DLQ

Isole falhas persistentes e recupere com controle

Pré-requisitos

Conheça SQS, tempo de visibilidade e deduplicação. O exercício usa uma política de fila ilustrativa e Node.js 24 para estimar o tempo de reprocessamento.

Use a DLQ para diagnóstico [1]

Uma dead-letter queue recebe mensagens que ultrapassam a política de recebimentos da fila de origem. Se o consumidor de comprovantes falha repetidamente, a mensagem sai do caminho principal e pode ser investigada. A presença na DLQ indica que o efeito precisa ser conferido. Ele pode já ter ocorrido antes de uma falha de exclusão.

Ajuste o limite de recebimentos

maxReceiveCount limita os recebimentos permitidos pela política da fila. Ele conta entregas da fila e não as chamadas internas do SDK. Um limite baixo pode isolar falhas curtas. Um limite alto pode gastar capacidade repetindo dados inválidos. Alinhe a configuração com visibilidade, retentativas internas e tempo máximo tolerável.

Observe idade e retenção

Em filas Standard, o prazo de expiração continua baseado no instante original de enfileiramento. Configure a retenção da DLQ maior que a da origem para dar margem de investigação. Monitore o número de mensagens disponíveis e investigue acúmulos. Se a ordem estrita de uma fila FIFO for uma regra de negócio, analise como isolar uma mensagem afeta as seguintes.

Reenvie depois de corrigir a causa [2]

Redrive move mensagens para processamento novamente. Antes, confirme se o consumidor entende a versão do evento e se o efeito ainda é válido. Comece com velocidade compatível com a capacidade disponível. O laboratório estima capacidade e não envia mensagens reais.

DLQ

Exercício resolvido • Recupere a geração de comprovantes

Enunciado e dados de partida

Há 2.400 mensagens na DLQ após uma incompatibilidade de esquema. Corrigida a leitura, o consumidor completa até 35 mensagens por segundo. O tráfego normal usa 25 por segundo. Configure o isolamento de falhas e escolha uma velocidade inicial de 8 por segundo para reprocessar com margem.

1. Modele o atributo RedrivePolicy da fila de origem

```
{
  "deadLetterTargetArn": "arn:aws:sqs:us-east-1:123456789012:agenda-dlq",
  "maxReceiveCount": 5
}
```

Leia a configuração

O ARN é fictício. Em uma fila de teste, substitua pelo ARN da DLQ correspondente e confira permissões, tipo e retenção. Com cinco recebimentos permitidos, o processamento normal ainda pode se recuperar de falhas curtas. O número cinco é uma decisão do exercício, a validar com métricas.

2. Salve estimativa.mjs e execute node estimativa.mjs

```
import assert from 'node:assert/strict';

function minutosParaEsvaziar(pendentes, capacidade, entrada, redrive) {
  const saldo = Math.min(redrive, capacidade - entrada);
  return saldo > 0 ? Math.ceil(pendentes / saldo / 60) : Infinity;
}

assert.equal(minutosParaEsvaziar(2400, 35, 25, 8), 5);
assert.equal(minutosParaEsvaziar(2400, 35, 35, 8), Infinity);
assert.equal(minutosParaEsvaziar(2400, 35, 25, 20), 4);
console.log('Orcamento de reprocessamento conferido');
```

DLQ

Confira o ritmo e o resultado do reenvio

3. Calcule a folga operacional

A capacidade livre é 35 − 25 = 10 mensagens por segundo. Com redrive de 8, o consumo estimado é 33 por segundo, com margem de 2. Para 2.400 mensagens, 2.400 ÷ 8 = 300 segundos ou 5 minutos. A estimativa supõe taxas constantes, nenhuma nova falha e capacidade mantida.

Caso de erro • Reenvie acima da capacidade

Com redrive de 20 e folga de 10, pelo menos 10 mensagens por segundo se acumulam enquanto o reenvio estiver ativo. O cálculo de quatro minutos arredonda para cima os 240 segundos necessários para o consumidor absorver todo o lote, incluindo o atraso na fila principal. Observe ambas as filas. A DLQ pode esvaziar antes de a agenda terminar o trabalho.

Variação • O tráfego normal subiu

Se o tráfego chegar a 35 por segundo, a folga se torna zero. O simulador retorna Infinity para indicar ausência de prazo finito enquanto as taxas se mantiverem. Pause ou reduza o reenvio e reavalie a capacidade antes de continuar.

Critérios para conferir

Verifique a causa corrigida com um lote representativo em uma fila de teste. Confira comprovantes gerados, duplicatas bloqueadas, aumento na fila de origem e mensagens que voltaram à DLQ. O sucesso é concluir o efeito esperado com atraso tolerável, além de esvaziar a DLQ.

Referências primárias

[1] AWS • Filas de mensagens mortas e retenção

[2] AWS • Redrive e velocidade de movimentação

DLQ

Reprocesse a DLQ após corrigir a causa

Separe correção de reprocessamento

Classifique falhas na DLQ por causa e versão do consumidor. Corrija e valide a leitura em homologação. Depois reintroduza em ritmo controlado, acompanhando a fila original, o volume de erros e o retorno à DLQ.

Exercício adicional

Um novo campo obrigatório levou mensagens antigas à DLQ. O consumidor já aceita as duas versões. Como avançar?

Resolução

Reprocesse primeiro um conjunto pequeno de mensagens representativas, acompanhe os resultados pelo ID e confirme os efeitos. Se estiver estável, aumente o ritmo e pause se as falhas voltarem.

Ao assistir ao vídeo

Observe o redrive e identifique para onde voltam as mensagens.

Vídeo complementar

Dead-Letter Queues in .NET with Amazon SQS (em inglês)