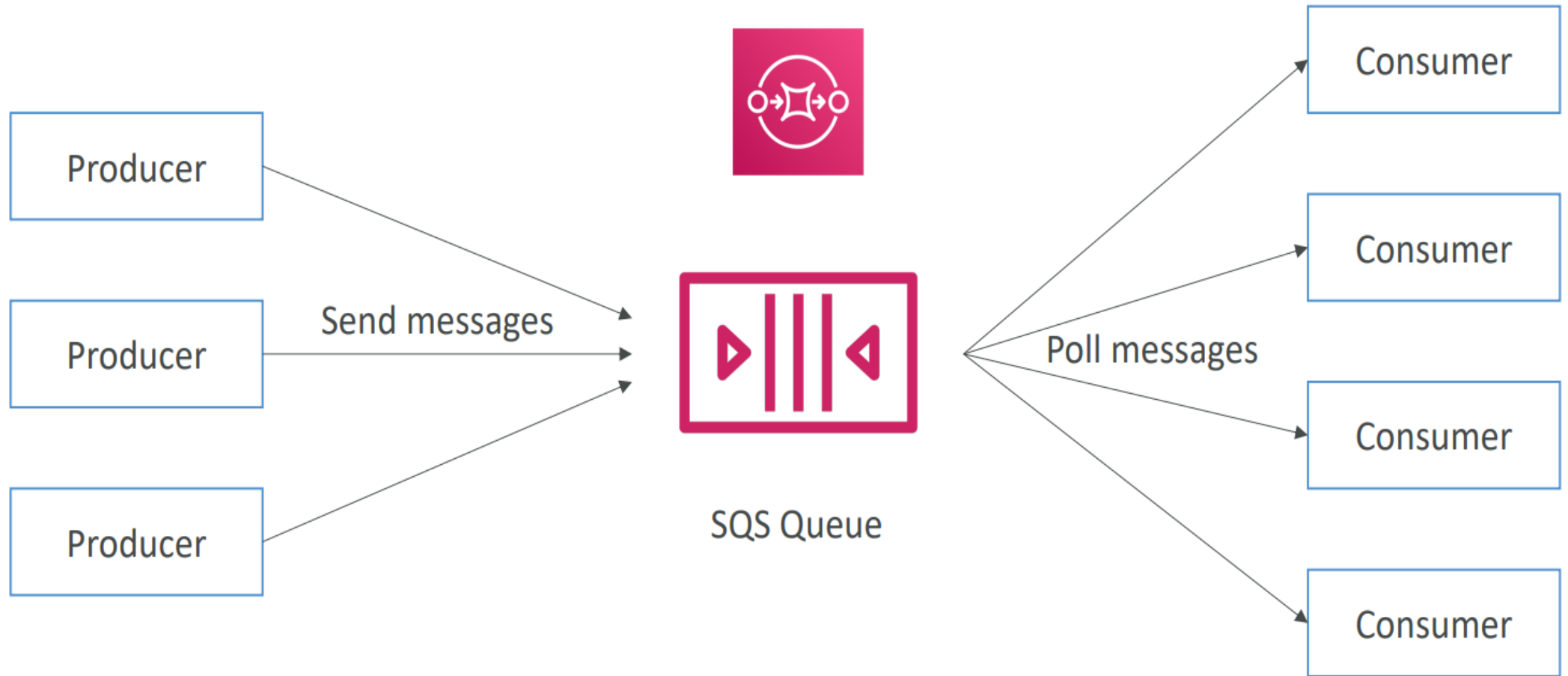


# Amazon SQS

# What is Queue?



# Amazon SQS : Standard Queue

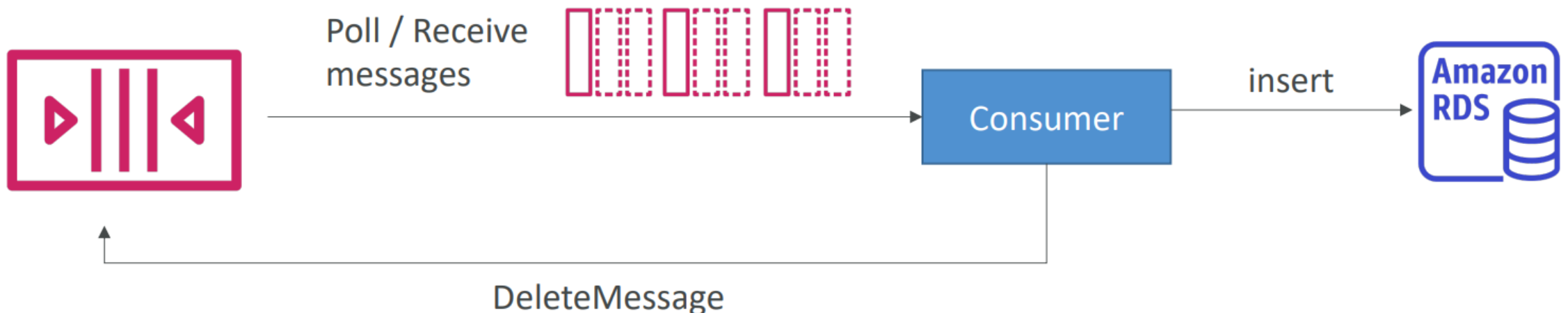
- Oldest offering (over 10 years old)
- Fully managed service, used to decouple applications
- Attributes:
  - Unlimited throughput, unlimited number of messages in queue
  - Default retention of messages: 4 days, maximum of 14 days
  - Low latency (<10ms on publish and receive)
  - Limitation of 256KB per message sent
- Can have duplicate messages
- Can have out of order messages

# Amazon SQS : Standard Queue

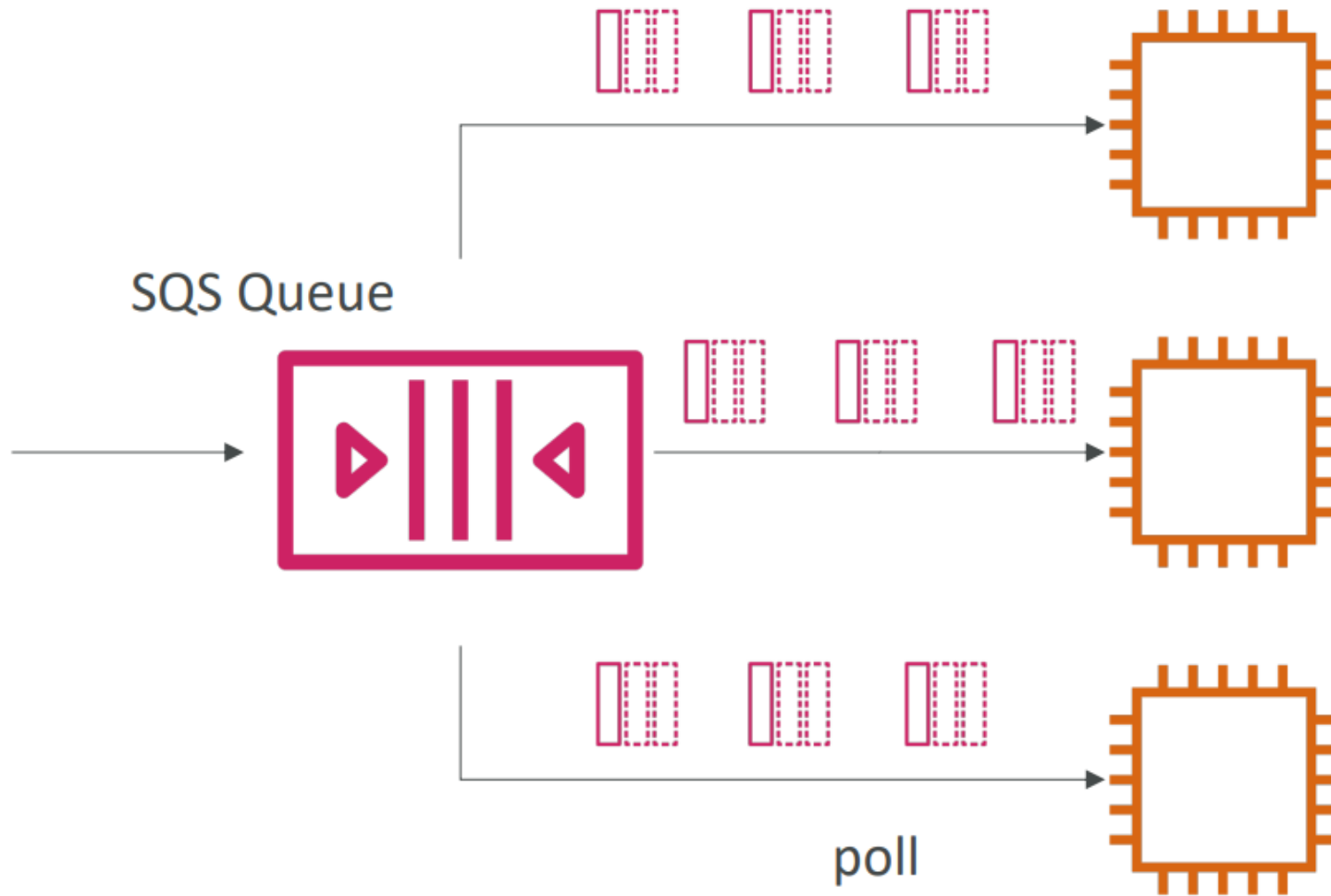
- Produced to SQS using the SDK (SendMessage API)
- The message is persisted in SQS until a consumer deletes it
- Message retention: default 4 days, up to 14 days
- Example: send an order to be processed
  - Order id
  - Customer id
  - Additional data

# Amazon SQS : Standard Queue

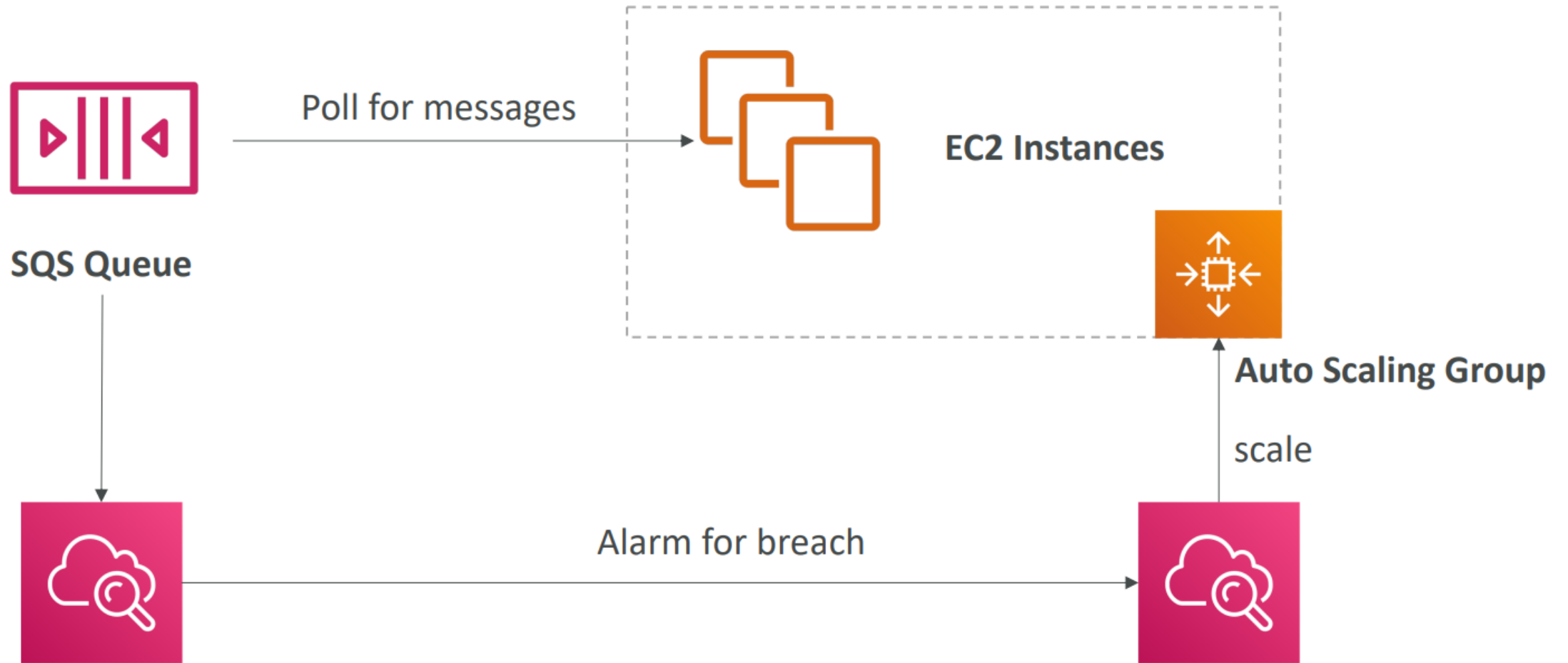
- Consumers (running on EC2 instances, servers, or AWS Lambda)...
- Poll SQS for messages (receive up to 10 messages at a time)
- Process the messages (example: insert the message into an RDS database)
- Delete the messages using the DeleteMessage API



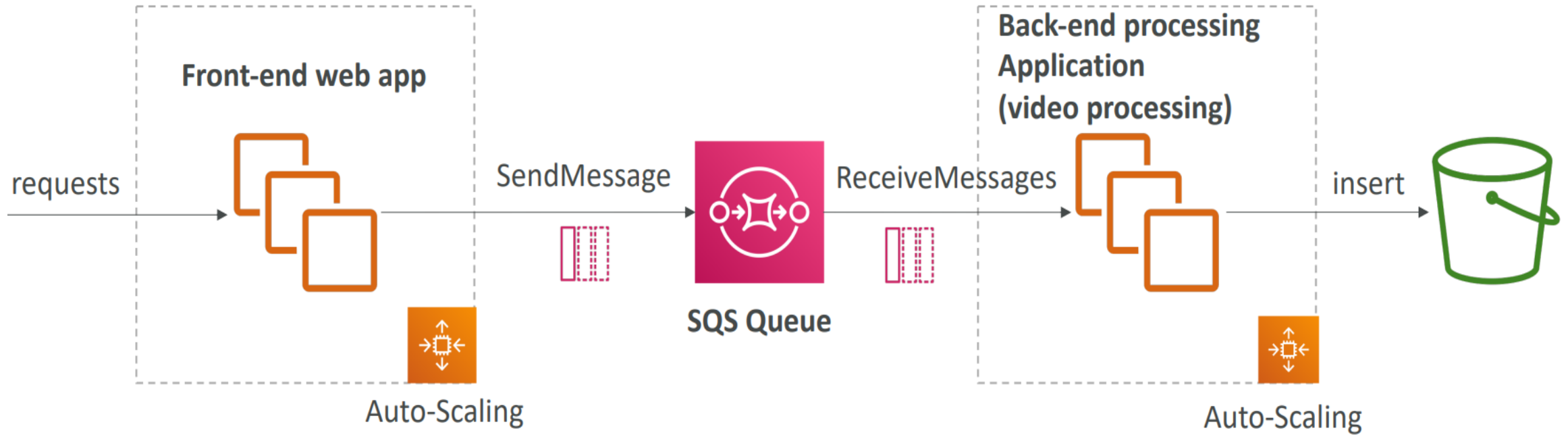
# SQS Processing with Multiple Instances



# SQS with AutoScaling Groups



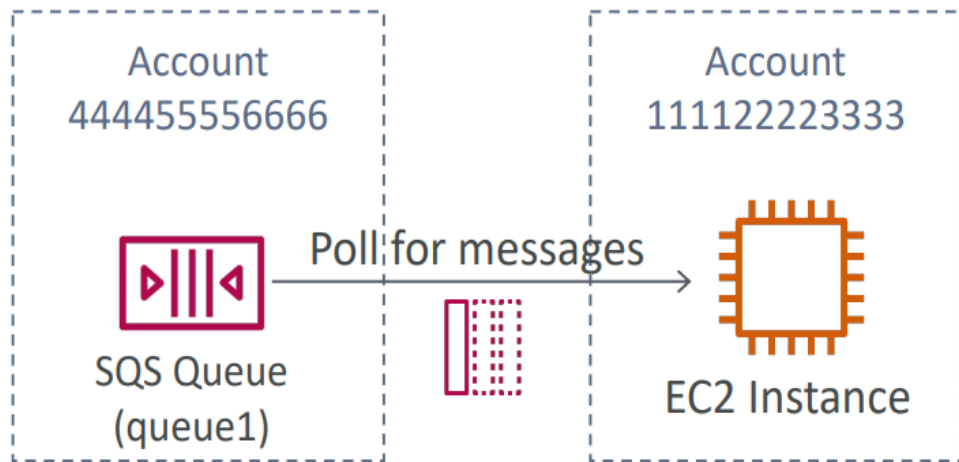
# SQS to decouple between application tiers





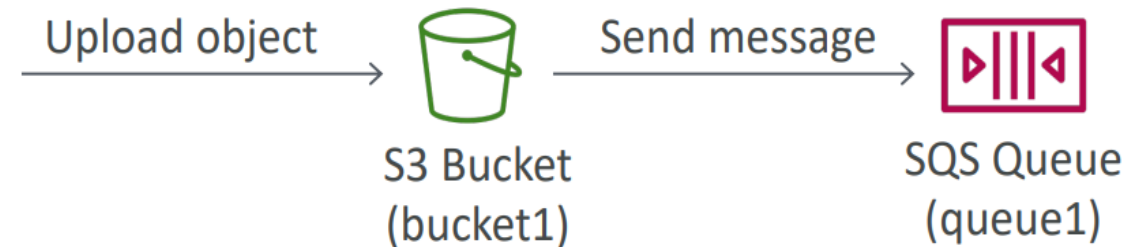
# SQS Access Control Policy

## Cross Account Access



```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Principal": { "AWS": [ "111122223333" ] },  
    "Action": [ "sqs:ReceiveMessage" ],  
    "Resource": "arn:aws:sqs:us-east-1:444455556666:queue1"  
  }]  
}
```

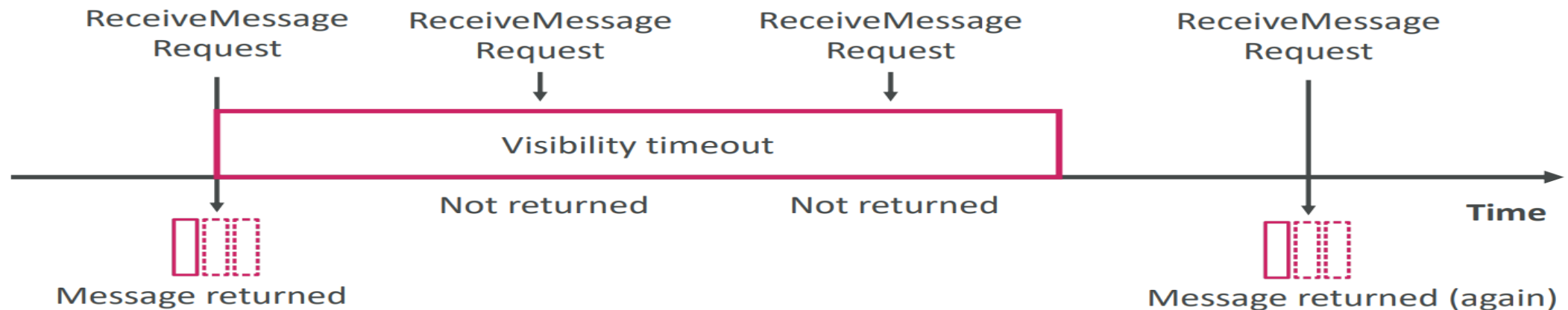
## Publish S3 Event Notifications To SQS Queue



```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Principal": { "AWS": "*" },  
    "Action": [ "sqs:SendMessage" ],  
    "Resource": "arn:aws:sqs:us-east-1:444455556666:queue1",  
    "Condition": {  
      "ArnLike": { "aws:SourceArn": "arn:aws:s3:*:*:bucket1" },  
      "StringEquals": { "aws:SourceAccount": "<bucket1_owner_account_id>" },  
    }  
  }]  
}
```

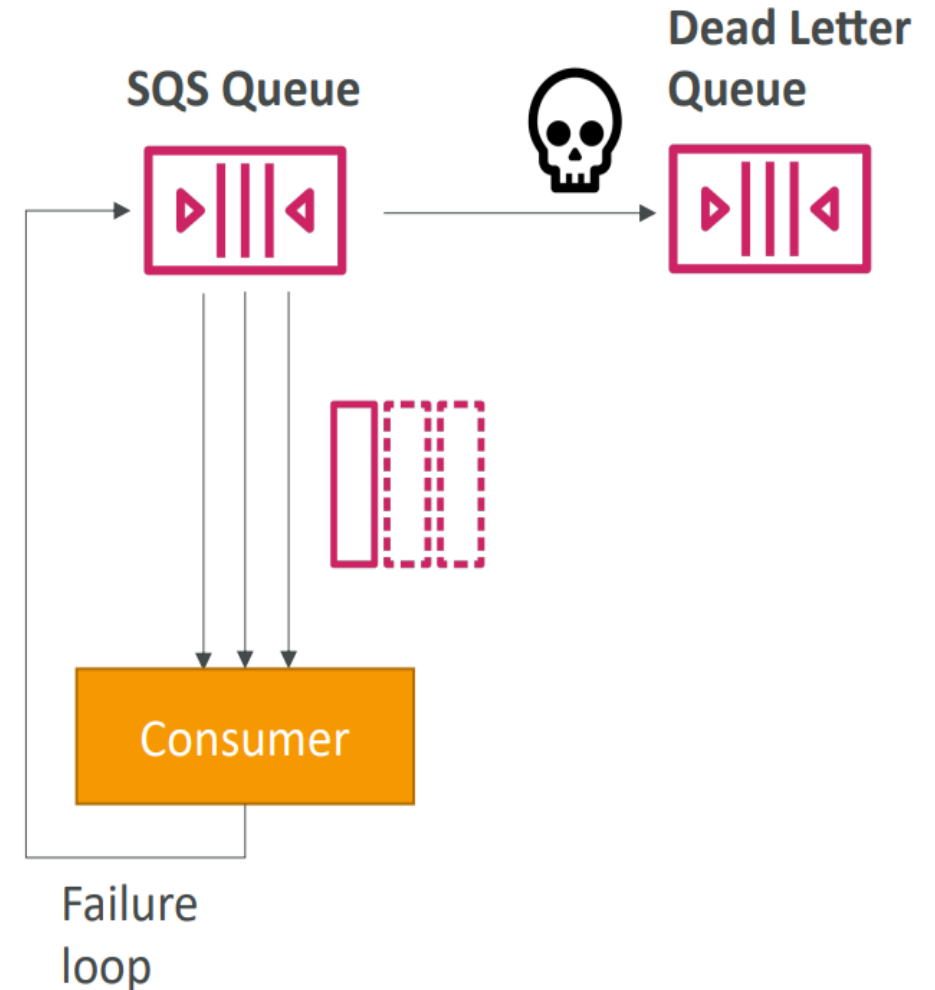
# SQS – Message Visibility Timeout

- After a message is polled by a consumer, it becomes invisible to other consumers
- By default, the “message visibility timeout” is 30 seconds
- That means the message has 30 seconds to be processed
- After the message visibility timeout is over, the message is “visible” in SQS



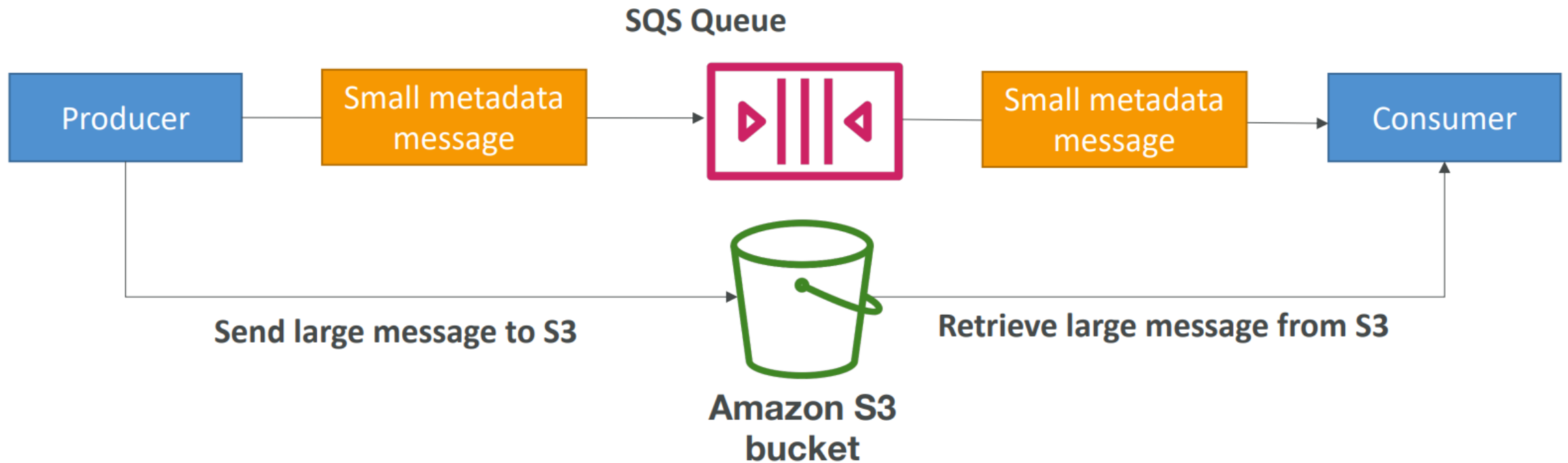
# Amazon SQS – Dead Letter Queue

- If a consumer fails to process a message within the Visibility Timeout... the message goes back to the queue!
- We can set a threshold of how many times a message can go back to the queue
- After the **MaximumReceives** threshold is exceeded, the message goes into a dead letter queue (DLQ)
- Useful for debugging!
- Make sure to process the messages in the DLQ before they expire:
  - Good to set a retention of 14 days in the DLQ



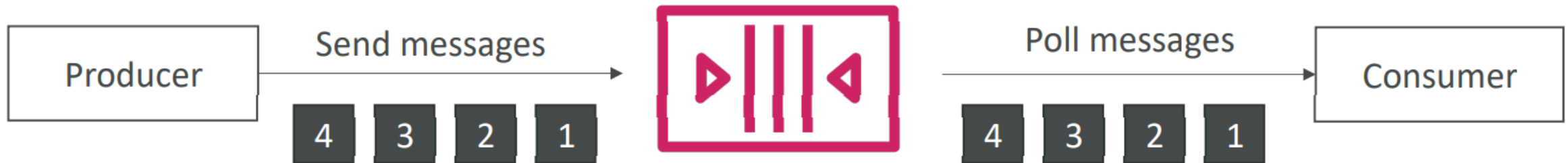
# SQS Extended Client

- Message size limit is 256KB, how to send large messages, e.g. 1GB?
- Using the SQS Extended Client (Java Library)



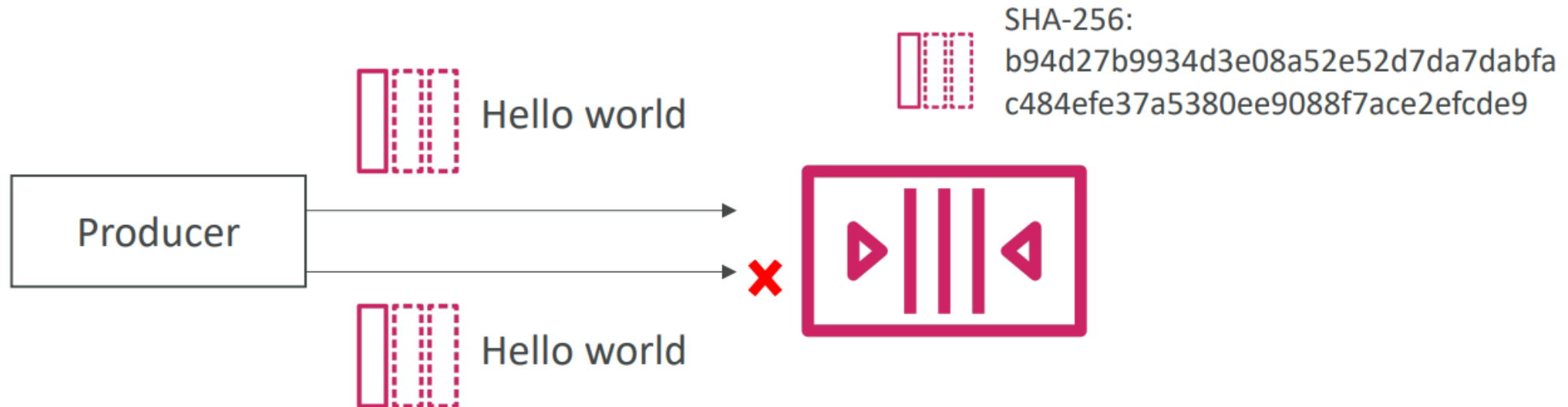
# Amazon SQS – FIFO Queue

- FIFO = First In First Out (ordering of messages in the queue)
- Limited throughput: 300 msg/s without batching, 3000 msg/s with
- Exactly-once send capability (by removing duplicates)
- Messages are processed in order by the consumer



# SQS FIFO – Deduplication

- De-duplication interval is 5 minutes
- Two de-duplication methods:
  - Content-based deduplication: will do a SHA-256 hash of the message body
  - Explicitly provide a Message Deduplication ID



# SQS FIFO – Message Grouping

- If you specify the same value of MessageGroupID in an SQS FIFO queue, you can only have one consumer, and all the messages are in order
- To get ordering at the level of a subset of messages, specify different values for MessageGroupID
  - Messages that share a common Message Group ID will be in order within the group
  - Each Group ID can have a different consumer (parallel processing!)
  - Ordering across groups is not guaranteed

