

**Che fine ha fatto il mio messaggio sqs?**

**Salvatore Caputi**

**28/03/2024 - AWS Venice User Group**

# Salvatore who?

# Salvatore who?



Solutions Architect @Pixartprinting - IT-MNF team



# Salvatore who?



Solutions Architect @Pixartprinting - IT-MNF team



Katori Shinto Ryu - Kenjutsu & laido beginner



# Salvatore who?



Solutions Architect @Pixartprinting - IT-MNF team

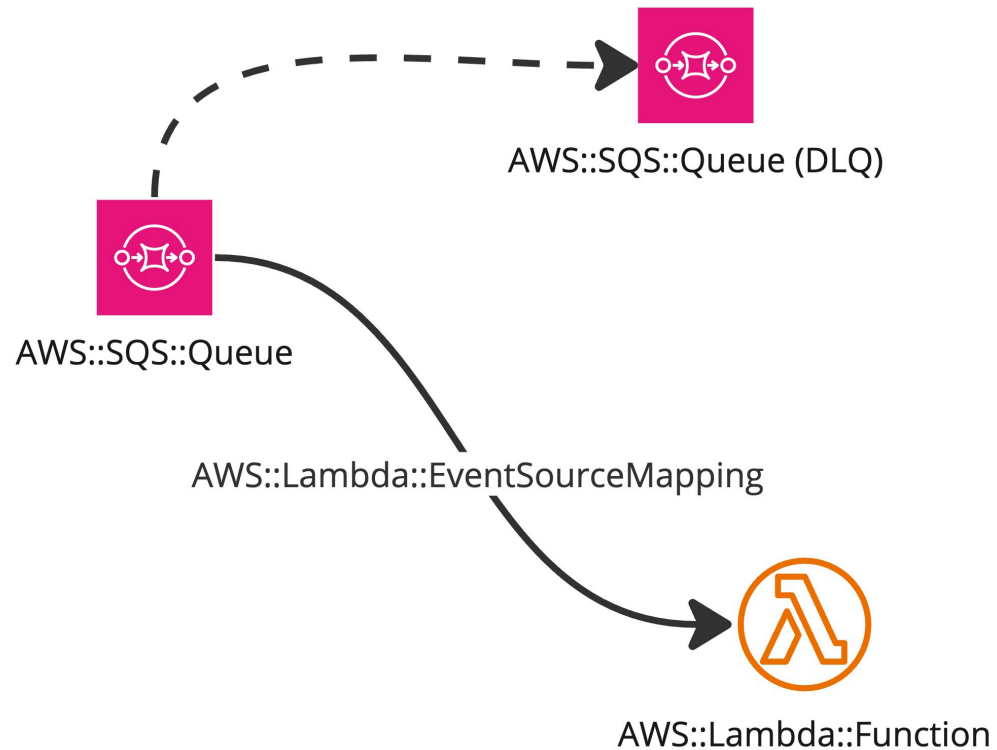


Katori Shinto Ryu - Kenjutsu & laido beginner

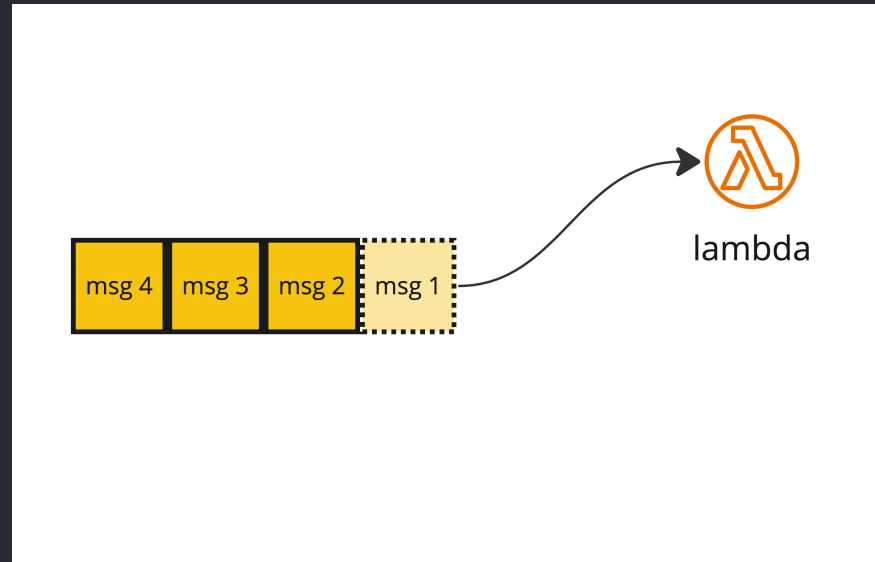


# The pattern

# The pattern



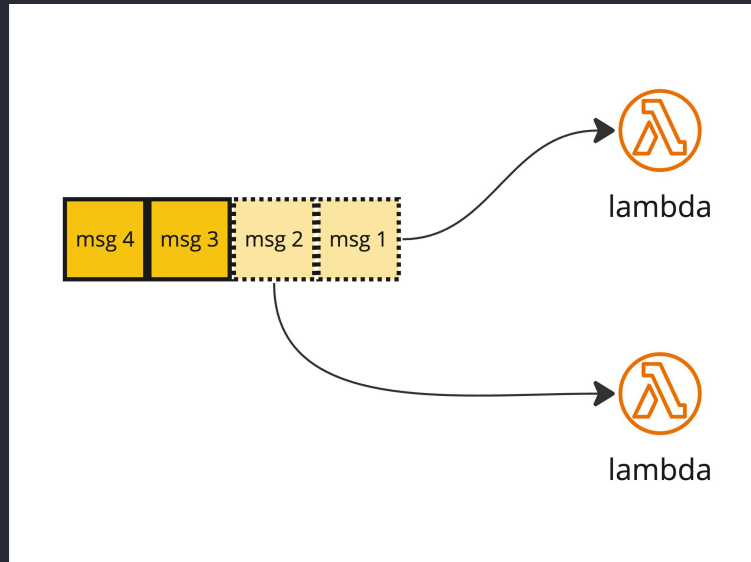
# Key concepts



As soon as the message has been *received*, it become invisible for the period set into *visibility timeout*

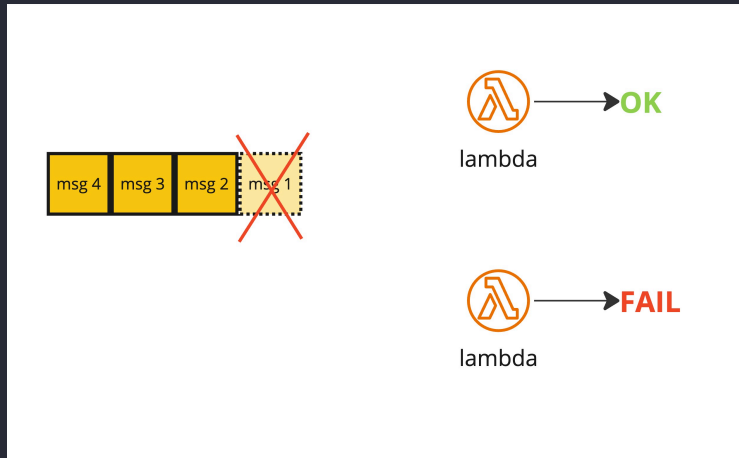


# Key concepts



*Reserved concurrency* sets the maximum number of concurrent executions of our function

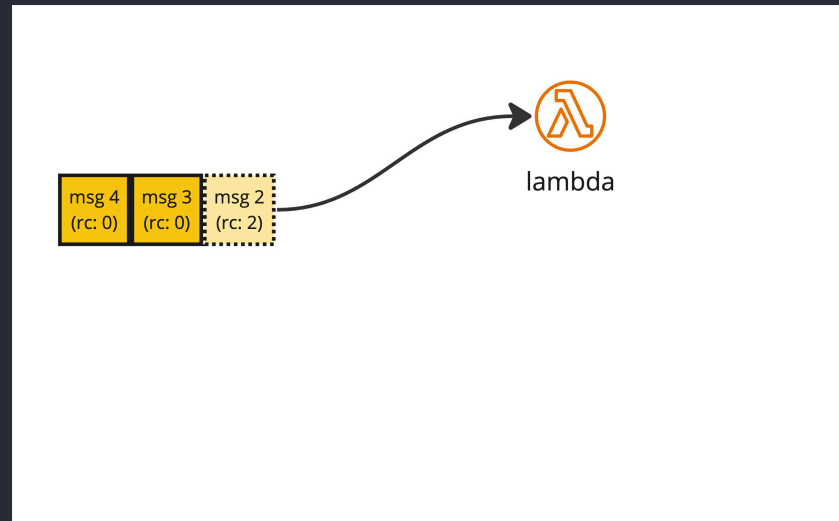
# Key concepts



If the lambda execution returns an error, the message is preserved.

Otherwise the message is *deleted*.

# Key concepts



*ReceiveCount* indicates how many times a message has been delivered. When the RC message value exceeds the *MaxReceiveCount* value, the message is moved to DLQ.

# The problem

# The problem

Users reported some messages did not reach their systems. This means messages were not elaborated by our lambda

# The analysis

# The analysis

No errors

# The analysis

No errors

Lot of messages into DLQ



# The analysis

No errors

Lot of messages into DLQ

We noticed "fresh" messages with  
*approximateReceiveCount* > 1

# The analysis

No errors

Lot of messages into DLQ

We noticed "fresh" messages with  
*approximateReceiveCount* > 1

We double checked nobody else was receiving those  
messages

# The analysis

So, what is happening to our sqs messages?

# The analysis



# The analysis

It seemed some messages were moved directly to DLQ without being elaborated

# Under the hood

# Under the hood

Lambda service uses long polling to check messages  
into SQS

# Under the hood

Lambda service uses long polling to check messages  
into SQS

It starts with 5 parallel executions



# Under the hood

Lambda service uses long polling to check messages into SQS

It starts with 5 parallel executions

It can scale, based on the rate of messages, up to 300 instances per minute (max 1000 instances)

# Under the hood

Lambda service uses long polling to check messages into SQS

It starts with 5 parallel executions

It can scale, based on the rate of messages, up to 300 instances per minute (max 1000 instances)

**Not related to reservedConcurency**

**Under the hood**

# Under the hood

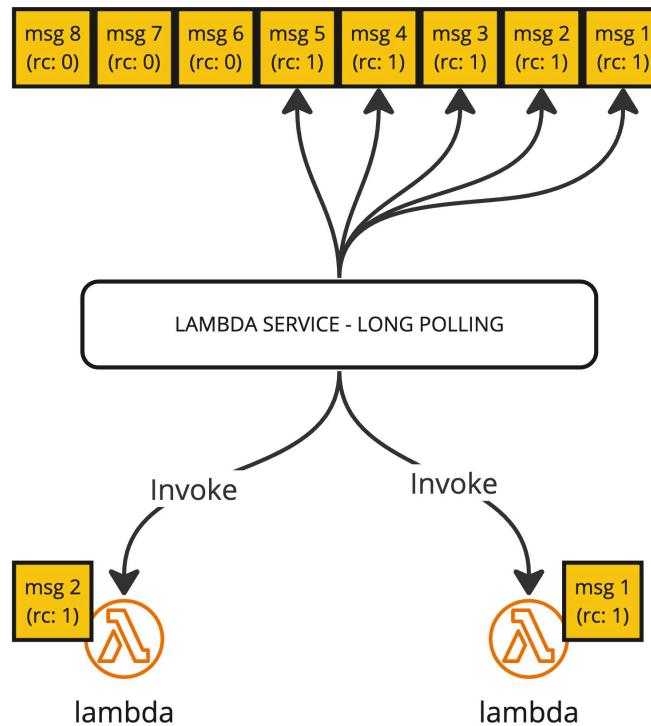
This means executions that exceeds the `reservedConcurrency` are throttled

# Under the hood

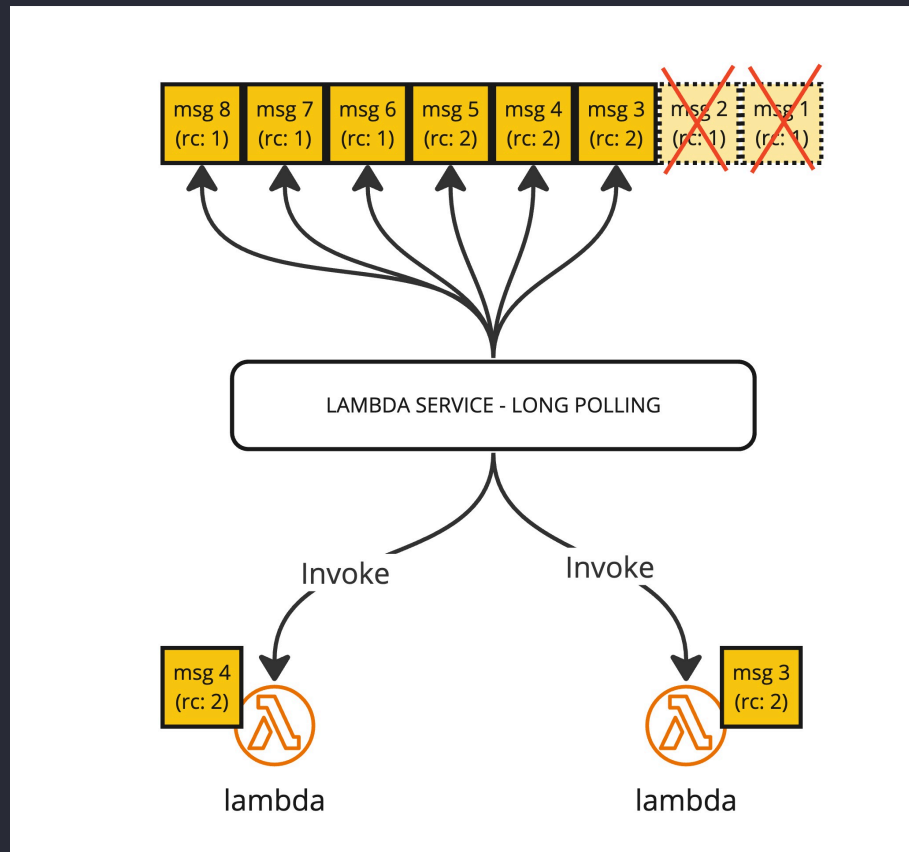
This means executions that exceeds the `reservedConcurrency` are throttled

This problem impacts mainly lambdas with a low `reservedConcurrency` (1-30)

# Under the hood



# Under the hood



# Mitigation - before January 2023

• **Prevalence** of **COVID-19** in **the UK** was **1.5%** in **January 2023**

• **Prevalence** of **COVID-19** in **the UK** was **1.5%** in **January 2023**

• **Prevalence** of **COVID-19** in **the UK** was **1.5%** in **January 2023**

• **Prevalence** of **COVID-19** in **the UK** was **1.5%** in **January 2023**

• **Prevalence** of **COVID-19** in **the UK** was **1.5%** in **January 2023**

• **Prevalence** of **COVID-19** in **the UK** was **1.5%** in **January 2023**



# Mitigation - before January 2023

Increase reservedConcurrency

# Mitigation - before January 2023

Increase `reservedConcurrency`

Set the queue's `visibilityTimeout` to at least 6 times the timeout that you configure on your function

# Mitigation - before January 2023

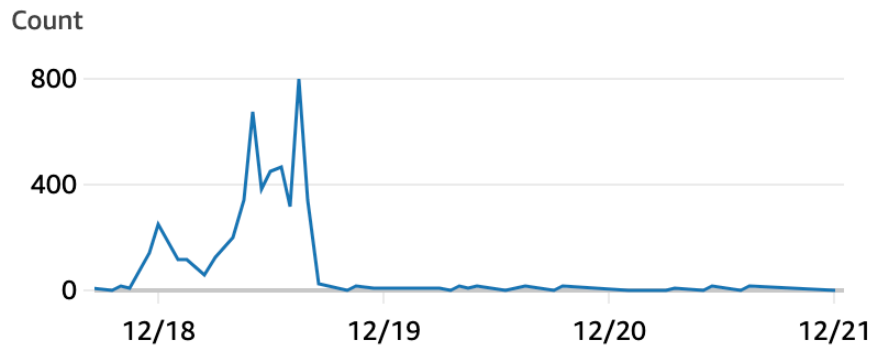
Increase `reservedConcurrency`

Set the queue's `visibilityTimeout` to at least 6 times the timeout that you configure on your function

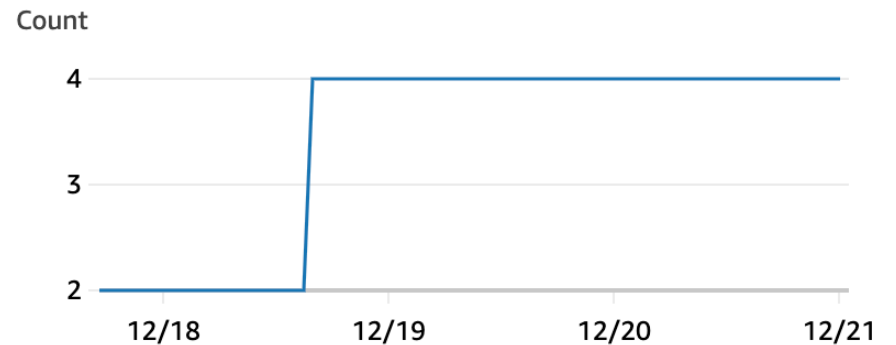
Set the `maxReceiveCount` on the queue's redrive policy to at least 5

# Mitigation - before January 2023

Throttles



Total concurrent executions



# **Solution - after January 2023**

# Solution - after January 2023

We can set the `maximumConcurrency` to the event source mapping

# Solution - after January 2023

We can set the `maximumConcurrency` to the event source mapping

Setting it lower than the `lambda reservedConcurrency` let us avert the *overpulling* problem

# Solution - after January 2023

We can set the `maximumConcurrency` to the event source mapping

Setting it lower than the `lambda reservedConcurrency` let us avert the *overpulling* problem

If you have more than one event source for the same lambda, the sum of all `maximumConcurrency` must be lower than `reservedConcurrency`



**Questions?**

# Thank you!

