



Amazon SQS AWS Lambda 이벤트 소스에 대한 이벤트 필터링 및 부분 배치 응답을 사용하여 애플리케이션 성능 향상

AWS 권장 가이드



AWS 권장 가이드: Amazon SQS AWS Lambda 이벤트 소스에 대한 이벤트 필터링 및 부분 배치 응답을 사용하여 애플리케이션 성능 향상

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

소개	1
목표 비즈니스 성과	1
Amazon SQS 이벤트 소스에 부분 배치 응답 사용의 이점	2
Amazon SQS 이벤트 소스에 Lambda 이벤트 필터 사용의 이점	3
Lambda 이벤트 필터의 사용 사례 예	3
부분 배치 응답 모범 사례	4
배치 처리	4
멥등성	4
Metrics	5
Snowball 안티 패턴 방지	6
Lambda 이벤트 필터링 모범 사례	7
일치하지 않는 메시지 누적 방지	7
FAQ	9
Lambda 이벤트 필터링과 부분 배치 응답을 기존 Lambda 및 Amazon SQS 애플리케이션에 적용 할 수 있나요?	9
Lambda 이벤트 필터링 및 부분 배치 응답 기능이 특정 프로그래밍 언어로 제한되나요?	9
이 기능을 AWS Cloud Development Kit (AWS CDK)와 함께 사용할 수 있나요?	9
리소스	10
문서 기록	11
.....	xii

Amazon SQS AWS Lambda 이벤트 소스에 대한 이벤트 필터링 및 부분 배치 응답을 사용하여 애플리케이션 성능 개선

Alan Oberto Jimenez, Amazon Web Services(AWS)

2022년 9월([문서 기록](#))

이 가이드에서는 Amazon Simple Queue Service(Amazon SQS) AWS Lambda 이벤트 소스에 대한 이벤트 필터링 및 부분 배치 응답을 구성하는 모범 사례를 다룹니다. 이 제어는 Lambda의 기본 메시지 배치 처리 동작과 관련된 두 가지 일반적인 성능 문제를 해결합니다.

- Lambda 함수가 전체 Amazon SQS 메시지 배치를 여러 번 재시도하도록 하는 포이즌 필 메시지
- Lambda 함수가 불필요하게 호출되게 하는 관련 없는 Amazon SQS 이벤트

이러한 제어를 설정하면 Amazon SQS 메시지 배치를 더 효율적으로 처리하고 서버리스 애플리케이션의 성능을 개선하도록 Lambda 함수를 구성할 수 있습니다.

목표 비즈니스 성과

이 가이드는 다음을 수행하는 데 도움이 됩니다.

- Lambda 함수가 Amazon SQS 메시지 배치를 처리하는 데 걸리는 시간을 줄여 서버리스 애플리케이션의 사용자 환경을 개선합니다.
- Lambda 함수가 처리할 수 없는 Amazon SQS 메시지를 분리하여 서버리스 아키텍처의 운영 기능과 전반적인 안정성을 개선합니다.
- 불필요한 Lambda 함수 호출 횟수를 줄여 서버리스 애플리케이션의 운영 비용을 줄입니다.
- Amazon SQS 메시지 배치를 처리할 때 Lambda 함수의 코드를 간소화하여 성능을 개선합니다.

Amazon SQS 이벤트 소스에 부분 배치 응답 사용의 이점

부분 배치 응답을 구성하면 Lambda 함수가 부분 Amazon SQS 메시지 배치를 처리하고 실패한 메시지만 재시도할 수 있습니다. 따라서 반복적인 데이터 전송이 필요하지 않고 처리량이 증가합니다.

기본적으로 Lambda 함수가 Amazon SQS 메시지 배치에서 메시지 하나를 처리하지 못하면 전체 배치가 대기열로 돌아갑니다. [가시성 시간 초과](#)가 발생하면 Lambda 함수는 메시지 배치를 다시 수신합니다. 함수가 유효한 메시지를 여러 번 처리하지 못하는 경우 Amazon SQS는 [DLQ\(Dead Letter Queue\)](#)(구성된 경우)로 메시지를 보냅니다.

이 기본 배치 처리 동작으로 인해 단일 실패(포이즌 필) 메시지로 인해 Lambda 함수가 메시지 배치를 여러 번 재시도할 수 있습니다. 이러한 메시지 일괄 재시도는 함수 코드가 [역동성](#)이 있고 메시지를 여러 번 처리할 수 있는 경우에도 애플리케이션 성능을 저하시킬 수 있습니다.

Amazon SQS 이벤트 소스에 Lambda 이벤트 필터 사용의 이점

Lambda 이벤트 필터를 구성하면 지정한 필터링 기준에 의해서만 함수를 호출할 수 있습니다. 이를 통해 함수에 대한 트래픽을 줄이고 코드를 간소화하여 비용을 절감할 수 있습니다.

이벤트 필터가 구성되지 않은 경우 Lambda 함수가 관련 없는 Amazon SQS 이벤트를 수신하여 함수가 불필요하게 호출될 수 있습니다. 이러한 관련 없는 메시지는 메시지 형식 문제(예: 누락된 필드) 또는 함수와 관련이 없는 값이 포함된 필드로 인해 발생할 수 있습니다.

Lambda 이벤트 필터의 사용 사례 예

100 USD 이상의 매출이 발생하면 고객 관리 팀에 알리는 애플리케이션이 있습니다. 그러나 모든 금액(5 USD, 10 USD 등)의 모든 판매가 조직의 Amazon SQS 대기열에 기록됩니다. 애플리케이션의 Lambda 함수가 대기열을 폴링할 때마다 애플리케이션의 비즈니스 로직과 관련이 없는 메시지를 수신합니다. 이 경우 관련 없는 메시지에 대한 처리 요금이 부과되며 이로 인해 애플리케이션의 운영 비용이 증가합니다. 이 상황에서 Lambda 이벤트 필터를 구성하면 Lambda 함수가 처리해야 하는 메시지 수가 제한되고 비용이 절감됩니다.

호텔 예약을 일괄 처리해야 하는 호텔 예약 애플리케이션의 경우 사용자 이름과 같은 특정 요청 필드는 운영 성공에 매우 중요합니다. 이벤트 필터가 없으면 개발자는 각 요청에 필드가 있는지 여부를 식별하는 로직을 애플리케이션에 코딩해야 합니다. 또한 애플리케이션 코드는 각 요청에 올바른 데이터 유형이 포함되어 있는지 확인할 수 있어야 합니다. Lambda 이벤트 필터를 사용하면 이 모든 로직을 Amazon SQS로 추상화하여 개발 노력과 코드 복잡성을 줄일 수 있습니다.

부분 배치 응답 구현에 대한 모범 사례

이 섹션에서는 Amazon SQS 이벤트 소스에 대한 부분 배치 응답을 구성하는 모범 사례를 제공합니다.

- 서버리스 애플리케이션 아키텍처에서 snowball 방지 패턴 생성을 방지하도록 DLQ(Dead Letter Queue)를 구성합니다. 자세한 내용은 [the section called “Snowball 안티 패턴 방지”\(을\)](#)를 참조하세요.
- 실패한 메시지만 표시하도록 Lambda 함수 이벤트 소스 매핑을 구성합니다. 이렇게 하려면 이벤트 소스 매핑을 구성할 때 FunctionResponseTypes 목록에 ReportBatchItemFailures를 추가합니다. FunctionResponseTypes Lambda 함수는 SQS에서 호출할 때 부분 배치 응답을 구현해야 합니다. 기본 제공 부분 AWS Lambda 배치 지원으로 SQS 메시지를 처리하는 배치 처리 유틸리티용 Powertools를 사용하는 것이 좋습니다.

배치 처리

- SQS에서 호출하는 Lambda 함수에서 부분 배치 응답을 구현합니다. AWS Lambda 배치 처리를 위한 Powertools 유틸리티를 사용하는 것이 좋습니다. 이 유틸리티는 기본 제공 부분 배치 응답 지원을 통해 SQS 메시지 처리를 처리합니다. 자세한 내용은 AWS Lambda 개발자 안내서의 [Amazon SQS 트리거를 사용하여 Lambda 함수에 대한 배치 항목 실패 보고를 참조하세요](#).

AWS Lambda 배치 처리 유틸리티용 Powertools

- [AWS Lambda 배치 처리 유틸리티용 Powertools\(Python\)](#)
- [AWS Lambda 배치 처리 유틸리티용 Powertools\(Java\)](#)
- [AWS Lambda 배치 처리 유틸리티용 Powertools\(.Net\)](#)
- [AWS Lambda 배치 처리 유틸리티용 Powertools\(TypeScript\)](#)

역등성

- DLQ(Dead Letter Queue)로 이동되기 전에 메시지가 소스 대기열로 배달되는 횟수를 정의합니다. 가장 가능성이 높은 장애 원인과 예상 복구 시간을 식별하여 정의한 값이 애플리케이션의 사용 사례에 맞는지 확인합니다. 재시도 횟수를 정의하려면 소스 대기열의 RedrivePolicy에서 maxReceiveCount 값을 구성합니다. 자세한 내용은 Amazon SQS API Reference의 [SetQueueAttributes](#)를 참조하세요.

또한 소스 대기열에 [Amazon Simple Queue Service](#) 배달 못한 편지 대기열 리드라이브 소개를 참조하세요.

- Lambda 코드가 멍등성이고 메시지를 여러 번 처리할 수 있는지 확인합니다. 멍등성을 구현하려면 Amazon AWS Lambda SQS 배치 내의 개별 작업을 지원하도록 함수의 코드를 준비하는 Idempotency Utility용 Powertools를 사용하는 것이 좋습니다. Amazon SQS 먼저 이벤트 소스 매핑에 ReportBatchItemFailures를 통합합니다. 자세한 내용은 AWS Lambda 개발자 안내서의 [부분 배치 응답 구현](#) 및 [Amazon SQS 메시지가 Lambda 함수를 두 번 이상 호출하지 않도록 하려면 어떻게 해야 합니까?](#)를 참조하세요.

AWS Lambda Idempotency 유틸리티용 Powertools

- [Powertools for AWS Lambda Idempotency Utility\(Python\)](#)
- [AWS Lambda Idempotency Utility용 Powertools\(Java\)](#)
- [AWS Lambda Idempotency Utility용 Powertools\(.Net\)](#)
- [AWS Lambda Idempotency Utility용 Powertools\(TypeScript\)](#)

Metrics

- 함수에 비즈니스 지표를 통합하여 작업 세부 정보 및 실패한 작업을 추적하려면 [임베디드 지표 형식](#)으로 SQS 이벤트를 처리하는 동안 운영 및 비즈니스 지표를 내보내는 aws-embedded-metrics 또는 Powertools for AWS Lambda Metrics Utility를 사용하는 것이 좋습니다.

AWS Lambda 지표 유틸리티용 Powertools

- [Powertools for AWS Lambda Metrics Utility\(Python\)](#)
- [Powertools for AWS Lambda Metrics Utility\(Java\)](#)
- [Powertools for AWS Lambda Metrics Utility\(.Net\)](#)
- [Powertools for AWS Lambda Metrics Utility\(TypeScript\)](#)
- [First-In-First-Out\(FIFO\)](#) 대기열을 사용하는 경우 함수는 첫 번째 실패 후 메시지 처리를 중단하고 실패한 메시지와 처리되지 않은 모든 메시지를 batchItemFailures로 반환해야 합니다. 이렇게 하면 대기열의 메시지 순서를 유지하는 데 도움이 됩니다.

Note

부분 배치 처리를 사용하는 애플리케이션의 전체 성능을 추적하려면 코드 수준 성능 추적이 필요합니다. 배치 처리가 구성된 후 Lambda 함수 호출은 일반적으로 처리 결과에 관계없이 성공합니다.

Snowball 안티 패턴 방지

Lambda와 Amazon SQS는 업스트림 마이크로서비스가 SQS 대기열에 쓰는 메시지를 제어할 수 없습니다. 처리할 수 없는 메시지가 있는 경우 별도의 [배달 못한 편지 대기열이 구성되지 않은 한 Lambda는 처리되지 않은 메시지를 소스 SQS 대기열에 반환합니다](#). 그런 다음 처리되지 않은 메시지는 Lambda 함수에 의해 재시도됩니다. 배달 못한 편지 대기열이 없는 경우 Amazon SQS 대기열로 반환된 처리되지 않은 메시지 수가 결국 대기열의 유효한 메시지보다 많아집니다.

각 연속 Lambda 호출이 문제를 악화시키는 이러한 유형의 눈덩이 안티 패턴은 다음과 같은 문제를 일으킬 수 있습니다.

- 작업이 처리하는 데 평소보다 오래 걸리거나 전혀 처리되지 않기 때문에 사용자 경험이 좋지 않음
- 비용 증가 - Amazon SQS 대기열의 메시지 수와 메시지 재시도 횟수의 기하급수적 증가에 비례
- 애플리케이션의 Lambda 컴퓨팅 용량 감소 또는 AWS 계정 함수에 호출 요청에 대한 제한이 없는 경우

부분 배치 응답을 구성할 때 눈덩이 안티 패턴을 생성하지 않으려면 배달 못한 편지 대기열도 생성하는 것이 가장 좋습니다. 이 별도의 대기열에는 성공적으로 처리되지 않은 메시지가 저장되어 애플리케이션의 처리되지 않은 메시지 수명 주기를 더 잘 관리할 수 있습니다.

자세한 내용은 [Amazon SQS 개발자 안내서의 Amazon SQS 콘솔을 사용하여 배달 못한 편지 대기열 구성](#)을 참조하세요. Amazon SQS

Lambda 이벤트 필터링 구현 모범 사례

다음은 [Amazon SQS 이벤트 소스에 대한 Lambda 이벤트 필터링](#) 구성 모범 사례입니다.

- [Lambda 서비스의 필터 규칙 구문](#), 특히 [Lambda가 Amazon SQS에 사용하는 필터 규칙 구문](#)을 숙지하세요. 그런 다음 비즈니스 로직에 따라 이벤트 필터를 설계합니다. 이벤트 필터의 효과는 필요한 경우에만 Lambda 함수를 호출할 수 있는 기능에 따라 달라집니다. 이를 위해서는 필터 규칙에 올바른 구문을 적용하여 규칙이 애플리케이션 비즈니스 로직의 요구 사항을 반영하도록 해야 합니다.
- 애플리케이션의 비즈니스 로직이 더 복잡한 일련의 필터 조건을 요구하는 상황에서는 다단계 필터링을 사용해야 합니다. 자세한 정보 및 예제 데이터 객체는 AWS Lambda 개발자 안내서의 [Lambda 이벤트 필터링](#)을 참조하세요.
- Amazon SQS 요청 파라미터에서 MessageRetentionPeriod 값을 정의하여 Amazon SQS에서 처리되지 않은 메시지를 보관할 기간을 제한합니다. 일반적으로 유효한 메시지를 처리하는 데 걸리는 시간을 식별하여 정의한 값이 애플리케이션의 사용 사례에 맞는지 확인합니다. 자세한 내용은 Amazon SQS API Reference의 [SetQueueAttributes](#)를 참조하세요. 이 가이드의 [일치하지 않는 메시지 누적 방지](#) 섹션도 참조하세요.
- Lambda 함수를 사용할 수 없어 유효한 메시지를 처리할 수 없는 상황에서는 Amazon SQS DLQ(Dead Letter Queue)를 구성합니다. 이 추가 Amazon SQS 대기열은 서버리스 애플리케이션 아키텍처에서 snowball 방지 패턴 생성을 방지하는 데 도움이 됩니다. 자세한 내용은 이 가이드의 [Snowball 안티 패턴 방지](#) 섹션을 참조하세요.
- Lambda 서비스에서 이벤트 필터 기능을 활성화해야 합니다. Lambda 함수를 처음 생성할 때 기능을 활성화해야 합니다. 지침은 AWS Lambda 개발자 안내서의 [이벤트 소스 매핑에 필터 기준 연결\(콘솔\)](#) 또는 [이벤트 소스 매핑에 필터 기준 연결\(AWS CLI\)](#)을 참조하세요.

일치하지 않는 메시지 누적 방지

이벤트 필터가 적절하게 설계되지 않은 경우 많은 양의 관련 없는 메시지가 Lambda 함수의 Amazon SQS 대기열을 빠르게 압도할 수 있습니다. 이 시나리오에서 Lambda 함수는 관련 메시지가 거의 없더라도 초기 메시지 배치에서 모든 메시지를 처리합니다. 함수가 관련 없는 추가 메시지를 처리하는 데 시간이 더 오래 걸리지만 Amazon SQS 대기열에는 관련 없는 메시지가 더 많이 추가됩니다. MessageRetentionPeriod 파라미터가 구성되지 않은 경우 이러한 새 메시지로 인해 함수가 다음 메시지 배치를 처리하는 데 훨씬 더 오래 걸립니다. 시간이 지나면 소스 대기열에 있는 관련 없는 메시지 수가 유효한 메시지 수보다 많아집니다.

MessageRetentionPeriod 파라미터는 Amazon SQS 대기열에 처리되지 않은 메시지를 보관하는 기간 (초 단위)을 제한합니다. 예를 들어 MessageRetentionPeriod 값이 3,600초이면 1시간 후에 대기열에서 처리되지 않은 모든 메시지가 제거됩니다.

지침은 Amazon SQS API Reference의 [SetQueueAttributes](#)를 참조하세요.

FAQ

Lambda 이벤트 필터링과 부분 배치 응답을 기존 Lambda 및 Amazon SQS 애플리케이션에 적용할 수 있나요?

예, 이 가이드에 설명된 구현 모범 사례를 따르기만 하면 됩니다.

Lambda 이벤트 필터링 및 부분 배치 응답 기능이 특정 프로그래밍 언어로 제한되나요?

아니요, 이 가이드에서 설명하는 기능은 Lambda가 현재 지원하는 모든 프로그래밍 언어에서 지원됩니다.

이 기능을 AWS Cloud Development Kit (AWS CDK)와 함께 사용할 수 있나요?

예, 이 가이드에서 설명하는 모든 기능은 AWS CDK에서 지원됩니다.

리소스

[FilterCriteria](#)(AWS Serverless Application Model 개발자 안내서)

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 [RSS 피드](#)를 구독하십시오.

변경 사항	설명	날짜
최초 게시	—	2022년 9월 30일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.