

# Simple Queue Service (SQS)

SQS, SNS, Kinesis and all the good stuff. How they work in general and together.

**SQS** is a web service that gives you access to a message queue that can be used to store messages while waiting for the messages to be processed. It is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates and another consumes.

In other words, a queue is a temporary repository for messages that are awaiting process.

You can also think of SQS as a buffer between two components, producing data and a component consuming data. This means that a queue consumes data. This means that a queue could potentially resolve issues that come about when a producer is producing at a larger rate than what the consumer can handle.

## Two types of Queues

1. Standard (default)
2. FIFO

### Standard Queues

Standard queues let you have nearly an unlimited number of transactions per second. Standard queues guarantee that a message is delivered at least once. They also provide best-effort ordering which ensures that messages are gradually delivered in the same order as they are received.

### FIFO Queues

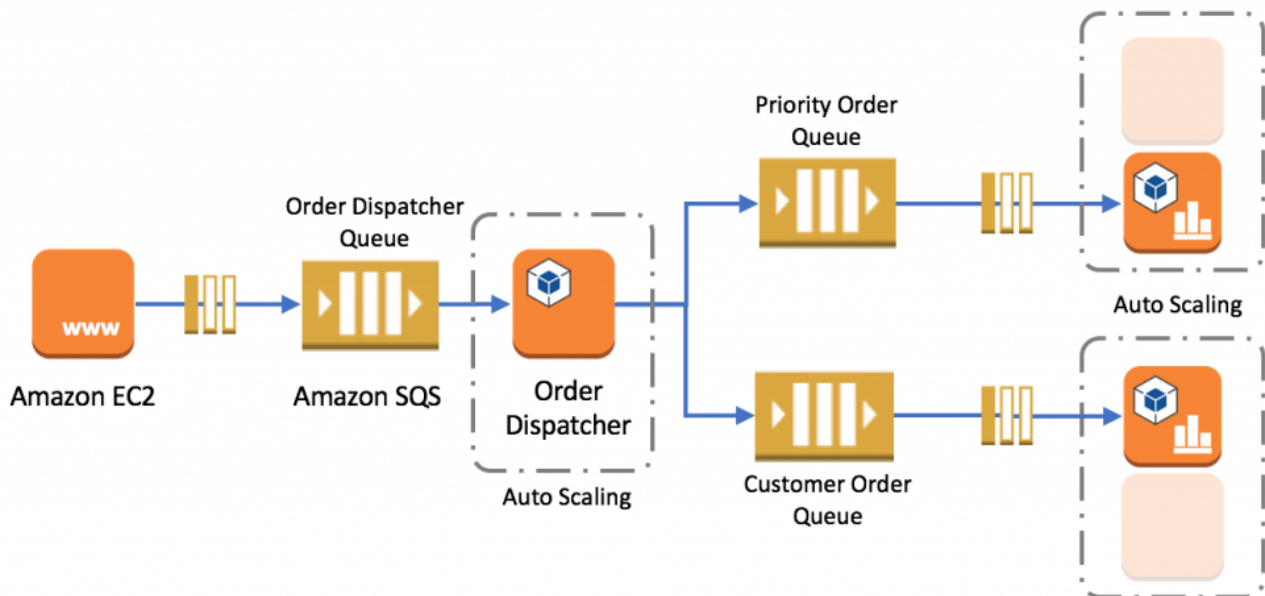
The most important feature of the FIFO (First In First Out) queue is the first in first out delivery coupled with exactly once processing.

The order in which messages are sent and received are strictly preserved and a message is delivered once and remains available until a consumer processes

it and deletes it.

**SQS** – is a pull based queue

The below outlines the parts of a distributed messaging system and explains the lifecycle of an Amazon SQS message.

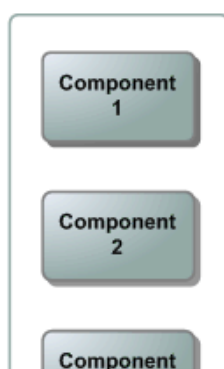


## Distributed Queues

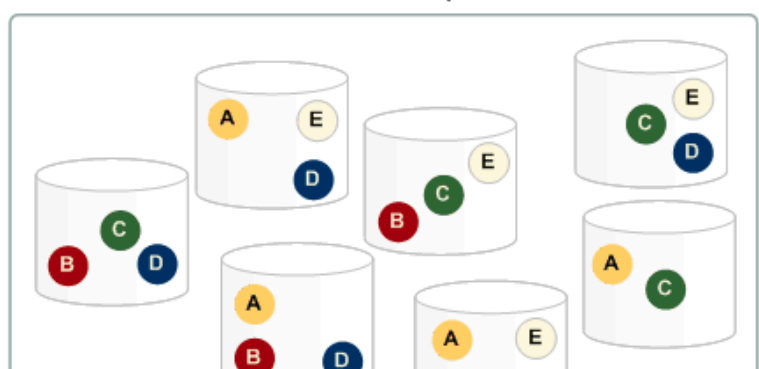
There are three main parts in a distributed messaging system: the components of your distributed system, your queue (distributed on Amazon SQS servers), and the messages in the queue.

In the following scenario, your system has several components that send messages to the queue and receive messages from the queue. The queue (which holds messages A through E) redundantly stores the messages across multiple Amazon SQS servers.

**Your Distributed System's Components**

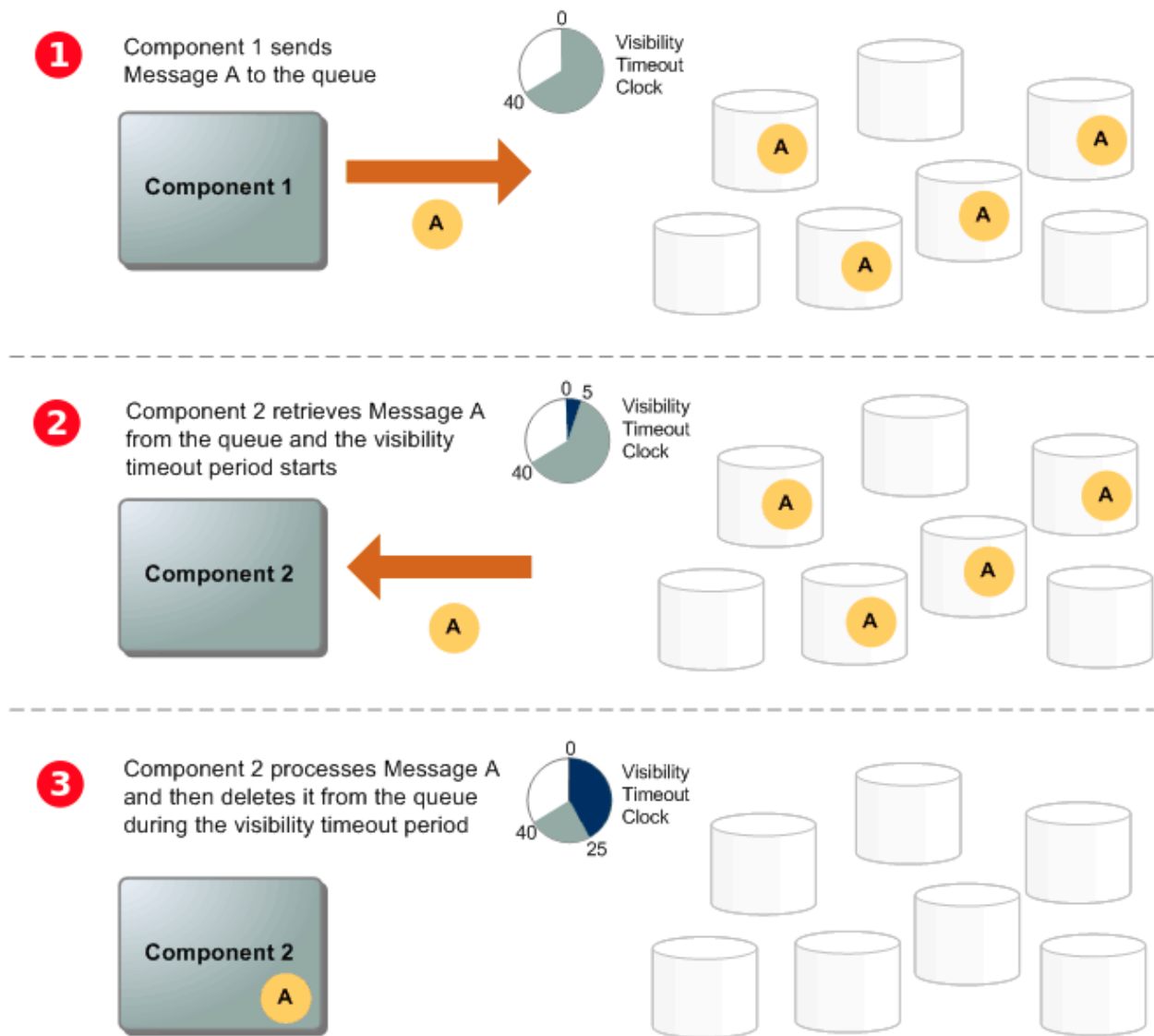


**Your Queue (Distributed on SQS Servers)**



## Message Lifecycle

The following scenario describes the lifecycle of an Amazon SQS message in a queue, from creation to deletion.



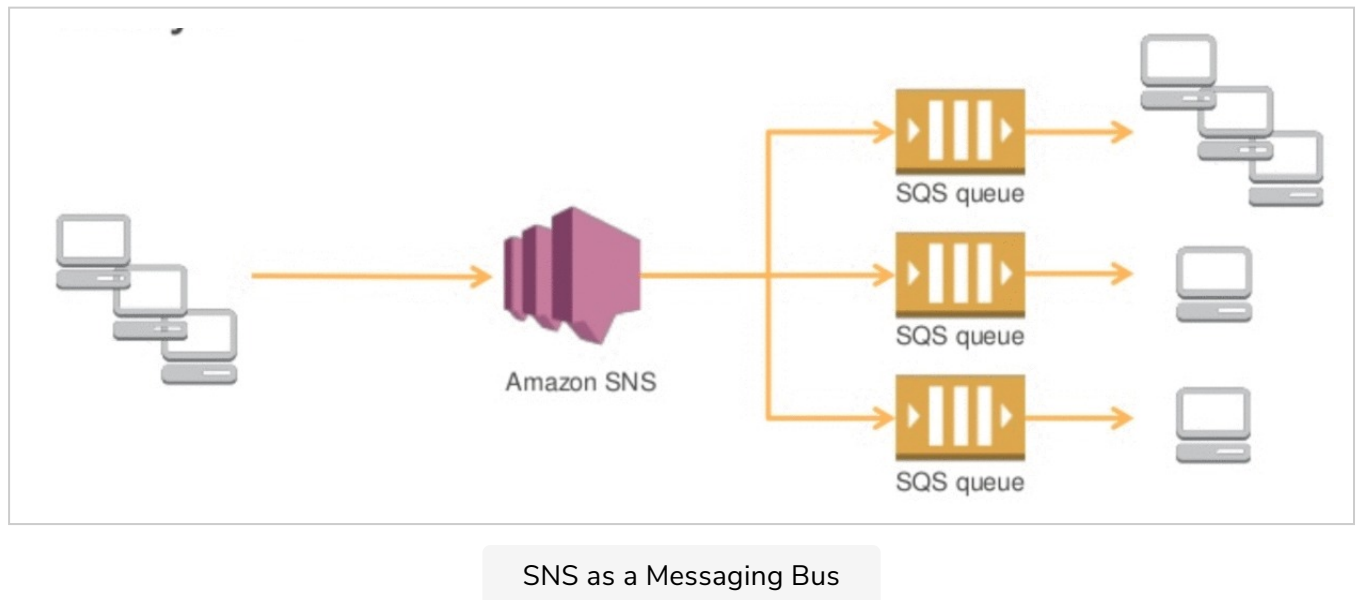
1. A producer (component 1) sends message A to a queue, and the message is distributed across the Amazon SQS servers redundantly.
2. When a consumer (component 2) is ready to process messages, it consumes messages from the queue, and message A is returned. While message A is being processed, it remains in the queue and isn't returned to subsequent receive requests for the duration of the visibility timeout.
3. The consumer (component 2) deletes message A from the queue to prevent the message from being received and processed again when the visibility timeout expires.

# SNS – Simple Notification Service

Is a web service that makes it easy to set up and operate and send notifications from the cloud. It provides the capacity to publish messages from an application and immediately deliver them to subscribers or other application

SNS – Push notification to mobile devices

1. Also has SMS text Capability
2. SNS can trigger email, HTTP, Lambda functions.



## Elastic Transcoder

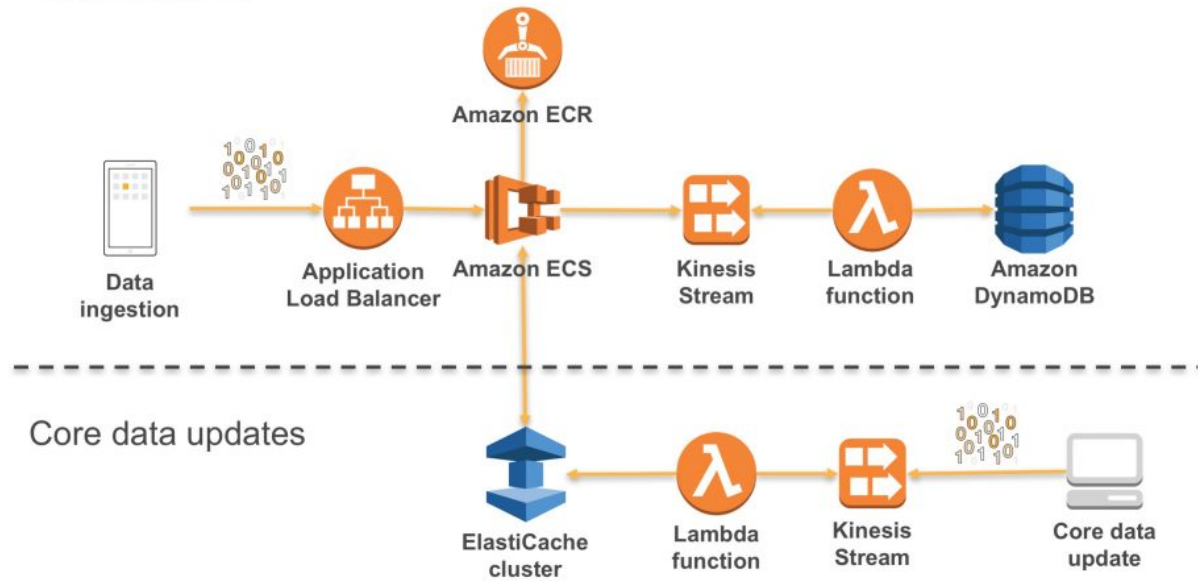
A service that converts any given video format into other formats specified by the user.

## Kinesis

**Streaming Data:** Data that is generated continuously by thousands or millions of data sources or data points that are captured and that needs to be picked up and processed.

1. Social Network data
2. Mobile games data
3. Stock Price data

## Data collection



Kinesis Data Collection & Core Data Updates

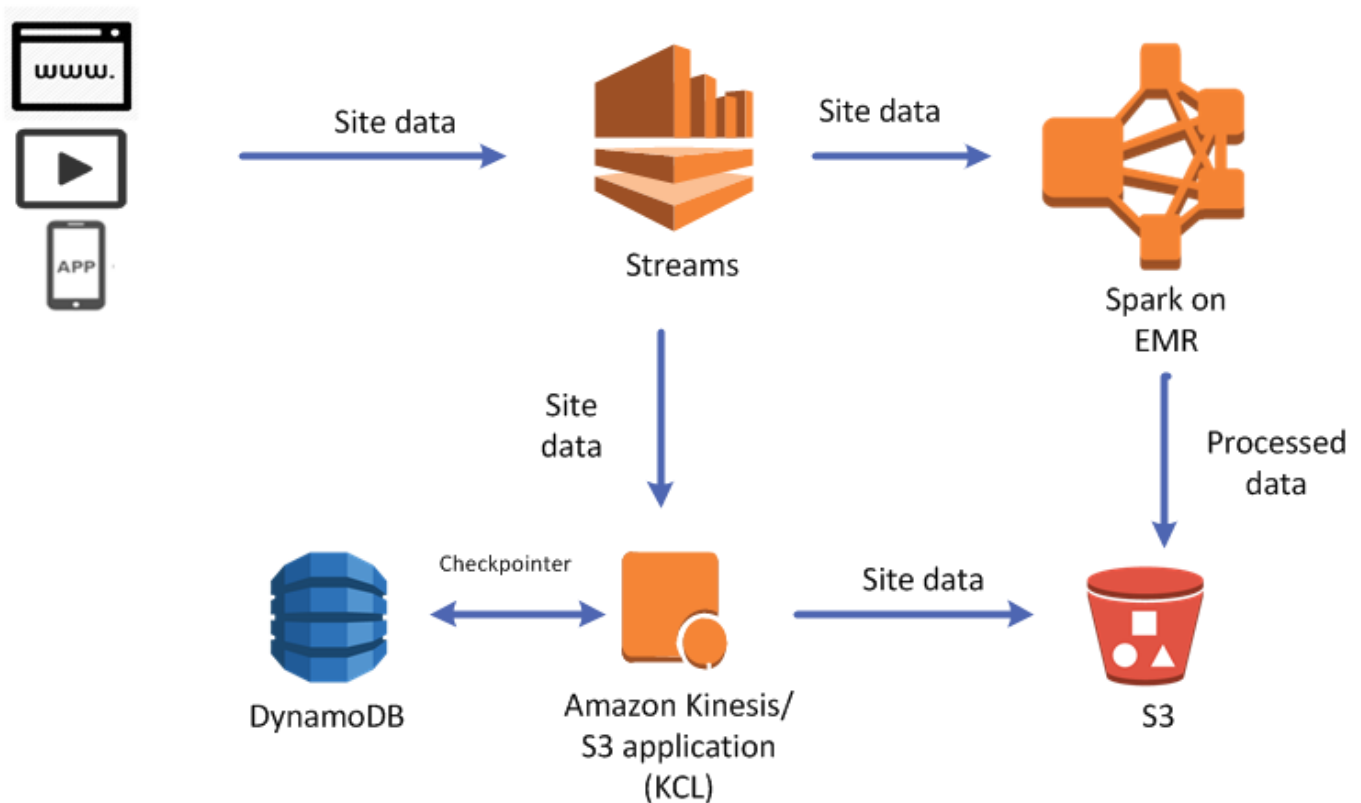
**Kinesis:** Amazon Kinesis is a platform on AWS to send your streaming data too. It is Amazon's version of Kafka.

Kinesis has 3 Services

1. Kinesis Streams
2. Kinesis Firehose
3. Kinesis Analytics

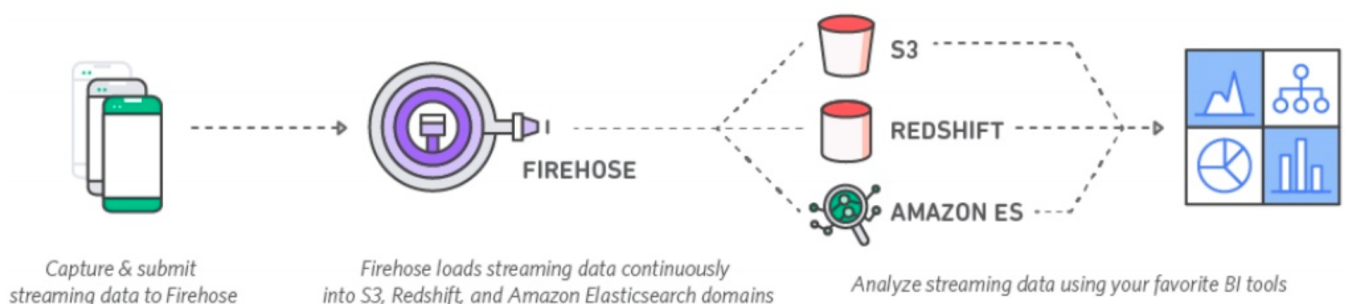
**Kinesis Streams** Consist of shards (cluster in Kafka)

You can have 5 Transactions per seconds for reads, a high data read rate.



Persist Streaming Data to Amazon S3 using Amazon Kinesis Firehose and AWS Lambda

**Kinesis Firehose** - is the easiest way to reliably load streaming data into data stores and analytics tools. It can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, enabling near real-time analytics with existing business intelligence tools and dashboards you're already using today.



**Kinesis Analytics** - is used to process streaming data in real time with standard SQL without having to learn new programming languages or processing frameworks. Amazon Kinesis Data Analytics enables you to query streaming data or build entire streaming applications using SQL, so that you can gain actionable insights and respond to your business and customer needs promptly.



Realtime Data Analytics