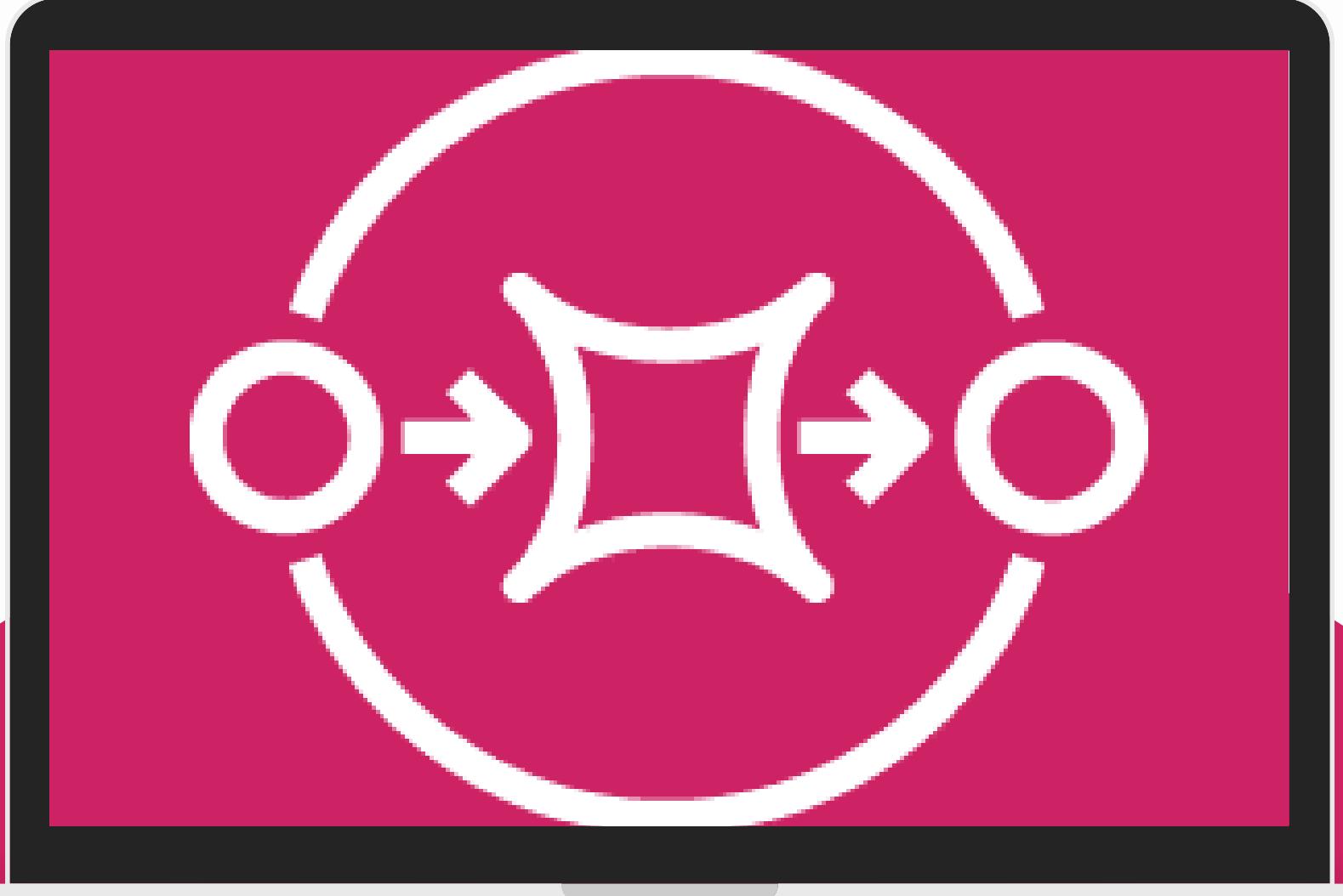


AWS SQS

AWS Simple Queueing Service



Overview

-  Introduction 01
-  Key Features 02
-  How it Works 03
-  Long Polling 04
-  Use Cases 05
-  Benefits 06

01)

INTRODUCTION



Introduction

A fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS offers two types of message queues: Standard queues and FIFO (First-In-First-Out) queues.

02)

KEY FEATURES

Key Features

Decoupling

Helps in decoupling the components of a cloud application, allowing each component to operate independently.

Scalability

Automatically scales to handle a virtually unlimited number of messages per second.

Fully Managed

AWS handles the underlying infrastructure and scaling, allowing you to focus on your application logic.

Key Features

Message Durability

Messages are stored redundantly across multiple availability zones (AZs).

Dead-Letter Queues

Helps in handling messages that can't be processed successfully by redirecting them to a dead-letter queue.

Visibility Timeout

Temporarily hides a message from other consumers while it is being processed, preventing multiple consumers from processing the same message.

Queue Types

Standard Queue

Provides maximum throughput, best-effort ordering, and at-least-once delivery.

FIFO Queue

Ensures that messages are processed exactly once, in the exact order they are sent.

Security

IAM

Supports AWS IAM policies for access control.

Encryption

Provides server-side encryption (SSE) to protect the contents of messages

03)

HOW IT WORKS

How it Works

Producer

- Sends messages to the SQS queue.
- Messages are stored in the queue until they are retrieved by consumers.

Consumer

- Retrieves and processes messages from the queue.
- Can be a single consumer or multiple consumers working in parallel.

Message Lifecycle

- Messages are retained in the queue for a configurable retention period (from 1 minute to 14 days).
- Once a consumer processes a message, it deletes the message from the queue.

04)

LONG POLLING

Long Polling

Long Polling in Amazon SQS helps efficiently retrieve messages by waiting for a message to arrive in the queue, reducing the need for constant polling.

How it Works

1. Extended Wait Time:

- When you send a receive message request, SQS can wait up to 20 seconds for a message to become available>
- This reduces the number of empty responses compared to short polling, which returns immediately.

•

2. Configuration:

- Long polling can be configured per request using the WaitTimeSeconds parameter.
- You can also set a default long polling wait time for a queue.

05) USE CASES

USE CASES

Microservices Communication

Allows microservices to communicate with each other asynchronously and scale independently.

Batch Processing

Enables batch processing of tasks by queuing them for processing by worker nodes.

Job Scheduling

Allows tasks to be scheduled and processed at a later time.

USE CASES

Load Leveling

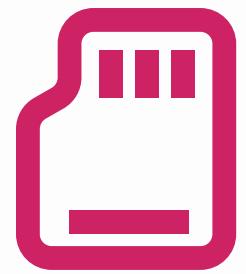
Smooths out intermittent heavy loads by queuing requests and processing them at a controlled rate.

Event-Driven Architecture

Facilitates building event-driven applications by decoupling event producers and consumers.

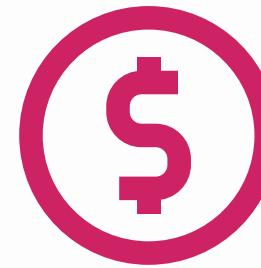
06) **BENEFITS**

Benefits



01) Reliability

Provides high availability and durability by storing messages redundantly across multiple AZs.



02) Flexibility

Supports a variety of use cases, from simple task queues to complex workflows



Ease of Use

Simple API for sending, receiving, and deleting messages.

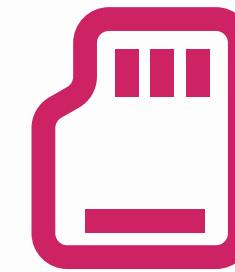
Easily integrates with other AWS services, such as AWS Lambda, AWS EC2, and AWS ECS.

Benefits



04) Cost Effective

Pay-as-you-go pricing model, with no upfront costs or minimum fees. Charges based on the number of requests and the amount of data transferred.

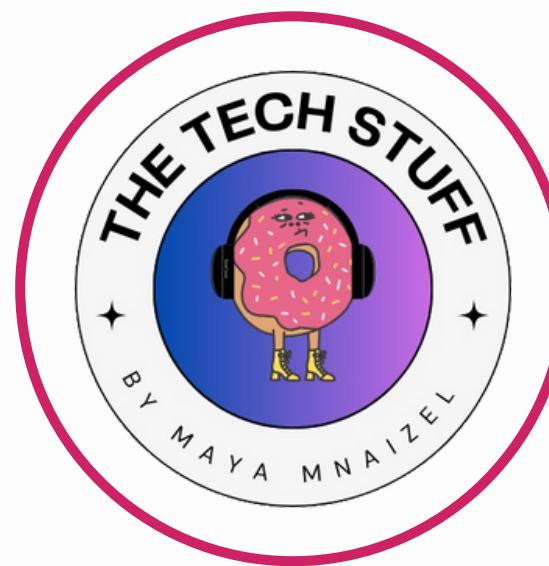


05) Security

End-to-end encryption ensures that messages are protected at rest and in transit.

IAM policies allow fine-grained access control to SQS resources.

THANK YOU!



The Tech Stuff



mayamnaizel2013@gmail.com