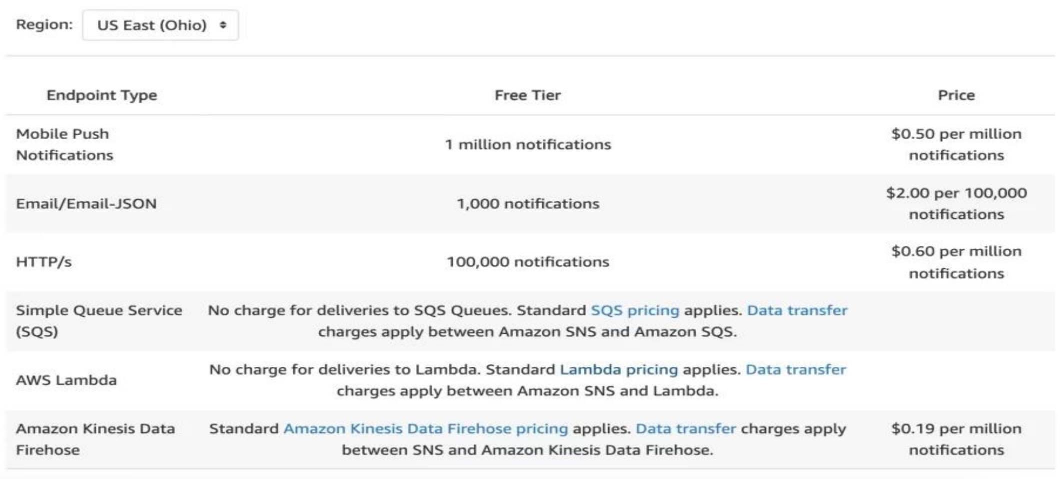

Dead-letter Queue
If an endpoint is unavailable, messages can be held in a dead-letter queue for analysis or reprocessing
Publisher
Publish messages from distributed systems, microservices, and other AWS services
Amazon SNS
Fully-managed pub/sub messaging and event-driven computing service
SNS Topic
Decouple message publishers from subscribers with topics
Message Filtering & Fanout
Filter messages according to subscription filter policies, and deliver them to subscribers
AWS Lambda
Amazon SQS
Amazon Kinesis Data Firehose
HTTP/S
Email
Subscribers
Receive messages in subscribing serverless functions, queues, microservices, delivery streams and more

❖ Amazon SNS Alternatives:

a. Amazon Kinesis Data Stream
b. Amazon Managed Queue Service (AWS MQS)
c. Apache Kafka
d. Twilio
e. Pusher

❖ Amazon SNS Pricing:

a. Publish action: each 64kb of request payload count as one request. So, 256kb payloadwill charged as four requests.
b. Mobile push notification: for e.g.: $0.50/ million request
c. SMS: price depends on country
d. E-mail: $2/100,000
e. HTTP/s notification: $ 0.60/ million
f. SQS and lambda calls are free. These are charged at SQS and lambda rates.
g. Data Transfer

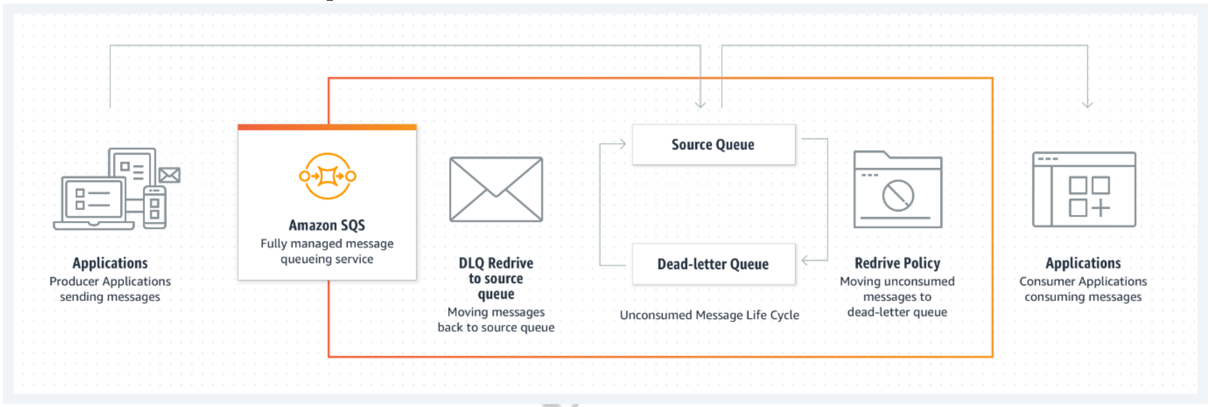


Region: US East (Ohio)

Endpoint Type	Free Tier	Price
Mobile Push Notifications	1 million notifications	$0.50 per million notifications
Email/Email-JSON	1,000 notifications	$2.00 per 100,000 notifications
HTTP/s	100,000 notifications	$0.60 per million notifications
Simple Queue Service (SQS)	No charge for deliveries to SQS Queues. Standard SQS pricing applies. Data transfer charges apply between Amazon SNS and Amazon SQS.	
AWS Lambda	No charge for deliveries to Lambda. Standard Lambda pricing applies. Data transfer charges apply between Amazon SNS and Lambda.	
Amazon Kinesis Data Firehose	Standard Amazon Kinesis Data Firehose pricing applies. Data transfer charges apply between SNS and Amazon Kinesis Data Firehose.	$0.19 per million notifications

SIMPLE QUEUE SERVICE (SQS)

- SQS is a fast, reliable, fully managed message queue service.
- It is a web service that gives you access to message queue that stores messages waiting to be processed.
- It offers a reliable highly scalable, hosted queue for storing messages between servers.
- It allows the decoupling of application components such that a failure in one component doesn't cause a bigger problem to application functionality. (Like in coupled application)
- Using SQS, you no longer need a highly available message cluster or the burden of running it.
- You can delete all the messages in an SQS queue without deleting the SQS Queue itself.
- You can use applications on EC2 instances to read and process the SQS queue message.
- You can use auto scaling to scale the EC2 fleet processing the SQS messages, as the queue size increases.
- These applications on EC2 instances can process the SQS message/ jobs then post the SQS results to other SQS queues or other AWS service.


Applications
Producer Applications sending messages
Amazon SQS
Fully managed message queueing service
DLQ Redrive to source queue
Moving messages back to source queue
Source Queue
Dead-letter Queue
Unconsumed Message Life Cycle
Redrive Policy
Moving unconsumed messages to dead-letter queue
Applications
Consumer Applications consuming messages

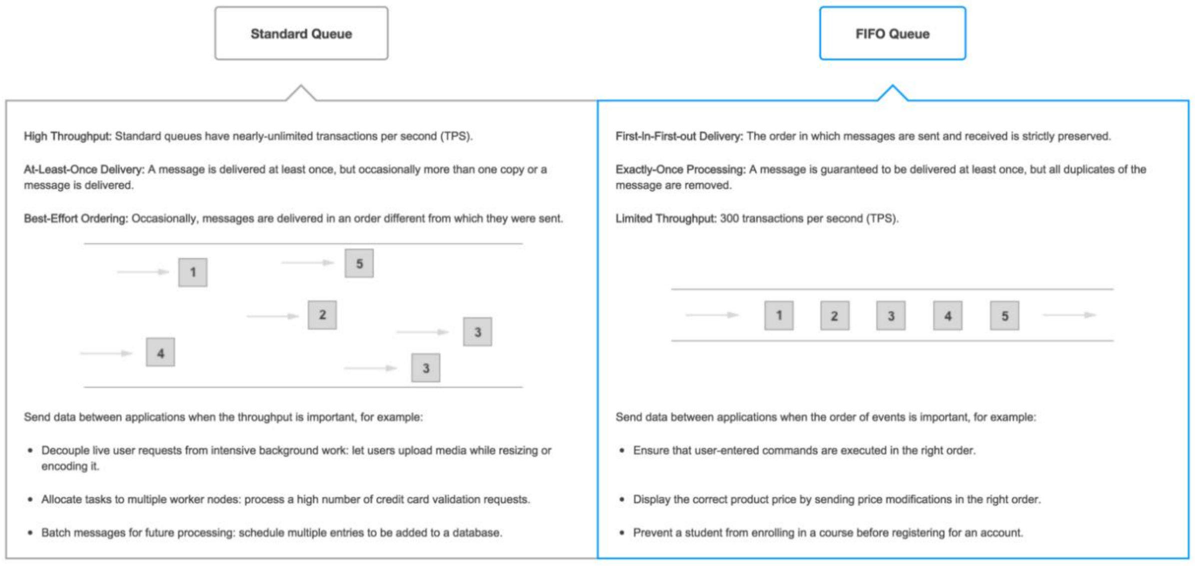
❖ Types of Amazon Queue Services:

- It is of two types such as:

  1. Standard Queue
  2. FIFO Queue

1. Standard Queue:
- High Throughput
- At Least One Delivery
- Duplicity is possible
- Best effort ordering

2. FIFO Queue:
- Limited throughput (300 TPS)
- Exactly one processing
- Duplicity not possible
- Strict ordering: first-in-first-out
- FIFO queue are limited to 300 transactions per second (TPS), but have all the capabilities of standard queue.


Standard Queue

High Throughput: Standard queues have nearly-unlimited transactions per second (TPS).

At-Least-Once Delivery: A message is delivered at least once, but occasionally more than one copy or a message is delivered.

Best-Effort Ordering: Occasionally, messages are delivered in an order different from which they were sent.

Send data between applications when the throughput is important, for example:

- Decouple live user requests from intensive background work: let users upload media while resizing or encoding it.
- Allocate tasks to multiple worker nodes: process a high number of credit card validation requests.
- Batch messages for future processing: schedule multiple entries to be added to a database.

FIFO Queue

First-In-First-out Delivery: The order in which messages are sent and received is strictly preserved.

Exactly-Once Processing: A message is guaranteed to be delivered at least once, but all duplicates of the message are removed.

Limited Throughput: 300 transactions per second (TPS).

Send data between applications when the order of events is important, for example:

- Ensure that user-entered commands are executed in the right order.
- Display the correct product price by sending price modifications in the right order.
- Prevent a student from enrolling in a course before registering for an account.

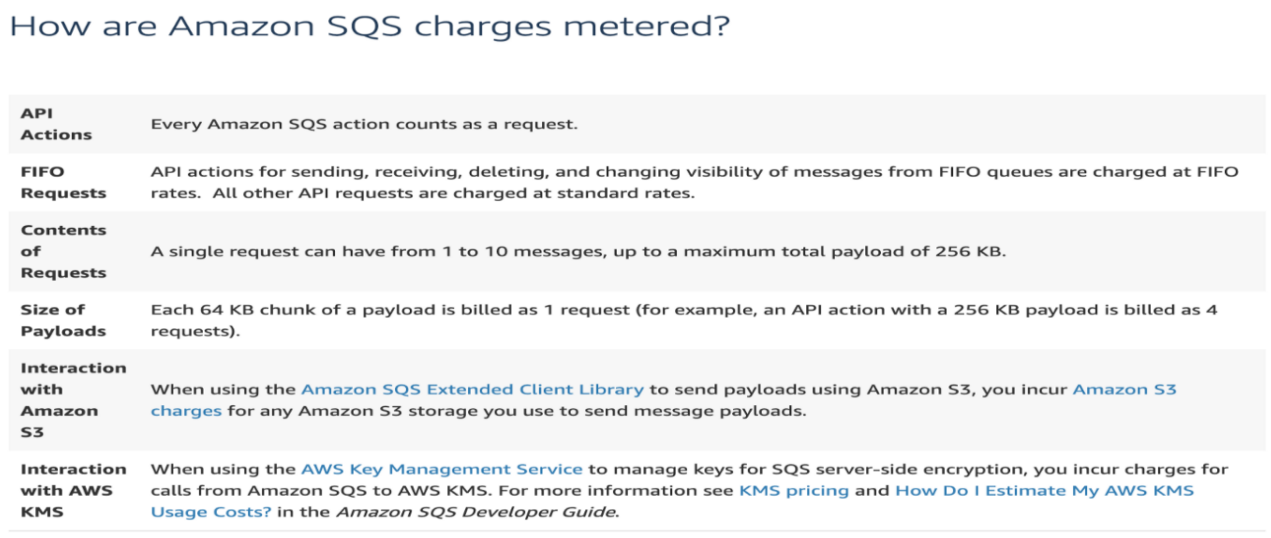
❖ SQS Pricing:

➢ The first 1 million monthly requests are free, after that pricing is according to regions.
➢ For e.g.: in Mumbai region:

- Standard Queue- $0.40/ million requests
- FIFO Queue- $0.50/ million requests

❖ How Amazon SQS charges:

1. **API action:** every Amazon SQS actions count as request.
2. **FIFO request:** API actions for sending, receiving, deleting and changing visibility of messages from FIFO queues are charged at FIFO rates.
3. **Contents of Request:** a single request can have from 1 to 10 messages, up to a maximum total payload of 256kb.
4. **Size of Payload:** each 64kn chunk of a payload is billed as 1 request. (For e.g.: API action with a 256kb payload is billed as 4 request)
5. **Interaction with Amazon S3.**
6. **Interaction with AWS KMS.**


How are Amazon SQS charges metered?

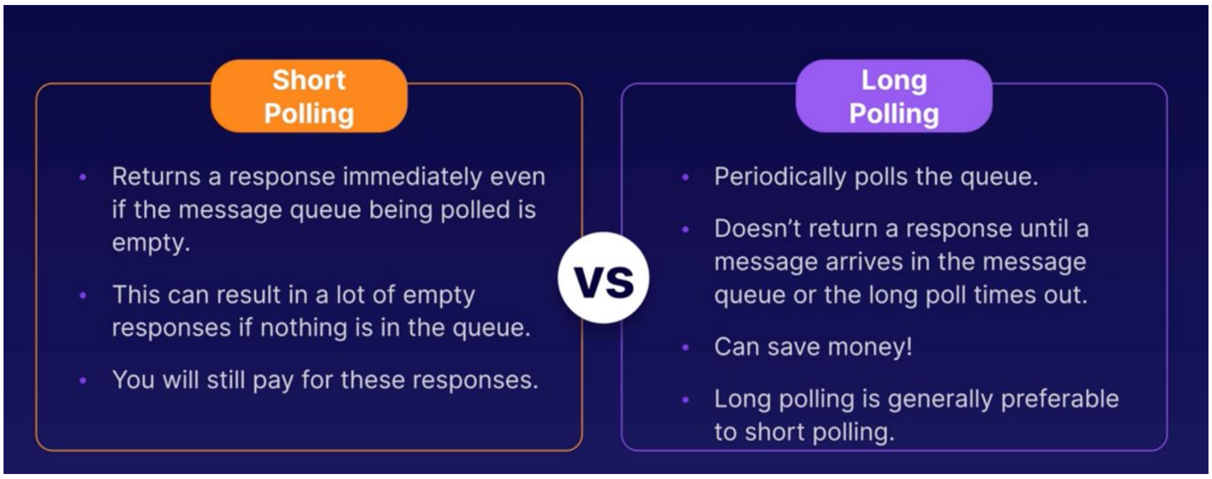
API Actions	Every Amazon SQS action counts as a request.
FIFO Requests	API actions for sending, receiving, deleting, and changing visibility of messages from FIFO queues are charged at FIFO rates. All other API requests are charged at standard rates.
Contents of Requests	A single request can have from 1 to 10 messages, up to a maximum total payload of 256 KB.
Size of Payloads	Each 64 KB chunk of a payload is billed as 1 request (for example, an API action with a 256 KB payload is billed as 4 requests).
Interaction with Amazon S3	When using the Amazon SQS Extended Client Library to send payloads using Amazon S3, you incur Amazon S3 charges for any Amazon S3 storage you use to send message payloads.
Interaction with AWS KMS	When using the AWS Key Management Service to manage keys for SQS server-side encryption, you incur charges for calls from Amazon SQS to AWS KMS. For more information see KMS pricing and How Do I Estimate My AWS KMS Usage Costs? in the *Amazon SQS Developer Guide*.

❖ Short Polling:

- ➢ A request is returned immediately even if the queue is empty.
- ➢ It doesn't wait for message to appear in the queue.
- ➢ It queries only a subset of the available servers for messages (based on weighted random distribution)
- ➢ Default by SQS
- ➢ Receive message wait time is set to 0.
- ➢ More request is used which implies higher cost.

❖ Long Polling:

- ➢ It is preferred to regular/ short polling. It uses fewer requests and request cost by: eliminating false empty responses by querying all the servers.
- ➢ Reduce the number of empty responses by allowing Amazon SQS to wait until a message is available in the queue before sending a response, Unless the connection timeout (20sec)
- ➢ Receive message wait time is set to a non-zero value (max 20sec)
- ➢ Billing is same for both polling's.



Short Polling

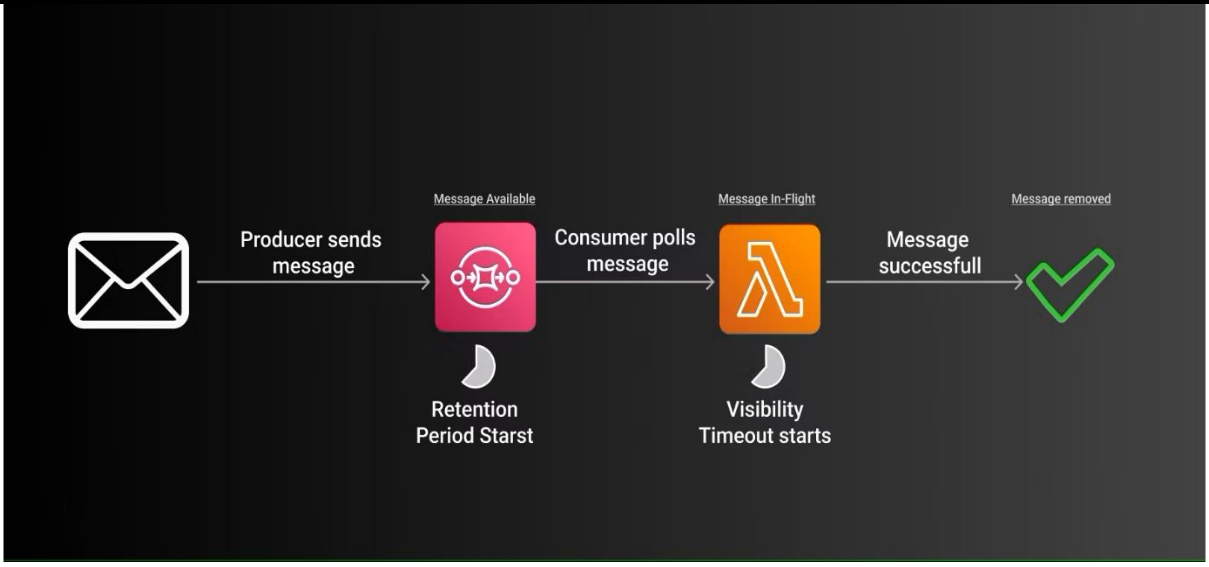
- Returns a response immediately even if the message queue being polled is empty.
- This can result in a lot of empty responses if nothing is in the queue.
- You will still pay for these responses.

VS

Long Polling

- Periodically polls the queue.
- Doesn't return a response until a message arrives in the message queue or the long poll times out.
- Can save money!
- Long polling is generally preferable to short polling.

❖ SQS Retention Period:



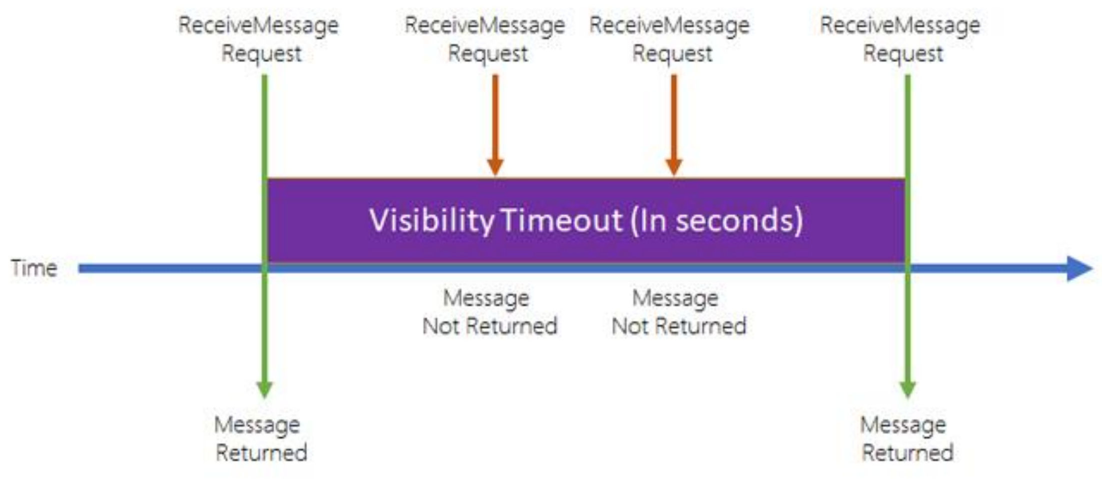
- SQS messages can remain in the queue for up to 14 days. (SQS retention period)
- Range is 1 min to 14 days. (Default is 4days)
- Once the maximum retention period of a message id reached, it will be deleted automatically from the queue.
- Messages can be sent to the queue and read from the queue simultaneously.
- SQS can be used with Dynamo DB, EC2, ECS, redshift, RDS, lambda, S3 to make distributed/ decoupled application.
- You can have multiple queues with different properties

❖ SQS Visibility Timeout:

- It is the duration of time a message is locked for read by other servers.
- Maximum is 12 hours and default are 30 sec.
- A server that read a message process it, can change the message visibility timeout if it needs more time to process the message.
- After a message is read, there are the following possibilities:
    a. An ACK is received that a message is processed, so it must be deleted from the queue to avoid duplicates.
    b. If a failure is received of the visibility timeout expires, the message will then be locked for read such that it can be read and processed by another servers.

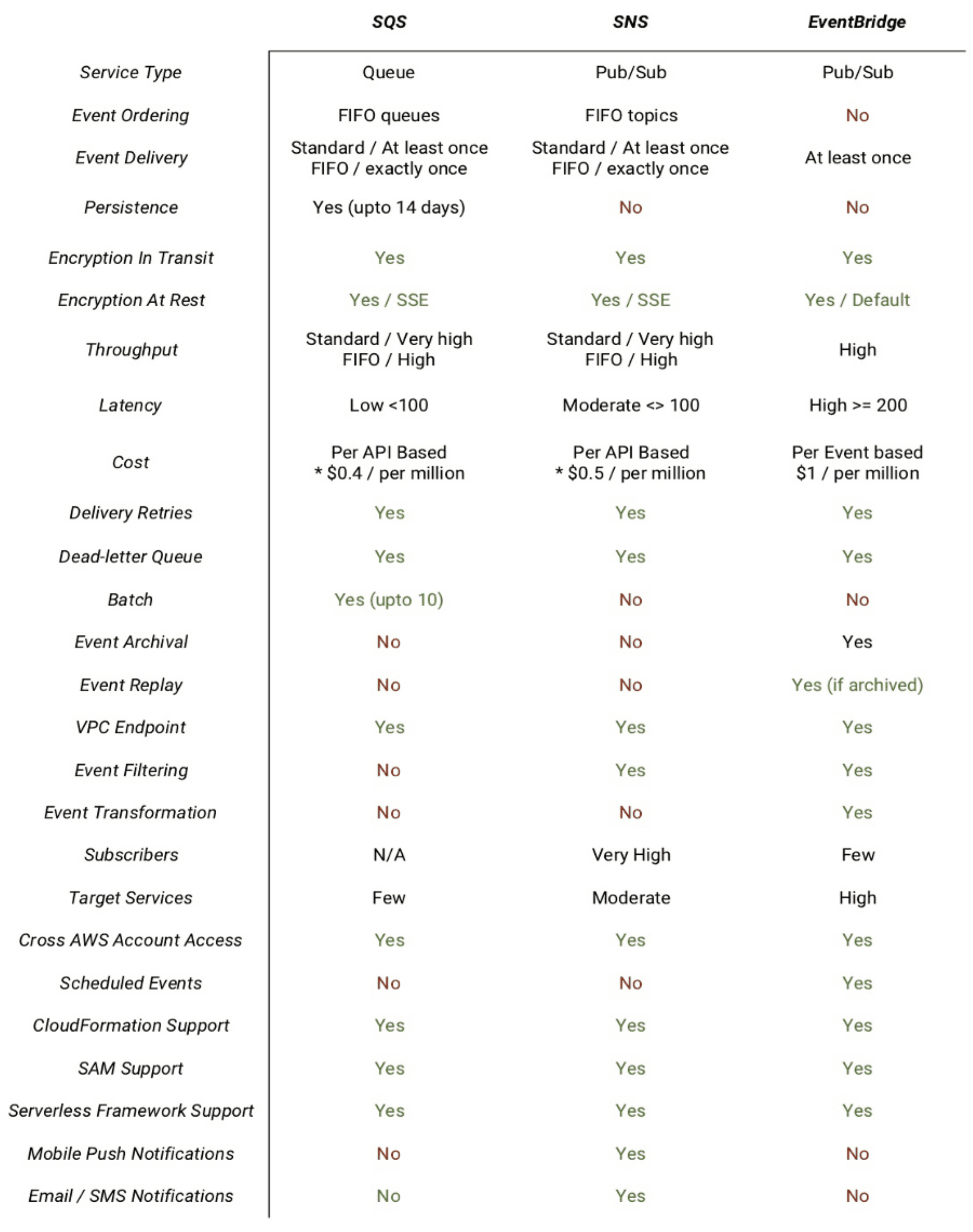
Delivery Delay: AWS SQS provides delivery delay options to postpone the delivery of new messages to a queue. If delivery delay is defined for a queue, any new messages will not be visible to the server for the duration of delay. The default (min) delay for a queue is 0 seconds. The maximum is 15 minutes.

Receive Message Wait Time: the default time is 0 seconds. This is maximum amount of timethat a long polling receive call will wait for a message to become available before returning anempty response (maximum value is 20sec).



❖ Dead Letter Queue:

- The main task of a dead letter queue is handling message failure. A dead letter queue lets you to set aside and isolated message that can't be processed correctly to determine why their processing didn't success.
- Don't use a dead letter queue with a FIFO queue, if you don't want to break the exact order of messages or operations.
- DLQ must be of the same type as the source queue. (Standard or FIFO)



	SQS	*SNS*	*EventBridge*
Service Type	Queue	Pub/Sub	Pub/Sub
Event Ordering	FIFO queues	FIFO topics	No
Event Delivery	Standard / At least once FIFO / exactly once	Standard / At least once FIFO / exactly once	At least once
Persistence	Yes (upto 14 days)	No	No
Encryption In Transit	Yes	Yes	Yes
Encryption At Rest	Yes / SSE	Yes / SSE	Yes / Default
Throughput	Standard / Very high FIFO / High	Standard / Very high FIFO / High	High
Latency	Low <100	Moderate <> 100	High >= 200
Cost	Per API Based * $0.4 / per million	Per API Based * $0.5 / per million	Per Event based $1 / per million
Delivery Retries	Yes	Yes	Yes
Dead-letter Queue	Yes	Yes	Yes
Batch	Yes (upto 10)	No	No
Event Archival	No	No	Yes
Event Replay	No	No	Yes (if archived)
VPC Endpoint	Yes	Yes	Yes
Event Filtering	No	Yes	Yes
Event Transformation	No	No	Yes
Subscribers	N/A	Very High	Few
Target Services	Few	Moderate	High
Cross AWS Account Access	Yes	Yes	Yes
Scheduled Events	No	No	Yes
CloudFormation Support	Yes	Yes	Yes
SAM Support	Yes	Yes	Yes
Serverless Framework Support	Yes	Yes	Yes
Mobile Push Notifications	No	Yes	No
Email / SMS Notifications	No	Yes	No

** Although SQS is cheaper than SNS, because of the per API metering, SQS is costlier with Lambda compared to SNS*