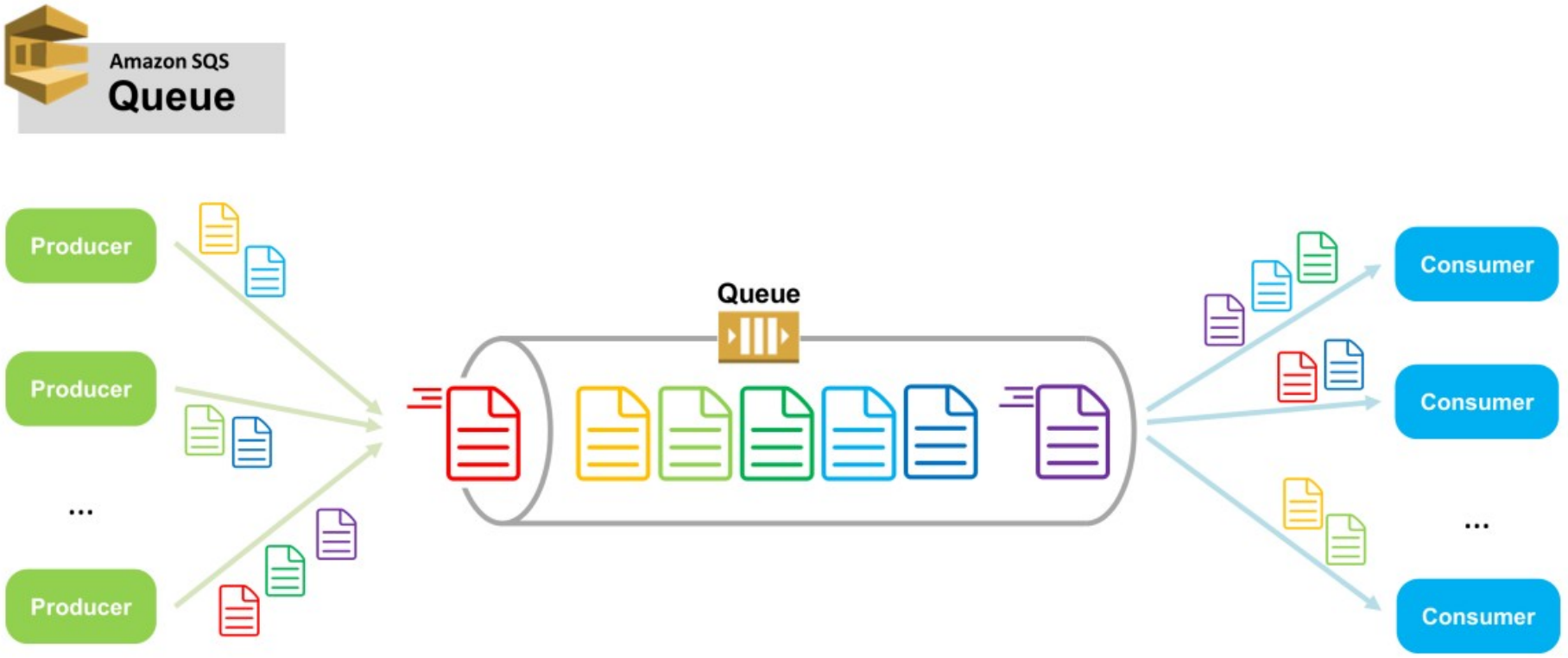
AWS SQS

What is AWS SQS?

- **Amazon** SQS is a web service that gives you access to a message queue that can be used to store messages while waiting for a computer to process them.
- Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component.
- A queue is a temporary repository for messages that are awaiting processing

SQS

SQS Aerial view



SQS

- Using Amazon SQS, you can decouple the components of an application so they run independently, easing message management between components.
- Any component of a distributed application can store messages in a fail-safe queue.
- Messages can contain up to 256 KB(max 2GB) of text in any format. Any component can later retrieve the messages programmatically using the Amazon SQS API

SQS

- The queue acts as a buffer between the component producing and saving data, and the component receiving the data for processing.

- This means the queue resolves issues that arise if the producer is producing work faster than the consumer can process it, or if the producer or consumer are only intermittently connected to the network.

Types of SQS queues

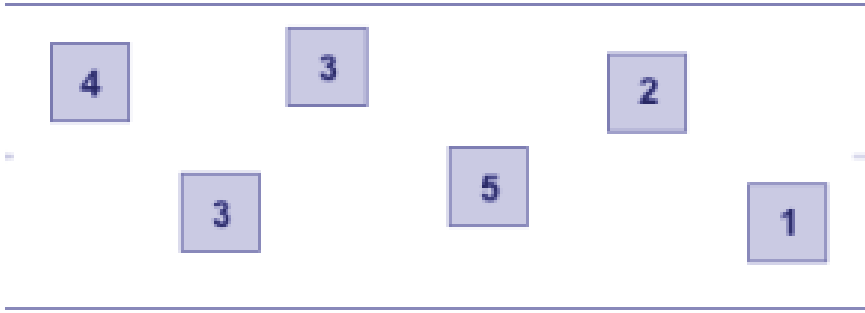
- Standard queues(default)
- FIFO queues

Types of SQS queues

Standard queues(default)

- Amazon SQS offers standard as the default queue type.
- A standard queue lets you have a nearly-unlimited number of transactions per second.
- Standard queues guarantee that a messages is delivered at least once.
- However, occasionally more than one copy of a message might be delivered out of order.
- Standard queues provides best-effort ordering which ensures that messages are generally delivered in the same order as they were sent. But no guarantee.

Standard Queues



FIFO Queue

FIFO Queue

- The FIFO Queue complements the standard queue. The most important features of this queue type are FIFO(First-in-First-Out) delivery and exactly-once processing.
- The order in which messages are send and received is strictly preserved and a message is delivered once and remains available until a consumer processes and deletes it.
- Duplicates are not introduced into the queue.
- FIFO queues also supports messages groups that allowed multiple ordered messages groups with in a single queue.
- FIFO queues are limited to 300 transactions per second(TPS) but have all the capabilities of standard queues.

FIFO Queues

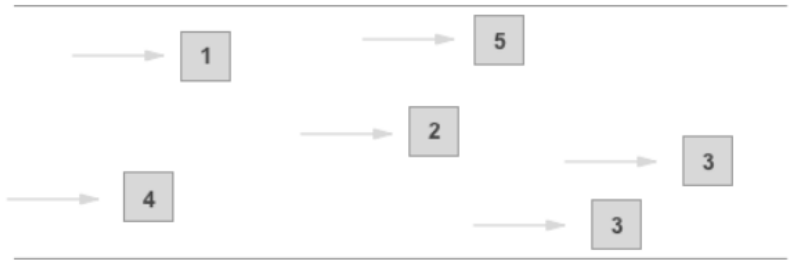
Standard Vs FIFO Queue

Standard Queue

Unlimited Throughput: Standard queues support a nearly unlimited number of transactions per second (TPS) per API action.

At-Least-Once Delivery: A message is delivered at least once, but occasionally more than one copy of a message is delivered.

Best-Effort Ordering: Occasionally, messages might be delivered in an order different from which they were sent.



Send data between applications when the throughput is important, for example:

- Decouple live user requests from intensive background work: let users upload media while resizing or encoding it.
- Allocate tasks to multiple worker nodes: process a high number of credit card validation requests.
- Batch messages for future processing: schedule multiple entries to be added to a database.

FIFO Queue

High Throughput: FIFO queues support up to 300 messages per second (300 send, receive, or delete operations per second). When you batch 10 messages per operation (maximum), FIFO queues can support up to 3,000 messages per second. To request a limit increase, file a support request.

First-In-First-out Delivery: The order in which messages are sent and received is strictly preserved.

Exactly-Once Processing: A message is delivered once and remains available until a consumer processes and deletes it. Duplicates are not introduced into the queue.

Send data between applications when the order of events is important, for example:

- Ensure that user-entered commands are executed in the right order.
- Display the correct product price by sending price modifications in the right order.
- Prevent a student from enrolling in a course before registering for an account.

SQS Exam Tips

- **SQS is** pull based, not push based.
- Messages are 256KB in size
- Messages can be kept in the queue from 1 minute to 14 days, default retention period is 4 days.
- Visibility Time Out is the amount of time that the message is in invisible in the SQS queue after a reader picks up that message. Provided the job is processed before the visibility time out expires, the message will then be deleted from the queue.
- IF the job is not processed with in that time, the message will become visible again and another reader will process it. This could result in the same message being delivered twice
- Visibility time out maximum is 12 hours

SQS Exam Tips

- **SQS g**uarantee that your messages will be at least processed once.
- Amazon SQS long polling is a way to retrieve messages from your amazon SQS queues. While the regular short polling return immediately(even if the message queue being polled is empty).
- Long polling doesn’t return a response until a message arrives in the message queue or the long poll timeout.
- While your queue is empty, long poll requests wait up to 20 seconds for the next text message to arrive. Long poll requests cost the same amount as regular requests.
- A queue can be created in any region.
- Each 64KB ‘chunk’ of payload is billed as 1 request.
- Messages can be sent and read simultaneously.

SQS Exam Tips

- **When** a message is received, it becomes “locked” while being processed. This keeps other computers from processing the message simultaneously.
- If the message processing fails, the lock will expire and the message will be available gain. In the case where the application needs more time for processing, the “lock” timeout can be changed automatically via the ChangeMessageVisibility operation.
- Developers can securely share SQS queues with other accounts. It can also be restricted by IP address and time of day
- SSE protects the content of messages in SQS queues using keys managed in KMS.
- Developers can handle stuck messages(messages that have not been successfully processed by a consumer) with Dead Letter Queues.
- When the maximum receive count is exceeded for a message it will be moved to DLQ associated with the original queue.
- Developers can setup separate consumer processes for DLQs which can help analyze and understand why messages are getting stuck. DLQ’s must be same type as STD or

Queue Attributes

Default Visibility Timeout	30	seconds	Value must be between 0 seconds and 12 hours.
Message Retention Period	4	days	Value must be between 1 minute and 14 days.
Maximum Message Size	256	KB	Value must be between 1 and 256 KB.
Delivery Delay	0	seconds	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time	0	seconds	Value must be between 0 and 20 seconds.
Content-Based Deduplication			

Dead Letter Queue Settings

Use Redrive Policy		
Dead Letter Queue		Value must be an existing queue name.
Maximum Receives		Value must be between 1 and 1000.

Server-Side Encryption (SSE) Settings

Use SSE			
AWS KMS Customer Master Key (CMK)			
Data Key Reuse Period			This value must be between 1 minute and 24 hours.