Section 13 : Cloud Integrations

Section Introduction:

- When we start deploying multiple applications they will inevitably need to communicate with one another.

- There are two patterns of appl^n communication
  1] Synchronous Communication
     (application to application)
  2] Asynchronous / Event Based
     (application to queue to application)

- Synchronous between applications can be problematic if there are sudden spike to traffic.

- What if you need to suddenly encode 1000 videos but usually it's 10?

- In that case, it's better to decouple your applications:
  1. using SQS: queue model
  2. using SNS: Publ Sub model
  3. using kinesis: real-time data streaming model

- These services can scale idependently from our application!

Amazon SQS - Simple Queue Service

Amazon SQS - Standard Queue

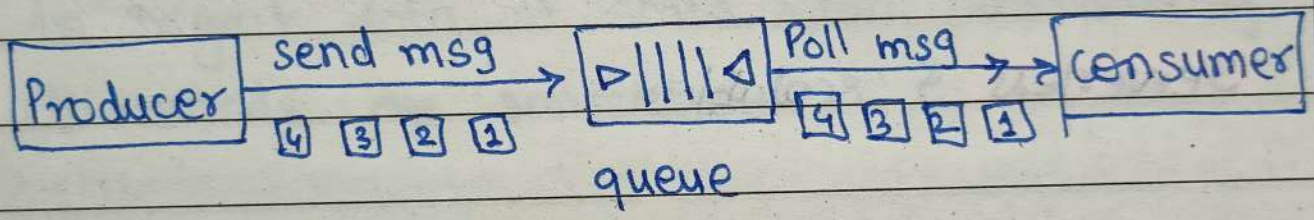
- Oldest AWS offering (over 10 years old)
- Fully managed service (- serverless), use to decouple applications.
- Scales from 1 message per second to 10,000s per second
- Default retention of messages: 4 days, maximum of 14 days.
- No limit to how many messages can be in the queue.
- Messages are deleted after they're read by consumers
- Low latency (< 10 ms on publish & receive)
- Consumers share the work to read messages & scale horizontally.

Amazon SQS - FIFO Queue

=

- FIFO :- First In First Out (Ordering of msges in the queue)

- messages are processed in order by the consumer



Amazon kinesis

- For the exam: kin
  kinesis = real-time big data streaming

- Managed service to collect, process & analyze real-time streaming data at any scale.

* extra info:-

① kinesis Data Streams:- low latency streaming to ingest data at scale from hundreds of thousands of sources.

② kinesis Data Firehouse:- load streams into S3, Redshift, ElasticSearch, etc....

③ Kinesis Data Analytics:- perform real-time analytics on streams using SQL.

④ Kinesis Video Streams:- monitor real-time video streams for analytics or ML.

Amazon SNS (Simple Notification service)

- The "event publishers" only sends message to one SNS topic.

- As many "event subscriber" as we want to listen to SNS topic notifications.

- Each subscriber to the topic will get all the messages.

- Upto 12,50,000 subscriptions per topic, 1,000,000 topics limit.

Amazon $MQ

- SQS, SNS are "Cloud-Native" services: proprietary protocols from AWS.

- Traditional applications running from on-premises may use open protocols such as: MQTT, AMQP, STOMP, Openwire, WSS,...

- When migrating to the cloud, instead of re-engineering the application to use SQS & SNS, we can use Amazon MQ.

- Amazon MQ is a managed message broker service for RabbitMQ & ActiveMQ.

- Amazon MQ doesn't "scale" as much as SQS/SNS

- Amazon MQ runs on servers, can run in multi-AZ with failover

- Amazon MQ has both que feature (~SQS) & topic features (~SNS)

Cloud Integration - Summary

① SQS :- $ Queue service in AWS
- Multiple Producers, messages are kept upto 14 days.
- Multiple consumers share the read & delete messages when done.
- Used to decouple applications in AWS.

② SNS :- Notification service in AWS.
- Subscribers : Email, Lambda, SQS, HTTP, Mobile....
- Multiple subscribers, send all messages to all of them
- No message retention.

③ Kinesis :- real-time data streaming, persistence & analysis.

④ Amazon MQ :- managed message broker for ActiveMQ & RabbitMQ in the cloud.
(MQTT, AMQP.... Protocols)