Cloud Integration

- When we start deploying multiple applications, they will inevitably need to communicate with one another
- Two Patterns:

1. Synchronous Communications (Application to Application)	2. Asynchronous /Event Based (Application to queue to Application)

Buying Service←--------→Shipping Service

Buying service→Queue→Shipping service

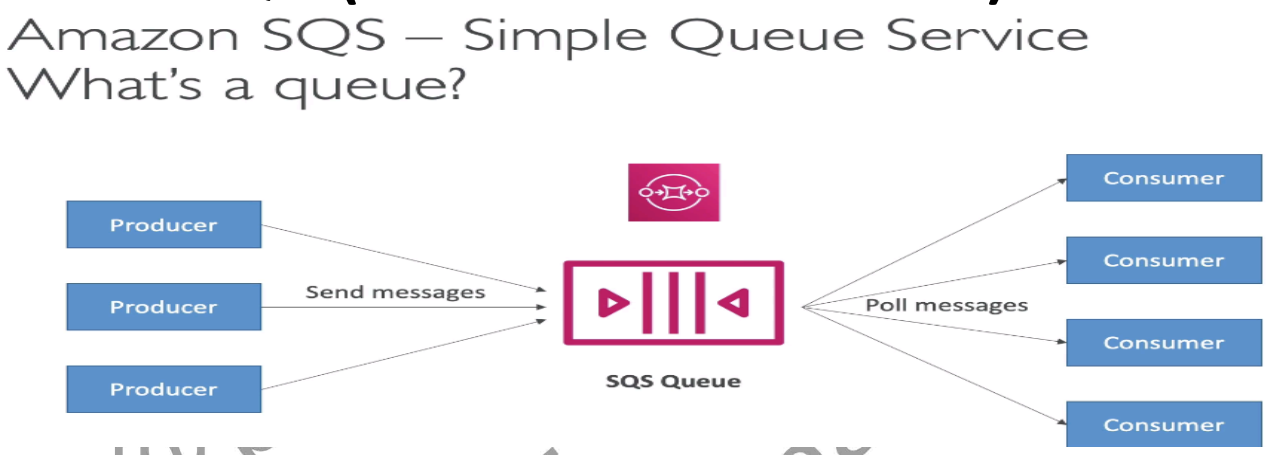
- Synchronous between application can be problematic if there are sudden spikes of traffic
- What if you need to suddenly encode 1000 video but usually is 10?

1. Using SQS: Queue Model

2. Using SNS: Pub/Sub Model

3. Using Kinesis: Real-Time Data Streaming Model (Out of Scope for the Exam)

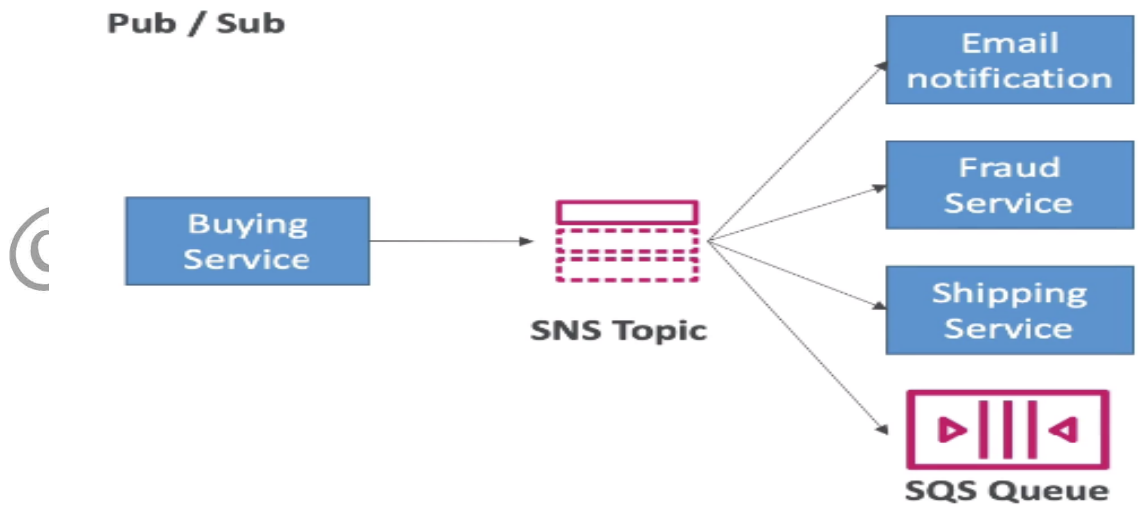
SQS: (Software As A Service)



- Oldest AWS offering (over 10 years old)
- Fully Managed Service (~Serverless), Use to decouple application
- Scales from I manage per second to 10,000s per second
- Default Retention of messages: 4 days, maximum of 14 days
- No limit how many messages on queue
- Message are deleted after they're read by consumers
- Low latency (<10 ms on publish and receive)
- Consumer share the work to read messages and Scale horizontally

SQS to Decouple between application tiers

Amazon SNS:



Amazon SNS

- The Event Publishers only sends message to one SNS topic
- As many Event subscribers as we want to listen to the SNS topic notifications
- Each subscriber to the topic will get all the messages
- Up to 10,000,000 subscribers per topic, 100,000 topics limit
-

SNS Subscribers can be:

- Http /HTTPS (with delivery retries – How many times)
- Emails, SMS message, Mobile notifications
- SQS queue (fan-out pattern), Lambda Functions(Write-your-own intergration)
-