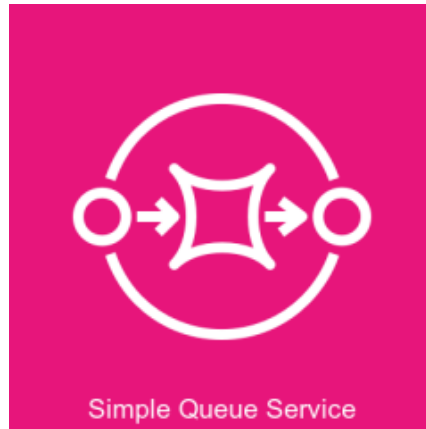


# Amazon Simple Queue Service (AWS SQS)



AWS SQS (Simple Queue Service) is a fully managed message queuing service that enables decoupling and asynchronous communication between microservices, distributed systems, and serverless applications.



**Producer (Publisher)** – Applications or services (EC2, Lambda, etc.) send messages to the SQS queue.

**SQS Queue (Standard/FIFO)** – Holds messages until consumers retrieve them.

**Consumers (Subscribers)** – Applications (EC2, Lambda, ECS, etc.) poll the queue to process messages.

## Key Features of AWS SQS

1. **Decoupling of Components**
  - Enables loose coupling between microservices and applications.
2. **Two Queue Types:**

- **Standard Queue** (default) – Best-effort ordering, high throughput.
- **FIFO Queue** – Ensures exactly-once message delivery and order preservation.
- 3. **Scalability & Reliability**
  - Can process millions of messages per second.
  - Distributed architecture ensures high availability.
- 4. **Message Retention**
  - Stores messages for **up to 14 days** (default is 4 days).
- 5. **Message Visibility Timeout**
  - Prevents other consumers from processing the same message while it's being processed.
- 6. **Dead Letter Queue (DLQ)**
  - Helps capture messages that fail processing multiple times.
- 7. **Long Polling & Short Polling**
  - **Long polling** reduces unnecessary API calls by waiting for messages.
- 8. **Security & Access Control**
  - Uses **IAM policies** for fine-grained access.
  - Supports **encryption at rest** using AWS KMS.
  - **VPC Endpoint** support for private communication.

## Use Cases

- **Asynchronous Processing** – Decouple services for better performance.
- **Order Processing Systems** – Ensure reliable, sequential order fulfillment.
- **Log Processing** – Queue log messages for batch processing.
- **Background Tasks** – Queue tasks to be processed later.
- **Retry Mechanisms** – Automatically retry failed jobs.

## How AWS SQS Works?

1. **Create an SQS Queue**
  - Choose **Standard** or **FIFO** queue type.
2. **Send Messages to Queue**
  - Producer sends messages to the queue.
3. **Process Messages by Consumers**
  - One or more consumers pull messages from the queue.
4. **Delete Messages After Processing**
  - Consumers delete messages after successful processing.

# AWS SQS vs. AWS SNS

Feature	SQS (Queue-based)	SNS (Pub/Sub)
Delivery Type	Pull-based	Push-based
Message Processing	One-to-one or one-to-many	One-to-many (fan-out)
Message Ordering	FIFO ensures ordering	No ordering guarantee
Use Case	Background jobs, event-driven processing	Real-time notifications, microservices communication

## Best Practices

- ✓ Use **FIFO queues** for exactly-once and ordered processing.
- ✓ Implement **Dead Letter Queues (DLQs)** to handle failed messages.
- ✓ Use **long polling** to reduce API calls and cost.
- ✓ Enable **server-side encryption** for sensitive data.
- ✓ Monitor queues using **Amazon CloudWatch**.