CLOUD COMPUTING APPLICATIONS

Amazon Simple Queue Service

Prof. Reza Farivar

AWS SQS

- Simple Queue Service
- **put-get-delete** paradigm
  - It requires a consumer to explicitly state that its data has finished processing the message it pulled from the queue
  - The message data is kept safe with the queue and gets deleted from the queue only after confirmation that it has been processed
- Multiple Producers and Consumers
- Pull model
  - The consumers must explicitly pull the queue
- Redundant infrastructure

Guaranteed message delivery: At Least Once

- when a process retrieves the message from the queue, it temporarily makes this message invisible
- When the client informs the queue that it has finished processing the message, SQS deletes this message from the queue
- If the client does not respond back to the queue in a specific amount of time, SQS makes it visible again
- Generally in order message delivery

Guaranteed message delivery: Exactly Once*

- * only true under limited conditions
- FIFO
  - queues work in conjunction with the publisher APIs to avoid introducing duplicate messages
  - Content-based deduplication:
    - SQS generates an SHA-256 hash of the body of the message to generate the content-based message deduplication ID
    - When an application sends a message to the FIFO queue with a message deduplication ID, it is used to deduplicate
  - Message order guarantee
- Bandwidth limits

Recommended reading: https://sookocheff.com/post/messaging/dissecting-sqs-fifo-queues/

Short Pull vs Long pull

- Short Pull: Returns empty if there is nothing in the queue
- What then? Pull again
  - AWS charges based on number of requests
- Solution: Long pull
  - `ReceiveMessageWaitTime`
    - The maximum amount of time that a long-polling receive call waits for a message to become available before returning an empty response

Handling Failures

- Visibility of messages in the Queue
  - A consumer has to explicitly “delete” a message after processing is done
  - Timeout period → message becomes visible again
- What if there is something wrong with a particular message?
  - Infinite loop of visible-invisible?
  - Dead Letter Queue (DLQ)
    - Store failed messages that are not successfully consumed by consumer processes
    - Isolate unconsumed or failed messages and subsequently troubleshoot these messages to determine the reason for their failures
  - Redrive Policy
    - If a queue fails to process a message a predefined number of times, that message is moved to the DLQ