Journey Into AWS

SQS - Simple Queue Service

Who am I?

👨 Guilherme Coelho

🏡 Jundiaí

🏢 Pagar.me

What is SQS?

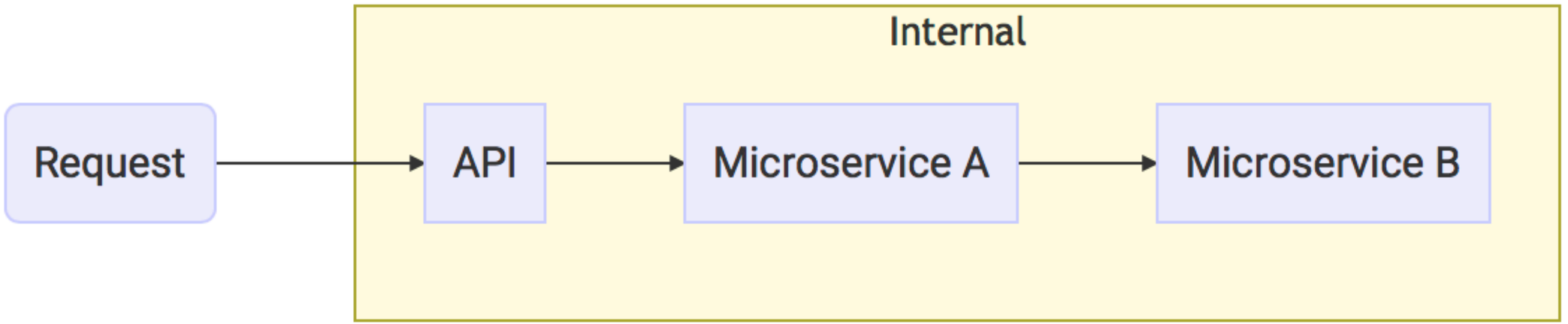
What is SQS (Simple Queue Service)?

- It's a **Service**!
- It's about **Queues**!
- It's **Simple**!

Why?

Why would you use it?

- 📪 Decoupling the components of an application.
- 🔋 Decoupling heavy-work processing.


Internal
Request
API
Microservice A
Microservice B

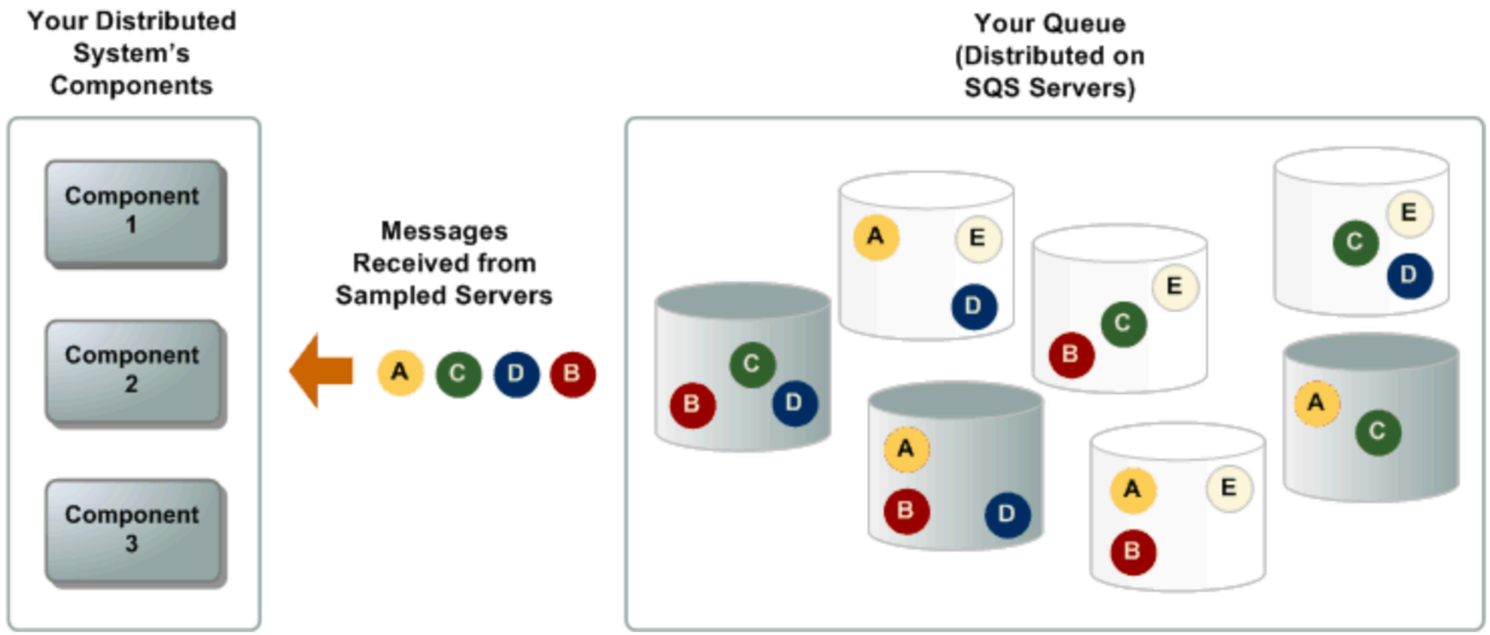
Anatomy of a Queue 💀🔬

Anatomy of a Queue 💀🔬

- `QueueUrl`
- Messages:
  - `MessageID`
  - `ReceiptHandle`

How Queues Work?

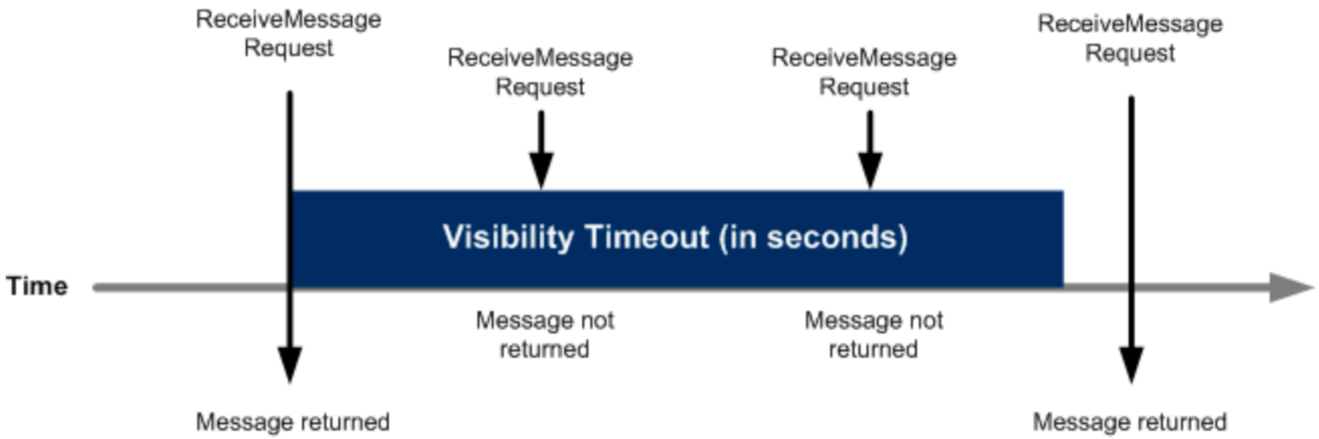
How Queues Work



At-Least-Once Delivery

- The same message can be delivered multiple times.
- Your application must have **Idempotency**.
- Idempotency: process multiple times, no harm done.

In-flight Messages and Visibility Timeout

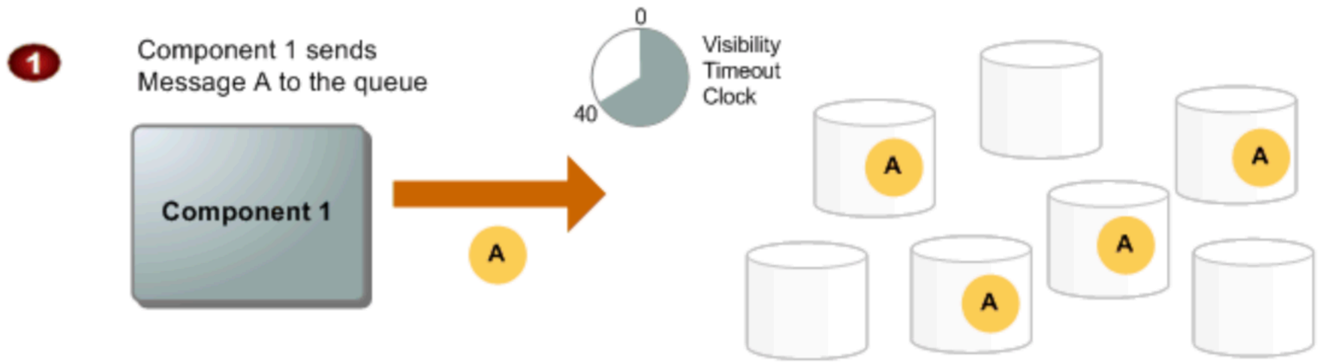


Changing Message Visibility

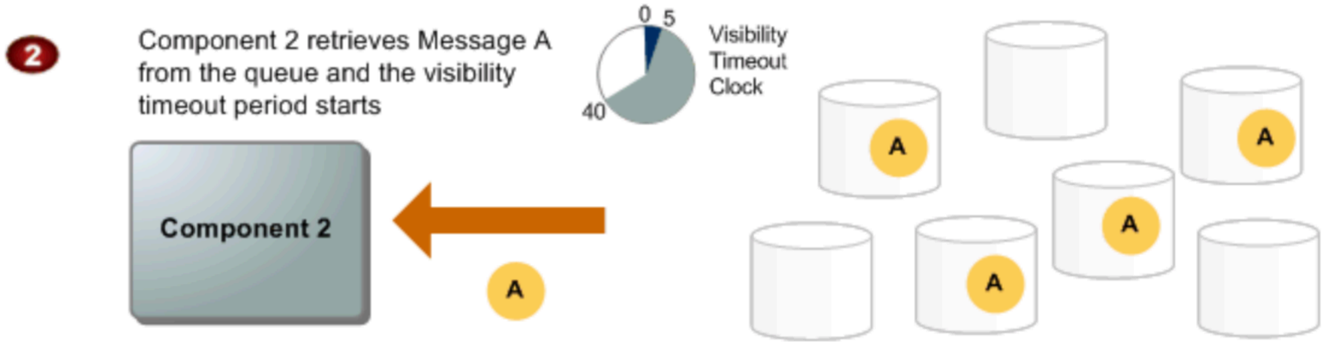
- Extending VisibilityTimeout
  - `ChangeMessageVisibility(15)`
- Terminating VisibilityTimeout
  - `ChangeMessageVisibility(0)`

Message Lifecycle

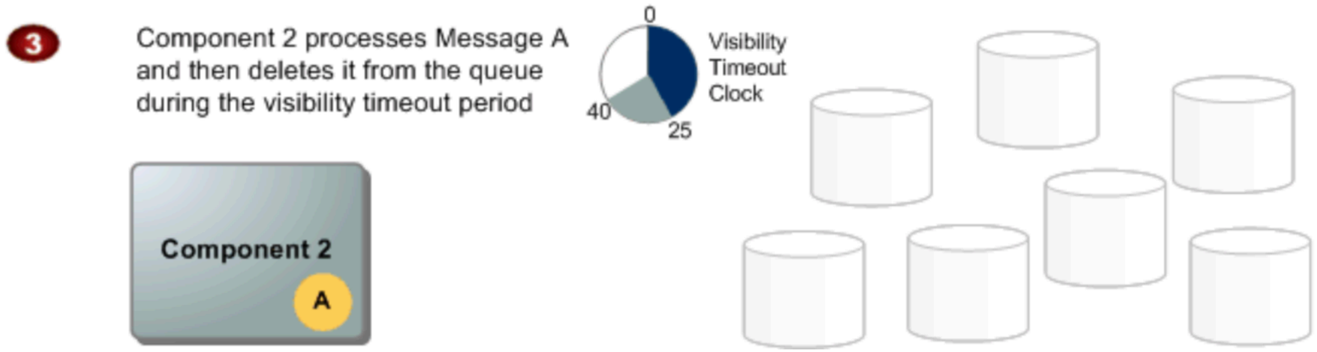
Message Lifecyle: post



Message Lifecycle: receive

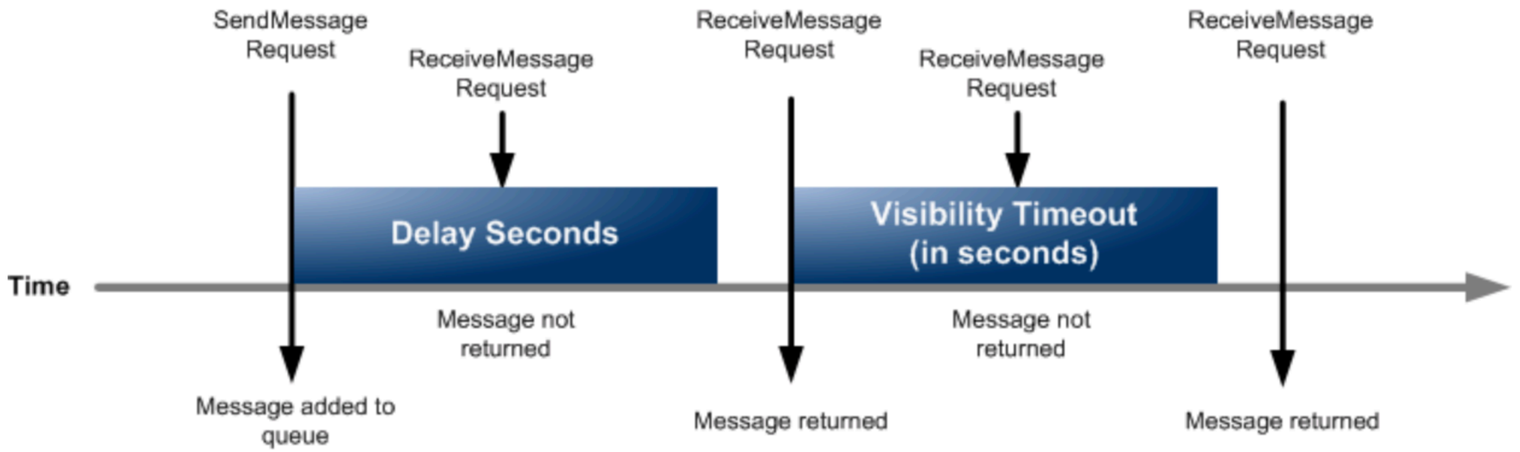


Message Lifecycle: delete



Delay Queues 🕑

Delay Queues 🕑



Dead Letter Queues 📪💀

Dead Letter Queues 📪💀

- Messages that can't be processed.
- `RedrivePolicy`
  - `deadLetterTargetArn`
  - `maxReceiveCount`

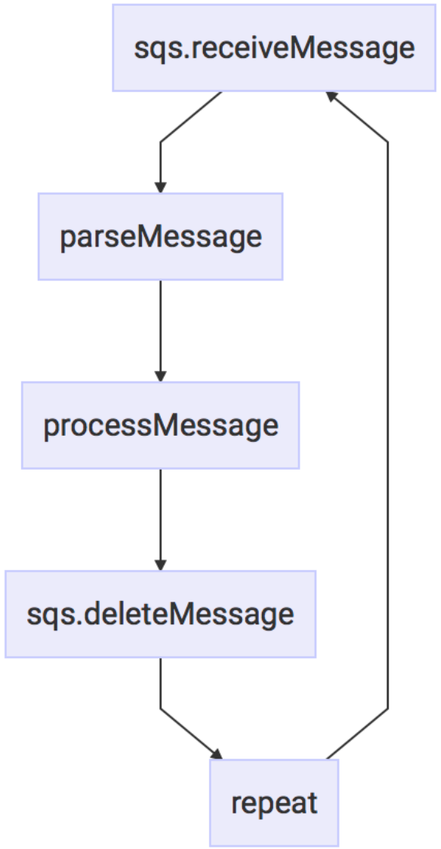
Short vs Long Polling 🏊

Short vs Long Polling 🏊

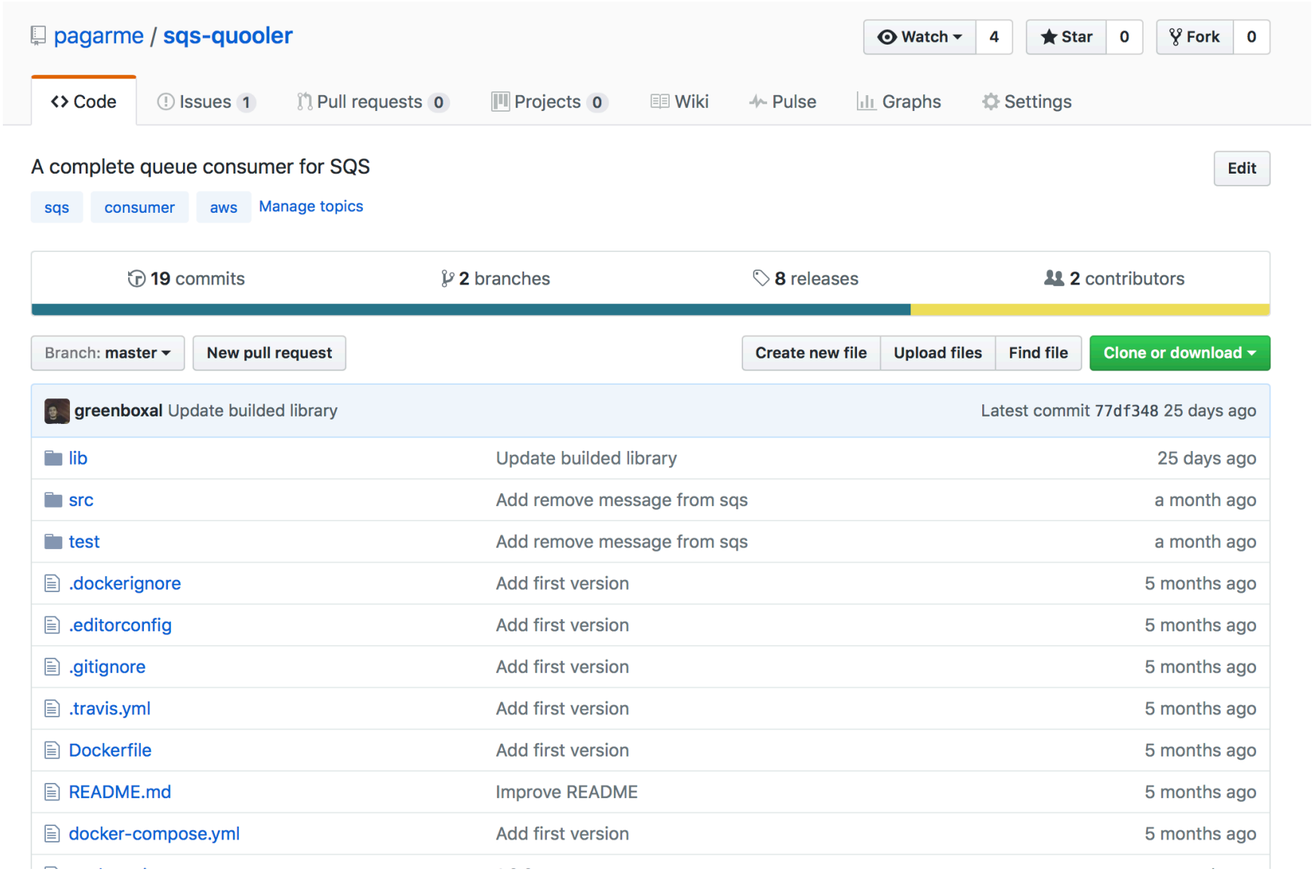
- **Short Polling (default)**: does not wait for messages to come.
- **Long Polling**: wait n seconds for messages to come.
  - Queries **all** the servers
  - Less empty responses
  - Reduced costs

Process Queue Pattern

Process Queue Pattern



We can make this easier 🔭



https://github.com/pagarme/sqs-quooler

sqs-quooler: installing

```
$ npm install --save sqs-quooler
```

sqs-quooler: importing

```
import { Queue } from 'sqs-quooler'

const myQueue = new Queue({
  endpoint: 'https://sqs.amazonaws.com/queue/my-queue'
  concurrency: 10
})
```

sqs-quooler: pushing items

```
myQueue.push({ id: 123 })
myQueue.push({ user: 'wx-78' })
```

sqs-quooler: processing the queue

```
const handler = (parsedItem, rawSQSMessage) => {
  // Process the item
  return processItem (parsedItem)
}

const options = {
  oneShot: false,
  keepMessage: false
}

myQueue.startProcessing(handler, options)
```

FIFO x Standard Queues

FIFO x Standard

	Standard	**FIFO**
Delivery	At-least-once	Exactly-once
Ordered	❌	✅
High Throughput	✅	❌ (300 t/s)
Price[1]	💵💵	💵💵💵

[1] Standard Queue Pricing: *$0,40 per 1 million requests*. [source]

🏈 Thanks 🏈

https://github.com/grvcoelho/talks