SQS – A Messaging service

Messaging

Simple Queue Service

Simple Notification Service

SES

Simple Queue Service (SQS)

Amazon SQS

queue

message

- Fully managed message queues for reliably communicating between applications
- A queue is a temporary repository of messages that are awaiting processing

Competitors:
- MSMQ by Microsoft
- MQ by IBM (mainly used for mainframe and modern servers)
- ActiveMQ by Apache

amazon web services

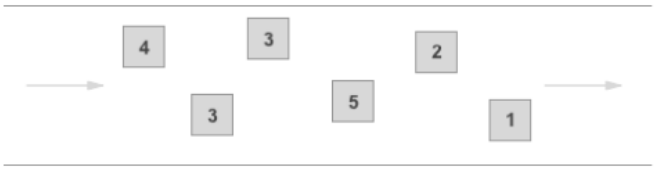
SQS (Message Queue Service)

- Amazon Simple Queue Service (Amazon SQS) is a reliable, highly-scalable hosted queue for storing messages as they travel between applications or micro-services.
- Amazon SQS moves data between distributed application components and helps you decouple these components.
- Messages can contain up to 256 kbytes of text in any format
- The queue acts as a buffer between producer and consumer
- Do you think queue will help if
  - Producer is producing faster than consumer can process?
  - What if producer and consumer are only intermittently connected?
  - What if consumer goes down for an hour. Will the messages be lost?

http://docs.aws.amazon.com/cli/latest/reference/sqs/index.html

SQS (Message Queue Service)

What type of Queue do you need?

Standard Queue	FIFO Queue
Available in all regions. **High Throughput** – Standard queues have nearly-unlimited transactions per second (TPS). **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.	Available in US West (Oregon) and US East (Ohio). **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. **Limited Throughput** – 300 transactions per second (TPS).
	
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.	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.

http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/Welcome.html

SQS (Message Queue Service)

What Can you Use Amazon SQS For?

Use Amazon SQS when you need each unique message to be consumed only once and for cases such as the following:

- **Decoupling the components of an application** – You have a queue of work items and want to track the successful completion of each item independently. Amazon SQS tracks the ACK/FAIL results, so the application does not have to maintain a persistent checkpoint or cursor. After a configured visibility timeout, Amazon SQS deletes acknowledged messages and redelivers failed messages.

- **Configuring individual message delay** – You have a job queue and you need to schedule individual jobs with a delay. With standard queues, you can configure individual messages to have a delay of up to 15 minutes.

- **Dynamically increasing concurrency or throughput at read time** – You have a work queue and want to add more consumers until the backlog is cleared. Amazon SQS requires no pre-provisioning.

- **Scaling transparently** – You buffer requests and the load changes as a result of occasional load spikes or the natural growth of your business. Because Amazon SQS can process each buffered request independently, Amazon SQS can scale transparently to handle the load without any provisioning instructions from you.

http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/Welcome.html

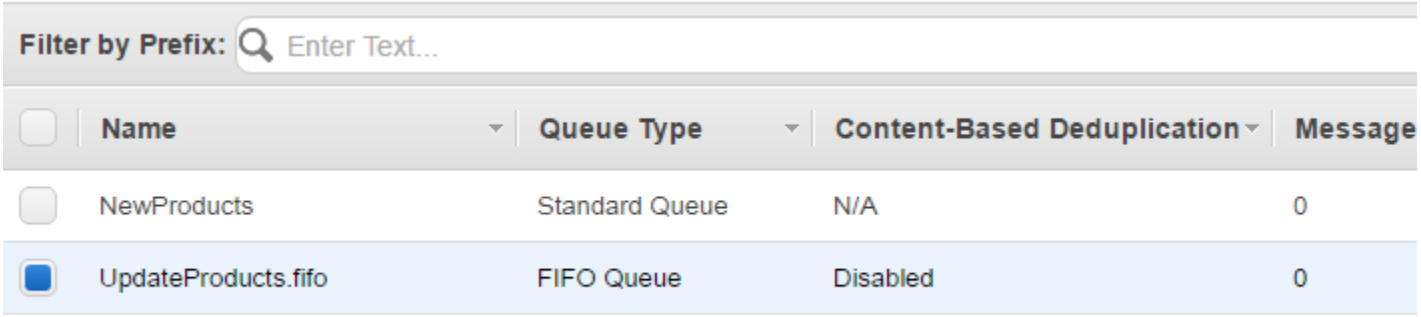
Exam Tips

- When asked what type of service to use and the keyword is decouple, most like SQS is the answer
- There is no priority field so in exam if there is a question about priority, most likely the answer is to create two queues. One for high priority and one for low priority. Consumer should always check messages first for high priority queue before going to another queue.
- Consumer usually ‘polls’ the queue and ‘pulls’ the messages.

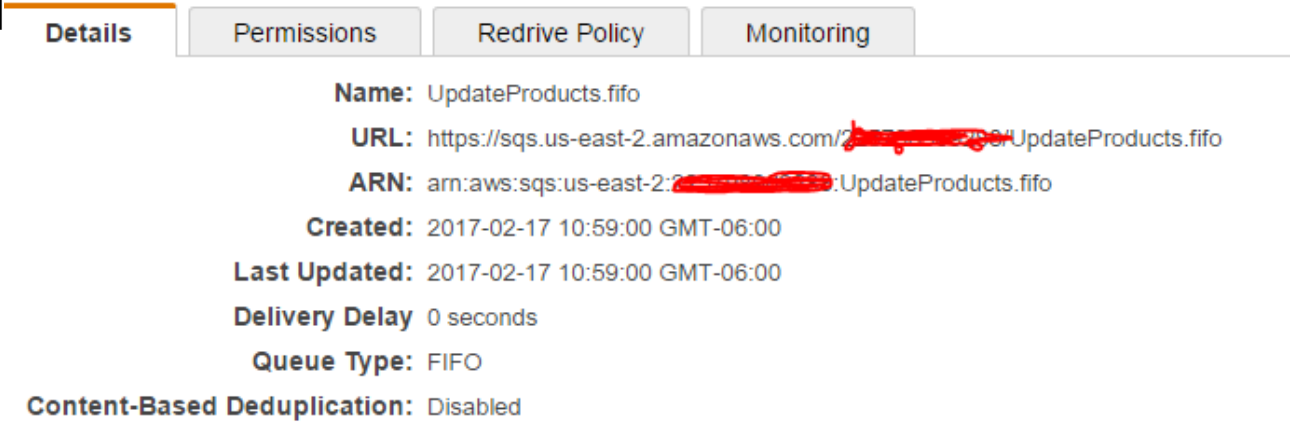
http://docs.aws.amazon.com/cli/latest/reference/sqs/index.html

SQS – Hands on

1. Created two queues, Standard and FIFO
2. Verify that after creating the queues, you can look at URL and ARN of your queue and your screen looks something like this



The **Queue Type** column helps you distinguish standard queues from FIFO q



Your queue's **URL** and **ARN** are displayed on the **Details** tab.

amazon web services