Lab 4–AWS SQS and SNS

Introduction to message queues for communication using AWS Simple Message Queue Service(SQS) and understanding publisher-subscriber messaging using AWS Simple Notification Service(SNS).

* Deliverables: 5 screenshots 4a-4e

# AWS SNS

* Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication
* **SNS** is a distributed **publish-subscribe** system.
* Messages are **pushed** to subscribers as and when they are sent by publishers to SNS.
* [Learn more here](https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc)

# Task 1 : Understand publisher subscriber pattern through application-to-person (A2P) messaging.

* Go to SNS from your console and create a topic
* Choose Standard as the type of your topic.
* Name the topic: [yourSRN]-[yourname] (eg: pes120180000-ram)
* Now, its time to create a subscription to the topic that you just created
* Create two subscriptions. Set the endpoint to your email and your friend’s mail with the Email protocol. Make sure to confirm subscription from the mail inboxes. Once the subscriptions are confirmed, take a screen shot of all subscriptions. **(4a)**
* Now you need to publish messages to your topic.
* While publishing messages to your topic, set the header to “Hello this is [yourname] from SNS! “ .
* Make sure to use Identical payload for all delivery, as your messages will be invariant of the different protocols.
* Send the message and check if you and your friend have received the mail. Take a screen shot of the mail received. **(4b)**

# AWS SQS

* **SQS** is distributed **queuing** system.
* Messages are NOT pushed to receivers.
* Receivers have to **poll or pull** messages from **SQS**.
* 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.
* Also note that SNS supports several end points such as email, SMS, http end point and SQS
* Learn in detail from [here](https://aws.amazon.com/sqs/)
* Learn about the basic architecture [here](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-basic-architecture.html)
* SQS offers two types of message queues.:
  + [Standard queues](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/standard-queues.html) offer maximum throughput, best-effort ordering, and at-least-once delivery.
  + SQS [FIFO queues](https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/FIFO-queues.html) are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

## Task 2 : Play around with SQS queues

* Make a standard queue and name it [yourname]-queue. Take a screenshot **(4c)**
* Send any message to the queue with a delay of 10 seconds.
* Poll for messages. Notice that the receive count gets incremented, if you poll for messages multiple times. Take a screenshot after polling multiple times. **(4d)**
* Unless you delete the messages, it remains in the queue.
* Delete the message from the queue.
* Send another message, and poll for messages.
* Try changing the queue to FIFO and experiment with it. Do you find any difference ?

# Fanout to Amazon SQS queues

* Using Amazon SNS and Amazon SQS together, messages can be delivered to applications that require immediate notification of an event, and also persisted in an Amazon SQS queue for other applications to process at a later time.
* When you subscribe an Amazon SQS queue to an Amazon SNS topic, you can publish a message to the topic and Amazon SNS sends an Amazon SQS message to the subscribed queue.
* The Amazon SQS message contains the subject and message that were published to the topic along with metadata about the message in a JSON document.

## Task 3 : Using Amazon SNS and Amazon SQS together

* Subscribe the queue which you created to the topic previously created.
* Publish a message to the topic and select custom payload for each delivery protocol.
* Go to your SQS queue and poll for messages. Take a screenshot of the json object which you received. **(4e)**

DELETE all topics, subscription and queues. You are done!!

Questions: (Optional to submit the answers)

* Why do you generally couple SNS and SQS together?
* What’s the point of having two different types of queues?