SQS & SNS

Amazon Simple Queue Service

- **Fully managed message queues for microservices, distributed systems, and serverless applications**
  - to decouple and
  - scale microservices,
  - distributed systems, and serverless applications.
- **SQS eliminates the complexity and overhead associated with managing and operating message-oriented middleware, and empowers developers to focus on differentiating work.**

Amazon Simple Queue Service

- **Using SQS, you can**
  - send,
  - store, and
  - receive messages between software components at any volume
- **without losing messages or requiring other services to be available.**

Amazon Simple Queue Service

- **Standard queues** offer maximum throughput, best-effort ordering, and at-least-once delivery.
- **SQS FIFO queues** are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

Amazon Simple Notification Service

- **Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.**

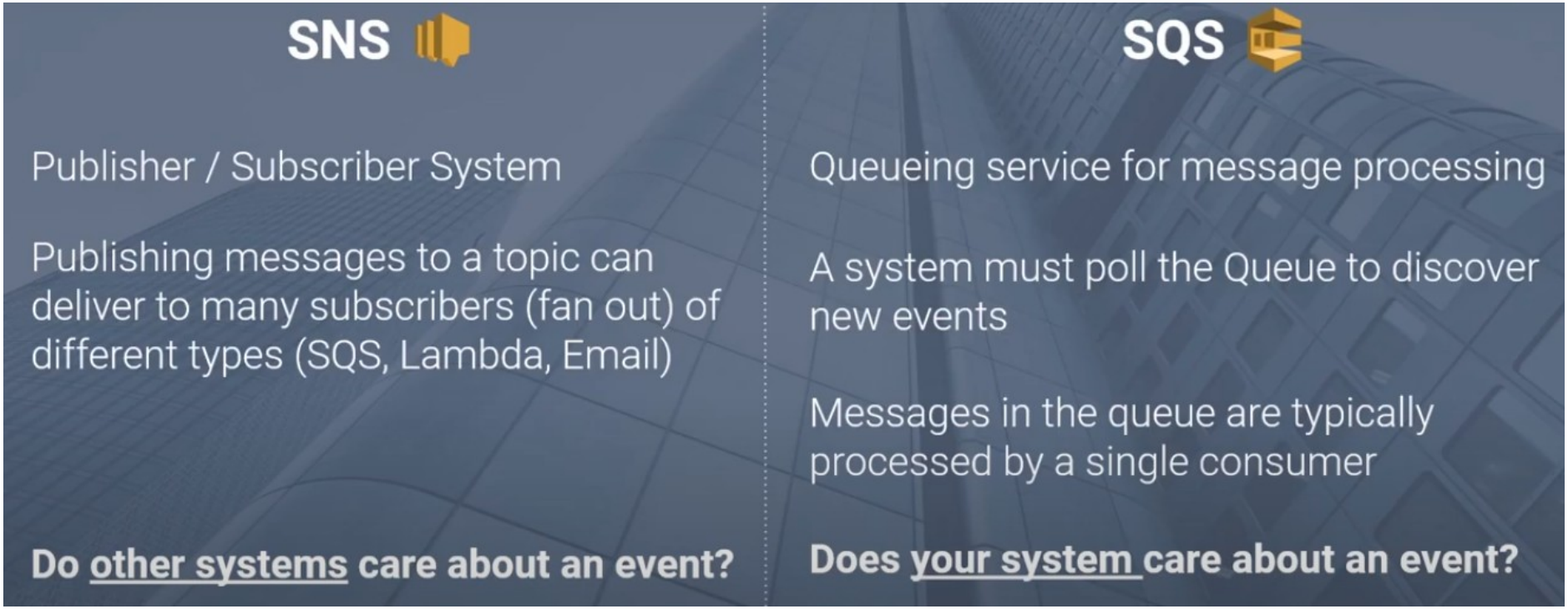
Amazon Simple Notification Service

- **Fully managed pub/sub messaging, SMS, email, and mobile push notifications**

Amazon Simple Notification Service

- **The A2A pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging between distributed systems, microservices, and event-driven serverless applications.**
- **Using Amazon SNS topics, your publisher systems can fanout messages to a large number of subscriber systems including Amazon SQS queues, AWS Lambda functions and HTTPS endpoints, for parallel processing, and Amazon Kinesis Data Firehose.**
- **The A2P functionality enables you to send messages to users at scale via SMS, mobile push, and email.**

SNS vs SQS


SNS
Publisher / Subscriber System
Publishing messages to a topic can deliver to many subscribers (fan out) of different types (SQS, Lambda, Email)
Do other systems care about an event?
SQS
Queueing service for message processing
A system must poll the Queue to discover new events
Messages in the queue are typically processed by a single consumer
Does your system care about an event?

Benefits

- **Eliminate administrative overhead**
- **Reliably deliver messages**
- **Keep sensitive data secure**
- **Scale elastically and cost-effectively**

Demo

Thank you!!!

References

- **https://docs.aws.amazon.com/index.html**