

* Messaging의 정의와 필요성

- 메시징은 애플리케이션(서비스)간의 통신을 목적으로 한 매개체인 메시지를 주고 받는 것을 의미함
 - 주문을 담당하는 애플리케이션과 배송을 담당하는 애플리케이션이 각각 있다면 원활한 서비스 제공을 위해 서로 통신을 해야함
 - 주문 애플리케이션과 배송 애플리케이션이 서로 주고 받는 데이터가 바로 메시지
 - 주문 애플리케이션과 배송 애플리케이션이 직접 메시지를 주고 받는 것을 동기식 통신이라 부름
 - 주문 애플리케이션과 배송 애플리케이션 사이에 매개체(Queue)를 두고 메시지를 주고 받는 것을 비동기식 통신이라 부름
 - 주문 / 배송 애플리케이션의 데이터 처리 스펙이 다르다면 통신에 문제가 생길 수 있음, 한쪽이 메시지를 처리하는 동안 다른 한쪽은 그 메시지를 처리하지 못하고 유실할 것임
 - 각각의 애플리케이션이 직접 통신하는 것을 방지하고(decouple) 매개체(Queue)를 통해 메시지를 주고 받는 것이 이러한 문제에 대한 대안이 될 수 있음
 - SQS, SNS, Kinesis가 이러한 분리 매개체에 해당함
 - SQS : Queue Model
 - SNS : Publish/Subscribe Model
 - Kinesis : Real-time Streaming Model

* Simple Notification Service(SNS)란?

- 어떠한 주제(Topic)에 대해 관련 있는 메시지 수신자(Subscriber, 구독자)에게만 메시지를 전송하는 서비스
 - 메시지 생산자(Publisher, 게시자)는 한 개의 SNS 주제에 대해서만 메시지를 송신함
 - 메시지 수신자는 해당 주제를 구독하는 구독자를 의미하며 그 주제를 구독하는 다수의 구독자에게 전달할 수 있고 송신된 메시지를 전달받게 됨
 - 주제는 메시지의 성격을 정의함
 - 메시지를 게시할 수 있는 대상과 받을 수 있는 대상
 - 암호화 여부
 - 메시지 전송 정책
 - 구독은 메시지를 전달받을 대상으로 아래와 같은 서비스들이 대상이 될 수 있음
 - HTTP(S) Endpoint
 - 이메일
 - Amazon SQS
 - Amazon Kinesis Data Firehose
 - Lambda(메시지를 전송하여 Lambda 호출 가능)
 - 모바일 애플리케이션
 - SQS와 마찬가지로 FIFO Topic을 통해 구독자들에게 메시지를 순서대로 전달할 수 있음
 - 'Push' 기반 서비스

* SNS + SQS Fan Out Pattern

- 메시지 생산자가 다수의 SQS Queue에 동일한 메시지를 보내고자 할 때, 전송 도중 일부 메시지가 누락될 가능성이 존재함
 - 메시지 생산자(애플리케이션)가 다수의 SQS Queue에 비정상적으로 종료되거나 전달에 실패할 경우, 모든 Queue에 전달할 수 없게 됨
 - 하나의 SNS 주제에 보내고자 하는 모든 SQS Queue을 구독자로 설정하고 메시지를 전송하면 위와 같은 문제없이 전달할 수 있음
 - 이것을 SNS + SQS Fan Out Pattern이라 부름
 - 기능 구현을 위해서는 SNS로 하여금 SQS에 쓰기 작업을 할 수 있도록 IAM 허용 정책을 설정해야 함
 - SQS와 마찬가지로 SNS FIFO를 통해 메시지를 순서대로 SQS Queue에 전달할 수 있음
 - 예시 : S3 Event(한 개의 이벤트 규칙) - > SNS Topic - > 다수의 SQS Queue

* 메시지 암호화

- HTTPS API를 이용해 전송 중인 메시지를 암호화할 수 있고, KMS Key를 통해 SNS 주제에 담긴 메시지를 암호화할 수 있음
- IAM Policy를 이용해 SQS API로의 접근을 통제할 수 있음