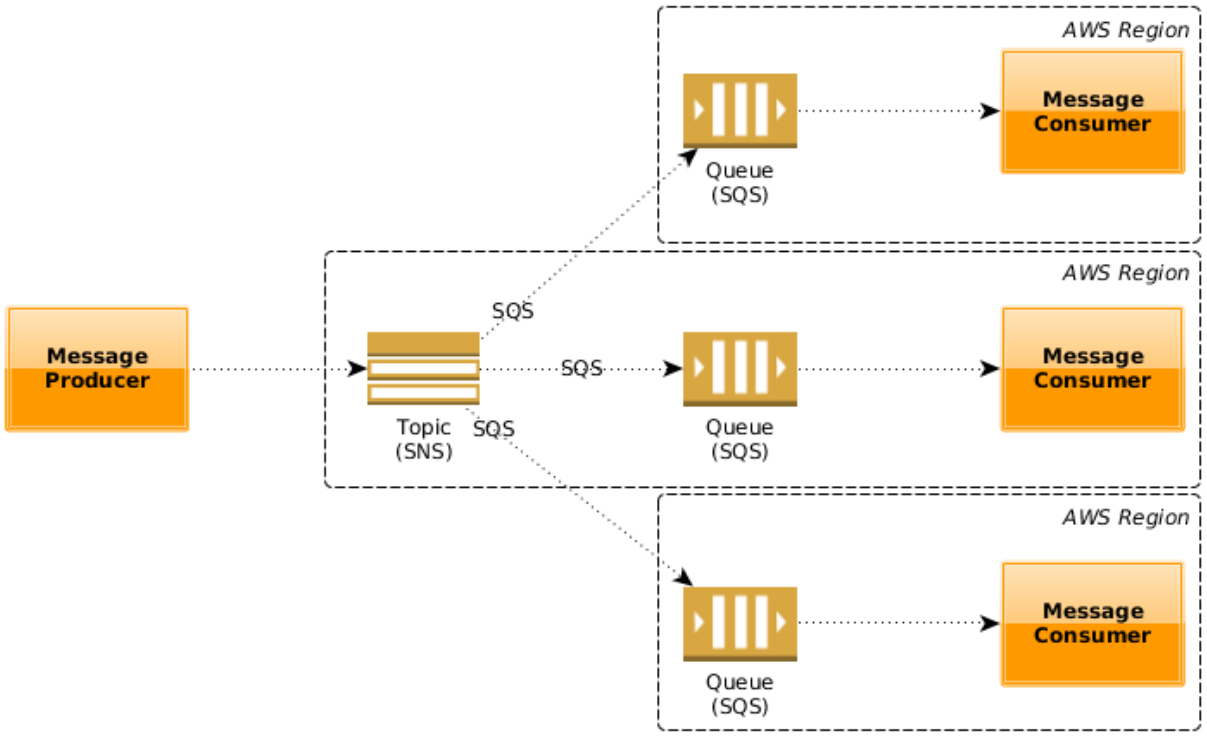
AWS — Difference between SQS and SNS

Comparisons: SQS vs SNS in AWS — Simple Notification Service and Simple Queue Service.



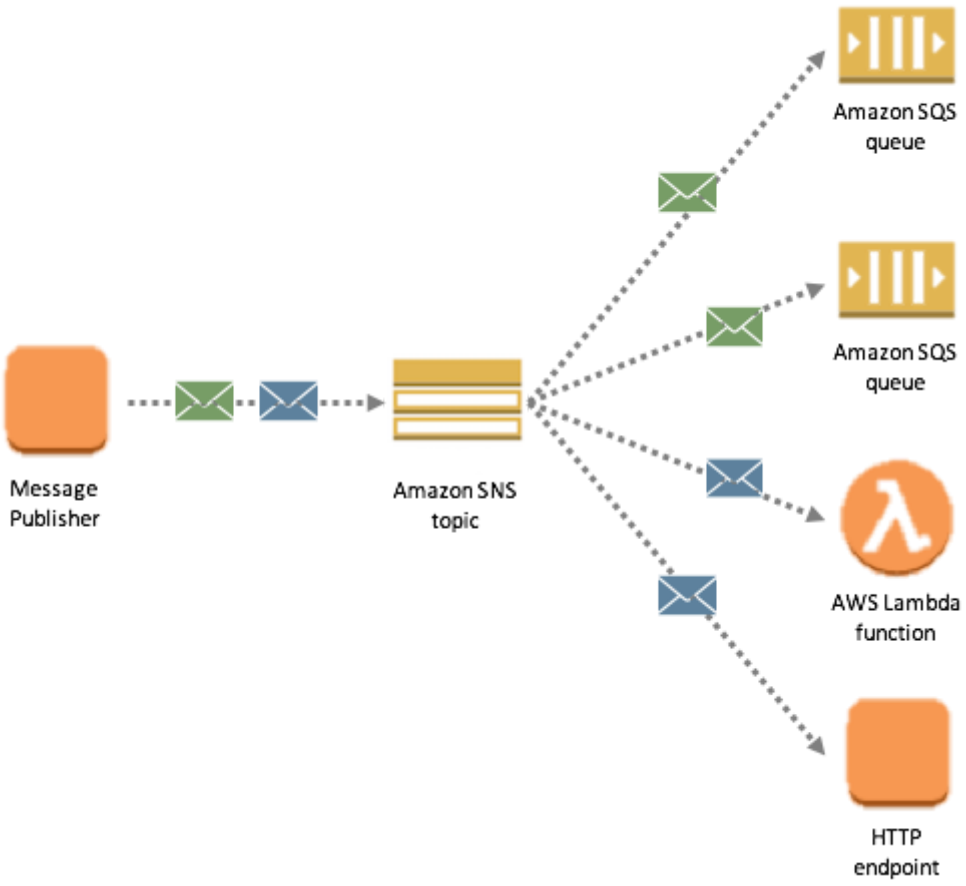
TL;DR

SQS and SNS are important components for scalable, large scale, distributed, cloud-based applications:

SNS is a distributed **publish-subscribe** service.

SQS is distributed **queuing** service.

SNS (Simple Notification Service)



SNS

Amazon SNS is a fast, flexible, fully managed push notification service that lets you send individual messages or to bulk messages to large numbers of recipients. Amazon SNS makes it simple and cost effective to send push notifications to mobile device

users, email recipients, or even send messages to other distributed services.

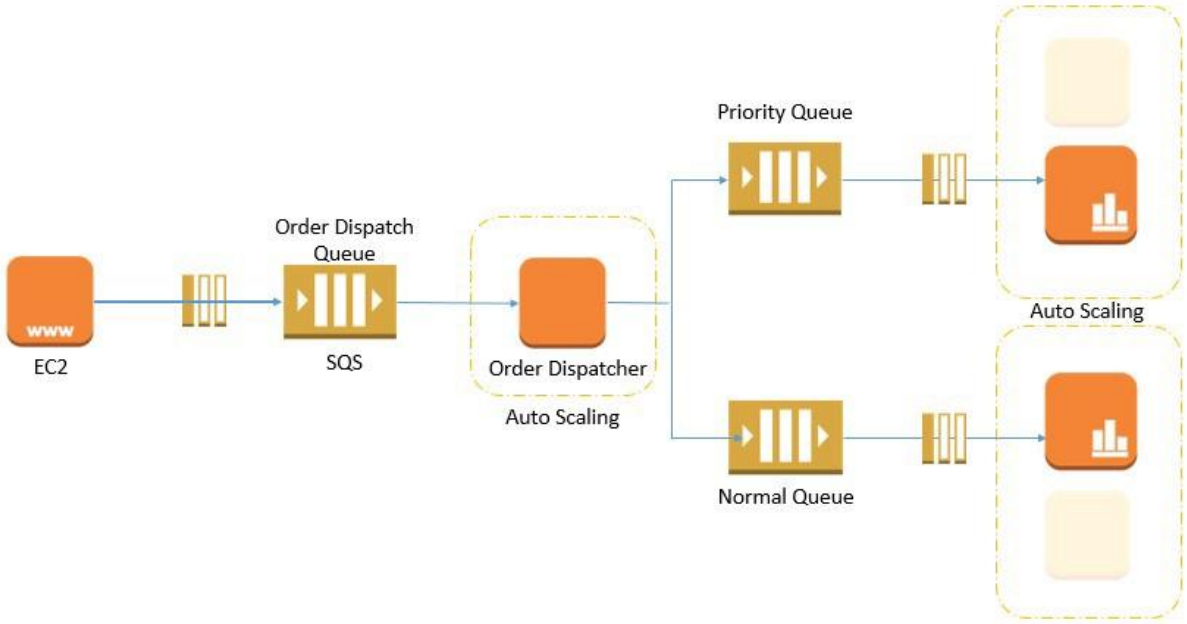
SNS is a distributed **publish-subscribe** system. Messages are **pushed** to subscribers as and when they are sent by publishers to SNS.

SNS supports several end points such as email, sms, http end point, and SQS. If you want an unknown number and type of subscribers to receive messages, you need SNS.

With Amazon SNS, you can send *push notifications* to Apple, Google, Fire OS, and Windows devices , as well as to Android devices in China with Baidu Cloud Push. You can use SNS to send SMS messages

to mobile device users in the US or to email recipients worldwide.

SQS (Simple Queue Service)



SQS

Amazon SQS is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.

SQS is distributed **queuing** system. Messages are not pushed to receivers. Receivers have to poll SQS to receive messages. Messages can be stored in SQS for a short duration of time (max 14 days).

Messages can't be received by multiple receivers at the same time. Any one receiver can receive a message, process, and delete it. Other receivers do not receive the same message later. Polling inherently introduces some latency in message delivery in SQS, unlike SNS where messages are immediately pushed to subscribers.

Key Differences

Entity Type

SQS : Queue (similar to JMS, MSMQ).

SNS: Topic-Subscriber (Pub/Sub system).

Message consumption

SQS: Pull Mechanism — Consumers poll messages from SQS.

SNS: Push Mechanism — SNS pushes messages to consumers.

Persistence

SQS: Messages are persisted for some duration is no consumer-available. The retention period value is from 1 minute to 14 days. The default is 4 days.

SNS: No persistence. Whichever consumer is present at the time of message arrival, get the message and the message is deleted. If no consumers are available then the message is lost.

In SQS the message delivery is guaranteed but in SNS it is not.

Consumer Type

SQS: All the consumers are supposed to be identical and hence process the messages in exact same way.

SNS: All the consumers are (supposed to be) processing the messages in different ways.

Use Cases

*Choose **SNS** if:*

- You would like to be able to publish and consume batches of messages.
- You would like to allow the same message to be processed in multiple ways.
- Multiple subscribers are needed.

*Choose **SQS** if:*

- You need a simple queue with no particular additional requirements.
- Decoupling two applications and allowing parallel asynchronous processing.
- Only one subscriber is needed.

*We can design a **fanout** pattern by using both SNS and SQS. In this pattern, a message published to an SNS topic is distributed to multiple SQS queues in parallel.*

Summary

SQS is mainly used to decouple applications. SNS distributes several copies of messages to several subscribers.

Here's a conclousion comparison of the two:

Entity Type

- SQS: Queue (Similar to JMS)
- SNS: Topic (Pub/Sub system)

Message consumption

- SQS: Pull Mechanism - Consumers poll and pull messages from SQS
- SNS: Push Mechanism - SNS Pushes messages to consumers

Use Case

- SQS: Decoupling two applications and allowing parallel asynchronous processing
- SNS: Fanout - Processing the same message in multiple ways

Persistence

- SQS: Messages are persisted for some (configurable) duration if no consumer is available (maximum two weeks), so the consumer does not have to be up when messages are added to the queue.
- SNS: No persistence. Whichever consumer is present at the time of message arrival gets the message and the message is deleted. If no consumers are available then the message is lost after a few retries.

Consumer Type

- SQS: All the consumers are typically identical and hence process the messages in the exact same way (each message is processed once by one consumer, though in rare cases messages may be resent)
- SNS: The consumers might process the messages in different ways

Sample applications

- SQS: Jobs framework: The Jobs are submitted to SQS and the consumers at the other end can process the jobs asynchronously. If the job frequency increases, the number of consumers can simply be increased to achieve better throughput.
- SNS: Image processing. If someone uploads an image to S3 then watermark that image, create a thumbnail and also send a *Thank You* email. In that case S3 can publish notifications to an SNS topic with three consumers listening to it. The first one watermarks the image, the second one creates a thumbnail and the third one sends a *Thank You* email. All of them receive the same message (image URL) and do their processing in parallel.