

Section Introduction

- When we start deploying multiple applications, they will inevitably need to communicate with one another
- There are two patterns of application communication



1.

Section Introduction

- Synchronous between applications can be problematic if there are sudden spikes of 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:
 - using SQS: queue model
 - using SNS: pub/sub model
 - using Kinesis: real-time data streaming model
- These services can scale independently from our application!

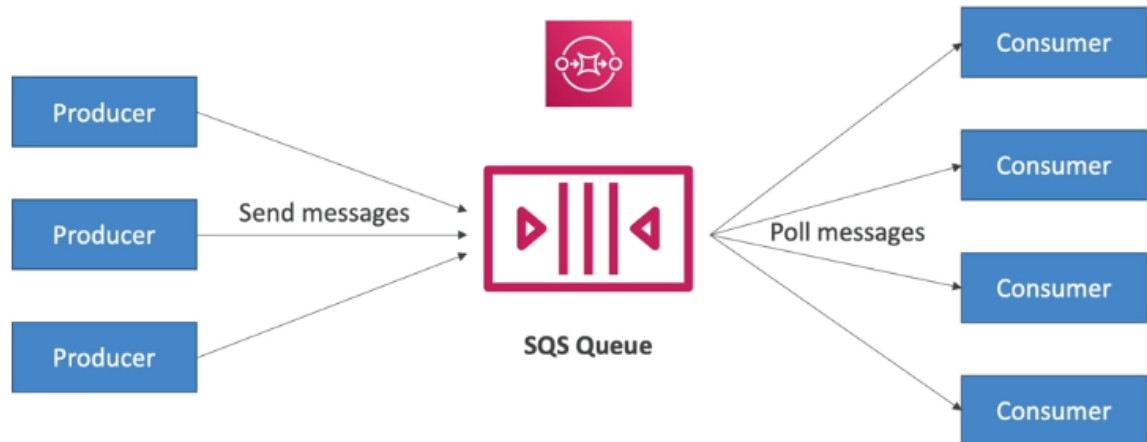
2.

3. In Synchronous communications..it will be difficult to manage..if its getting more traffic

4. Here we will have deeper look in SQS/SNS

Amazon SQS – Simple Queue Service

What's a queue?



1.

SQS stores messages. Queues are used to decouple message producers from message consumers. This is one way to architect for scale and reliability.

Let's say you've built a mobile voting app for a popular TV show and 5 to 25 million viewers are all voting at the same time (at the end of each performance). How are you going to handle that many votes in such a short space of time (say, 15 seconds)? You *could* build a significant web server tier and database back-end that could handle millions of messages per second but that would be expensive, you'd have to pre-provision for maximum expected workload, and it would not be resilient (for example to database failure or throttling). If few people voted then you're overpaying for infrastructure; if voting went crazy then votes could be lost.

A better solution would use some queuing mechanism that decoupled the voting apps from your service where the vote queue was highly scalable so it could happily absorb 10 messages/sec or 10 *million* messages/sec. Then you would have an application tier pulling messages from that queue as fast as possible to tally the votes.

2.

Messages in SQS will automatically deleted after processing completed

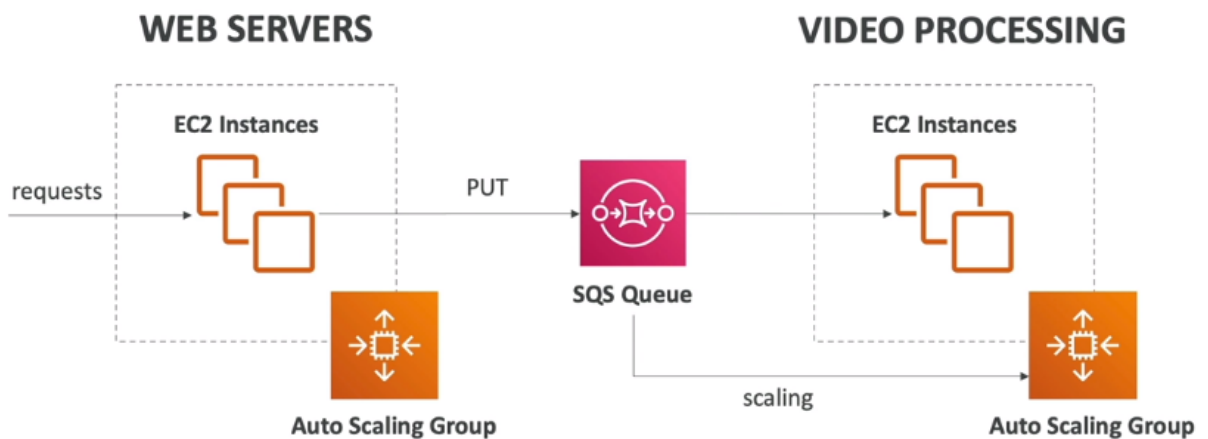
Amazon SQS – Standard Queue



- 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 and receive)
- Consumers share the work to read messages & scale horizontally

3.

SQS to decouple between application tiers



4.

Amazon SQS – FIFO Queue

- FIFO = First In First Out (ordering of messages in the queue)



- Messages are processed in order by the consumer
- 5.
 6. So when you have a normal SQS queue consumers can read messages altogether and they could be in different orders.
 7. But with Amazon SQS FIFO queues the message are going to be in order and it's just a feature you need to remember for the exam.

SQS HandsOn

1. Just go with the flow..if any doubts refer boogle

Amazon Kinesis

Amazon Kinesis



- For the exam: Kinesis = real-time big data streaming
- Managed service to collect, process, and analyze real-time streaming data at any scale
- Too detailed for the Cloud Practitioner exam but good to know:
 - Kinesis Data Streams: low latency streaming to ingest data at scale from hundreds of thousands of sources
 - Kinesis Data Firehose: load streams into S3, Redshift, ElasticSearch, etc...
 - Kinesis Data Analytics: perform real-time analytics on streams using SQL
 - Kinesis Video Streams: monitor real-time video streams for analytics or ML

1.

Sure. Here are some examples of how Amazon Kinesis can be used:

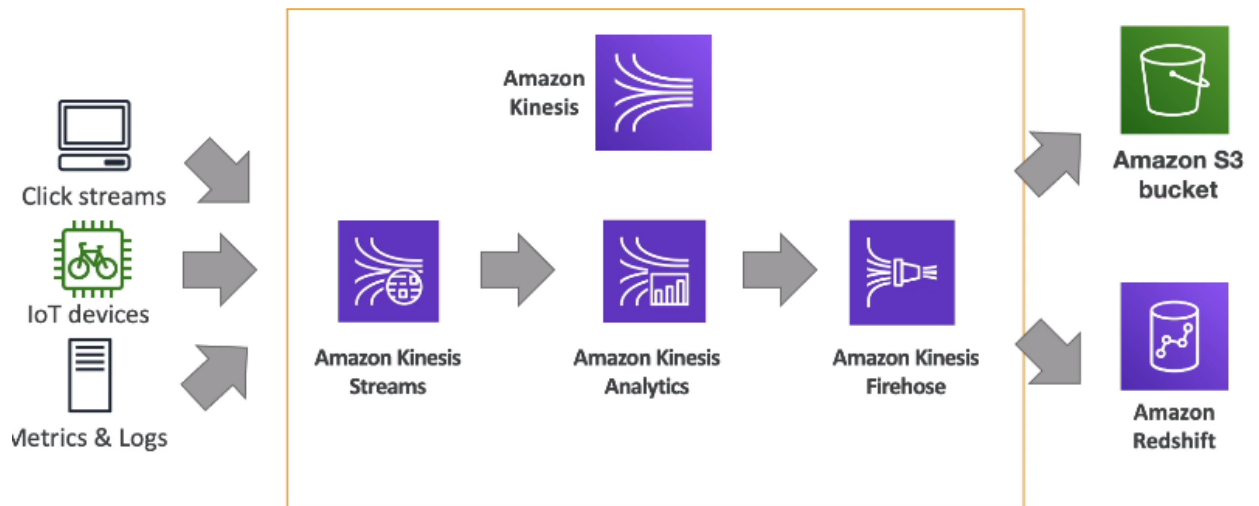
- **Streaming analytics:** Kinesis can be used to perform streaming analytics on data. This can be used to identify trends, patterns, and anomalies in real time. For example, Kinesis can be used to track customer behavior on a website or app, or to monitor the health of a machine in a factory.
- **Real-time applications:** Kinesis can be used to build real-time applications, such as fraud detection, live video streaming, and social media analytics. For example, Kinesis can be used to detect fraud in real time by tracking credit card transactions, or to live stream a sporting event to millions of viewers.
- **Machine learning:** Kinesis can be used to feed data into machine learning models. This can be used to train models to make predictions in real time. For example, Kinesis can be used to train a model to predict customer churn, or to predict the likelihood of a machine failure.


Here are some more specific examples:

- A social media company could use Kinesis to stream data from its social media platforms. This data could then be used to track trends, identify patterns, and detect anomalies in real time.

2.

Kinesis (high level overview)

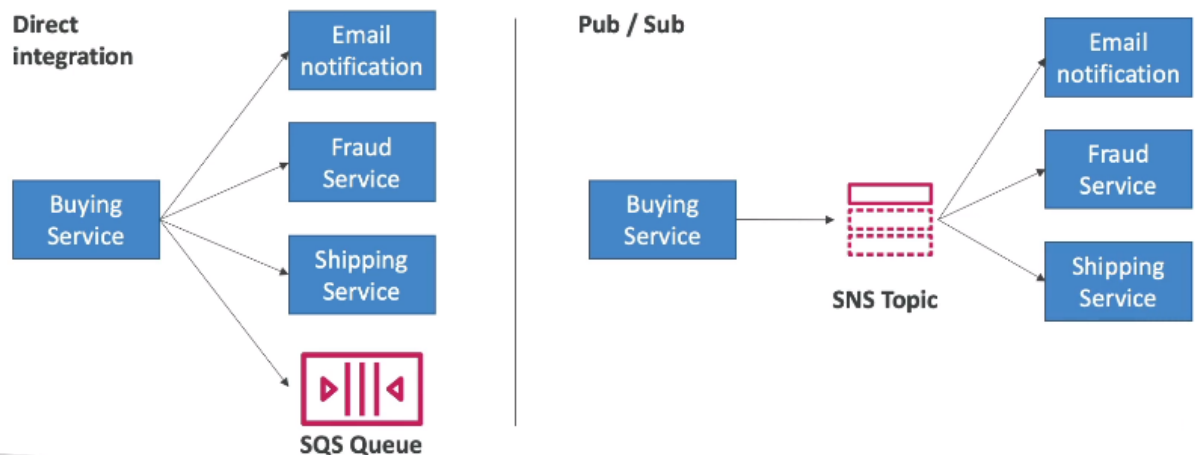


3. 
4. So we have Amazon Kinesis streams that are going to be getting data from click streams, IoT devices, metrics, log servers, all these kind of things. Then we can use Kinesis Data Analytics if we wanted to analyze this data
5. and produce output in real-time. And then we could use Kinesis Firehose to send these outputs directly into destinations such as an Amazon S3 buckets or an Amazon Redshift database where we can analyze this data and perform more analytics down the road if we wanted to.

SNS Overview:

Amazon SNS

- What if you want to send one message to many receivers?

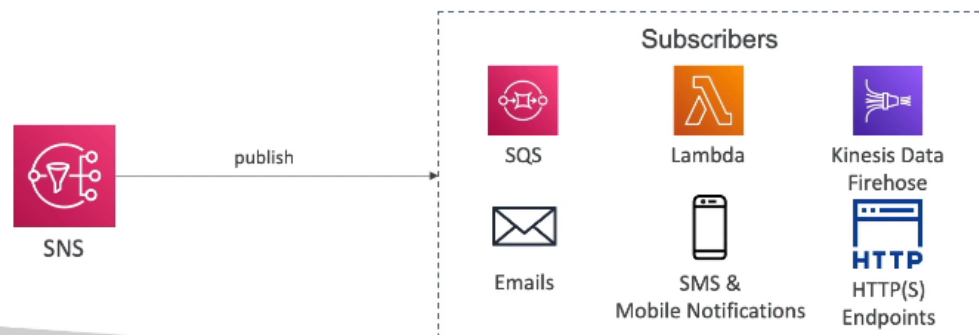


1.

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 12,500,000 subscriptions per topic, 100,000 topics limit



2.

3. So bottom line is, any time in the exam you see notification, publish subscribe, subscribers, et cetera, et cetera. Then think about Amazon SNS.

SNS Hands On:

1. Practicals example is..sending one email to the subscription list(to many recipients at once)...refer online for practicals

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 and SNS, we can use Amazon MQ
- Amazon MQ is a managed message broker service for



- 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 queue feature (~SQS) and topic features (~SNS)

1. _____

Summary

Integration Section – Summary

- SQS:
 - Queue service in AWS
 - Multiple Producers, messages are kept up to 14 days
 - Multiple Consumers share the read and delete messages when done
 - Used to decouple applications in AWS
- SNS:
 - Notification service in AWS
 - Subscribers: Email, Lambda, SQS, HTTP, Mobile...
 - Multiple Subscribers, send all messages to all of them
 - No message retention
- Kinesis: real-time data streaming, persistence and analysis
- Amazon MQ: managed message broker for ActiveMQ and RabbitMQ in the cloud (MQTT, AMQP. protocols)

1. _____