Amazon CloudWatch 로그

사용 설명서

Amazon CloudWatch 로그: 사용 설명서

Table of Contents

Amazon CloudWatch Logs이란 무엇입니까? 1
Features 1
관련 AWS 서비스 1
Pricing 2
개념 2
설정 4
Amazon Web Services(AWS)에 가입 4
Amazon CloudWatch 콘솔에 로그인 4
명령행 인터페이스 설정 4
시작하기 5
통합 CloudWatch 에이전트를 사용하여 CloudWatch Logs 시작하기 5
이전 CloudWatch Logs 에이전트를 사용하여 CloudWatch Logs 시작하기 5
CloudWatch Logs 에이전트 필수 조건 6
빠른 시작 실행 중인 EC2 Linux 인스턴스에 에이전트 설치 6
빠른 시작 Launch 시 EC2 Linux 인스턴스에 에이전트 설치 10
빠른 시작 사용 CloudWatch Logs Windows Server 2016 인스턴스 13
빠른 시작 Windows Server 2012 및 Windows Server 2008 인스턴스에서 CloudWatch Logs 사용 20
빠른 시작 에이전트 설치 AWS OpsWorks 26
CloudWatch Logs 에이전트 상태 보고 30
CloudWatch Logs 에이전트 시작 31
CloudWatch Logs 에이전트 중지 31
빠른 시작 사용 AWS CloudFormation 시작하기 CloudWatch Logs 31
CloudWatch Logs Insights로 로그 데이터 분석 33
지원되는 로그 및 검색되는 필드 33
JSON 로그의 필드 34
튜토리얼: 샘플 쿼리 실행 및 수정 35
샘플 쿼리 실행 36
샘플 쿼리 수정 36
샘플 쿼리에 필터 명령 추가 37
튜토리얼: 집계 함수를 사용하여 쿼리 실행 37
튜토리얼: 로그 필드별로 그룹화된 시각화를 생성하는 쿼리 실행 38
튜토리얼: 시계열 시각화를 생성하는 쿼리 실행 38
쿼리 구문 39
지원되는 쿼리 명령 39
필터 명령에서 일치 및 정규식 42
쿼리에 별칭 사용 42
쿼리에서 주석 사용 43
지원되는 연산 및 함수 43
그래프로 로그 데이터 시각화 48
시계열 데이터 시각화 48
필드별로 그룹화된 로그 데이터 시각화 48
쿼리 저장 및 재실행 49
샘플 쿼리 50
대시보드에 쿼리 추가 또는 쿼리 결과 내보내기 53
실행 중인 쿼리 또는 쿼리 기록 보기 53
로그 그룹 및 로그 스트림 작업 54
로그 그룹 생성 54
로그 그룹에 로그 보내기 54
로그 데이터 보기 54
필터 패턴을 사용하여 로그 데이터 검색 55
콘솔을 이용하여 로그 항목 검색 55
AWS CLI를 이용하여 로그 항목 검색 55
지표에서 로그로 피벗 적용 56
Troubleshooting 56

로그 데이터 보존 기간 변경 57
로그 그룹 태그 지정 57
태그 기본 사항 57
태그 지정을 사용하여 비용 추적 58
태그 제한 58
AWS CLI를 사용하여 로그 그룹에 태그 지정 58
CloudWatch Logs API를 사용하여 로그 그룹에 태그 지정 59
AWS KMS를 사용하여 로그 데이터 암호화 59
Limits 59
단계 1. 생성 AWS KMS CMK 60
단계 2. CMK에 대한 권한 설정 60
단계 3. 로그 그룹을 CMK와 연결 62
단계 4. CMK에서 로그 그룹 연결 해제 62
KMS 키 및 암호화 컨텍스트 62
특정 AWS 서비스에서 로깅 활성화 64
필터를 사용하여 로그 이벤트에서 지표 생성 65
Concepts 65
필터 및 패턴 구문 66
로그 이벤트에서 일치하는 단어 검색 66
일치가 발견될 경우 지표 값 변경 방법 설정 72
로그 항목에서 발견된 수치 값 게시 72
지표 필터 생성 73
예: 로그 이벤트 수 73
예: 용어의 카운트 발생 74
예: HTTP 404 코드 계산 75
예: HTTP 4xx 코드 개수 76
예: Apache Log에서 필드 추출 77
지표 필터 나열 78
지표 필터 삭제 79
구독을 통한 로그 데이터 실시간 처리 80
Concepts 80
구독 필터 사용 81
예 1: 구독 필터 Kinesis 81
예 2: 구독 필터 AWS Lambda 84
실시예 3: 구독 필터 Amazon Kinesis Data Firehose 87
구독과 교차 계정 로그 데이터 공유 91
대상 생성 92
구독 필터 생성 95
로그 이벤트 이동 검사 95
런타임 시 대상 멤버십 수정 96
로그 직접 보내기 Amazon S3 또는 Kinesis Data Firehose 98
Amazon S3로 로그 데이터 내보내기 99
Concepts 99
콘솔을 사용하여 Amazon S3으로 로그 데이터 내보내기 100
단계 1. Amazon S3 버킷 생성 100
단계 2. 생성 IAM 에 대한 전체 액세스 권한이 있는 사용자 Amazon S3 및 CloudWatch Logs 100
단계 3. 다음에 대한 권한 설정 Amazon S3 버킷 101
단계 4. 내보내기 작업 생성 102
AWS CLI를 사용하여 Amazon S3으로 로그 데이터 내보내기 102
단계 1. Amazon S3 버킷 생성 103
단계 2. 생성 IAM 에 대한 전체 액세스 권한이 있는 사용자 Amazon S3 및 CloudWatch Logs 103
단계 3. 다음에 대한 권한 설정 Amazon S3 버킷 104
단계 4. 내보내기 작업 생성 105
단계 5. 내보내기 작업 설명 106
단계 6. 내보내기 작업 취소 107
데이터를 Amazon ES로 스트리밍 108
사전 요구 사항 108

Amazon ES에 대한 로그 그룹 구독 108
로그를 게시하는 AWS 서비스 110
보안 112
데이터 보호 112
저장 데이터 암호화 113
전송 중 데이터 암호화 113
Identity and Access Management 113
Authentication 113
액세스 제어 114
액세스 관리 개요 115
자격 증명 기반 정책(IAM 정책) 사용 118
CloudWatch Logs 권한 참조 문서 123
서비스 연결 역할 사용 126
규정 준수 확인 128
복원성 128
인프라 보안 128
인터페이스 VPC 엔드포인트 129
가용성 129
CloudWatch Logs용 VPC 엔드포인트 생성 129
VPC와 CloudWatch Logs 간의 연결 테스트 130
CloudWatch Logs VPC 엔드포인트에 대한 액세스 제어 130
VPC 컨텍스트 키에 대한 지원 131
API 호출 로깅 132
CloudTrail의 CloudWatch Logs 정보 132
로그 파일 항목 이해 133
에이전트 참조 135
에이전트 구성 파일 135
HTTP 프록시와 함께 CloudWatch Logs 에이전트 사용 139
CloudWatch Logs 에이전트 구성 파일 분류 139
CloudWatch Logs 에이전트 FAQ 140
CloudWatch 지표를 통한 사용량 모니터링 143
CloudWatch Logs 지표 143
CloudWatch Logs 지표의 차원 144
Service Quotas 145
문서 기록 147
AWS Glossary 149
..... cl

Amazon CloudWatch Logs이란 무엇입니까?

Amazon CloudWatch Logs를 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, AWS CloudTrail, Route 53 및 기타 소스에서 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다.

CloudWatch Logs는 확장성이 뛰어난 단일 서비스에서 사용하는 모든 시스템, 애플리케이션 및 AWS 서비스에서 로그를 중앙 집중화할 수 있습니다. 따라서 이들을 손쉽게 확인하여 특정 오류 코드나 패턴을 검색하거나 특정 필드를 기준으로 필터링하거나 향후 분석을 위해 안전하게 보관할 수 있습니다. CloudWatch Logs을 사용하면 소스에 관계 없이 모든 로그를 시간에 따라 정렬된 일관된 단일 흐름의 형태로 확인할 수 있습니다. 또한 다른 차원을 기준으로 이들을 쿼리 및 정렬하고, 특정 필드에 따라 그룹화하며, 강력한 쿼리 언어를 이용해 사용자 지정 계산을 생성하고, 대시보드에 로그 데이터를 시각화할 수 있습니다.

Features

- 데이터 쿼리 – CloudWatch Logs Insights를 사용하여 로그 데이터를 대화식으로 검색하고 분석할 수 있습니다. 쿼리를 수행하여 운영 문제에 보다 효율적이고 효과적으로 대응할 수 있습니다. CloudWatch Logs Insights에는 몇 가지 간단하지만 강력한 명령을 사용하여 특별히 제작된 쿼리 언어가 포함되어 있습니다. 시작하는 데 도움이 되는 샘플 쿼리, 명령 설명, 쿼리 자동 완성 및 로그 필드 검색이 제공됩니다. 여러 가지 유형의 AWS 서비스 로그에 대한 샘플 쿼리가 포함되어 있습니다. 시작하려면 CloudWatch Logs Insights로 로그 데이터 분석 (p. 33)단원을 참조하십시오.
- Amazon EC2 인스턴스 로그 모니터링 – CloudWatch Logs를 사용하면 로그 데이터를 통해 애플리케이션과 시스템을 모니터링할 수 있습니다. 예를 들어 CloudWatch Logs에서는 애플리케이션 로그에서 발생하는 오류의 수를 추적하고 오류 비율이 지정한 임계값을 초과할 때마다 알림을 전송할 수 있습니다. CloudWatch Logs는 모니터링하는 데 로그 데이터를 사용하므로 코드를 변경할 필요가 없습니다. 예를 들어 특정 리터럴 문자(예: "NullReferenceException")에 대한 애플리케이션 로그를 모니터링하거나 로그 데이터의 특정 위치(예: Apache 액세스 로그의 "404" 상태 코드)에서 리터럴 문자의 출현 횟수를 계산할 수 있습니다. 검색할 단어가 발견되면 CloudWatch Logs는 지정한 CloudWatch 지표로 데이터를 보고합니다. 로드 데이터는 전송 시는 물론 저장 시에도 암호화됩니다. 시작하려면 CloudWatch Logs 시작하기 (p. 5)단원을 참조하십시오.
- AWS CloudTrail 기록 이벤트 모니터링 – CloudWatch에서 경보를 생성하고, CloudTrail에서 포착된 특정 API 활동에 대한 알림을 수신하며, 이러한 알림을 사용하여 문제 해결을 할 수 있습니다. 시작하려면 다음을 참조하십시오. 전송 중 CloudTrail 이벤트 대상 CloudWatch Logs 에서 AWS CloudTrail User Guide.
- 로그 보존 – 기본적으로 로그는 무기한으로 저장되고 만료 기간이 없습니다. 로그 그룹별로 보존 정책을 조정할 수 있고 무기한 보존 유지 또는 10년 및 하루 중 보존 기간을 선택합니다.
- 로그 데이터 아카이브 – CloudWatch Logs를 사용하여 내구성이 뛰어난 스토리지에 로그 데이터를 저장할 수 있습니다. CloudWatch Logs 에이전트를 사용하면 호스트에서 로그 서비스로 로테이션 된 로그 데이터와 로테이션 되지 않은 로그 데이터를 모두 쉽고 빠르게 전송할 수 있습니다. 그런 다음 필요할 때 원시 로그 데이터에 액세스할 수 있습니다.
- 로그 Route 53 DNS 쿼리 – CloudWatch Logs를 사용하여 Route 53 가 수신하는 DNS 쿼리에 대한 정보를 기록할 수 있습니다. 자세한 내용은 을 참조하십시오. DNS 쿼리 로깅 에서 Amazon Route 53 개발자 안내서.

관련 AWS 서비스

다음 서비스가 CloudWatch Logs와 함께 사용됩니다.

- AWS CloudTrail은 AWS Management 콘솔, AWS Command Line Interface(AWS CLI) 및 기타 서비스를 통해 이루어진 호출을 포함하여 계정에서 CloudWatch Logs API에 대한 호출을 모니터링할 수 있도록 합니다. CloudTrail 로깅이 활성화되면 CloudTrail는 계정에서 API 호출을 포착하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 각 로그 파일에는 하나 이상의 레코드가 포함될 수 있으며, 요청을 충족하기 위해 수행해야 하는 작업의 수에 따라 그 수가 결정됩니다. 에 대한 추가 정보 AWS CloudTrail, 참조 정의 AWS CloudTrail? 에서 AWS CloudTrail User Guide. CloudWatch가 CloudTrail 로그 파일에 기록하는 데이터 유형의 예제는 AWS CloudTrail에서 Amazon CloudWatch Logs API 호출 로깅 (p. 132) 단원을 참조하십시오.
- AWS Identity and Access Management (IAM)은(는) 사용자를 위해 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있는 웹 서비스입니다. IAM을 사용하여 AWS 리소스를 사용할 수 있는 사람을 제어(인증)하고 이들이 사용할 수 있는 리소스 및 그 사용 방법을 제어(권한 부여)합니다. 자세한 내용은 을 참조하십시오. 정의 IAM? 에서 IAM 사용 설명서.
- Amazon Kinesis Data Streams는 신속하고 지속적인 데이터 인테이크 및 집계를 위해 사용할 수 있는 웹 서비스입니다. 사용되는 데이터 유형으로는 IT 인프라 로그 데이터, 애플리케이션 로그, 소셜 미디어, 시장 데이터 피드, 웹 클릭스트림 데이터 등이 있습니다. 데이터 인테이크 및 처리에 대한 응답이 실시간으로 이루어지기 때문에 일반적으로 간소화된 방식으로 처리가 됩니다. Amazon Kinesis Data StreamsAmazon Kinesis Data Streams 개발자 안내서자세한 정보는 의 란 무엇입니까? 단원을 참조하십시오.
- AWS Lambda는 새 정보에 신속하게 응답하는 애플리케이션을 구축하는 데 사용할 수 있는 웹 서비스입니다. Lambda 함수 형태로 애플리케이션 코드를 업로드하면 Lambda는 고가용성 컴퓨팅 인프라에서 코드를 실행하고 서버 및 운영 체제 유지 관리, 용량 프로비저닝 및 자동 조정, 코드 및 보안 패치 배포, 코드 모니터링 및 로깅 등 모든 컴퓨팅 리소스 관리를 수행합니다. Lambda가 지원하는 언어 중 하나로 코드를 공급하기만 하면 됩니다. AWS LambdaAWS Lambda Developer Guide자세한 정보는 의 란 무엇입니까? 단원을 참조하십시오.

Pricing

AWS 가입 시 무상으로 CloudWatch Logs를 시작할 수 있는 AWS 프리 티어를 제공합니다.

표준 요금은 CloudWatch Logs를 사용하여 다른 서비스가 저장한 로그(예: Amazon VPC 흐름 로그 및 Lambda 로그)에 적용됩니다.

자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

Amazon CloudWatch Logs 개념

CloudWatch Logs의 이해 및 사용에 핵심이 되는 용어 및 개념에 대한 설명은 다음과 같습니다.

로그 이벤트

로그 이벤트는 모니터링 중인 애플리케이션 또는 리소스에 기록된 일부 활동에 대한 레코드입니다. CloudWatch Logs가 파악한 로그 이벤트 레코드에는 이벤트가 발생한 시점에 대한 타임스탬프와 원시 이벤트 메시지 등 두 개의 속성이 포함되어 있습니다. 각 메시지는 UTF-8로 인코딩되어야 합니다.

로그 스트림

로그 스트림은 동일한 소스를 공유하는 로그 이벤트 시퀀스입니다. 보다 구체적으로 말하자면, 로그 스트림은 모니터링 중인 애플리케이션 인스턴스나 리소스에서 나온 이벤트의 시퀀스를 표시하는 데 주로 사용됩니다. 예를 들어 로그 스트림은 특정 호스트의 Apache 액세스 로그에 연결될 수 있습니다. 로그 스트림이 더 이상 필요하지 않으면 aws logs delete-log-stream 명령을 사용하여 이를 삭제할 수 있습니다.

로그 그룹

로그 그룹은 동일한 보존 기간, 모니터링 및 액세스 제어 설정을 공유하는 로그 스트림 그룹을 정의합니다. 각 로그 스트림은 하나의 로그 그룹에 속해야 합니다. 예를 들어, 각 호스트의 Apache 액세스 로그에

대해 별도의 로그 스트림이 있으면 로그 스트림을 `MyWebsite.com/Apache/access_log`라는 하나의 로그 그룹으로 묶을 수 있습니다.

하나의 로그 그룹에서 포함할 수 있는 로그 스트림의 수에는 제한이 없습니다.

지표 필터

지표 필터를 사용하여 수집된 이벤트에서 지표 관찰값을 추출하고 이를 CloudWatch 지표의 데이터 요소로 변환할 수 있습니다. 지표 필터는 로그 그룹에 할당이 되고, 로그 그룹에 할당된 모든 필터들은 로그 스트림에 적용됩니다.

보존 기간 설정

보존 기간 설정은 CloudWatch Logs에 로그 이벤트를 보관하는 기간을 설정하는 데 사용할 수 있습니다. 기간이 만료된 로그 이벤트는 자동으로 삭제됩니다. 지표 필터와 마찬가지로 보존 기간 설정 역시 로그 그룹에 할당이 되며, 로그 그룹에 할당된 보존 기간은 로그 스트림에 적용됩니다.

설정

Amazon CloudWatch Logs를 사용하려면 AWS 계정이 있어야 합니다. AWS 계정이 있어야 서비스(예: Amazon EC2)를 사용해 웹 기반 인터페이스인 CloudWatch 콘솔에서 확인 가능한 로그를 생성할 수 있습니다. 뿐만 아니라 AWS Command Line Interface(AWS CLI)를 설치 및 구성할 수 있습니다.

Amazon Web Services(AWS)에 가입

AWS 계정을 생성하면 모든 AWS 서비스에 자동으로 계정이 등록됩니다. 사용한 서비스에 대해서만 지불하면 됩니다.

이미 AWS 계정이 있다면 다음 단계로 건너뛰십시오. AWS 계정이 없는 경우에는 아래 단계를 수행하여 계정을 만드십시오.

AWS 계정에 가입하려면 다음을 수행합니다.

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   등록 절차 중 전화를 받고 전화 키패드를 사용하여 확인 코드를 입력하는 과정이 있습니다.

Amazon CloudWatch 콘솔에 로그인

Amazon CloudWatch 콘솔에 로그인하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 변경합니다. 탐색 모음에서 AWS 리소스가 상주하는 리전을 선택합니다.
3. 탐색 창에서 로그를 선택합니다.

명령행 인터페이스 설정

AWS CLI를 사용하여 CloudWatch Logs 작업을 수행할 수 있습니다.

AWS CLI의 설치 및 구성 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS 명령줄 인터페이스를 사용한 설정을 참조하십시오.

CloudWatch Logs 시작하기

Amazon EC2 인스턴스 및 온프레미스 서버의 로그를 CloudWatch Logs로 수집할 수 있도록 AWS는 두 가지 옵션을 제공합니다.

- 권장 사항 – 통합 CloudWatch 에이전트. 에이전트 하나로 로그와 고급 지표를 모두 수집할 수 있습니다. Windows Server를 실행하는 서버를 포함하여 운영 체제 전반에 걸쳐 지원을 제공합니다. 또한 이 에이전트는 우수한 성능을 제공합니다.

  통합 에이전트를 통해 CloudWatch 지표를 수집하는 경우 인게스트 표시 여부에 대해 추가 시스템 지표를 수집할 수도 있습니다. 또한 `StatsD` 또는 `collectd`를 사용하는 사용자 지정 지표 수집을 지원합니다.

  자세한 내용은 설치 CloudWatch 에이전트 in the Amazon CloudWatch 사용 설명서.
- 지원 사항(더 이상 사용되지 않을 예정임) - Linux를 실행하는 서버에서만 로그 수집을 지원하는 이전의 CloudWatch Logs 에이전트. 이미 해당 에이전트를 사용하고 있으면 계속 사용할 수 있습니다. 그러나 이전 에이전트는 Python 2.7, 3.0 및 3.3이 필요합니다. 현재 EC2 인스턴스는 이러한 버전의 Python을 사용하지 않으며 이러한 버전은 더 이상 사용되지 않으며 패치가 적용되지 않으므로 통합 CloudWatch 에이전트로 마이그레이션하는 것이 좋습니다.

  CloudWatch Logs 에이전트에서 통합 CloudWatch 에이전트로 마이그레이션하려면 통합 에이전트의 구성 마법사를 통해 현재 CloudWatch Logs 에이전트 구성 파일을 읽고 새 에이전트를 설정하여 동일한 로그를 수집할 수 있습니다. 마법사에 대한 자세한 내용은 다음을 참조하십시오. 생성 CloudWatch 마법사가 포함된 에이전트 구성 파일 in the Amazon CloudWatch 사용 설명서.

내용

- 통합 CloudWatch 에이전트를 사용하여 CloudWatch Logs 시작하기 (p. 5)
- 이전 CloudWatch Logs 에이전트를 사용하여 CloudWatch Logs 시작하기 (p. 5)
- 빠른 시작 사용 AWS CloudFormation 시작하기 CloudWatch Logs (p. 31)

통합 CloudWatch 에이전트를 사용하여 CloudWatch Logs 시작하기

통합 사용에 대한 자세한 정보 CloudWatch 에이전트 시작 CloudWatch Logs, 참조: 다음에서 메트릭 및 로그 수집 Amazon EC2 인스턴스 및 온프레미스 서버 CloudWatch 에이전트 in the Amazon CloudWatch 사용 설명서. 이 섹션에 나열된 단계를 완료하여 에이전트를 설치, 구성 및 시작합니다. 에이전트를 사용하여 CloudWatch 지표를 수집하지 않는 경우 지표를 참조하는 모든 섹션을 무시할 수 있습니다.

현재 기존 CloudWatch Logs 에이전트를 사용하고 있으며 새로운 통합 에이전트를 사용하기 위해 마이그레이션하려는 경우 새 에이전트 패키지에 포함된 마법사를 사용하는 것이 좋습니다. 이 마법사는 현재 CloudWatch Logs 에이전트 구성 파일을 읽고 동일한 로그를 수집하도록 CloudWatch 에이전트를 설정할 수 있습니다. 마법사에 대한 자세한 내용은 다음을 참조하십시오. 생성 CloudWatch 마법사가 포함된 에이전트 구성 파일 in the Amazon CloudWatch 사용 설명서.

이전 CloudWatch Logs 에이전트를 사용하여 CloudWatch Logs 시작하기

CloudWatch Logs 에이전트를 사용하면 Linux 또는 Windows Server를 실행하는 Amazon EC2 인스턴스에서 나온 로그 데이터와 AWS CloudTrail에서 나온 로그 이벤트를 게시할 수 있습니다. 대신 CloudWatch 통합

에이전트를 사용하여 로그 데이터를 게시하는 것이 좋습니다. 새 에이전트에 대한 자세한 내용은 다음을 참조하십시오. 다음에서 메트릭 및 로그 수집 Amazon EC2 인스턴스 및 온프레미스 서버 CloudWatch 에이전트 in the Amazon CloudWatch 사용 설명서. 또는 이전 CloudWatch Logs 에이전트를 계속 사용할 수 있습니다.

내용

- CloudWatch Logs 에이전트 필수 조건 (p. 6)
- 빠른 시작 설치 및 구성 CloudWatch Logs 실행 중인 EC2 Linux 인스턴스에 대한 에이전트 (p. 6)
- 빠른 시작 설치 및 구성 CloudWatch Logs 시작 시 EC2 Linux 인스턴스의 에이전트 (p. 10)
- 빠른 시작 활성화 Amazon EC2 로그 전송을 위해 Windows Server 2016을 실행하는 인스턴스 CloudWatch Logs 사용 CloudWatch Logs 에이전트 (p. 13)
- 빠른 시작 Windows Server 2012 및 Windows Server 2008을 실행하는 Amazon EC2 인스턴스가 CloudWatch Logs로 로그를 전송하도록 설정 (p. 20)
- 빠른 시작 설치 CloudWatch Logs 에이전트 사용 AWS OpsWorks 셰프 (p. 26)
- CloudWatch Logs 에이전트 상태 보고 (p. 30)
- CloudWatch Logs 에이전트 시작 (p. 31)
- CloudWatch Logs 에이전트 중지 (p. 31)

CloudWatch Logs 에이전트 필수 조건

CloudWatch Logs 에이전트에는 Python 버전 2.7, 3.0 또는 3.3과 다음과 같은 버전의 Linux가 필요합니다.

- Amazon Linux버전 2014.03.02 이상 Amazon Linux 2는 지원되지 않습니다.
- Ubuntu Server 버전 12.04, 14.04 또는 16.04
- CentOS 버전 6, 6.3, 6.4, 6.5 또는 7.0
- Red Hat Enterprise Linux(RHEL) 버전 6.5 또는 7.0
- Debian 8.0

빠른 시작 설치 및 구성 CloudWatch Logs 실행 중인 EC2 Linux 인스턴스에 대한 에이전트

> **Tip**
>
> CloudWatch는 EC2 인스턴스 및 온프레미스 서버에서 로그와 지표를 모두 수집할 수 있는 통합 에이전트가 새롭게 추가되었습니다. 이미 이전 CloudWatch Logs 에이전트의 사용을 중단하였다면 새롭게 통합된 CloudWatch 에이전트 사용을 권장합니다. 자세한 정보는 CloudWatch Logs 시작하기 (p. 5) 단원을 참조하십시오.
> 이번 단원의 나머지에서는 이전 CloudWatch Logs 에이전트의 사용에 대해서 설명하겠습니다.

실행 중인 EC2 Linux 인스턴스에서 이전 CloudWatch Logs 에이전트 구성

기존 EC2 인스턴스에서 CloudWatch Logs 에이전트 설치 프로그램을 사용하여 CloudWatch Logs 에이전트를 설치 및 구성할 수 있습니다. 설치가 완료되면 로그가 자동으로 인스턴스에서 에이전트 설치 도중 생성한 로그 스트림으로 흐릅니다. 에이전트는 인스턴트가 시작되었고 비활성화를 할 때까지 실행이 유지되는지 확인합니다.

에이전트를 사용하는 외에도 AWS CLI, CloudWatch Logs SDK 또는 CloudWatch Logs API를 사용하여 로그 데이터를 게시할 수도 있습니다. AWS CLI는 명령줄에서, 또는 스크립트를 통해 데이터를 게시하기에 가장

적합합니다. CloudWatch Logs SDK는 애플리케이션에서 직접 로그 데이터를 게시하거나 자체적으로 로그 게시 애플리케이션을 구축하기에 가장 적합합니다.

단계 1. 구성 IAM 의 역할 또는 사용자 CloudWatch Logs

CloudWatch Logs 에이전트는 IAM 역할 및 사용자를 지원합니다. 인스턴스가 이미 IAM 역할에 연결되어 있으면 아래의 IAM 정책이 포함되어 있는지 확인합니다. 인스턴스에 IAM 역할이 할당되어 있지 않으면 다음 단계에서 IAM 자격 증명을 사용하거나 해당 인스턴스에 IAM 역할을 할당할 수 있습니다. 자세한 내용은 IAM 역할을 인스턴스에 연결을 참조하십시오.

CloudWatch Logs에 대한 IAM 역할이나 사용자를 구성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Roles(역할)를 선택합니다.
3. 역할 이름을 선택하여 역할을 선택합니다(이름 옆의 확인란은 선택하지 않음).
4. 정책 연결에서 정책 생성을 선택합니다.

   새 브라우저 탭 또는 창이 열립니다.
5. JSON 탭을 선택하고 다음 JSON 정책 문서를 입력합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "*"
    ]
  }
 ]
}
```

6. 작업이 완료되면 정책 검토를 선택합니다. 정책 검사기가 모든 구문 오류를 보고합니다.
7. 정책 검토 페이지에서 생성하는 정책에 대한 이름과 설명(선택 사항)을 입력합니다. 정책 요약을 검토하여 정책에 부여된 권한을 확인합니다. 그런 다음 [Create policy]를 선택하여 작업을 저장합니다.
8. 브라우저 탭 또는 창을 닫고, 역할의 권한 추가 페이지로 돌아옵니다. 새로 고침을 선택한 후 새 정책을 선택하여 역할에 연결합니다.
9. Attach Policy(정책 연결)를 선택합니다.

단계 2. 설치 및 구성 CloudWatch Logs 기존 Amazon EC2 인스턴스

CloudWatch Logs 에이전트 설치 프로세스는 Amazon EC2 인스턴스가 Amazon Linux, Ubuntu, CentOS 또는 Red Hat 중 무엇을 실행하느냐에 따라 다릅니다. 인스턴스에서 Linux 버전에 적합한 단계를 사용합니다.

기존 Amazon Linux 인스턴스에 CloudWatch Logs을 설치 및 구성하려면

2014년 9월에 출시된 Amazon Linux AMI 버전부터는 CloudWatch Logs 에이전트를 awslogs 패키지에서 RPM 설치 프로그램으로 사용할 수 있습니다. 이전 버전의 Amazon Linux은 `sudo yum update -y` 명령을 통해 인스턴스를 업데이트하여 awslogs 패키지에 액세스할 수 있습니다. CloudWatch Logs 설치 프로그램을 사용하는 대신 RPM으로 awslogs 패키지를 설치하면 CloudWatch Logs 에이전트를 수동으로 재설치할 필요 없이 인스턴스가 AWS에서 정기적으로 패키지에 대한 업데이트 및 패치를 수신할 수 있습니다.

Warning

이전에 Python 스크립트를 사용해 에이전트를 설치한 경우에는 RPM 설치 방법을 이용해 CloudWatch Logs 에이전트를 업데이트하지 마십시오. 이로 인해 구성 문제가 발생하여 CloudWatch Logs 에이전트가 CloudWatch에 로그를 전송하지 못할 수 있습니다.

1. Amazon Linux 인스턴스에 연결합니다. 자세한 내용은 인스턴스에 연결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.

   연결 문제에 대한 자세한 내용은 다음을 참조하십시오. 인스턴스에 연결하는 문제 해결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.
2. Amazon Linux 인스턴스를 업데이트하여 패키지 리포지토리에서 최신 변경 사항을 찾아냅니다.

   ```
   sudo yum update -y
   ```
3. `awslogs` 패키지를 설치합니다. 이는 Amazon Linux 인스턴스에서 awslogs를 설치하는 데 있어 권장되는 방법입니다.

   ```
   sudo yum install -y awslogs
   ```
4. `/etc/awslogs/awslogs.conf` 파일을 편집하여 추적할 로그를 구성합니다. 이 파일을 편집하는 방법에 대한 자세한 내용은 CloudWatch Logs 에이전트 참조 (p. 135) 단원을 참조하십시오.
5. 기본적으로 `/etc/awslogs/awscli.conf`는 us-east-1 리전을 가리킵니다. 다른 리전으로 로그를 푸시하려면 `awscli.conf` 파일을 편집하고 해당 리전을 지정합니다.
6. `awslogs` 서비스를 시작합니다.

   ```
   sudo service awslogs start
   ```

   Amazon Linux 2를 실행 중인 경우 다음 명령과 함께 `awslogs` 서비스를 시작합니다.

   ```
   sudo systemctl start awslogsd
   ```
7. (선택 사항) 서비스 시작 시 기록된 오류에 대해서는 `/var/log/awslogs.log` 파일을 확인하십시오.
8. (선택 사항) 다음 명령을 실행하여 시스템 부팅 때마다 `awslogs` 서비스를 시작합니다.

   ```
   sudo chkconfig awslogs on
   ```

   Amazon Linux 2를 실행 중인 경우 다음 명령으로 각 시스템 부팅에서 서비스를 시작합니다.

   ```
   sudo systemctl enable awslogsd.service
   ```
9. 에이전트가 몇 분 동안 실행된 이후에 CloudWatch 콘솔에 새로 생성된 로그 그룹 및 로그 스트림이 나타납니다.

   자세한 정보는 CloudWatch Logs에 전송된 로그 데이터 보기 (p. 54) 단원을 참조하십시오.

기존의 Ubuntu Server, CentOS 또는 Red Hat 인스턴스에 CloudWatch Logs을 설치 및 구성하려면

Ubuntu Server, CentOS 또는 Red Hat을 실행하는 AMI를 사용하고 있는 경우에는 다음 절차를 이용해 인스턴스에 CloudWatch Logs 에이전트를 수동으로 설치합니다.

1. EC2 인스턴스에 연결합니다: 자세한 내용은 인스턴스에 연결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.

연결 문제에 대한 자세한 내용은 다음을 참조하십시오. 인스턴스에 연결하는 문제 해결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.

2. 두 옵션 중 하나를 사용하여 CloudWatch Logs 에이전트 설치 프로그램을 실행합니다. 인터넷에서 직접 실행하거나 파일을 다운로드하여 독립적으로 실행할 수 있습니다.

   Note

   CentOS 6.x, Red Hat 6.x, 또는 Ubuntu 12.04를 사용 중인 경우 독립 실행형 설치 관리자의 다운로드 및 실행 단계를 사용합니다. 이들 시스템에서는 CloudWatch Logs 에이전트를 인터넷에서 직접 설치하는 것이 지원되지 않습니다.

   Note

   아래 명령을 실행하기 전에 Ubuntu에서 `apt-get update`를 실행합니다.

   인터넷에서 직접 실행하려면 다음 명령을 사용하고 프롬프트의 메시지를 따릅니다.

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
   ```

   ```
   sudo python ./awslogs-agent-setup.py --region us-east-1
   ```

   위의 명령이 작동하지 않으면 다음 작업을 시도합니다.

   ```
   sudo python3 ./awslogs-agent-setup.py --region us-east-1
   ```

   단독으로 다운로드 및 실행하려면 다음 명령을 사용하고 프롬프트의 메시지를 따릅니다.

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
   ```

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/AgentDependencies.tar.gz
    -O
   ```

   ```
   tar xvf AgentDependencies.tar.gz -C /tmp/
   ```

   ```
   sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/
   AgentDependencies
   ```

   us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1 리전을 지정하여 CloudWatch Logs 에이전트를 설치할 수 있습니다.

   Note

   최신 버전과 `awslogs-agent-setup`의 버전 이력에 대한 자세한 내용은 CHANGELOG.txt를 참조하십시오.

   CloudWatch Logs 에이전트 설치 프로그램은 설정 단계에서 특정 정보를 요구합니다. 시작에 앞서 모니터링할 로그 파일과 타임스탬프 형식을 알아야 합니다. 또한 다음 정보를 이미 확보하고 있어야 합니다.

항목	설명
AWS 액세스 키 ID	IAM 역할을 사용하고 있는 경우에는 Enter 키를 누릅니다. 아니면 AWS 액세스 키 ID를 입력합니다.
AWS 보안 액세스 키	IAM 역할을 사용하고 있는 경우에는 Enter 키를 누릅니다. 아니면 AWS 보안 액세스 키를 입력합니다.
기본 리전 이름	Enter를 누릅니다. 기본값은 us-east-2입니다. 이를 us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1로 설정할 수 있습니다.
기본 출력 형식	공백으로 남겨 두고 Enter를 누릅니다.
업로드할 로그 파일의 경로	전송할 로그 데이터가 포함된 파일의 위치입니다. 설치 프로그램이 경로를 제안합니다.
대상 로그 그룹 이름	로그 그룹의 이름. 설치 프로그램이 로그 그룹 이름을 제안합니다.
대상 로그 스트림 이름	기본적으로 이 이름은 호스트 이름입니다. 설치 프로그램이 호스트 이름을 제안합니다.
타임스탬프 형식	지정된 로그 파일 내의 타임스탬프 형식을 지정합니다. 사용자 지정을 선택해서 자체 형식을 지정합니다.
초기 위치	데이터가 업로드된 방법입니다. 이를 start_of_file로 설정하면 데이터 파일에 모든 것이 업로드됩니다. end_of_file로 설정하면 새로 추가된 데이터만 업로드됩니다.

이러한 단계가 완료되면 설치 프로그램이 또 다른 로그 파일 구성에 대해 묻습니다. 각 로그 파일에서 원하는 횟수 만큼 프로세스를 실행할 수 있습니다. 더 이상 모니터링할 로그 파일이 없고 설치 프로그램에 또 다른 로그를 설정하라는 프롬프트 메시지가 나타나면 N을 선택합니다. 에이전트 구성 파일을 설정하는 방법에 대한 자세한 내용은 CloudWatch Logs 에이전트 참조 (p. 135) 단원을 참조하십시오.

> **Note**
>
> 단일 로그 스트림으로 데이터를 전송하기 위해 여러 개의 로그 소스를 구성하는 것이 불가능합니다.

3. 에이전트가 몇 분 동안 실행된 이후에 CloudWatch 콘솔에 새로 생성된 로그 그룹 및 로그 스트림이 나타납니다.

   자세한 정보는 CloudWatch Logs에 전송된 로그 데이터 보기 (p. 54) 단원을 참조하십시오.

빠른 시작 설치 및 구성 CloudWatch Logs 시작 시 EC2 Linux 인스턴스의 에이전트

> **Tip**
>
> 이 단원에서 설명하는 이전 CloudWatch Logs 에이전트는 사용 중단되는 과정에 있습니다. 로그와 지표 둘 다를 수집할 수 있는 새로운 통합 CloudWatch 에이전트를 대신 사용하는 것이 좋습니다. 또한 이전 CloudWatch Logs 에이전트에는 Python 3.3 또는 이전 버전이 필요하며 이러한 버전은 기본적으로 새 EC2 인스턴스에 설치되지 않습니다. 통합 CloudWatch 에이전트에 대한 자세한 내용은 CloudWatch 에이전트 설치를 참조하십시오.
> 이번 단원의 나머지에서는 이전 CloudWatch Logs 에이전트의 사용에 대해서 설명하겠습니다.

EC2 Linux 인스턴스를 시작할 때 이전 CloudWatch Logs 에이전트 설치

시작 시 파라미터 정보를 인스턴스에 전달할 수 있게 해주는 Amazon EC2 기능인 Amazon EC2 사용자 데이터를 사용하여 해당 인스턴스에 CloudWatch Logs 에이전트를 설치 및 구성합니다. CloudWatch Logs 에이전트 설치 및 구성 정보를 Amazon EC2로 전달하려면 Amazon S3 버킷 같은 네트워크 위치에 구성 파일을 제공할 수 있습니다.

단일 로그 스트림으로 데이터를 전송하기 위해 여러 개의 로그 소스를 구성하는 것이 불가능합니다.

Prerequisite

모든 로그 그룹 및 로그 스트림을 설명하는 에이전트 구성 파일을 생성합니다. 모니터링할 로그 파일을 비롯해 로그 파일을 업로드할 로그 그룹 및 로그 스트림을 설명하는 텍스트 파일입니다. 에이전트는 이 구성 파일을 사용하여 여기에 설명된 모든 로그 파일들에 대한 모니터링 및 업로드를 시작합니다. 에이전트 구성 파일을 설정하는 방법에 대한 자세한 내용은 CloudWatch Logs 에이전트 참조 (p. 135) 단원을 참조하십시오.

다음은 Amazon Linux를 위한 에이전트 구성 파일 예제입니다.

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

다음은 Ubuntu을 위한 에이전트 구성 파일 예제입니다.

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

IAM 역할을 구성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Policies를 선택한 후 Create Policy를 선택합니다.
3. 정책 생성 페이지의 Create Your Own Policy(자체 정책 생성)에서 선택을 선택합니다. 사용자 지정 정책 생성에 대한 자세한 내용은 다음을 참조하십시오. Amazon EC2에 대한 IAM 정책 in the Linux 인스턴스용 Amazon EC2 사용 설명서.
4. 정책 검토 페이지의 정책 이름에 정책 이름을 입력합니다.
5. 정책 문서에 다음 정책을 붙여 넣습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
```

```
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::myawsbucket/*"
            ]
        }
    ]
}
```

6. 정책 생성을 선택합니다.
7. 탐색 창에서 [Roles], [Create New Role]을 선택합니다.
8. Set Role Name(역할 이름 설정) 페이지에서 역할 이름을 입력한 다음 다음 단계를 선택합니다.
9. On 역할 유형 선택 페이지, 선택 선택 다음 날짜 아마존 EC2.
10. 정책 연결 페이지의 테이블 헤더에서 정책 유형과 Customer Managed(고객 관리형)를 선택합니다.
11. 생성한 IAM 정책을 선택하고 다음 단계를 선택합니다.
12. [Create Role]을 선택합니다.

    자세한 정보는 IAM 사용자 및 정책 IAM 사용자 및 그룹 and IAM 정책 관리 in the IAM 사용 설명서.

새 인스턴스를 시작하고 CloudWatch Logs을 활성화하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. [Launch Instance]를 선택합니다.

   자세한 내용은 인스턴스 시작 에서 Linux 인스턴스용 Amazon EC2 사용 설명서.
3. On 1단계: Amazon Machine Image(AMI) 선택 페이지, 실행할 Linux 인스턴스 유형을 선택한 다음 2단계: 인스턴스 유형 선택 페이지, 선택 다음: 인스턴스 세부 정보 구성

   확실히 클라우드-초기화 Amazon Machine Image(AMI)에 포함되어 있습니다. Amazon Linux amis, 그리고 Ubuntu와 RHEL의 amis는 이미 클라우드-init, 그러나 centos 및 기타 amis는 AWS Marketplace 은(는) 에 없습니다.
4. On 3단계: 인스턴스 세부 정보 구성 페이지, IAM 역할, 를 선택합니다. IAM 을(를) 생성했습니다.
5. 고급 세부 정보의 사용자 데이터에 다음 스크립트를 붙여 넣습니다. 그런 다음, -c 옵션의 값을 에이전트 구성 파일의 위치로 변경하여 해당 스크립트를 업데이트합니다.

```
#!/bin/bash
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
chmod +x ./awslogs-agent-setup.py
./awslogs-agent-setup.py -n -r us-east-1 -c s3://DOC-EXAMPLE-BUCKET1/my-config-file
```

6. 인스턴스를 더 변경하고 시작 설정을 검토한 다음 시작을 선택합니다.
7. 에이전트가 몇 분 동안 실행된 이후에 CloudWatch 콘솔에 새로 생성된 로그 그룹 및 로그 스트림이 나타납니다.

   자세한 정보는 CloudWatch Logs에 전송된 로그 데이터 보기 (p. 54) 단원을 참조하십시오.

빠른 시작 활성화 Amazon EC2 로그 전송을 위해 Windows Server 2016을 실행하는 인스턴스 CloudWatch Logs 사용 CloudWatch Logs 에이전트

Tip

CloudWatch는 EC2 인스턴스 및 온프레미스 서버에서 로그와 지표를 모두 수집할 수 있는 통합 에이전트가 새롭게 추가되었습니다. 새롭게 통합된 CloudWatch 에이전트의 사용을 권장합니다. 자세한 정보는 CloudWatch Logs 시작하기 (p. 5) 단원을 참조하십시오.
이번 단원의 나머지에서는 이전 CloudWatch Logs 에이전트의 사용에 대해서 설명하겠습니다.

이전 CloudWatch Logs 에이전트를 사용하여 Windows Server 2016을 실행하는 Amazon EC2 인스턴스가 CloudWatch Logs로 로그를 전송하도록 설정

Windows Server 2016을 실행하는 인스턴스가 CloudWatch Logs로 로그를 전송하도록 설정하는 방법은 여러 가지가 있습니다. 이 단원의 단계에서는 시스템 관리자 Run Command를 사용합니다. 다른 방법에 대한 자세한 내용은 Amazon CloudWatch로 로그, 이벤트, 성능 카운터 전송 단원을 참조하십시오.

단계

- 샘플 구성 파일 다운로드 (p. 13)
- CloudWatch에 JSON 파일 구성 (p. 13)
- 시스템 관리자를 위한 IAM 사용자 및 역할 만들기 (p. 19)
- 시스템 관리자 사전 조건 확인 (p. 19)
- 인터넷 액세스 확인 (p. 19)
- 시스템 관리자 Run Command를 사용하여 CloudWatch Logs 활성화 (p. 19)

샘플 구성 파일 다운로드

다음 샘플 파일을 컴퓨터에 다운로드합니다. AWS.EC2.Windows.cloudwatch.json.

CloudWatch에 JSON 파일 구성

구성 파일에서 선택 항목을 지정하여 CloudWatch로 전송할 로그를 결정합니다. 이 파일을 만들고 선택 항목을 지정하는 프로세스를 완료하는 데 30분 이상 걸릴 수 있습니다. 이 작업을 한 번 완료한 후 모든 인스턴스에 구성 파일을 재사용할 수 있습니다.

단계

- 단계 1. CloudWatch 로그 사용 (p. 13)
- 단계 2. 설정 구성 CloudWatch (p. 14)
- 단계 3. 전송할 데이터 구성 (p. 14)
- 단계 4. 흐름 제어 구성 (p. 18)
- 단계 5. JSON 콘텐츠 저장 (p. 19)

단계 1. CloudWatch 로그 사용

JSON 파일의 상단에서 `IsEnabled`를 "false"에서 "true"로 변경합니다.

```
"IsEnabled": true,
```

단계 2. 설정 구성 CloudWatch

자격 증명, 리전, 로그 그룹 이름, 로그 스트림 네임스페이스를 지정합니다. 그러면 인스턴스가 CloudWatch Logs로 로그 데이터를 보낼 수 있습니다. 동일한 로그 데이터를 여러 위치로 보내려는 경우, ID가 고유하고 (예: "CloudWatchLogs2" 및 "CloudWatchLogs3") ID별로 리전이 다른 섹션을 추가할 수 있습니다.

CloudWatch Logs로 로그 데이터를 보내기 위해 설정을 구성하려면

1. JSON 파일에서 `CloudWatchLogs` 섹션을 찾습니다.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. `AccessKey` 및 `SecretKey` 필드는 비워둡니다. IAM 역할을 사용하여 자격 증명을 구성합니다.
3. `Region`에 로그 데이터를 보내려는 리전을 입력합니다(예: `us-east-2`).
4. `LogGroup`에 로그 그룹의 이름을 입력합니다. 이 이름은 CloudWatch 콘솔의 로그 그룹 화면에 표시됩니다.
5. `LogStream`에 대상 로그 스트림을 입력합니다. 이 이름은 CloudWatch 콘솔의 로그 그룹 > 스트림 화면에 표시됩니다.

   `{instance_id}`를 사용하는 경우 기본적으로 로그 스트림 이름은 이 인스턴스의 인스턴스 ID입니다.

   미리 존재하지 않는 로그 스트림 이름을 지정하면 CloudWatch Logs에서 이 이름을 자동으로 생성합니다. 리터럴 문자열, 미리 정의된 변수 `{instance_id}`, `{hostname}` 및 `{ip_address}`, 또는 이들의 조합을 사용하여 로그 스트림 이름을 정의할 수 있습니다.

단계 3. 전송할 데이터 구성

이벤트 데이터, ETW(Windows용 이벤트 추적) 데이터 및 기타 로그 데이트를 CloudWatch Logs로 전송할 수 있습니다.

CloudWatch Logs로 Windows 애플리케이션 이벤트 로그 데이터를 보내려면

1. JSON 파일에서 `ApplicationEventLog` 섹션을 찾습니다.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **1** - 오류 메시지만 업로드됩니다.
   - **2** - 경고 메시지만 업로드됩니다.

- **4** - 정보 메시지만 업로드됩니다.

값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **3**을 지정하면 오류 메시지(**1**)와 경고 메시지(**2**)가 업로드됩니다. 값 **7**을 지정하면 오류 메시지(**1**), 경고 메시지(**2**) 및 정보 메시지(**4**)가 업로드됩니다.

CloudWatch Logs로 보안 로그 데이터를 보내려면

1. JSON 파일에서 `SecurityEventLog` 섹션을 찾습니다.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. 모든 메시지를 업로드하려면 `Levels`에 **7**을 입력합니다.

CloudWatch Logs로 시스템 이벤트 로그 데이터를 보내려면

1. JSON 파일에서 `SystemEventLog` 섹션을 찾습니다.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **1** - 오류 메시지만 업로드됩니다.
   - **2** - 경고 메시지만 업로드됩니다.
   - **4** - 정보 메시지만 업로드됩니다.

   값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **3**을 지정하면 오류 메시지(**1**)와 경고 메시지(**2**)가 업로드됩니다. 값 **7**을 지정하면 오류 메시지(**1**), 경고 메시지(**2**) 및 정보 메시지(**4**)가 업로드됩니다.

CloudWatch Logs로 다른 유형의 이벤트 로그 데이터를 보내려면

1. JSON 파일에서 새 섹션을 추가합니다. 각 섹션에는 고유한 `Id`가 있어야 합니다.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
```

```
        "Levels": "7"
    }
},
```

2. `Id`에 업로드할 로그의 이름을 입력합니다(예: **`WindowsBackup`**).
3. `LogName`에 업로드할 로그의 이름을 입력합니다. 로그 이름은 다음과 같이 확인할 수 있습니다.

   a. 이벤트 뷰어를 엽니다.

   b. 탐색 창에서 Applications and Services Logs(응용 프로그램 및 서비스 로그)를 선택합니다.

   c. 로그로 이동한 다음 작업, 속성을 선택합니다.
4. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **`1`** - 오류 메시지만 업로드됩니다.
   - **`2`** - 경고 메시지만 업로드됩니다.
   - **`4`** - 정보 메시지만 업로드됩니다.

   값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **`3`**을 지정하면 오류 메시지(**`1`**)와 경고 메시지(**`2`**)가 업로드됩니다. 값 **`7`**을 지정하면 오류 메시지(**`1`**), 경고 메시지(**`2`**) 및 정보 메시지(**`4`**)가 업로드됩니다.

CloudWatch Logs로 Windows용 이벤트 추적 데이터를 보내는 방법

ETW(Windows용 이벤트 추적)는 애플리케이션이 로그를 기록할 수 있는 효율적이고 세부적인 로깅 메커니즘을 제공합니다. 로깅 세션을 시작하고 중지할 수 있는 세션 관리자가 각 ETW를 제어합니다. 각 세션에는 한 공급자와 하나 또는 그 이상의 소비자가 있습니다.

1. JSON 파일에서 `ETW` 섹션을 찾습니다.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. `LogName`에 업로드할 로그의 이름을 입력합니다.
3. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **`1`** - 오류 메시지만 업로드됩니다.
   - **`2`** - 경고 메시지만 업로드됩니다.
   - **`4`** - 정보 메시지만 업로드됩니다.

   값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **`3`**을 지정하면 오류 메시지(**`1`**)와 경고 메시지(**`2`**)가 업로드됩니다. 값 **`7`**을 지정하면 오류 메시지(**`1`**), 경고 메시지(**`2`**) 및 정보 메시지(**`4`**)가 업로드됩니다.

CloudWatch Logs로 사용자 지정 로그(텍스트 기반 로그 파일)를 보내려면

1. JSON 파일에서 `CustomLogs` 섹션을 찾습니다.

```
{
```

```
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. `LogDirectoryPath`에서 인스턴스에 로그가 저장되어 있는 경로를 입력합니다.
3. `TimestampFormat`에 사용할 타임스탬프 형식을 입력합니다. 지원되는 값에 대한 자세한 내용은 MSDN의 사용자 지정 날짜 및 시간 형식 문자열 주제를 참조하십시오.

   Important

   원본 로그 파일에는 각 로그 줄의 시작 부분에 타임스탬프가 있어야 하고 타임스탬프 뒤에는 공백이 있어야 합니다.
4. `Encoding`에 사용할 파일 인코딩을 입력합니다(예: UTF-8). 지원되는 값 목록은 MSDN에서 Encoding Class 항목을 참조하십시오.

   Note

   표시 이름이 아니라 인코딩 이름을 사용하십시오.
5. (선택 사항) `Filter`에 로그 이름의 접두사를 입력합니다. 모든 파일을 모니터링하려면 이 파라미터를 공백으로 둡니다. 지원되는 값에 대한 자세한 내용은 MSDN의 FileSystemWatcherFilter 속성 주제를 참조하십시오.
6. (선택 사항) `CultureName`에 타임스탬프가 기록되는 로캘을 입력합니다. `CultureName`이 공백이면 기본적으로 Windows 인스턴스에서 현재 사용 중인 것과 같은 로캘로 설정됩니다. 자세한 내용은 MSDN의 제품 동작 주제에 있는 표의 `Language tag` 열을 참조하십시오.

   Note

   The `div`, `div-MV`, `hu`, 그리고 `hu-HU` 값은 지원되지 않습니다.
7. (선택 사항) `TimeZoneKind`, 유형 `Local` 또는 `UTC`. 시간대 정보가 로그의 시간 스탬프에 포함되지 않을 때 시간대 정보를 제공하도록 설정할 수 있습니다. 이 파라미터가 공백으로 남겨져 있고 타임스탬프에 표준 시간대 정보가 포함되어 있지 않으면, CloudWatch Logs가 기본적으로 현지 표준 시간대로 설정됩니다. 타임스탬프에 표준 시간대 정보가 이미 포함되어 있는 경우 이 파라미터는 무시됩니다.
8. (선택 사항) `LineCount`에 로그 파일을 식별할 헤더의 줄 수를 입력합니다. 예를 들어 IIS 로그 파일에 있는 헤더들은 사실상 동일합니다. `5`를 입력하면 로그 파일 헤더에서 처음 나오는 세 줄을 읽어 식별하는 식입니다. IIS 로그 파일에서 처음 나오는 세 줄은 날짜와 타임스탬프이지만, 로그 파일 간에 타임스탬프가 반드시 다르지는 않습니다. 이러한 이유로, 로그 파일에 고유한 지문을 남기기 위해 실제 로그 데이터를 한 줄 이상 포함하는 것이 좋습니다.

CloudWatch Logs로 IIS 로그 데이터를 보내려면

1. JSON 파일에서 `IISLog` 섹션을 찾습니다.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
```

```
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. `LogDirectoryPath`에 개별 사이트에 대해 IIS 로그를 저장할 폴더를 입력합니다(예: `C:\inetpub\logs\LogFiles\W3SVC`*n*).

   Note

   W3C 로그 형식만 지원됩니다. IIS, NCSA 및 사용자 지정 형식은 지원되지 않습니다.

3. `TimestampFormat`에 사용할 타임스탬프 형식을 입력합니다. 지원되는 값에 대한 자세한 내용은 MSDN의 사용자 지정 날짜 및 시간 형식 문자열 주제를 참조하십시오.
4. `Encoding`에 사용할 파일 인코딩을 입력합니다(예: UTF-8). 지원되는 값에 대한 자세한 내용은 MSDN의 인코딩 클래스 주제를 참조하십시오.

   Note

   표시 이름이 아니라 인코딩 이름을 사용하십시오.

5. (선택 사항) `Filter`에 로그 이름의 접두사를 입력합니다. 모든 파일을 모니터링하려면 이 파라미터를 공백으로 둡니다. 지원되는 값에 대한 자세한 내용은 MSDN의 FileSystemWatcherFilter 속성 주제를 참조하십시오.
6. (선택 사항) `CultureName`에 타임스탬프가 기록되는 로캘을 입력합니다. `CultureName`이 공백이면 기본적으로 Windows 인스턴스에서 현재 사용 중인 것과 같은 로캘로 설정됩니다. 지원되는 값에 대한 자세한 내용은 MSDN의 제품 동작 주제에 있는 표의 `Language tag` 열을 참조하십시오.

   Note

   The `div`, `div-MV`, `hu`, 그리고 `hu-HU` 값은 지원되지 않습니다.

7. (선택 사항) `TimeZoneKind`, 입력 `Local` 또는 `UTC`. 시간대 정보가 로그의 시간 스탬프에 포함되지 않을 때 시간대 정보를 제공하도록 설정할 수 있습니다. 이 파라미터가 공백으로 남겨져 있고 타임스탬프에 표준 시간대 정보가 포함되어 있지 않으면, CloudWatch Logs가 기본적으로 현지 표준 시간대로 설정됩니다. 타임스탬프에 표준 시간대 정보가 이미 포함되어 있는 경우 이 파라미터는 무시됩니다.
8. (선택 사항) `LineCount`에 로그 파일을 식별할 헤더의 줄 수를 입력합니다. 예를 들어 IIS 로그 파일에 있는 헤더들은 사실상 동일합니다. **5**를 입력하면 로그 파일 헤더에서 처음 나오는 다섯 줄을 읽어 식별하는 식입니다. IIS 로그 파일에서 처음 나오는 세 줄은 날짜와 타임스탬프이지만, 로그 파일 간에 타임스탬프가 반드시 다르지는 않습니다. 이러한 이유로, 로그 파일에 고유한 지문을 남기기 위해 실제 로그 데이터를 한 줄 이상 포함하는 것이 좋습니다.

단계 4. 흐름 제어 구성

각 데이터 형식의 `Flows` 섹션에 해당 대상이 있어야 합니다. 예를 들어 사용자 지정 로그, ETW 로그 및 시스템 로그를 CloudWatch Logs로 전송하려면 `Flows` 섹션에 `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs`를 추가합니다.

Warning

유효하지 않은 단계를 추가하면 흐름이 차단됩니다. 예를 들어, 디스크 측정치 단계를 추가하지만 인스턴스에 디스크가 없는 경우 흐름의 모든 단계가 차단됩니다.

같은 로그 파일을 두 개 이상의 대상으로 보낼 수 있습니다. 예를 들어, 애플리케이션 로그를 `CloudWatchLogs` 섹션에 정의된 두 개의 대상으로 보내려면 `Flows` 섹션에 `ApplicationEventLog, (CloudWatchLogs,CloudWatchLogs2)`를 추가합니다.

흐름 제어를 구성하는 방법

1. `AWS.EC2.Windows.CloudWatch.json` 파일에서 `Flows` 섹션을 찾습니다.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. `Flows`에서 업로드할 각 데이터 형식(예: `ApplicationEventLog`) 및 대상(예: `CloudWatchLogs`)을 추가합니다.

단계 5. JSON 콘텐츠 저장

이제 JSON 파일 편집이 끝났습니다. 이것을 저장하여 다른 창의 텍스트 편집기에 파일 내용을 붙여 넣습니다. 이 절차의 이후 단계에서 이 파일 내용이 필요할 것입니다.

시스템 관리자를 위한 IAM 사용자 및 역할 만들기

시스템 관리자 Run Command를 사용할 때 인스턴스 자격 증명에 대한 IAM 역할이 필요합니다. 이 역할은 시스템 관리자가 인스턴스에 대한 작업을 수행하도록 허용합니다. 시스템 관리자를 구성하고 실행하기 위해 고유한 IAM 사용자 계정을 만들 수도 있습니다. 자세한 내용은 다음에 대한 보안 역할 구성 시스템 관리자 in the AWS 시스템 관리자 사용 설명서. 첨부하는 방법에 대한 정보는 IAM 기존 인스턴스에 대한 역할을 참조하십시오. 연결 IAM 인스턴스에 대한 역할 in the Windows 인스턴스용 Amazon EC2 사용 설명서.

시스템 관리자 사전 조건 확인

시스템 관리자 Run Command를 사용하여 CloudWatch Logs와의 통합을 구성하기 전에, 인스턴스가 최소 요구 사항을 충족하는지 확인해야 합니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 시스템 관리자 사전 조건을 참조하십시오.

인터넷 액세스 확인

로그 및 이벤트 데이터를 CloudWatch로 보내려면 Amazon EC2 Windows Server 인스턴스와 관리형 인스턴스에 아웃바운드 인터넷 액세스 권한이 있어야 합니다. 인터넷 액세스를 구성하는 방법에 대한 자세한 내용은 다음을 참조하십시오. 인터넷 게이트웨이 in the Amazon VPC 사용 설명서.

시스템 관리자 Run Command를 사용하여 CloudWatch Logs 활성화

Run Command를 사용하면 요청 시 인스턴스의 구성을 관리할 수 있습니다. 시스템 관리자 문서, 파라미터를 지정하고 하나 이상의 인스턴스에 명령을 실행합니다. 인스턴스의 SSM 에이전트는 명령을 처리하고 지정된 대로 인스턴스를 구성합니다.

Run Command를 사용하여 CloudWatch Logs와 통합을 구성하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. https://console.aws.amazon.com/systems-manager에서 SSM 콘솔을 엽니다.
3. 탐색 창에서 명령 실행을 선택합니다.
4. Run a command를 선택합니다.
5. Command document에서 AWS-ConfigureCloudWatch를 선택합니다.

6. 대상 인스턴스에서 CloudWatch Logs와 통합할 인스턴스를 선택합니다. 이 목록에서 인스턴스가 보이지 않는 경우, Run Command에 대해 구성되지 않은 인스턴스일 수 있습니다. 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 시스템 관리자 사전 조건을 참조하십시오.
7. 상태에서 활성을 선택합니다.
8. 속성에 이전 작업에서 생성한 JSON 내용을 복사하여 붙여 넣습니다.
9. 나머지 옵션 필드를 완성하고 Run을 선택합니다.

다음 절차를 이용해 Amazon EC2 콘솔에서 명령 실행 결과를 확인합니다.

콘솔에서 명령 출력을 보는 방법

1. 명령을 선택합니다.
2. 출력 탭을 선택합니다.
3. 출력 보기를 선택합니다. 명령 출력 페이지에 명령 실행 결과가 표시됩니다.

빠른 시작 Windows Server 2012 및 Windows Server 2008을 실행하는 Amazon EC2 인스턴스가 CloudWatch Logs로 로그를 전송하도록 설정

> Tip
>
> CloudWatch는 EC2 인스턴스 및 온프레미스 서버에서 로그와 지표를 모두 수집할 수 있는 통합 에이전트가 새롭게 추가되었습니다. 새롭게 통합된 CloudWatch 에이전트의 사용을 권장합니다. 자세한 정보는 CloudWatch Logs 시작하기 (p. 5) 단원을 참조하십시오.
> 이번 단원의 나머지에서는 이전 CloudWatch Logs 에이전트의 사용에 대해서 설명하겠습니다.

Windows Server 2012 및 Windows Server 2008을 실행하는 Amazon EC2 인스턴스가 CloudWatch Logs로 로그를 전송하도록 설정

다음 단계를 사용하면 Windows Server 2012 및 Windows Server 2008을 실행하는 인스턴스가 로그를 CloudWatch Logs로 보내도록 설정할 수 있습니다.

샘플 구성 파일 다운로드

다음 샘플 JSON 파일을 컴퓨터에 다운로드합니다. AWS.EC2.Windows.cloudwatch.json. 다음 단계에서 이 파일을 편집합니다.

CloudWatch에 JSON 파일 구성

JSON 구성 파일에서 선택 항목을 지정하여 CloudWatch로 전송할 로그를 결정합니다. 이 파일을 만들고 선택 항목을 지정하는 프로세스를 완료하는 데 30분 이상 걸릴 수 있습니다. 이 작업을 한 번 완료한 후 모든 인스턴스에 구성 파일을 재사용할 수 있습니다.

단계

- 단계 1. CloudWatch 로그 사용 (p. 21)
- 단계 2. 설정 구성 CloudWatch (p. 21)
- 단계 3. 전송할 데이터 구성 (p. 21)
- 단계 4. 흐름 제어 구성 (p. 25)

단계 1. CloudWatch 로그 사용

JSON 파일의 상단에서 `IsEnabled`를 "false"에서 "true"로 변경합니다.

```
"IsEnabled": true,
```

단계 2. 설정 구성 CloudWatch

자격 증명, 리전, 로그 그룹 이름, 로그 스트림 네임스페이스를 지정합니다. 그러면 인스턴스가 CloudWatch Logs로 로그 데이터를 보낼 수 있습니다. 동일한 로그 데이터를 여러 위치로 보내려는 경우, ID가 고유하고 (예: "CloudWatchLogs2" 및 "CloudWatchLogs3") ID별로 리전이 다른 섹션을 추가할 수 있습니다.

CloudWatch Logs로 로그 데이터를 보내기 위해 설정을 구성하려면

1. JSON 파일에서 `CloudWatchLogs` 섹션을 찾습니다.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. `AccessKey` 및 `SecretKey` 필드는 비워둡니다. IAM 역할을 사용하여 자격 증명을 구성합니다.
3. `Region`에 로그 데이터를 보내려는 리전을 입력합니다(예: `us-east-2`).
4. `LogGroup`에 로그 그룹의 이름을 입력합니다. 이 이름은 CloudWatch 콘솔의 로그 그룹 화면에 표시됩니다.
5. `LogStream`에 대상 로그 스트림을 입력합니다. 이 이름은 CloudWatch 콘솔의 로그 그룹 > 스트림 화면에 표시됩니다.

   `{instance_id}`를 사용하는 경우 기본적으로 로그 스트림 이름은 이 인스턴스의 인스턴스 ID입니다.

   미리 존재하지 않는 로그 스트림 이름을 지정하면 CloudWatch Logs에서 이 이름을 자동으로 생성합니다. 리터럴 문자열, 미리 정의된 변수 `{instance_id}`, `{hostname}` 및 `{ip_address}`, 또는 이들의 조합을 사용하여 로그 스트림 이름을 정의할 수 있습니다.

단계 3. 전송할 데이터 구성

이벤트 데이터, ETW(Windows용 이벤트 추적) 데이터 및 기타 로그 데이트를 CloudWatch Logs로 전송할 수 있습니다.

CloudWatch Logs로 Windows 애플리케이션 이벤트 로그 데이터를 보내려면

1. JSON 파일에서 `ApplicationEventLog` 섹션을 찾습니다.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
```

```
    }
},
```

2. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **1** - 오류 메시지만 업로드됩니다.
   - **2** - 경고 메시지만 업로드됩니다.
   - **4** - 정보 메시지만 업로드됩니다.

   값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **3**을 지정하면 오류 메시지(**1**)와 경고 메시지(**2**)가 업로드됩니다. 값 **7**을 지정하면 오류 메시지(**1**), 경고 메시지(**2**) 및 정보 메시지(**4**)가 업로드됩니다.

CloudWatch Logs로 보안 로그 데이터를 보내려면

1. JSON 파일에서 `SecurityEventLog` 섹션을 찾습니다.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. 모든 메시지를 업로드하려면 `Levels`에 **7**을 입력합니다.

CloudWatch Logs로 시스템 이벤트 로그 데이터를 보내려면

1. JSON 파일에서 `SystemEventLog` 섹션을 찾습니다.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **1** - 오류 메시지만 업로드됩니다.
   - **2** - 경고 메시지만 업로드됩니다.
   - **4** - 정보 메시지만 업로드됩니다.

   값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **3**을 지정하면 오류 메시지(**1**)와 경고 메시지(**2**)가 업로드됩니다. 값 **7**을 지정하면 오류 메시지(**1**), 경고 메시지(**2**) 및 정보 메시지(**4**)가 업로드됩니다.

CloudWatch Logs로 다른 유형의 이벤트 로그 데이터를 보내려면

1. JSON 파일에서 새 섹션을 추가합니다. 각 섹션에는 고유한 `Id`가 있어야 합니다.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. `Id`에 업로드할 로그의 이름을 입력합니다(예: **`WindowsBackup`**).
3. `LogName`에 업로드할 로그의 이름을 입력합니다. 로그 이름은 다음과 같이 확인할 수 있습니다.

   a. 이벤트 뷰어를 엽니다.

   b. 탐색 창에서 Applications and Services Logs(응용 프로그램 및 서비스 로그)를 선택합니다.

   c. 로그로 이동한 다음 작업, 속성을 선택합니다.
4. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **`1`** - 오류 메시지만 업로드됩니다.
   - **`2`** - 경고 메시지만 업로드됩니다.
   - **`4`** - 정보 메시지만 업로드됩니다.

   값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **`3`**을 지정하면 오류 메시지(**`1`**)와 경고 메시지(**`2`**)가 업로드됩니다. 값 **`7`**을 지정하면 오류 메시지(**`1`**), 경고 메시지(**`2`**) 및 정보 메시지(**`4`**)가 업로드됩니다.

CloudWatch Logs로 Windows용 이벤트 추적 데이터를 보내는 방법

ETW(Windows용 이벤트 추적)는 애플리케이션이 로그를 기록할 수 있는 효율적이고 세부적인 로깅 메커니즘을 제공합니다. 로깅 세션을 시작하고 중지할 수 있는 세션 관리자가 각 ETW를 제어합니다. 각 세션에는 한 공급자와 하나 또는 그 이상의 소비자가 있습니다.

1. JSON 파일에서 `ETW` 섹션을 찾습니다.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. `LogName`에 업로드할 로그의 이름을 입력합니다.
3. `Levels`에서 업로드할 메시지의 유형을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

   - **`1`** - 오류 메시지만 업로드됩니다.
   - **`2`** - 경고 메시지만 업로드됩니다.
   - **`4`** - 정보 메시지만 업로드됩니다.

   값을 적절히 조합하여 두 가지 이상의 메시지 유형을 포함할 수 있습니다. 예를 들어 값 **`3`**을 지정하면 오류 메시지(**`1`**)와 경고 메시지(**`2`**)가 업로드됩니다. 값 **`7`**을 지정하면 오류 메시지(**`1`**), 경고 메시지(**`2`**) 및 정보 메시지(**`4`**)가 업로드됩니다.

CloudWatch Logs로 사용자 지정 로그(텍스트 기반 로그 파일)를 보내려면

1. JSON 파일에서 `CustomLogs` 섹션을 찾습니다.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. `LogDirectoryPath`에서 인스턴스에 로그가 저장되어 있는 경로를 입력합니다.
3. `TimestampFormat`에 사용할 타임스탬프 형식을 입력합니다. 지원되는 값에 대한 자세한 내용은 MSDN의 사용자 지정 날짜 및 시간 형식 문자열 주제를 참조하십시오.

   Important

   원본 로그 파일에는 각 로그 줄의 시작 부분에 타임스탬프가 있어야 하고 타임스탬프 뒤에는 공백이 있어야 합니다.
4. `Encoding`에 사용할 파일 인코딩을 입력합니다(예: UTF-8). 지원되는 값에 대한 자세한 내용은 MSDN의 인코딩 클래스 주제를 참조하십시오.

   Note

   표시 이름이 아니라 인코딩 이름을 사용하십시오.
5. (선택 사항) `Filter`에 로그 이름의 접두사를 입력합니다. 모든 파일을 모니터링하려면 이 파라미터를 공백으로 둡니다. 지원되는 값에 대한 자세한 내용은 MSDN의 FileSystemWatcherFilter 속성 주제를 참조하십시오.
6. (선택 사항) `CultureName`에 타임스탬프가 기록되는 로캘을 입력합니다. `CultureName`이 공백이면 기본적으로 Windows 인스턴스에서 현재 사용 중인 것과 같은 로캘로 설정됩니다. 지원되는 값에 대한 자세한 내용은 MSDN의 제품 동작 주제에 있는 표의 `Language tag` 열을 참조하십시오.

   Note

   The `div`, `div-MV`, `hu`, 그리고 `hu-HU` 값은 지원되지 않습니다.
7. (선택 사항) `TimeZoneKind`, 유형 `Local` 또는 `UTC`. 시간대 정보가 로그의 시간 스탬프에 포함되지 않을 때 시간대 정보를 제공하도록 설정할 수 있습니다. 이 파라미터가 공백으로 남겨져 있고 타임스탬프에 표준 시간대 정보가 포함되어 있지 않으면, CloudWatch Logs가 기본적으로 현지 표준 시간대로 설정됩니다. 타임스탬프에 표준 시간대 정보가 이미 포함되어 있는 경우 이 파라미터는 무시됩니다.
8. (선택 사항) `LineCount`에 로그 파일을 식별할 헤더의 줄 수를 입력합니다. 예를 들어 IIS 로그 파일에 있는 헤더들은 사실상 동일합니다. **5**를 입력하면 로그 파일 헤더에서 처음 나오는 세 줄을 읽어 식별하는 식입니다. IIS 로그 파일에서 처음 나오는 세 줄은 날짜와 타임스탬프이지만, 로그 파일 간에 타임스탬프가 반드시 다르지는 않습니다. 이러한 이유로, 로그 파일에 고유한 지문을 남기기 위해 실제 로그 데이터를 한 줄 이상 포함하는 것이 좋습니다.

CloudWatch Logs로 IIS 로그 데이터를 보내려면

1. JSON 파일에서 `IISLog` 섹션을 찾습니다.

```
{
```

```
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. `LogDirectoryPath`에 개별 사이트에 대해 IIS 로그를 저장할 폴더를 입력합니다(예: `C:\inetpub\logs\LogFiles\W3SVC`*n*).

   Note

   W3C 로그 형식만 지원됩니다. IIS, NCSA 및 사용자 지정 형식은 지원되지 않습니다.

3. `TimestampFormat`에 사용할 타임스탬프 형식을 입력합니다. 지원되는 값에 대한 자세한 내용은 MSDN의 사용자 지정 날짜 및 시간 형식 문자열 주제를 참조하십시오.
4. `Encoding`에 사용할 파일 인코딩을 입력합니다(예: UTF-8). 지원되는 값에 대한 자세한 내용은 MSDN의 인코딩 클래스 주제를 참조하십시오.

   Note

   표시 이름이 아니라 인코딩 이름을 사용하십시오.

5. (선택 사항) `Filter`에 로그 이름의 접두사를 입력합니다. 모든 파일을 모니터링하려면 이 파라미터를 공백으로 둡니다. 지원되는 값에 대한 자세한 내용은 MSDN의 FileSystemWatcherFilter 속성 주제를 참조하십시오.
6. (선택 사항) `CultureName`에 타임스탬프가 기록되는 로캘을 입력합니다. `CultureName`이 공백이면 기본적으로 Windows 인스턴스에서 현재 사용 중인 것과 같은 로캘로 설정됩니다. 지원되는 값에 대한 자세한 내용은 MSDN의 제품 동작 주제에 있는 표의 `Language tag` 열을 참조하십시오.

   Note

   The `div`, `div-MV`, `hu`, 그리고 `hu-HU` 값은 지원되지 않습니다.

7. (선택 사항) `TimeZoneKind`, 입력 `Local` 또는 `UTC`. 시간대 정보가 로그의 시간 스탬프에 포함되지 않을 때 시간대 정보를 제공하도록 설정할 수 있습니다. 이 파라미터가 공백으로 남겨져 있고 타임스탬프에 표준 시간대 정보가 포함되어 있지 않으면, CloudWatch Logs가 기본적으로 현지 표준 시간대로 설정됩니다. 타임스탬프에 표준 시간대 정보가 이미 포함되어 있는 경우 이 파라미터는 무시됩니다.
8. (선택 사항) `LineCount`에 로그 파일을 식별할 헤더의 줄 수를 입력합니다. 예를 들어 IIS 로그 파일에 있는 헤더들은 사실상 동일합니다. **5**를 입력하면 로그 파일 헤더에서 처음 나오는 다섯 줄을 읽어 식별하는 식입니다. IIS 로그 파일에서 처음 나오는 세 줄은 날짜와 타임스탬프이지만, 로그 파일 간에 타임스탬프가 반드시 다르지는 않습니다. 이러한 이유로, 로그 파일에 고유한 지문을 남기기 위해 실제 로그 데이터를 한 줄 이상 포함하는 것이 좋습니다.

단계 4. 흐름 제어 구성

각 데이터 형식의 `Flows` 섹션에 해당 대상이 있어야 합니다. 예를 들어 사용자 지정 로그, ETW 로그 및 시스템 로그를 CloudWatch Logs로 전송하려면 `Flows` 섹션에 `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs`를 추가합니다.

Warning

유효하지 않은 단계를 추가하면 흐름이 차단됩니다. 예를 들어, 디스크 측정치 단계를 추가하지만 인스턴스에 디스크가 없는 경우 흐름의 모든 단계가 차단됩니다.

같은 로그 파일을 두 개 이상의 대상으로 보낼 수 있습니다. 예를 들어, 애플리케이션 로그를 `CloudWatchLogs` 섹션에 정의된 두 개의 대상으로 보내려면 `Flows` 섹션에 `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)`를 추가합니다.

흐름 제어를 구성하는 방법

1. `AWS.EC2.Windows.CloudWatch.json` 파일에서 `Flows` 섹션을 찾습니다.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. `Flows`에서 업로드할 각 데이터 형식(예: `ApplicationEventLog`) 및 대상(예: `CloudWatchLogs`)을 추가합니다.

이제 JSON 파일 편집이 끝났습니다. 이후 단계에서 사용하게 됩니다.

에이전트 시작

활성화하려면 Amazon EC2 Windows Server 2012 또는 Windows Server 2008을 실행하는 인스턴스 CloudWatch Logs, EC2Config 서비스를 사용합니다(`EC2Config.exe`). 인스턴스는 EC2Config 4.0 이상이어야 하며 이 절차를 사용할 수 있습니다. 이전 버전의 EC2Config를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오. EC2Config 3.x 또는 그 이전 버전을 사용하여 구성 CloudWatch in the Windows 인스턴스용 Amazon EC2 사용 설명서

EC2Config 4.x를 사용하여 CloudWatch를 구성하려면

1. 이 절차에서 앞서 편집한 AWS.EC2.Windows.CloudWatch.json 파일의 인코딩을 확인합니다. BOM 없는 UTF-8 인코딩만 지원됩니다. Windows Server 2008 - 2012 R2 인스턴스에서 다음 폴더에 파일을 저장합니다. . `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.
2. Windows Services 제어판을 사용하거나 PowerShell에서 다음 명령을 사용하여 SSM 에이전트(`AmazonSSMAgent.exe`)를 시작하거나 다시 시작합니다.

```
PS C:\> Restart-Service AmazonSSMAgent
```

SSM 에이전트가 다시 시작한 후 구성 파일을 감지하고 CloudWatch 통합에 맞게 인스턴스를 구성합니다. 로컬 구성 파일에서 파라미터와 설정을 변경할 경우 SSM 에이전트를 다시 시작하여 변경 사항을 선택해야 합니다. 인스턴스에서 CloudWatch 통합을 비활성화하려면 `IsEnabled`를 `false`로 변경하고 구성 파일에 변경 사항을 저장합니다.

빠른 시작 설치 CloudWatch Logs 에이전트 사용 AWS OpsWorks 셰프

타사 시스템 및 클라우드 인프라 자동화 도구인 AWS OpsWorks 및 Chef를 사용하여 CloudWatch Logs 에이전트를 설치하고 로그 스트림을 생성할 수 있습니다. Chef는 컴퓨터에 소프트웨어를 설치 및 구성하기 위해 작성하는 "레시피"와 구성 및 정책 배포 작업을 수행하기 위한 레시피 모음인 "쿡북"을 사용합니다. 자세한 내용은 Chef를 참조하십시오.

아래의 Chef 레시피 예제는 각 EC2 인스턴스에서 하나의 로그 파일을 모니터링하는 방법을 보여줍니다. 레시피는 로그 그룹으로 스택 이름을 사용하고 로그 스트림 이름으로 인스턴스의 호스트 이름을 사용합니다.

여러 개의 로그 파일을 모니터링하려면 여러 로그 그룹 및 로그 스트림을 생성하도록 레시피를 확장해야 합니다.

단계 1. 사용자 지정 레시피 만들기

레시피를 저장할 리포지토리를 생성합니다. AWS OpsWorks는 Git 및 Subversion을 지원합니다. 아니면 Amazon S3에 아카이브를 저장할 수 있습니다. 쿡북 저장소의 구조는 다음과 같습니다. 쿡북 리포지토리 in the AWS OpsWorks User Guide. 아래의 예는 쿡북이 이름이라고 가정합니다. `logs`. install.rb recipe는 CloudWatch Logs 에이전트. 또한 쿡북 예제(CloudWatchLogs-Cookbooks.zip)를 다운로드할 수도 있습니다.

다음 코드가 포함된 metadata.rb라는 파일을 생성합니다.

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

CloudWatch Logs 구성 파일을 생성합니다.

```
#config.rb

template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

CloudWatch Logs 에이전트 다운로드 및 설치:

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end

remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py"
  mode "0755"
end

  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> 위의 예에서, 교체 *region* 다음 중 하나를 가지고 있습니다. us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1.

에이전트 설치가 실패할 경우 `python-dev` 패키지가 설치되어 있는지 확인하십시오. 설치되어 있지 않은 경우 다음 명령을 사용한 후 에이전트 설치를 다시 시도하십시오.

```
sudo apt-get -y install python-dev
```

이 레시피는 기록할 파일 같이 다양한 속성을 지정하기 위해 수정이 가능한 cwlogs.cfg.erb 템플릿 파일을 사용합니다. 이들 속성에 대한 자세한 내용은 CloudWatch Logs 에이전트 참조 (p. 135) 단원을 참조하십시오.

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state


## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
#
#log_stream_name = {instance_id}
#
## Format specifier for timestamp parsing.
#
#datetime_format = %b %d %H:%M:%S
#
#

[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

템플릿은 스택 구성 및 배포 JSON에서 해당되는 속성을 참조하여 스택 이름과 호스트 이름을 얻습니다. 기록할 파일을 지정하는 속성은 cwlogs 쿡북의 default.rb 속성 파일(logs/attributes/default.rb)에 정의되어 있습니다.

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

단계 2. 생성 AWS OpsWorks 스택

1. https://console.aws.amazon.com/opsworks/에서 AWS OpsWorks 콘솔을 엽니다.
2. OpsWorks 대시보드에서 스택 추가를 선택하여 AWS OpsWorks 스택을 생성합니다.
3. Add stack(스택 추가) 화면에서 Chef 11 stack(Chef 11 스택)을 선택합니다.
4. 스택 이름에 이름을 입력합니다.
5. Use custom Chef Cookbooks(사용자 지정 Chef 쿡북 사용)에서 예를 선택합니다.
6. Repository type(리포지토리 유형)에서 사용할 리포지토리 유형을 선택합니다. 위의 예제를 사용하는 경우에는 Http Archive(Http 아카이브)를 선택합니다.
7. 리포지토리 URL에 이전 단계에서 생성한 쿡북이 저장된 리포지토리를 입력합니다. 위의 예제를 사용하는 경우에는 **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`**를 입력합니다.
8. Add stack(스택 추가)을 선택해서 스택을 생성합니다.

단계 3. 확장 IAM 역할

AWS OpsWorks 인스턴스에서 CloudWatch Logs을 사용하려면 인스턴스에서 사용되는 IAM 역할을 확대해야 합니다.

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Policies를 선택한 후 Create Policy를 선택합니다.
3. 정책 생성 페이지의 Create Your Own Policy(자체 정책 생성)에서 선택을 선택합니다. 사용자 지정 정책 생성에 대한 자세한 내용은 다음을 참조하십시오. Amazon EC2에 대한 IAM 정책 in the Linux 인스턴스용 Amazon EC2 사용 설명서.
4. 정책 검토 페이지의 정책 이름에 정책 이름을 입력합니다.
5. 정책 문서에 다음 정책을 붙여 넣습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

6. 정책 생성을 선택합니다.
7. 탐색 창에서 역할을 선택한 다음, 콘텐츠 창의 역할 이름에서 AWS OpsWorks 스택에서 사용되는 인스턴스 역할의 이름을 선택합니다. 스택 설정에서 스택에서 사용되는 역할을 찾을 수 있습니다(기본값 `aws-opsworks-ec2-role`).

   Note

   확인란이 아니라 사용자 이름을 선택합니다.
8. 권한 탭의 Managed Policies(관리형 정책)에서 정책 연결을 선택합니다.
9. 정책 연결 창의 테이블 헤더(필터 및 검색 옆에 있음)에서 정책 유형과 Customer Managed Policies(고객 관리형 정책)를 선택합니다.
10. 고객 관리형 정책에서 위에서 생성한 IAM 정책을 선택하고 정책 연결을 선택합니다.

    자세한 정보는 IAM 사용자 및 정책 IAM 사용자 및 그룹 and IAM 정책 관리 in the IAM 사용 설명서.

단계 4. 레이어 추가

1. https://console.aws.amazon.com/opsworks/에서 AWS OpsWorks 콘솔을 엽니다.
2. 탐색 창에서 계층을 선택합니다.
3. 콘텐츠 창에서 계층을 선택하고 Add layer(계층 추가)를 선택합니다.
4. OpsWorks 탭의 Layer type(계층 유형)에서 사용자 지정을 선택합니다.
5. 이름 및 Short name(짧은 이름) 필드에서 계층의 긴 이름과 짧은 이름을 입력한 다음 Add layer(계층 추가)를 선택합니다.

6. 레시피 탭의 Custom Chef Recipes(사용자 지정 Chef 레시피)에는 Setup, Configure, Deploy, Undeploy, Shutdown 등 AWS OpsWorks 수명 주기 이벤트에 해당하는 여러 개의 제목이 있습니다. AWS OpsWorks는 연관된 레시피를 실행하는 인스턴스 수명 주기의 중요 지점에서 이러한 이벤트들을 트리거합니다.

   Note

   위의 제목들이 보이지 않으면 Custom Chef Recipes(사용자 지정 Chef 레시피)로 가서 편집을 선택합니다.
7. 입력 로그::config, logs::install 다음 날짜 설정, 선택 + 목록에 추가한 다음 저장.

   AWS OpsWorks는 인스턴스 부팅이 끝나면 즉시 이 계층의 새 인스턴스 각각에서 이 레시피를 실행합니다.

단계 5. 인스턴스 추가

이 계층은 인스턴스를 구성하는 방법을 제어만 합니다. 해당 계층에 몇몇 인스턴스를 추가하고 이를 시작해야 합니다.

1. https://console.aws.amazon.com/opsworks/에서 AWS OpsWorks 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택한 다음 계층으로 가서 + 인스턴스를 선택합니다.
3. 기본 설정을 수락하고 Add Instance(인스턴스 추가)를 선택해서 해당 계층에 인스턴스를 추가합니다.
4. 해당 행의 작업 열에서 시작을 클릭해서 인스턴스를 시작합니다.

   AWS OpsWorks는 새 EC2 인스턴스를 시작하고 CloudWatch Logs을 구성합니다. 준비가 되면 인스턴스가 온라인 상태로 바뀝니다.

단계 6. 로그 보기

에이전트가 몇 분 동안 실행된 이후에 CloudWatch 콘솔에 새로 생성된 로그 그룹 및 로그 스트림이 나타납니다.

자세한 정보는 CloudWatch Logs에 전송된 로그 데이터 보기 (p. 54) 단원을 참조하십시오.

CloudWatch Logs 에이전트 상태 보고

다음 절차를 사용하여 EC2 인스턴스에 CloudWatch Logs 에이전트의 상태를 보고합니다.

에이전트 상태를 보고하려면

1. EC2 인스턴스에 연결합니다: 자세한 내용은 인스턴스에 연결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.

   연결 문제에 대한 자세한 내용은 다음을 참조하십시오. 인스턴스에 연결하는 문제 해결 in the Linux 인스턴스용 Amazon EC2 사용 설명서
2. 명령 프롬프트에서 다음 명령을 입력합니다:

   ```
   sudo service awslogs status
   ```

   Amazon Linux 2를 실행 중인 경우 다음 명령을 입력합니다.

   ```
   sudo service awslogsd status
   ```
3. CloudWatch Logs 에이전트에 오류, 경고 또는 문제가 없는지 /var/log/awslogs.log 파일을 확인합니다.

CloudWatch Logs 에이전트 시작

설치 이후에 EC2 인스턴스의 CloudWatch Logs 에이전트가 자동으로 시작되지 않았거나 에이전트가 중단된 경우에는 아래 절차를 사용하여 에이전트를 시작할 수 있습니다.

에이전트를 시작하려면

1. EC2 인스턴스에 연결합니다: 자세한 내용은 인스턴스에 연결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.

   연결 문제에 대한 자세한 내용은 다음을 참조하십시오. 인스턴스에 연결하는 문제 해결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.
2. 명령 프롬프트에서 다음 명령을 입력합니다:

   ```
   sudo service awslogs start
   ```

   Amazon Linux 2를 실행 중인 경우 다음 명령을 입력합니다.

   ```
   sudo service awslogsd start
   ```

CloudWatch Logs 에이전트 중지

다음 절차를 사용하여 EC2 인스턴스에서 CloudWatch Logs 에이전트를 중지합니다.

에이전트를 중지하려면

1. EC2 인스턴스에 연결합니다: 자세한 내용은 인스턴스에 연결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.

   연결 문제에 대한 자세한 내용은 다음을 참조하십시오. 인스턴스에 연결하는 문제 해결 in the Linux 인스턴스용 Amazon EC2 사용 설명서.
2. 명령 프롬프트에서 다음 명령을 입력합니다:

   ```
   sudo service awslogs stop
   ```

   Amazon Linux 2를 실행 중인 경우 다음 명령을 입력합니다.

   ```
   sudo service awslogsd stop
   ```

빠른 시작 사용 AWS CloudFormation 시작하기 CloudWatch Logs

AWS CloudFormation은 AWS 리소스를 JSON 형식으로 설명 및 프로비저닝할 수 있게 해줍니다. 이 방법은 AWS 리소스 모음을 하나의 단위로 관리할 수 있고 리전에서 AWS 리소스를 손쉽게 복제할 수 있다는 장점이 있습니다.

AWS CloudFormation을 사용하여 AWS를 프로비저닝할 때 사용할 AWS 리소스를 설명하는 템플릿을 생성합니다. 다음은 로그 그룹과 404 출현 횟수를 계산해서 로그 그룹에 전송하는 지표 필터를 생성하는 템플릿 코드 조각의 예제입니다.

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code = 404,
 size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    }
}
```

다음은 기본적인 예제입니다. AWS CloudFormation을 사용하여 훨씬 풍부하게 CloudWatch Logs 배포를 설정할 수 있습니다. 템플릿 예제에 대한 자세한 내용은 Amazon CloudWatch Logs 템플릿 스니펫 in the AWS CloudFormation 사용 설명서. 시작하는 방법에 대한 자세한 내용은 다음을 참조하십시오. 시작하기 AWS CloudFormation in the AWS CloudFormation 사용 설명서.

CloudWatch Logs Insights로 로그 데이터 분석

CloudWatch Logs Insights를 사용하면 Amazon CloudWatch Logs에서 로그 데이터를 대화식으로 검색해 분석할 수 있습니다. 운영상의 문제에 보다 효율적이고 효과적으로 대처할 수 있도록 쿼리를 수행할 수 있습니다. 문제가 발생하면 CloudWatch Logs Insights를 사용해 잠재적인 원인을 식별하고 배포된 수정 사항을 확인할 수 있습니다.

CloudWatch Logs Insights에는 간단하지만 강력한 몇 가지 명령을 포함한 특수 쿼리 언어가 포함되어 있고, CloudWatch Logs Insights는 샘플 쿼리, 명령 설명, 쿼리 자동 작성 및 로그 필드 검색 기능을 제공해 손쉽게 시작할 수 있도록 지원합니다. 여러 가지 유형의 AWS 서비스 로그에 대한 샘플 쿼리가 포함되어 있습니다.

CloudWatch Logs Insights는 AWS 서비스(예: Amazon Route 53, AWS Lambda, AWS CloudTrail 및 Amazon VPC)의 로그와 로그 이벤트를 JSON으로 출력하는 모든 애플리케이션 또는 사용자 지정 로그에서 필드를 자동으로 검색합니다.

CloudWatch Logs Insights를 사용하면 2018년 11월 5일 이후 CloudWatch Logs로 전송된 로그 데이터를 검색할 수 있습니다.

단일 요청은 최대 20개의 로그 그룹을 쿼리할 수 있습니다. 쿼리가 완료되지 않은 경우 15분 후에 쿼리가 시간 초과됩니다. 쿼리 결과는 7일 동안 사용할 수 있습니다.

생성한 쿼리를 저장할 수 있습니다. 그러면 복잡한 쿼리를 실행해야 할 때마다 다시 만들지 않고도 실행할 수 있습니다.

> Important
>
> 네트워크 보안 팀에서 웹 소켓 사용을 허용하지 않는 경우 현재 CloudWatch 콘솔의 CloudWatch Logs Insights 부분에 액세스할 수 없습니다. 다음을 사용할 수 있습니다. CloudWatch Logs Insights 쿼리 기능 사용 APIs. 자세한 내용은 을 참조하십시오. StartQuery 에서 Amazon CloudWatch Logs API Reference.

내용

- 지원되는 로그 및 검색되는 필드 (p. 33)
- 튜토리얼: 샘플 쿼리 실행 및 수정 (p. 35)
- 튜토리얼: 집계 함수를 사용하여 쿼리 실행 (p. 37)
- 튜토리얼: 로그 필드별로 그룹화된 시각화를 생성하는 쿼리 실행 (p. 38)
- 튜토리얼: 시계열 시각화를 생성하는 쿼리 실행 (p. 38)
- CloudWatch Logs Insights 쿼리 구문 (p. 39)
- 그래프로 로그 데이터 시각화 (p. 48)
- CloudWatch Logs Insights 쿼리 저장 및 재실행 (p. 49)
- 샘플 쿼리 (p. 50)
- 대시보드에 쿼리 추가 또는 쿼리 결과 내보내기 (p. 53)
- 실행 중인 쿼리 또는 쿼리 기록 보기 (p. 53)

지원되는 로그 및 검색되는 필드

CloudWatch Logs Insights는 모든 유형의 로그를 지원합니다. CloudWatch Logs로 전송되는 모든 로그의 경우 다음 5가지 시스템 필드가 자동으로 생성됩니다.

- `@message`에는 구문 분석되지 않은 원시 로그 이벤트가 포함되어 있습니다. 이는 `message` 필드 InputLogevent.
- `@timestamp`에는 로그 이벤트 `timestamp` 필드에 포함된 이벤트 타임스탬프가 포함되어 있습니다. 이는 `timestamp` 필드 InputLogevent.
- `@ingestionTime`에는 로그 이벤트가 CloudWatch Logs에서 수신된 시간이 포함되어 있습니다.
- `@logStream`에는 로그 이벤트가 추가된 로그 스트림의 이름이 포함되어 있습니다. 로그 스트림은 로그를 생성한 프로세스와 동일한 프로세스를 통해 로그를 그룹화하는 데 사용됩니다.
- `@log` 은(는) 다음 형식의 로그 그룹 식별자입니다. *`account-id`*`:`*`log-group-name`*. 이는 여러 로그 그룹의 쿼리에 유용하며, 특정 이벤트가 에 속하는 로그 그룹을 식별하는 데 유용합니다.

CloudWatch Logs Insights는 생성하는 필드의 시작 부분에 @ 기호를 삽입합니다.

다수의 로그 유형에 대해 CloudWatch Logs 역시 로그에 포함된 로그 필드를 자동으로 검색합니다. 다음 표에는 이러한 자동 검색 필드가 나와 있습니다.

CloudWatch Logs Insights가 자동으로 검색하지 않는 필드가 포함된 다른 로그 유형에 대해서는 **`parse`** 명령을 사용해 쿼리에 사용할 임시 필드를 추출해 생성할 수 있습니다. 자세한 내용은 CloudWatch Logs Insights 쿼리 구문 (p. 39) 단원을 참조하십시오.

검색한 로그 필드의 이름이 `@` 문자로 시작하면 CloudWatch Logs Insights는 해당 필드의 앞에 `@`를 추가로 붙여 표시합니다. 예를 들어, 로그 필드 이름이 `@example.com`이면 이 필드 이름은 `@@example.com`으로 표시됩니다.

로그 유형	검색된 로그 필드
Amazon VPC 흐름 로그	`@timestamp`, `@logStream`, `@message`, `accountId`, `endTime`, `interfaceId`, `logStatus`, `startTime`, `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`
Route 53 로그	`@timestamp`, `@logStream`, `@message`, `edgeLocation`, `hostZoneId`, `protocol`, `queryName`, `queryTimestamp`, `queryType`, `resolverIp`, `responseCode`, `version`
Lambda 로그	`@timestamp`, `@logStream`, `@message`, `@requestId`, `@duration`, `@billedDuration`, `@type`, `@maxMemoryUsed`, `@memorySize` 만약 Lambda 로그 라인에는 X-Ray 추적 ID에는 다음 필드도 포함됩니다. `@xrayTraceId` 및 `@xraySegmentId`. CloudWatch Logs Insights는 Lambda 로그에서 로그 필드를 자동으로 검색하지만 각 로그 이벤트에 포함된 첫 번째 JSON 조각에 대해서만 검색합니다. Lambda 로그 이벤트에 JSON 조각이 여러 개 포함된 경우 **`parse`** 명령을 사용하여 로그 필드를 구문 분석하고 추출할 수 있습니다. 자세한 정보는 JSON 로그의 필드 (p. 34) 단원을 참조하십시오.
CloudTrail 로그 JSON 형식의 로그	자세한 정보는 JSON 로그의 필드 (p. 34) 단원을 참조하십시오.
기타 로그 유형	`@timestamp`, `@ingestionTime`, `@logStream`, `@message`, `@log`.

JSON 로그의 필드

CloudWatch Logs Insights에서는 점 표기법을 사용하여 중첩된 JSON 필드를 나타냅니다. 다음 JSON 이벤트 예제에서 JSON 객체 `userIdentity`의 필드 `type`은 `userIdentity.type`으로 표시됩니다.

JSON 배열은 필드 이름 및 값 목록으로 평면화됩니다. 예를 들어, `requestParameters.instancesSet`의 첫 번째 항목에 대한 `instanceId` 값을 지정하려면 `requestParameters.instancesSet.items.0.instanceId`를 사용합니다.

CloudWatch Logs Insights는 JSON 로그에서 최대 100개의 로그 이벤트 필드를 추출할 수 있습니다. 추출되지 않은 기타 필드의 경우 parse 명령을 사용하여 메시지 필드의 구문 분석되지 않은 원시 로그 이벤트에서 이러한 필드를 구문 분석할 수 있습니다.

```
{ "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
```

튜토리얼: 샘플 쿼리 실행 및 수정

다음 자습서에서는 사용자가 CloudWatch Logs Insights를 시작할 수 있도록 도와줍니다. 샘플 쿼리를 실행한 다음 수정해 다시 실행하는 방법을 살펴봅니다.

쿼리를 실행하려면 CloudWatch Logs에 이미 저장된 로그가 있어야 합니다. CloudWatch Logs를 이미 사용 중이고 로그 그룹과 로그 스트림이 설정되어 있으면 시작할 준비가 된 것입니다. 또한 AWS CloudTrail, Amazon Route 53 또는 Amazon VPC 등과 같은 서비스를 사용하고 있고 이러한 서비스의 로그가 CloudWatch Logs로 전송되도록 설정한 경우에는 이미 로그가 있을 수도 있습니다. CloudWatch Logs에 로그를 전송하는 방법에 대한 자세한 정보는 CloudWatch Logs 시작하기 (p. 5) 단원을 참조하십시오.

CloudWatch Logs Insights의 쿼리가 로그 이벤트에서 일련의 필드를 반환하거나 로그 이벤트에 대해 수행된 수학적 집계 또는 다른 작업의 결과를 반환합니다. 이 자습서에서는 로그 이벤트 목록을 반환하는 쿼리를 보여줍니다.

샘플 쿼리 실행

샘플 쿼리를 실행하여 시작합니다.

CloudWatch Logs Insights 샘플 쿼리를 실행하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.

   페이지 상단 근처에 쿼리 편집기가 있습니다. CloudWatch Logs Insights를 처음으로 열면 이 상자에는 최신 로그 이벤트 20개를 반환하는 기본 쿼리가 포함되어 있습니다.
3. 쿼리 편집기 위에서 쿼리할 로그 그룹을 하나 이상 선택합니다. 로그 그룹을 쉽게 찾을 수 있도록 검색 창에 텍스트를 입력하여 CloudWatch Logs가 검색 창에 일치하는 로그 그룹을 표시하게 할 수 있습니다.

   로그 그룹을 선택하면 CloudWatch Logs Insights가 로그 그룹의 데이터에서 필드를 자동으로 검색합니다. 이러한 검색된 필드를 보려면 페이지 오른쪽의 필드를 선택합니다.
4. (선택 사항) 오른쪽 위에 있는 시간 선택기를 사용하여 쿼리할 기간을 선택합니다.
5. 실행을 선택합니다.

   쿼리의 결과가 표시됩니다. 이 예제에서 결과는 모든 유형의 최신 로그 이벤트 20개입니다.

   또한 CloudWatch Logs는 이 로그 그룹의 로그 이벤트를 시간의 흐름에 따라 보여주는 막대 그래프도 표시합니다. 이 막대 그래프는 단순히 테이블에 표시된 이벤트가 아니라 쿼리 및 시간 범위와 일치하는 로그 그룹 내 이벤트의 분포를 보여줍니다.
6. 반환되는 로그 이벤트 중 하나의 모든 필드를 보려면 해당 로그 이벤트 왼쪽에 있는 아이콘을 선택합니다.

샘플 쿼리 수정

이 자습서에서는 최신 로그 이벤트 50개를 표시하도록 샘플 쿼리를 수정합니다.

이전 자습서를 아직 실행하지 않은 경우 지금 실행하는 것이 좋습니다. 이 자습서에는 이전 자습서를 마친 지점에서 시작합니다.

> **Note**
>
> 일부 샘플 쿼리 제공 CloudWatch Logs Insights 사용 `head` 또는 `tail` 명령 대신 `limit`. 이러한 명령은 사용 중단 중이며 다음으로 대체되었습니다. `limit`. 사용 `limit` 대신 `head` 또는 `tail` 를 작성합니다.

CloudWatch Logs Insights 샘플 쿼리를 수정하려면

1. 쿼리 편집기에서 20을 50으로 변경한 다음 실행을 선택합니다.

   새 쿼리의 결과가 표시됩니다. 기본 시간 범위 내에서 로그 파일에 데이터가 충분하다고 가정하고 이제 로그 이벤트 50개가 나열됩니다.
2. (선택 사항) 생성한 쿼리를 저장할 수 있습니다. 이 쿼리를 저장하려면 저장을 선택합니다. 자세한 정보는 CloudWatch Logs Insights 쿼리 저장 및 재실행 (p. 49) 단원을 참조하십시오.

샘플 쿼리에 필터 명령 추가

이 자습서에서는 쿼리 편집기에서 쿼리를 보다 과감하게 변경하는 방법에 대해 살펴봅니다. 이 자습서에서는 검색된 로그 이벤트의 필드를 기반으로 이전 쿼리의 결과를 필터링합니다.

이전 자습서를 아직 실행하지 않은 경우 지금 실행하는 것이 좋습니다. 이 자습서에는 이전 자습서를 마친 지점에서 시작합니다.

이전 쿼리에 필터 명령을 추가하려면

1. 필터링할 필드를 결정합니다. 지난 15분 동안 선택한 로그 그룹에 포함된 로그 이벤트에서 CloudWatch Logs가 감지한 가장 일반적인 필드와 각 필드가 나타나는 로그 이벤트의 비율을 보려면 페이지 오른쪽에서 필드 선택합니다.

   특정 로그 이벤트에 포함된 필드를 확인하려면 해당 행 왼쪽에 있는 아이콘을 선택합니다.

   더 awsRegion 필드는 로그에 있는 이벤트에 따라 로그 이벤트에 나타날 수 있습니다. 이 튜토리얼의 나머지 부분에서는 awsRegion 을 필터 필드로 사용하지만 해당 필드를 사용할 수 없는 경우 다른 필드를 사용할 수 있습니다.
2. 쿼리 편집기 상자에서 50 뒤에 커서를 놓고 Enter를 누릅니다.
3. 새 줄에 먼저 |(파이프 문자)와 공백을 입력합니다. CloudWatch Logs Insights 쿼리 내 명령은 파이프 문자로 구분해야 합니다.
4. **`filter awsRegion="us-east-1"`**를 입력합니다.
5. 실행을 선택합니다.

   다시 쿼리를 실행하면 이제 새 필터와 일치하는 최신 결과 50개가 표시됩니다.

   다른 필드를 필터링했는데 오류 결과가 표시되면 해당 필드 이름을 이스케이프해야 할 수 있습니다. 필드 이름에 영숫자가 아닌 문자가 포함되어 있으면 필드 이름의 앞/뒤에 백틱 문자(`)를 입력해야 합니다(예: **`` `error-code`="102" ``**).

   영숫자가 아닌 문자를 포함하는 필드 이름에 백틱 문자를 사용해야 하지만 값은 그렇지 않습니다. 값은 항상 따옴표(“) 안에 포함됩니다.

CloudWatch Logs Insights에는 여러 가지 명령과 정규식, 수학적 연산 및 통계적 연산 지원을 비롯하여 강력한 쿼리 기능이 있습니다. 자세한 정보는 CloudWatch Logs Insights 쿼리 구문 (p. 39) 단원을 참조하십시오.

튜토리얼: 집계 함수를 사용하여 쿼리 실행

이 자습서에서는 로그 레코드에 대한 집계 함수 실행 결과를 반환하는 쿼리를 실행합니다.

집계 쿼리를 실행하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 쿼리 편집기 위에서 로그 그룹을 하나 이상 선택합니다. 로그 그룹을 쉽게 찾을 수 있도록 검색 창에 텍스트를 입력하면 CloudWatch Logs가 검색 창에 일치하는 로그 그룹을 표시합니다.
4. 쿼리 편집기에서 현재 표시되어 있는 쿼리를 삭제하고 다음을 입력한 후 실행을 선택합니다. 교체 필드 이름 필드 이름과 함께 표시되는 필드 페이지 오른쪽에 있습니다.

```
stats count(*) by fieldname
```

결과에는 CloudWatch Logs가 수신했고, 사용자가 선택한 필드 이름에 대해 각기 다른 값을 포함하고 있는 로그 그룹의 로그 이벤트 수가 표시됩니다.

튜토리얼: 로그 필드별로 그룹화된 시각화를 생성하는 쿼리 실행

를 사용하는 쿼리를 실행할 때 `stats` 기능을 사용하여 반환된 결과를 로그 항목의 하나 이상의 필드 값으로 그룹화할 수 있습니다. 결과를 막대 차트, 원형 차트, 선 그래프 또는 스택 영역 그래프로 볼 수 있습니다. 이렇게 하면 로그에서 추세를 보다 효율적으로 시각화할 수 있습니다.

시각화를 위한 쿼리를 실행하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 쿼리할 로그 그룹을 하나 이상 선택합니다.
4. 쿼리 편집기에서 현재 내용을 삭제하고 다음 `stats` 함수를 입력한 후 쿼리 실행을 선택합니다.

```
stats count(*) by @logStream
    | limit 100
```

결과는 로그 그룹에서 각 로그 스트림에 대한 로그 이벤트 수를 보여줍니다. 결과는 100개 행으로 제한됩니다.

5. Visualization(시각화) 탭을 선택합니다.
6. 선 옆에 있는 화살표를 선택한 다음 막대를 선택합니다.

막대 차트가 나타나고 로그 그룹의 각 로그 스트림에 대한 막대가 표시됩니다.

튜토리얼: 시계열 시각화를 생성하는 쿼리 실행

를 사용하는 쿼리를 실행할 때 `bin()` 기능을 사용하여 반환된 결과를 기간별로 그룹화하면 결과를 선 그래프, 스택 영역 그래프, 원형 차트 또는 막대 차트로 볼 수 있습니다. 따라서 시간 경과에 따른 로그 이벤트의 추세를 보다 효율적으로 시각화할 수 있습니다.

시각화를 위한 쿼리를 실행하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 쿼리할 로그 그룹을 하나 이상 선택합니다.
4. 쿼리 편집기에서 현재 내용을 삭제하고 다음 `stats` 함수를 입력한 후 쿼리 실행을 선택합니다.

```
stats count(*) by bin(30s)
```

결과에는 CloudWatch Logs가 30초 동안 매번 수신한 로그 그룹의 로그 이벤트 수가 표시됩니다.

5. Visualization(시각화) 탭을 선택합니다.

결과가 선 그래프로 표시됩니다. 막대 차트, 원형 차트 또는 누적 영역 차트로 전환하려면 라인 를 클릭합니다().

CloudWatch Logs Insights 쿼리 구문

CloudWatch Logs Insights에서는 로그 그룹에 대한 쿼리를 수행하는 데 사용할 수 있는 쿼리 언어를 지원합니다. 각 쿼리에는 Unix 스타일 파이프 문자(|)로 구분된 쿼리 명령을 한 개 이상 포함할 수 있습니다.

정규식, 산술 연산, 비교 연산, 숫자 함수, 날짜/시간 함수, 문자열 함수 및 일반 함수 등을 비롯하여 다양한 지원 함수 및 연산과 함께 6가지 쿼리 명령이 지원됩니다.

주석도 지원됩니다. 쿼리에서 # 문자로 시작되는 행은 무시됩니다.

다음으로 시작하는 필드 @ 기호 생성자 CloudWatch Logs 통찰력. 해당 필드에 대한 자세한 내용은 CloudWatch Logs 자동 검색 및 생성, 보기 지원되는 로그 및 검색되는 필드 (p. 33).

CloudWatch Logs Insights 쿼리 명령

다음 표에는 기본적인 예제와 함께 지원되는 6가지 쿼리 명령이 나와 있습니다. 더욱 복잡한 샘플 쿼리는 샘플 쿼리 (p. 50) 단원을 참조하십시오.

Command	설명	예제:

display	쿼리 결과에 표시할 필드를 지정합니다. 쿼리에서 이 명령을 두 번 이상 지정하면 마지막에 지정한 필드만 사용됩니다.	다음 예에서는 필드를 사용합니다. `@message` 임시 필드를 생성합니다 `loggingType` 및 `loggingMessage` 쿼리에 사용할 수 있습니다. 이벤트를 필터링하여 `ERROR` 을(를) `loggingType`, 그러나 그런 다음 `loggingMessage` 필드에 표시됩니다.
		`fields @message`
		`  | parse @message "[*] *" as loggingType, loggingMessage`
		`  | filter loggingType = "ERROR"`
		`  | display loggingMessage`

fields	표시할 로그 이벤트에서 지정된 필드를 검색합니다.
	
	필드 명령 내의 함수 및 작업을 사용하여 표시할 필드 값을 수정하고 쿼리의 나머지 부분에서 사용할 새 필드를 작성할 수 있습니다.
		다음 예는 필드를 표시합니다. `foo-bar`, `action`, 및 차이의 절대값 `f3` 및 `f4` 로그 그룹의 모든 로그 이벤트에 대해.
		`fields` `` `foo-bar` ``, `action`, `abs(f3-f4)`
		
		다음 예는 임시 필드를 생성하고 표시합니다. `opStatus`. 의 값 `opStatus` 각 로그 항목에 대한 는 `Operation` 및 `StatusCode` 필드, 해당 값 사이에 하이픈 사용.
		`fields concat(Operation, '-', StatusCode) as opStatus`

filter	하나 이상의 조건을 기반으로 쿼리 결과를 필터링합니다. 에서 다양한 연산자 및 식을 사용할 수 있습니다. **filter** 명령. 자세한 정보는 the section called “필터 명령에서 일치 및 정규식” (p. 42) 단원을 참조하십시오.	다음 예에서는 필드를 검색합니다. `f1`, `f2`, 및 `f3` 모든 로그 이벤트의 값이 2000을 초과하는 경우 `duration` 필드.

Command	설명	예제:

		`fields f1, f2, f3 | filter (duration>2000)`
		
		다음 예제도 유효한 쿼리이지만 결과에 별도의 필드가 표시되지 않습니다. 대신, 결과는 `@timestamp` 모든 로그 데이터를 `@message` 모든 로그 이벤트에 대한 필드 (기간이 2000보다 큼)
		
		`filter (duration>2000)`
		
		다음 예에서는 필드를 검색합니다. `f1` 및 `f2` 모든 로그 이벤트에 대해 `f1` 10 또는 `f3` 은 (는) 25보다 큽니다.
		
		`fields f1, f2 | filter (f1=10 or f3>25)`
		
		다음 예제에서는 필드가 `statusCode` 은 (는) 200에서 299 사이의 값을 가집니다.
		
		`fields f1 | filter statusCode like /2\d\d/`
		
		다음 예제에서는 다음을 포함하는 로그 이벤트를 반환합니다. `statusCode` "300", "400" 또는 "500" 중 선택할 수 있습니다.
		
		`fields @timestamp, @message | filter statusCode in [300,400,500]`
		
		이 마지막 예제는 `Type` 값이 "foo", "bar" 또는 "1"인 필드.
		
		`fields @timestamp, @message | filter Type not in ["foo","bar",1]`

`stats`	로그 필드의 값을 기반으로 집계 통계를 계산합니다. `stats`를 사용하는 경우 `by`를 사용하여 통계를 계산할 때 데이터를 그룹화하는데 사용할 기준을 하나 이상 지정할 수 있습니다. `sum()`, `avg()`, `count()`, `min()` 및 `max()`를 포함하나 여러 가지 통계 연산자가 지원됩니다.	다음 예제는 의 평균값을 계산합니다. `f1` 의 각 고유 값에 대해 `f2`. `stats avg (f1) by f2`

Command	설명	예제:
sort	검색된 로그 이벤트를 정렬합니다. 오름차순(asc) 및 내림차순(desc)이 둘 다 지원됩니다.	다음 예에서는 반환된 이벤트를 다음 값을 기준으로 내림차순으로 정렬합니다. f1, 및 은 필드를 표시합니다. f1, f2, 및 f3. `fields f1, f2, f3 | sort f1 desc`
limit	쿼리에서 반환되는 로그 이벤트 수를 지정합니다. 이 옵션을 사용하여 결과를 소수로 제한하여 관련 결과의 소집합을 볼 수 있습니다. 1,000에서 10,000 사이의 숫자와 limit를 함께 사용하여 콘솔에 표시되는 쿼리 결과 행 수를 기본값인 1,000행보다 크게 늘릴 수도 있습니다. 제한을 지정하지 않으면 쿼리의 기본값은 최대 1,000개 행을 표시합니다.	다음 예제에서는 다음 값을 기준으로 이벤트를 내림차순으로 정렬합니다. @timestamp, 및 은 필드를 표시합니다. f1 및 f2 정렬 순서별 첫 25개 이벤트. 이 경우, 정렬 순서는 타임스탬프를 기준으로 최신 타임스탬프부터 표시되므로, 최근 25개 이벤트가 반환됩니다. `sort @timestamp desc | limit 25 | display f1, f2`

parse	로그 필드에서 데이터를 추출하고 쿼리에서 추가로 처리할 수 있는 하나 이상의 임시 필드를 작성합니다. parse 는 glob 표현식과 정규 표현식을 모두 허용합니다.
	glob 표현식의 경우, 문자의 각 변수가 별표(*)로 대체될 수 있는 상수 문자열로 parse 명령을 제공합니다(큰따옴표 또는 작은따옴표로 문자를 묶음). 이는 as 다음의 별칭을 키워드로 하여 위치 순서로 임시 필드로 추출됩니다.
	정규식을 슬래시(/)로 둘러쌉니다. 표현식에는 추출될 일치 스트링의 각 부분이 네임드 캡처링 그룹에 묶여있습니다. 네임드 캡처링 그룹의 예는 name이 이름이고, .*가 패턴인 (?<name>.*)입니다.
		단일 로그 라인을 예로 사용합니다.
		`25 May 2019 10:24:39,474 [ERROR] {foo=2, bar=data} The error was: DataIntegrityException`
		다음 두 가지 parse 식은 각각 다음을 수행합니다. 임시 필드 level, config, 및 exception 이(가) 생성되었습니다. level 은(는) 다음 값을 가집니다. ERROR, config 은(는) 다음 값을 가집니다. {foo=2, bar=data}, 및 exception 은(는) 다음 값을 가집니다. DataIntegrityException. 첫 번째 예제에서는 glob 식을 사용하고, 두 번째 예제에서는 정규식을 사용합니다.
		`parse @message "[*] * The error was: *" as level, config, exception`
		`parse @message /\[(?<level>\S+)\]\s+(?<config>\{.*\})\s+The error was: (?<exception>\S+)/`
		다음 예제에서는 정규식을 사용하여 로그 필드 @message에서 임시 필드 @user2, @method2 및 @latency2를 추출하고 @method2 및 @user2의 고유한 개별 조합에 대한 평균 지연 시간을 반환합니다.
		`parse @message /user=(?<user2>.*?), method:(?<method2>.*?), latency := (?<latency2>.*?)/ | stats avg(@latency2) by @method2, @user2`

이전 테이블의 쿼리 명령에 대한 메모

다음 규칙, 지침 및 팁은 이전 표의 쿼리 명령에 적용됩니다.

- @ 기호, 마침표(.) 및 영숫자 문자 이외의 문자가 포함된 로그 필드가 쿼리에 명명되어 있으면 해당 필드 이름은 백틱 문자(`)로 둘러싸야 합니다. 예를 들어 `foo-bar` 필드 이름은 영숫자가 아닌 문자를 포함하므로 백틱 문자로 묶어야 합니다.
- 둘 다 **`fields`** 및 **`display`** 쿼리 결과에 표시할 필드를 지정하는 데 사용됩니다. 이 두 가지의 차이점은 다음과 같습니다.
  - 귀하는 **`display`** 를 클릭하여 결과에 표시할 필드를 지정합니다. 다음을 사용할 수 있습니다. **`fields`** 명령 사용 (으)로 키워드를 사용하여 함수 및 로그 이벤트의 필드를 사용하여 새 임시 필드를 만듭니다. 예를 들어, `fields ispresent(resolverArn) as isRes` 임시 필드 생성 `isRes` 쿼리의 나머지 부분에서 사용할 수 있습니다. 의 값 `isRes` 은(는) `resolverArn` 로그 이벤트 의 검색된 필드입니다.
  - 여러 개의 **`fields`** 명령 및 은(는) **`display`** 명령, 모든 **`fields`** `commands are displayed.`
  - 여러 개의 **`display`** 명령, 마지막에 지정된 필드만 **`display`** `command are displayed.`

필터 명령에서 일치 및 정규식

비교 연산자(=, !=, <, <=, >, >=), 부울 연산자(and, or, 및 not) 및 정규식이 **`filter`** 명령.

in을 사용하여 집합 소속을 테스트할 수 있습니다. 즉시 확인할 요소가 있는 배열을 배치하십시오. in. 다음을 사용할 수 있습니다. not 함께 in. 문자열 일치 사용 in 은(는) 완전한 문자열 일치 이어야 합니다.

하위 문자열을 기준으로 필터링하려면 다음을 사용할 수 있습니다. like 또는 =~ (동등한 부호 뒤에 물결무늬가 이어짐) filter 명령. 다음을 사용하여 하위 문자열 일치 like 또는 =~, 하위 문자열을 묶어 이중 또는 단일 인용 부호로 일치시키십시오. 정규식 일치를 수행하려면, 슬래시와 일치시킬 표현식을 둘러싸십시오. 쿼리가 설정한 기준과 일치하는 로그 이벤트만 반환합니다.

예제:

다음 세 가지 예는 f1 단어 포함 Exception. 처음 두 예제에서는 정규식 을 사용합니다. 세 번째 예제에서는 부분 문자열 일치를 사용합니다. 세 예제 모두 대/소문자를 구별합니다.

```
fields f1, f2, f3 | filter f1 like /Exception/
```

```
fields f1, f2, f3 | filter f1 =~ /Exception/
```

```
fields f1, f2, f3 | filter f1 like "Exception"
```

다음 예에서는 "예외"에 대한 검색을 대소문자를 구분하지 않도록 변경합니다.

```
fields f1, f2, f3 | filter f1 like /(?i)Exception/
```

다음 예제에서는 정규식을 사용합니다. 모든 이벤트를 반환합니다. f1 정확히 Exception. 쿼리는 대소문자를 구분하지 않습니다.

```
fields f1, f2, f3 | filter f1 =~ /^(?i)Exception$/
```

쿼리에 별칭 사용

as를 사용하여 쿼리에서 별칭을 하나 이상 생성할 수 있습니다. 별칭은 fields, stats 및 sort 명령에서 지원됩니다.

로그 필드에 대한 별칭과 연산 및 함수의 결과에 대한 별칭을 생성할 수 있습니다.

예제:

다음 예제는 쿼리 명령에서 별칭 사용을 보여줍니다.

```
fields abs(myField) as AbsoluteValuemyField, myField2
```

`myField`의 절대값을 `AbsoluteValuemyField`로 반환하고 필드 `myField2`도 반환합니다.

```
stats avg(f1) as myAvgF1 | sort myAvgF1 desc
```

`f1`의 값 평균을 `myAvgF1`으로 계산하여 해당 값을 기준으로 내림차순으로 반환합니다.

쿼리에서 주석 사용

`#` 문자를 사용하여 쿼리에서 행을 주석 처리할 수 있습니다. `#` 문자로 시작되는 행은 무시됩니다. 이 기능을 쿼리를 문서화하는 경우 또는 단일 호출에서 해당 행을 삭제하지 않고 복잡한 쿼리의 일부를 일시적으로 무시하려는 경우 유용할 수 있습니다.

다음 예제에서 두 번째 행이 무시됩니다.

```
fields @timestamp, @message
    # | filter @message like /delay/
    | limit 20
```

지원되는 연산 및 함수

이 쿼리 언어는 다음 표에 나와 있는 다양한 유형의 연산 및 함수를 지원합니다.

비교 연산

비교 연산은 `filter` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다. 비교 연산은 모든 데이터 형식을 인수로 수락하고 부울 결과를 반환합니다.

```
= != < <= > >=
```

부울 연산자

부울 연산자를 사용할 수 있습니다. **`and`**, **`or`**, 및 **`not`**. 이러한 부울 연산자는 부울 값을 반환하는 기능에서만 사용할 수 있습니다.

산술 연산

산술 연산을 `filter` 및 `fields` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다. 산술 연산은 숫자 데이터 형식을 인수로 수락하고 숫자 결과를 반환합니다.

연산	설명
`a + b`	덧셈
`a - b`	뺄셈
`a * b`	곱셈
`a / b`	나눗셈
`a ^ b`	지수화. `2 ^ 3` 반품 `8`

연산	설명
`a % b`	나머지 또는 계수. `10 % 3` 반품 `1`

숫자 연산

숫자 연산을 `filter` 및 `fields` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다. 숫자 연산은 숫자 데이터 형식을 인수로 수락하고 숫자 결과를 반환합니다.

연산	결과 유형	설명
`abs(a: number)`	숫자	절대값입니다.
`ceil(a: number)`	숫자	천장값으로 반올림합니다(a의 값보다 큰 수 중 가장 작은 정수).
`floor(a: number)`	숫자	바닥값으로 반올림합니다(a) 값보다 작은 수 중 가장 큰 정수).
`greatest(a: number, ...numbers: number[])`	숫자	가장 큰 값을 반환합니다.
`least(a: number, ...numbers: number[])`	숫자	가장 작은 값을 반환합니다.
`log(a: number)`	숫자	자연 로그입니다.
`sqrt(a: number)`	숫자	제곱근입니다.

일반 함수

일반 함수를 `filter` 및 `fields` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다.

Function	결과 유형	설명
`ispresent(fieldName: LogField)`	boolean	이 필드가 존재하는 경우 `true`를 반환합니다.
`coalesce(fieldName: LogField, ...fieldNames: LogField[])`	LogField	목록에서 null이 아닌 첫 번째 값을 반환합니다.

문자열 함수

문자열 함수를 `filter` 및 `fields` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다.

Function	결과 유형	설명
`isempty(fieldName: string)`	boolean	필드가 누락되어 있거나 빈 문자열일 경우 `true`를 반환합니다.
`isblank(fieldName: string)`	boolean	필드가 누락되어 있거나 빈 문자열이거나 빈 공백만 포함된 경우 `true`를 반환합니다.

Function	결과 유형	설명
`concat(str: string, ...strings: string[])`	string	문자열을 연결합니다.

`ltrim(str: string)`
 	
`ltrim(str: string, subStr: string)`
	string	문자열의 왼쪽에서 공백을 제거합니다. 함수에 두 번째 문자열 인수가 있는 경우 다음 문자를 제거합니다. `subStr` 왼쪽에서 `str`. 예를 들어, `ltrim("xyZfooxyZ","xyZ")` 반품 `"fooxyZ"`.

`rtrim(str: string)`
 	
`rtrim(str: string, subStr: string)`
	string	문자열의 오른쪽에서 공백을 제거합니다. 함수에 두 번째 문자열 인수가 있는 경우 다음 문자를 제거합니다. `subStr` 오른쪽에서 `str`. 예를 들어, `rtrim("xyZfooxyZ","xyZ")` 반품 `"xyZfoo"`.

`trim(str: string)`
 	
`trim(str: string, subStr: string)`
	string	문자열의 양쪽 끝에서 공백을 제거합니다. 함수에 두 번째 문자열 인수가 있는 경우 다음 문자를 제거합니다. `subStr` 양쪽에서 `str`. 예를 들어, `trim("xyZfooxyZ","xyZ")` 반품 `"foo"`.

`strlen(str: string)`	숫자	문자열 길이를 Unicode 코드 포인트로 반환합니다.
`toupper(str: string)`	string	문자열을 대문자로 변환합니다.
`tolower(str: string)`	string	문자열을 소문자로 변환합니다.

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	string	숫자 인수가 지정한 인덱스의 하위 문자열을 문자열 끝에 반환합니다. 함수에 두 번째 숫자 인수가 있는 경우 해당 인수에는 검색되는 하위 문자열의 길이가 포함됩니다. 예를 들어, `substr("xyZfooxyZ",3, 3)`는 `"foo"`을 반환합니다.

`replace(str: string, searchValue: string, replaceValue: string)`	string	모든 인스턴스 대체 `searchValue` 에서 `str` 함께 `replaceValue`. 예를 들어, `replace("foo","o","0")` 반품 `"f00"`.
`strcontains(str: string, searchValue: string)`	숫자	`str`에 `searchValue`와 0이 포함되어 있으면 1을 반환합니다.

날짜/시간 함수

날짜/시간 함수를 `filter` 및 `fields` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다. 이러한 함수를 사용해 집계 함수가 포함된 쿼리에 대한 시간 버킷을 생성할 수 있습니다.

날짜/시간 함수의 일부로 숫자 다음에 분을 나타내는 `m` 또는 시간을 나타내는 `h`로 구성된 기간을 사용할 수 있습니다. 예를 들어, `10m`은 10분을, `1h`는 1시간을 나타냅니다.

Function	결과 유형	설명
`bin(period: Period)`	Timestamp	`@timestamp`의 값을 지정한 기간으로 반올림한 다음 자릅니다.
`datefloor(timestamp: Timestamp, period: Period)`	Timestamp	타임스탬프를 지정한 기간으로 자릅니다. 예를 들어, `datefloor(@timestamp, 1h)`는 `@timestamp`의 모든 값을 해당 시간 아래로 자릅니다.
`dateceil(timestamp: Timestamp, period: Period)`	Timestamp	타임스탬프를 지정한 기간으로 반올림한 다음 자릅니다. 예를 들어, `dateceil(@timestamp, 1h)`는 `@timestamp`의 모든 값을 해당 시간 위로 자릅니다.
`fromMillis(fieldName: number)`	Timestamp	입력 필드를 Unix Epoch 밀리초로 해석하여 타임스탬프로 변환합니다.
`toMillis(fieldName: Timestamp)`	숫자	지정된 필드에 있는 타임스탬프를 Unix Epoch 밀리초를 나타내는 숫자로 변환합니다.

IP 주소 함수

IP 주소 문자열 함수를 `filter` 및 `fields` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다.

Function	결과 유형	설명
`isValidIp(fieldName: string)`	boolean	반품 `true` 필드가 유효한 경우 IPv4 또는 IPv6 주소.
`isValidIpV4(fieldName: string)`	boolean	반품 `true` 필드가 유효한 경우 IPv4 주소.
`isValidIpV6(fieldName: string)`	boolean	반품 `true` 필드가 유효한 경우 IPv6 주소.
`isIpInSubnet(fieldName: string, subnet: string)`	boolean	반품 `true` 필드가 유효한 경우 IPv4 또는 IPv6 지정된 v4 또는 v6 서브넷 내의 주소입니다. 서브넷을 지정할 때는 `192.0.2.0/24` 또는 `2001:db8::/32`와 같은 CIDR 표기법을 사용합니다.
`isIpv4InSubnet(fieldName: string, subnet: string)`	boolean	반품 `true` 필드가 유효한 경우 IPv4 지정된 v4 서브넷 내의 주소입니다. 서브넷을 지정할 때는 `192.0.2.0/24`와 같은 CIDR 표기법을 사용합니다.
`isIpv6InSubnet(fieldName: string, subnet: string)`	boolean	반품 `true` 필드가 유효한 경우 IPv6 지정된 v6 서브넷 내의 주소입니다. 서브넷을 지정할 때는 `2001:db8::/32`와 같은 CIDR 표기법을 사용합니다.

통계 집계 함수

집계 함수를 `stats` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다.

Function	결과 유형	설명
`avg(fieldName: NumericLogField)`	숫자	지정된 필드의 값 평균입니다.

`count()`
 	
`count(fieldName: LogField)`
	숫자	로그 이벤트를 카운트합니다. `count()` (또는 `count(*)`) 쿼리에 의해 반환된 모든 이벤트를 카운트하는 동안 `count(fieldName)` 지정된 필드 이름을 포함하는 모든 레코드를 카운트합니다.

`count_distinct(fieldName: LogField)`	숫자	필드에 대해 고유한 값의 개수를 반환합니다. 필드의 카디널리티가 매우 높은 경우(고유한 값이 많이 포함되어 있음) `count_distinct`가 반환하는 값은 근사치입니다.
`max(fieldName: LogField)`	LogFieldValue	쿼리된 로그에서 이 로그 필드에 대한 최대값입니다.
`min(fieldName: LogField)`	LogFieldValue	쿼리된 로그에서 이 로그 필드에 대한 최소값입니다.
`pct(fieldName: LogFieldValue, percent: number)`	LogFieldValue	백분위수는 데이터 세트에서 값의 상대적 위치를 나타냅니다. 예를 들어, `pct(@duration, 95)`는 `@duration`의 값 중 95퍼센트가 이 값보다 낮고 5퍼센트는 이 값보다 큰 `@duration` 값을 반환합니다.
`stddev(fieldName: NumericLogField)`	숫자	지정된 필드의 값에 대한 표준 편차입니다.
`sum(fieldName: NumericLogField)`	숫자	지정된 필드의 값 합계입니다.

통계 비집계 기능

비집계 함수를 `stats` 명령에서 사용하고 다른 함수의 인수로 사용할 수 있습니다.

Function	결과 유형	설명
`earliest(fieldName: LogField)`	LogField	쿼리된 로그에서 가장 이른 타임스탬프가 있는 로그 이벤트에서 `fieldName`의 값을 반환합니다.
`latest(fieldName: LogField)`	LogField	쿼리된 로그에서 최신 타임스탬프가 있는 로그 이벤트에서 `fieldName`의 값을 반환합니다.
`sortsFirst(fieldName: LogField)`	LogField	쿼리된 로그에서 가장 빨리 정렬된 `fieldName`의 값을 반환합니다.
`sortsLast(fieldName: LogField)`	LogField	쿼리된 로그에서 가장 늦게 정렬된 `fieldName`의 값을 반환합니다.

그래프로 로그 데이터 시각화

막대 차트, 선형 차트 및 누적 영역 차트와 같은 시각화를 사용하여 로그 데이터의 패턴을 보다 효율적으로 식별할 수 있습니다. CloudWatch Logs Insights는 `stats` 함수와 하나 이상의 집계 함수를 사용하는 쿼리에 대한 시각화를 생성합니다. 자세한 정보는 Aggregation Functions in the Stats Command (p. 46) 단원을 참조하십시오.

이러한 쿼리는 모두 막대 차트를 생성할 수 있습니다. 쿼리가 `bin()` 함수를 사용하여 필드별로 데이터를 그룹화하는 경우 선형 차트 및 누적 영역 차트도 볼 수 있습니다.

주제

- 시계열 데이터 시각화 (p. 48)
- 필드별로 그룹화된 로그 데이터 시각화 (p. 48)

시계열 데이터 시각화

시계열 시각화는 다음과 같은 특성을 가진 쿼리에 사용할 수 있습니다.

- 쿼리에 집계 함수가 하나 이상 포함되어 있습니다. 자세한 정보는 Aggregation Functions in the Stats Command (p. 46) 단원을 참조하십시오.
- 쿼리가 `bin()` 함수를 사용하여 필드 하나를 기준으로 데이터를 그룹화합니다.

이러한 쿼리는 선 차트, 누적 영역 차트, 막대 차트 및 파이 차트를 생성할 수 있습니다.

예제:

전체 자습서는 the section called “튜토리얼: 시계열 시각화를 생성하는 쿼리 실행” (p. 38) 단원을 참조하십시오.

다음은 시계열 시각화에 사용할 수 있는 쿼리 예제입니다.

다음 쿼리는 5분마다 생성된 데이터 포인트와 함께 `myfield1` 필드의 평균 값에 대한 시각화를 생성합니다. 각 데이터 포인트는 이전 5분 동안 생성된 로그의 `myfield1` 값 평균을 집계한 것입니다.

```
stats avg(myfield1) by bin(5m)
```

다음 쿼리는 5분마다 생성된 데이터 포인트와 함께 여러 필드를 기준으로 세 값의 시각화를 생성합니다. 쿼리가 집계 함수를 포함하고 있고 그룹화 필드로 `bin()`을 사용하기 때문에 시각화가 생성됩니다.

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

선형 차트 및 누적 영역 차트 제한 사항

로그 항목 정보를 집계하지만 `bin()` 함수를 사용하지 않는 쿼리는 막대 차트를 생성할 수 있습니다. 그러나 이러한 쿼리는 선형 차트 또는 누적 영역 차트를 생성할 수 없습니다. 이러한 쿼리 유형에 대한 자세한 내용은 the section called “필드별로 그룹화된 로그 데이터 시각화” (p. 48) 단원을 참조하십시오.

필드별로 그룹화된 로그 데이터 시각화

`stats` 함수와 하나 이상의 집계 함수를 사용하는 쿼리에 대해 막대 차트를 생성할 수 있습니다. 자세한 정보는 Aggregation Functions in the Stats Command (p. 46) 단원을 참조하십시오.

시각화를 보려면 쿼리를 실행합니다. 그런 다음 Visualization(시각화) 탭을 선택하고 Line(선형) 옆의 화살표를 선택한 다음 Bar(막대)를 선택합니다. 막대 차트에서는 시각화가 최대 100개 막대로 제한됩니다.

예제:

전체 자습서는 the section called “튜토리얼: 로그 필드별로 그룹화된 시각화를 생성하는 쿼리 실행” (p. 38) 단원을 참조하십시오. 다음 단락에는 필드별 시각화에 대한 더 많은 예제 쿼리가 포함되어 있습니다.

다음 VPC 흐름 로그 쿼리는 각 대상 주소에 대해 세션당 전송된 평균 바이트 수를 찾습니다.

```
stats avg(bytes) by dstAddr
```

각 결과 값에 대해 둘 이상의 막대가 포함된 차트를 생성할 수도 있습니다. 예를 들어 다음 VPC 흐름 로그 쿼리는 각 대상 주소에 대해 세션당 전송된 평균 및 최대 바이트 수를 찾습니다.

```
stats avg(bytes), max(bytes) by dstAddr
```

다음 쿼리는 쿼리 유형에 대해 Amazon Route 53 쿼리 로그 수를 찾습니다.

```
stats count(*) by queryType
```

CloudWatch Logs Insights 쿼리 저장 및 재실행

쿼리를 작성한 후 나중에 다시 실행할 수 있도록 저장할 수 있습니다. 저장된 쿼리는 폴더 구조로 유지되므로 체계적으로 관리할 수 있습니다. 계정별로 리전당 최대 1,000개의 CloudWatch Logs Insights 쿼리를 저장할 수 있습니다.

쿼리를 저장하려면 `logs:PutQueryDefinition` 권한이 있는 역할에 로그인해야 합니다. 저장된 쿼리의 목록을 보려면 `logs:DescribeQueryDefinitions` 권한이 있는 역할에 로그인해야 합니다.

쿼리를 저장하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 쿼리 편집기에서 쿼리를 작성합니다.
4. Save를 선택합니다.

   저장 버튼이 표시되지 않으면 CloudWatch Logs 콘솔의 새 디자인으로 전환해야 합니다. 그렇게 하려면 다음을 수행하십시오.

   a. 탐색 창에서 로그 그룹을 선택합니다.
   b. 새 디자인 사용해 보기를 선택합니다.
   c. 탐색 창에서 Insights를 선택하고 이 절차의 3단계로 돌아갑니다.
5. 쿼리의 이름을 입력합니다.
6. (선택 사항) 쿼리를 저장할 폴더를 선택합니다. 새로 생성을 선택하여 폴더를 만듭니다. 새 폴더를 만드는 경우 폴더 이름에 슬래시(/) 문자를 사용하여 폴더 구조를 정의할 수 있습니다. 예를 들어 새 폴더의 이름을 **`folder-level-1/folder-level-2`**로 지정하면 **`folder-level-1`**이라는 최상위 폴더가 만들어지고 그 폴더 안에 **`folder-level-2`**라는 다른 폴더가 만들어집니다. 쿼리가 **`folder-level-2`**에 저장됩니다.
7. (선택 사항) 쿼리의 로그 그룹 또는 쿼리 텍스트를 변경합니다.
8. Save를 선택합니다.

저장된 쿼리를 실행하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 오른쪽에서 쿼리를 선택합니다.
4. 저장된 쿼리 목록에서 쿼리를 선택합니다. 이 쿼리가 쿼리 편집기에 나타납니다.
5. 실행을 선택합니다.

저장된 쿼리의 새 버전을 저장하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 오른쪽에서 쿼리를 선택합니다.
4. 저장된 쿼리 목록에서 쿼리를 선택합니다. 이 쿼리가 쿼리 편집기에 나타납니다.
5. 쿼리를 수정합니다. 작업을 확인하기 위해 쿼리를 실행해야 하는 경우 쿼리 실행을 선택합니다.
6. 새 버전을 저장할 준비가 되면 작업, 다른 이름으로 저장을 선택합니다.
7. 쿼리의 이름을 입력합니다.
8. (선택 사항) 쿼리를 저장할 폴더를 선택합니다. 새로 생성을 선택하여 폴더를 만듭니다. 새 폴더를 만드는 경우 폴더 이름에 슬래시(/) 문자를 사용하여 폴더 구조를 정의할 수 있습니다. 예를 들어 새 폴더의 이름을 **`folder-level-1/folder-level-2`**로 지정하면 **`folder-level-1`**이라는 최상위 폴더가 만들어지고 그 폴더 안에 **`folder-level-2`**라는 다른 폴더가 만들어집니다. 쿼리가 **`folder-level-2`**에 저장됩니다.
9. (선택 사항) 쿼리의 로그 그룹 또는 쿼리 텍스트를 변경합니다.
10. Save를 선택합니다.

쿼리를 삭제하려면 `logs:DeleteQueryDefinition` 권한이 있는 역할에 로그인해야 합니다.

저장된 쿼리를 편집 또는 삭제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 오른쪽에서 쿼리를 선택합니다.
4. 저장된 쿼리 목록에서 쿼리를 선택합니다. 이 쿼리가 쿼리 편집기에 나타납니다.
5. 작업, 편집 또는 작업, 삭제를 선택합니다.

샘플 쿼리

이 단원에는 CloudWatch Logs Insights의 이점을 보여주는 예제 쿼리가 포함되어 있습니다.

일반 쿼리

다음은 최근에 추가된 로그 이벤트 25개를 찾습니다.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

다음은 시간당 발생한 예외 수 목록을 가져옵니다.

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
```

```
    | sort exceptionCount desc
```

다음은 예외에 해당되지 않는 로그 이벤트 목록을 가져옵니다.

```
fields @message | filter @message not like /Exception/
```

질의 대상 Lambda 로그

과다 프로비저닝된 메모리 양을 확인합니다.

```
filter @type = "REPORT"
    | stats max(@memorySize / 1024 / 1024) as provisonedMemoryMB,
        min(@maxMemoryUsed / 1024 / 1024) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1024 / 1024) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1024 / 1024) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

지연 보고서를 생성합니다.

```
filter @type = "REPORT" |
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

질의 대상 Amazon VPC 흐름 로그

다음은 호스트 간에 상위 15개의 패킷 전송을 찾습니다.

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

지정된 서브넷의 호스트에 대해 상위 15개 바이트 전송을 찾습니다.

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

다음은 데이터 전송 프로토콜로 UDP를 사용하는 IP 주소를 찾습니다.

```
filter protocol=17 | stats count(*) by srcAddr
```

다음은 캡처 기간 중 흐름 레코드를 건너뛴 IP 주소를 찾습니다.

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

질의 대상 Route 53 로그

다음은 시간당 레코드 배포를 쿼리 유형별로 찾습니다.

```
stats count(*) by queryType, bin(1h)
```

다음은 요청 수가 가장 많은 10 DNS 해석기 10개를 찾습니다.

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

다음은 서버가 DNS 요청을 완료하지 못한 도메인 및 하위 도메인별 레코드 수를 찾습니다.

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

질의 대상 CloudTrail 로그

다음은 각 서비스, 이벤트 유형 및 AWS 리전에 대한 로그 항목 수를 찾습니다.

```
stats count(*) by eventSource, eventName, awsRegion
```

다음은 지정된 AWS 리전에서 시작 또는 중지한 Amazon EC2 호스트를 찾습니다.

```
filter (eventName="StartInstances" or eventName="StopInstances") and region="us-east-2"
```

AWS 영역, 사용자 이름 및 ARNs 새로 만든 IAM 사용자.

```
filter eventName="CreateUser"
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

다음은 API `UpdateTrail`을 호출하는 중 예외가 발생한 레코드 수를 찾습니다.

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

분석 명령의 예제

glob 표현식을 사용하여 로그 필드 `@message`에서 임시 필드 `@user`, `@method` 및 `@latency`를 추출하고 `@method` 및 `@user`의 고유한 개별 조합에 대한 평균 지연 시간을 반환합니다.

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

정규식을 사용하여 로그 필드 `@message`에서 임시 필드 `@user2`, `@method2` 및 `@latency2`를 추출하고 `@method2` 및 `@user2`의 고유한 개별 조합에 대한 평균 지연 시간을 반환합니다.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(@latency2) by @method2,
    @user2
```

휘발성 필드인 `loggingTime`, `loggingType` 및 `loggingMessage` 필드를 추출하고 `ERROR` 또는 `INFO` 문자열이 포함된 이벤트를 기록하도록 필터링한 다음 `ERROR` 문자열이 포함된 이벤트에 대해 `loggingMessage` 및 `loggingType` 필드만 표시합니다.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

대시보드에 쿼리 추가 또는 쿼리 결과 내보내기

쿼리를 실행한 후 CloudWatch 대시보드에 쿼리를 추가하거나 클립보드에 쿼리 결과를 복사할 수 있습니다.

대시보드에 추가한 쿼리는 대시보드를 로드할 때마다 그리고 대시보드를 새로 고칠 때마다 실행됩니다. 이러한 쿼리는 10개 동시 CloudWatch Logs Insights 쿼리.

대시보드에 쿼리 결과를 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 로그 그룹을 하나 이상 선택하고 쿼리를 실행합니다.
4. 대시보드에 추가를 선택합니다.
5. 대시보드를 선택하거나 새로 생성을 선택하여 쿼리 결과에 대한 새 대시보드를 생성합니다.
6. 쿼리 결과에 사용할 위젯 유형을 선택합니다.
7. 위젯의 이름을 입력합니다.
8. 대시보드에 추가를 선택합니다.

쿼리 결과를 클립보드로 복사하거나 쿼리 결과를 다운로드하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 로그 그룹을 하나 이상 선택하고 쿼리를 실행합니다.
4. 결과 내보내기를 선택한 다음 원하는 옵션을 선택합니다.

실행 중인 쿼리 또는 쿼리 기록 보기

현재 진행 중인 쿼리와 최신 쿼리 기록을 볼 수 있습니다.

현재 실행 중인 쿼리에는 대시보드에 추가한 쿼리가 포함됩니다. 귀하는 10개로 동시에 제한됩니다. CloudWatch Logs Insights 계정당 쿼리(대시보드에 추가된 쿼리 포함)

최근 쿼리 기록을 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Insights를 선택합니다.
3. 새 디자인의 CloudWatch Logs 콘솔을 사용하는 경우 기록을 선택합니다. 이전 디자인을 사용하는 경우 작업, 이 계정에 대한 쿼리 기록 보기를 선택합니다.

   최근 쿼리 목록이 나타납니다. 쿼리를 선택하고 실행을 선택하여 이 중 하나를 다시 실행할 수 있습니다.

   상태 아래의 CloudWatch Logs에서 현재 실행 중인 쿼리가 진행 중으로 표시됩니다.

로그 그룹 및 로그 스트림 작업

로그 스트림은 동일한 소스를 공유하는 로그 이벤트 시퀀스입니다. CloudWatch Logs에서 각 별도의 로그 소스가 별도의 로그 스트림을 구성합니다.

로그 그룹은 동일한 보존 기간, 모니터링 및 액세스 제어 설정을 공유하는 로그 스트림의 그룹입니다. 로그 그룹을 정의하고 각 그룹에 배치할 스트림을 지정할 수 있습니다. 하나의 로그 그룹에서 포함할 수 있는 로그 스트림의 수에는 제한이 없습니다.

이 단원에 나오는 절차를 사용하여 로그 그룹 및 로그 스트림 작업을 수행합니다.

CloudWatch Logs에 로그 그룹을 생성합니다.

Amazon CloudWatch Logs User Guide의 이전 단원에 나온 단계들을 사용하여 Amazon EC2 인스턴스에 CloudWatch Logs 에이전트를 설치하면 해당 프로세스의 일부로 로그 그룹이 생성됩니다. CloudWatch 콘솔에서 직접 로그 그룹을 생성할 수도 있습니다.

사용자 그룹을 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 작업을 선택한 후 로그 그룹 생성을 선택합니다.
4. 로그 그룹의 이름을 입력한 다음 로그 그룹 생성을 선택합니다.

로그 그룹에 로그 보내기

CloudWatch Logs 여러 로부터의 로그 이벤트를 자동으로 AWS 서비스. 다른 로그 이벤트를 CloudWatch Logs 다음 방법 중 하나를 사용합니다.

- CloudWatch 에이전트— 통합 CloudWatch 에이전트는 두 메트릭과 로그를 모두 보낼 수 CloudWatch Logs. 설치 및 사용에 대한 정보 CloudWatch 에이전트, 참조 메트릭 및 로그 수집 Amazon EC2 인스턴스 및 온프레미스 서버 CloudWatch 에이전트 in the Amazon CloudWatch 사용 설명서.
- AWS CLI—The 로그-로그-이벤트 로그 이벤트 일괄 업로드 CloudWatch Logs.
- 프로그램적으로— The 이벤트 API를 통해 로그 이벤트의 배치를 CloudWatch Logs.

CloudWatch Logs에 전송된 로그 데이터 보기

CloudWatch Logs 에이전트가 CloudWatch Logs로 전송한 로그 데이터를 스트림별로 확인하고 스크롤할 수 있습니다. 확인할 로그 데이터에 대해 시간 범위를 지정할 수 있습니다.

로그 데이터를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 로그 그룹에서 스트림을 확인할 로그 그룹을 선택합니다.

4. 로그 그룹 목록에서 보려는 로그 그룹의 이름을 선택합니다.
5. 로그 스트림 목록에서 보려는 로그 스트림의 이름을 선택합니다.
6. 로그 데이터의 표시 방법을 변경하려면 다음 중 하나를 수행합니다.

   - 단일 로그 이벤트를 확장하려면 해당 로그 이벤트 옆에 있는 화살표를 선택합니다.
   - 로그 이벤트 목록에서 모든 로그 이벤트를 확장하고 이를 일반 텍스트 버전으로 보려면 텍스트를 선택합니다.
   - 로그 이벤트를 필터링하려면 검색 필드에 원하는 검색 필터를 입력합니다. 자세한 정보는 필터를 사용하여 로그 이벤트에서 지표 생성 (p. 65) 단원을 참조하십시오.
   - 지정된 날짜 및 시간 범위의 로그 데이터를 보려면 검색 필터 옆의 날짜 및 시간 옆에 있는 화살표를 선택합니다. 날짜 및 시간 범위를 지정하려면 절대를 선택합니다. 미리 정의된 분, 시간, 일 또는 주 수를 선택하려면 상대를 선택합니다. UTC와 현지 시간대 간을 전환할 수도 있습니다.

필터 패턴을 사용하여 로그 데이터 검색

필터 및 패턴 구문 (p. 66)를 사용하여 로그 데이터를 검색할 수 있습니다. 로그 그룹 내의 모든 로그 스트림을 검색하거나 AWS CLI를 사용하여 특정 로그 스트림을 검색할 수 있습니다. 각각의 검색이 실행되면서 데이터의 다음 페이지를 검색하거나 검색을 계속할 수 있도록 발견된 데이터의 첫 페이지와 토큰이 반환됩니다. 결과가 반환되지 않은 경우에는 계속 검색을 할 수 있습니다.

검색 범위를 제한하기 위해 쿼리하고자 하는 시간 범위를 설정할 수 있습니다. 처음에는 시간 범위를 넓게 잡아서 관심 있는 로그 줄이 어디에 있는지 확인한 다음, 시간 범위를 좁혀서 관심 있는 시간 범위로 로그에 대한 뷰의 범위를 지정할 수 있습니다.

로그에서 추출된 지표에서 해당 로그로 직접 피벗을 적용할 수도 있습니다.

콘솔을 이용하여 로그 항목 검색

콘솔을 이용하여 지정된 기준을 충족하는 로그 항목을 검색할 수 있습니다.

콘솔을 이용하여 로그 항목을 검색하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 로그 그룹에서 검색할 로그 스트림이 포함된 로그 그룹의 이름을 선택합니다.
4. 로그 스트림에서 검색할 로그 스트림의 이름을 선택합니다.
5. 로그 이벤트에서 사용할 필터 구문을 입력합니다.

콘솔을 이용하여 시간 범위에 대한 모든 로그 항목을 검색하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 로그 그룹에서 검색할 로그 스트림이 포함된 로그 그룹의 이름을 선택합니다.
4. 로그 그룹 검색을 선택합니다.
5. 로그 이벤트에서 날짜 및 시간 범위를 선택하고 필터 구문을 입력합니다.

AWS CLI를 이용하여 로그 항목 검색

AWS CLI를 이용하여 지정된 기준을 충족하는 로그 항목을 검색할 수 있습니다.

AWS CLI를 이용하여 로그 항목을 검색하려면

명령 프롬프트에서 아래 filter-log-events 명령을 실행합니다. `--filter-pattern`을 사용하여 지정된 필터 패턴으로 결과를 제한하고 `--log-stream-names`을 사용하여 지정된 로그 그룹으로 결과를 제한합니다.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] --filter-pattern VALID_METRIC_FILTER_PATTERN]
```

AWS CLI를 이용하여 지정된 시간 범위에 대한 모든 로그 항목을 검색하려면

명령 프롬프트에서 아래 filter-log-events 명령을 실행합니다.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-time 1482217558365]
 [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

지표에서 로그로 피벗 적용

콘솔의 다른 부분에서 특정 로그 항목으로 이동할 수 있습니다.

대시보드 위젯에서 로그로 이동하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. 대시보드를 선택합니다.
4. 위젯에서 로그 보기 아이콘을 선택한 다음 이 시간 범위에서 로그 보기를 선택합니다. 지표 필터가 하나 이상 있는 경우 목록에서 하나를 선택합니다. 지표 필터의 수가 목록에 표시할 수 있는 것보다 많은 경우에는 더 많은 지표 필터를 선택하고 지표 필터를 선택 또는 검색합니다.

지표에서 로그로 이동하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 모든 지표 탭의 검색 필드에 지표 이름을 입력하고 Enter 키를 누릅니다.
4. 검색 결과에서 지표를 하나 이상 선택합니다.
5. 작업과 로그 보기를 선택합니다. 지표 필터가 하나 이상 있는 경우 목록에서 하나를 선택합니다. 지표 필터의 수가 목록에 표시할 수 있는 것보다 많은 경우에는 더 많은 지표 필터를 선택하고 지표 필터를 선택 또는 검색합니다.

Troubleshooting

검색 완료에 너무 많은 시간이 소요

로그 데이터가 많은 경우에는 검색을 완료하는 데 긴 시간이 소요될 수 있습니다. 다음과 같은 방법으로 검색 속도를 높일 수 있습니다.

- AWS CLI를 사용 중인 경우 관심이 있는 로그 스트림으로만 검색 범위를 제한할 수 있습니다. 예를 들어 로그 그룹에 1,000개의 로그 스트림이 있지만 관련성이 있다고 판단되는 로그 스트림 3개만 찾아보고 싶다면 AWS CLI를 사용하여 해당 로그 그룹 내 3개의 로그 스트림으로만 검색을 제한할 수 있습니다.
- 더 짧고 세분화된 시간 범위를 사용하면 검색할 데이터 양을 줄이고 쿼리 속도를 높일 수 있습니다.

CloudWatch Logs에서 로그 데이터 보존 기간을 변경

기본적으로 CloudWatch Logs에서 로그 데이터는 무기한으로 저장됩니다. 하지만 로그 그룹에서 로그 데이터를 저장할 기간을 구성할 수 있습니다. 현재 보존 기간 설정보다 오래된 데이터는 모두 자동으로 삭제됩니다. 언제든지 로그 그룹별로 로그 보존 기간을 변경할 수 있습니다.

로그 보존 기간 설정을 변경하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 업데이트할 로그 그룹을 찾습니다.
4. 해당 로그 그룹의 이벤트 만료 시점 열에서 만기 없음 같은 현재 보존 기간 설정을 선택합니다.
5. 보존 기간 편집의 보존 기간에서 로그 보존 기간 값을 선택한 다음 확인을 선택합니다.

Amazon CloudWatch Logs에서 로그 그룹에 태그 지정

태그의 형태로 Amazon CloudWatch Logs에서 생성한 로그 그룹에 자체 메타데이터를 할당할 수 있습니다. 태그는 로그 그룹에 대해 정의된 키-값 페어입니다. 태그를 사용하면 간단하면서도 효과적으로 AWS 리소스를 관리하고 결제 데이터를 포함하여 데이터를 구성할 수 있습니다.

내용

- 태그 기본 사항 (p. 57)
- 태그 지정을 사용하여 비용 추적 (p. 58)
- 태그 제한 (p. 58)
- AWS CLI를 사용하여 로그 그룹에 태그 지정 (p. 58)
- CloudWatch Logs API를 사용하여 로그 그룹에 태그 지정 (p. 59)

태그 기본 사항

AWS CLI 또는 CloudWatch Logs API를 사용하여 다음 작업을 완료합니다.

- 생성 시 로그 그룹에 태그를 추가합니다.
- 기존 로그 그룹에 태그를 추가합니다.
- 로그 그룹에 대한 태그를 나열합니다.
- 로그 그룹에서 태그를 제거합니다.

태그를 사용하여 로그 그룹을 분류할 수 있습니다. 예를 들어 용도, 소유자 또는 환경을 기준별로 로그 그룹을 분류할 수 있습니다. 각 태그에 대해 키와 값이 정의되기 때문에 특정 요구를 충족하는 사용자 지정 범주 세트를 생성할 수 있습니다. 예를 들어, 태그 세트를 정의하여 소유자와 연관 애플리케이션에 따라 로그 그룹을 추적할 수 있습니다. 다음은 태그의 몇 가지 예제입니다.

- 프로젝트 프로젝트 이름
- Owner Name
- 용도 로드 테스트.
- 애플리케이션 애플리케이션 이름

- Environment 프로덕션

태그 지정을 사용하여 비용 추적

태그를 사용하여 AWS 비용을 분류 및 추적할 수 있습니다. 로그 그룹 같은 AWS 리소스에 태그를 적용할 때 AWS 비용 할당 보고서에는 태그별로 집계된 사용 내역 및 비용이 포함됩니다. 비즈니스 범주를 나타내는 태그(예: 비용 센터, 애플리케이션 이름 또는 소유자)를 적용하여 여러 서비스에 대한 비용을 정리할 수 있습니다. 자세한 내용은 사용자 지정 청구 보고서에 비용 할당 태그 사용 in the AWS Billing and Cost Management 사용 설명서.

태그 제한

태그에 적용되는 제한은 다음과 같습니다.

기본 제한

- 로그 그룹당 최대 태그 수는 50개입니다.
- 태그 키와 값은 대/소문자를 구분합니다.
- 삭제된 로그 그룹에 대해 태그를 변경하거나 편집할 수 없습니다.

태그 키 제한

- 각 태그 키는 고유해야 합니다. 이미 사용 중인 키를 가진 태그를 추가하면 기존 키-값 페어에 새 태그가 덮어쓰기 됩니다.
- 태그 키에 `aws:` 접두사는 사용하지 마십시오. 이 접두사는 AWS용으로 예약되어 있습니다. AWS는 이 접두사로 시작되는 태그를 생성하지만, 사용자는 이를 편집 또는 삭제할 수 없습니다.
- 태그 키의 길이는 유니코드 1~128자여야 합니다.
- 태그 키는 다음 문자로 구성되어야 합니다. 유니코드 문자, 숫자, 백인 공간 및 특수 문자: `_ . / = + - @`.

태그 값 제한

- 태그 값의 길이는 유니코드 0~225자여야 합니다.
- 태그 값은 공백 상태로 둘 수 있습니다. 그렇지 않으면 다음 문자로 구성해야 합니다. 유니코드 문자, 숫자, 백인 공간 및 다음 특수 문자: `_ . / = + - @`.

AWS CLI를 사용하여 로그 그룹에 태그 지정

AWS CLI를 사용하여 태그를 추가, 나열 및 제거할 수 있습니다. 예제는 다음 설명서를 참조하십시오.

create-log-group

로그 그룹을 생성합니다. 로그 그룹 생성 시 태그를 선택에 따라 추가할 수 있습니다.

tag-log-group

지정된 로그 그룹에 대한 태그를 추가 또는 업데이트합니다.

list-tags-log-group

지정된 로그 그룹에 대한 태그를 나열합니다.

untag-log-group

지정된 로그 그룹에 대한 태그를 제거합니다.

CloudWatch Logs API를 사용하여 로그 그룹에 태그 지정

CloudWatch Logs API를 사용하여 태그를 추가, 나열 및 제거할 수 있습니다. 예제는 다음 설명서를 참조하십시오.

CreateLogGroup

로그 그룹을 생성합니다. 로그 그룹 생성 시 태그를 선택에 따라 추가할 수 있습니다.

TagLogGroup

지정된 로그 그룹에 대한 태그를 추가 또는 업데이트합니다.

ListTagsLogGroup

지정된 로그 그룹에 대한 태그를 나열합니다.

UntagLogGroup

지정된 로그 그룹에 대한 태그를 제거합니다.

AWS KMS를 사용하여 CloudWatch Logs에서 로그 데이터를 암호화

로그 그룹 데이터는 항상 CloudWatch Logs에서 암호화됩니다. 선택적으로 이 암호화에 AWS AWS Key Management Service를 사용할 수 있습니다. 이렇게 하면 AWS KMS(AWS KMS) 고객 마스터 키(CMK)를 사용하여 암호화가 수행됩니다. 로그 그룹을 생성할 때나 로그 그룹이 존재하는 경우에는 CMK를 로그 그룹에 연결하면 로그 그룹 수준에서 AWS KMS를 사용한 암호화가 활성화됩니다.

Important

CloudWatch Logs에서는 이제 `kms:EncryptionContext:aws:logs:arn`을 키로 사용하고 로그 그룹의 ARN을 해당 키의 값으로 사용하여 암호화 컨텍스트를 지원합니다. CMK로 이미 암호화된 로그 그룹이 있고 단일 계정 및 로그 그룹에서 CMK를 사용하도록 제한하려면 IAM 정책에 조건을 포함하는 새 CMK를 할당해야 합니다. 자세한 정보는 KMS 키 및 암호화 컨텍스트 (p. 62) 단원을 참조하십시오.

CMK를 로그 그룹에 연결하고 나면 로그 데이터에서 새로 수집된 모든 데이터를 CMK를 사용해 암호화할 수 있습니다. 이 데이터는 보존 기간 전반에 걸쳐 암호화된 형식으로 저장됩니다. CloudWatch Logs는 요청이 발생할 때마다 이 데이터를 해독합니다. 암호화된 데이터 요청이 발생할 때마다 CloudWatch Logs는 CMK에 대한 권한을 가지고 있어야 합니다.

로그 그룹에서 CMK가 연결 해제되고 나면 CloudWatch Logs는 로그 그룹에서 새로 수집된 데이터에 대한 암호화를 중단합니다. 이전에 수집된 모든 데이터는 암호화를 유지합니다.

Important

CloudWatch Logs는 대칭 CMK만 지원합니다. 비대칭 CMK를 사용하여 로그 그룹의 데이터를 암호화하지 마십시오. 자세한 내용은 대칭 및 비대칭 키 사용을 참조하십시오.

Limits

- 다음 단계를 수행하려면 다음 권한이 있어야 합니다. `kms:CreateKey`, `kms:GetKeyPolicy`, 그리고 `kms:PutKeyPolicy`.
- 로그 그룹에서 CMK를 연결하거나 연결 해제하고 난 후 이러한 변경이 적용되기까지 최대 5분의 시간이 소요될 수 있습니다.

- 연결된 CMK에 대한 CloudWatch Logs 액세스를 취소하거나 연결된 CMK를 삭제한 경우에는 CloudWatch Logs에서 암호화된 데이터를 더 이상 검색할 수 없습니다.
- CloudWatch 콘솔을 사용하여 CMK를 로그 그룹에 연결할 수 없습니다.

단계 1. 생성 AWS KMS CMK

AWS KMS CMK를 생성하려면 아래 create-key 명령을 사용하십시오.

```
aws kms create-key
```

이 명령의 출력 화면에는 CMK의 키 ID와 Amazon 리소스 이름(ARN)이 포함됩니다. 다음은 예제 출력입니다.

```
{
    "KeyMetadata": {
        "KeyId": "6f815f63-e628-448c-8251-e40cb0d29f59",
        "Description": "",
        "Enabled": true,
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012"
    }
}
```

단계 2. CMK에 대한 권한 설정

기본적으로 모든 AWS KMS CMK는 비공개입니다. 리소스 소유자만 이를 사용하여 데이터를 암호화 및 해독할 수 있습니다. 그러나 리소스 소유자가 원한다면 다른 사용자 및 리소스에게 CMK에 대한 액세스 권한을 부여할 수 있습니다. 이 단계에서는 CMK를 사용할 보안 주체 권한을 CloudWatch 서비스에 제공합니다. 서비스 보안 주체는 CMK가 저장된 리전과 동일한 리전에 있어야 합니다.

가장 좋은 방법은 키 사용을 지정된 AWS 계정 또는 로그 그룹으로만 제한하는 것이 좋습니다.

먼저, 아래 get-key-policy 명령을 사용하여 CMK에 대한 기본 정책을 `policy.json`로 저장합니다.

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./policy.json
```

텍스트 편집기에서 `policy.json` 파일을 열고 다음 설명 중 하나에서 굵은 글꼴로 표시된 섹션을 추가합니다. 기존 설명과 새 설명을 쉼표로 구분합니다. 이러한 설명은 `Condition` 섹션을 사용하여 KMS 키의 보안을 강화합니다. 자세한 정보는 KMS 키 및 암호화 컨텍스트 (p. 62) 단원을 참조하십시오.

이 예제의 `Condition` 섹션에서는 키를 단일 로그 그룹 ARN으로 제한합니다.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
```

```
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:log-group:log-group-name"
                }
            }
        }
    ]
}
```

이 예제의 `Condition` 섹션에서는 KMS 사용을 지정된 계정으로 제한하지만 모든 로그 그룹에 사용할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:*"
                }
            }
        }
    ]
}
```

마지막으로 아래 put-key-policy 명령을 사용하여 업데이트된 정책을 추가합니다.

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://policy.json
```

단계 3. 로그 그룹을 CMK와 연결

로그 그룹을 생성할 때와 그 이후에 CMK를 로그 그룹에 연결할 수 있습니다.

로그 그룹에 이미 CMK가 연결되어 있는지 확인하려면 다음 describe-log-groups 명령을 사용합니다.

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

출력에 `kmsKeyId` 필드가 포함된 경우 로그 그룹은 해당 필드의 값에 대해 표시된 키와 연결됩니다.

생성 시 로그 그룹에 CMK를 연결하려면

다음과 같이 create-log-group 명령을 사용합니다.

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

기존 로그 그룹에 CMK를 연결하려면

다음과 같이 associate-kms-key 명령을 사용합니다.

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

단계 4. CMK에서 로그 그룹 연결 해제

로그 그룹에 연결된 CMK의 연결을 해제하려면 아래 disassociate-kms-key 명령을 사용하십시오.

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

KMS 키 및 암호화 컨텍스트

KMS 키와 암호화된 로그 그룹의 보안을 강화하기 위해 CloudWatch Logs에서는 이제 로그 그룹 ARN을 로그 데이터를 암호화하는 데 사용되는 암호화 컨텍스트의 일부로 배치합니다. 암호화 컨텍스트는 추가 인증 데이터로 사용되는 키-값 쌍의 집합입니다. 암호화 컨텍스트를 사용하면 IAM 정책 조건을 사용하여 AWS 계정 및 로그 그룹별로 KMS 키에 대한 액세스를 제한할 수 있습니다. 자세한 내용은 암호화 컨텍스트 and IAM JSON 정책 요소: Condition

암호화된 각 로그 그룹에 대해 서로 다른 CMK 키를 사용하는 것이 좋습니다.

이전에 암호화한 로그 그룹이 있고 해당 로그 그룹에만 작동하는 새 CMK를 사용하도록 로그 그룹을 변경하려는 경우 다음 단계를 수행합니다.

CMK를 해당 로그 그룹으로 제한하는 정책에 따라 사용하도록 암호화된 로그 그룹을 변환하려면

1. 다음 명령을 입력하여 로그 그룹의 현재 CMK의 ARN을 찾습니다.

   ```
   aws logs describe-log-groups
   ```

   출력에는 다음 줄이 포함됩니다. ARN을 기록해 둡니다. 7단계에서 사용해야 합니다.

   ```
   ...
   ```

```
"kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
cdef-0123-456789abcdef"
...
```

2. 다음 명령을 입력하여 새 CMK를 생성합니다.

```
aws kms create-key
```

3. 다음 명령을 입력하여 새 키의 정책을 `policy.json` 파일에 저장합니다.

```
aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
policy.json
```

4. 텍스트 편집기를 사용하여 `policy.json`을 열고 `Condition` 표현식을 정책에 추가합니다.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT-ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:REGION:ACCOUNT-
ID:log-
group:LOG-GROUP-NAME"
                }
            }
        }
    ]
}
```

5. 다음 명령을 입력하여 업데이트된 정책을 새 CMK에 추가합니다.

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. 다음 명령을 입력하여 정책을 로그 그룹과 연결합니다.

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

CloudWatch Logs은 이제 새 CMK를 사용하여 모든 새 데이터를 암호화합니다.

7. 그런 다음 이전 CMK에서 `Decrypt`를 제외한 모든 권한을 취소합니다. 먼저 다음 명령을 입력하여 이전 정책을 검색합니다.

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text > ./
policy.json
```

8. 텍스트 편집기를 사용하여 `policy.json`을 열고 `Action` 목록에서 `kms:Decrypt*`를 제외한 모든 값을 제거합니다.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::REGION:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt*"
            ],
            "Resource": "*"
        }
    ]
}
```

9. 다음 명령을 입력하여 업데이트된 정책을 이전 CMK에 추가합니다.

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

특정 AWS 서비스에서 로깅 활성화

일부 AWS 서비스에서 CloudWatch Logs에 로그를 보내려면 CloudWatch Logs에 로그를 보낼 서비스 권한을 부여하는 CloudWatch Logs 리소스 정책을 사용하거나 생성해야 합니다. 이 문제는 Amazon API Gateway, AWS Step Functions 및 Amazon Managed Streaming for Apache Kafka에 영향을 줄 수 있습니다.

이러한 서비스는 리소스 정책에서 전송하는 각 로그 그룹을 나열해야 하며 CloudWatch Logs 리소스 정책은 5120자로 제한됩니다. 따라서 많은 수의 로그 그룹에 로그를 전송하는 서비스가 이 제한에 도달할 수 있습니다.

이 문제를 완화하기 위해 CloudWatch Logs에서는 다른 AWS 서비스에서 사용하는 리소스 정책의 크기를 모니터링하고, 정책이 5120자의 크기 제한에 가까워진다는 것을 감지하면 CloudWatch Logs에서 해당 서비스에 대한 리소스 정책의 `/aws/vendedlogs/*`를 자동으로 활성화합니다. 그런 다음 이러한 서비스에 대해 `/aws/vendedlogs/`로 시작하는 이름을 가진 로그 그룹을 사용하여 로그 구독을 생성하기 시작할 수 있습니다.

필터를 사용하여 로그 이벤트에서 지표 생성

CloudWatch Logs 에이전트가 Amazon CloudWatch로 로그 데이터를 게시하기 시작하면 하나 이상의 지표 필터를 생성하여 로그 데이터에 대한 검색 및 필터링을 시작할 수 있습니다. 지표 필터는 CloudWatch Logs에 전송될 때 로그 데이터에서 검색할 단어와 패턴을 정의합니다. CloudWatch Logs는 이러한 지표 필터를 사용하여 로그 데이터를 그래프 처리를 하거나 경보를 설정할 수 있는 숫자 CloudWatch 지표로 전환합니다. 이러한 지표를 보거나 경보를 설정할 때 백분위수 통계를 포함하여 어떤 유형의 CloudWatch 통계도 사용이 가능합니다.

> Note
>
> 지표 값이 음수가 아닌 경우에만 지표의 백분위수 통계가 지원됩니다. 지표 필터를 음수를 보고할 수 있도록 설정하면 값이 음수인 경우에는 해당 지표의 백분위수 통계를 사용할 수 없습니다. 자세한 내용은 백분위수를 참조하십시오.

필터는 데이터를 소급해서 필터링하지 않습니다. 필터는 필터가 생성된 이후에 발생한 이벤트에 대한 지표 데이터 요소만 게시합니다. 필터링된 결과는 처음 50개 줄을 반환하는데, 필터링된 결과에 대한 타임스탬프가 지표 생성 시간보다 이른 경우에는 결과가 표시가 되지 않습니다.

내용

- Concepts (p. 65)
- 필터 및 패턴 구문 (p. 66)
- 지표 필터 생성 (p. 73)
- 지표 필터 나열 (p. 78)
- 지표 필터 삭제 (p. 79)

Concepts

각 지표 필터는 다음과 같은 키 요소들로 이루어져 있습니다.

필터 패턴

CloudWatch Logs가 각 로그 이벤트에서 데이터를 해석하는 방법을 심볼로 설명한 것입니다. 예를 들어 로그 항목에는 타임스탬프, IP 주소, 문자열 등이 포함될 수 있습니다. 이러한 패턴을 사용하여 로그 파일에서 검색할 내용을 지정합니다.

척도 이름

모니터링된 로그 정보가 게시되는 CloudWatch 지표의 이름입니다. 예를 들어 ErrorCount라는 지표에 게시할 수 있습니다.

지표 네임스페이스

새 CloudWatch 지표에 대한 대상 네임스페이스입니다.

지표 값

일치하는 로그가 발견될 때마다 지표에 게시하는 숫자 값입니다. 예를 들어, 특정 단어(예: "Error")의 출현 횟수를 계산할 경우 각 출현마다 이 값이 "1"이 됩니다. 전송된 바이트를 계산하는 경우 로그 이벤트에서 발견된 실제 바이트 수만큼 증가시킬 수 있습니다.

기본값

일치하는 로그가 발견되지 않은 기간 동안 지표 필터에 보고되는 값입니다. 이 값을 0으로 설정하면 모든 기간에서 데이터가 보고되어 데이터가 없는 기간 때문에 지표가 "불규칙"해지는 것을 방지할 수 있습니다.

필터 및 패턴 구문

지표 필터를 사용하여 로그 이벤트에서 일치하는 단어, 구문 또는 값을 검색할 수 있습니다. 지표 필터가 로그 이벤트에서 단어, 구문 또는 값 중 하나를 찾으면 CloudWatch 지표의 값을 증가시킬 수 있습니다. 예를 들어 로그 이벤트에서 지표 필터를 생성하여 ERROR라는 단어를 검색하고 출현 횟수를 계산할 수 있습니다.

지표 필터는 웹 요청에 대한 지연 시간과 같이 공백으로 구분된 로그 이벤트에서도 숫자 값을 추출할 수 있습니다. 다음 예제에서 로그로부터 추출된 실제 숫자 값만큼 지표 값을 증가시킬 수 있습니다.

또한 조건 연산자 및 와일드카드를 사용하여 정확히 일치되는 값을 찾을 수도 있습니다. 지표 필터를 생성하기 전에 CloudWatch 콘솔에서 검색 패턴을 테스트할 수 있습니다. 아래 단원에는 지표 필터 구문이 보다 자세하게 설명되어 있습니다.

로그 이벤트에서 일치하는 단어 검색

로그 이벤트에서 단어를 검색하려면 이 단어를 지표 필터 패턴으로 사용합니다. 지표 필터 패턴에 여러 개의 단어를 지정할 수 있지만, 모든 단어들은 일치 항목이 될 수 있도록 로그 이벤트에 나타나야 합니다. 지표 필터는 대소문자를 구분합니다.

영숫자나 밑줄 이외의 문자가 포함된 지표 필터 단어는 큰따옴표("") 안에 위치해야 합니다.

단어를 제외하려면 단어 앞에 마이너스 부호(-)를 붙입니다.

예 1 모든 것을 일치시킵니다

"" 필터 패턴은 모든 로그 이벤트와 일치합니다.

예 2 단일 용어

"ERROR" 필터 패턴은 다음과 같이 이 단어를 포함하는 로그 이벤트 메시지와 일치합니다.

- [ERROR] 치명적인 예외가 발생
- ERRORCODE로 종료: 1.

예 3 용어 포함 및 용어 제외

앞의 예제에서 필터 패턴을 "ERROR" - "Exiting"으로 변경하면 "Exiting with ERRORCODE: -1"라는 로그 이벤트 메시지가 제외가 됩니다.

예 4 여러 용어

"ERROR Exception" 필터 패턴은 다음과 같이 두 단어를 모두 포함하는 로그 이벤트 메시지와 일치합니다.

- [ERROR] IllegalArgumentException이 발생
- [ERROR] 처리되지 않은 예외

"Failed to process the request" 필터 패턴은 다음과 같이 모든 단어를 모두 포함하는 로그 이벤트 메시지와 일치합니다.

- [WARN] 요청 처리에 실패
- [ERROR] 계속할 수 없음: 요청을 처리하지 못했습니다.

OR 패턴 일치

OR 패턴 일치를 사용하여 텍스트 기반 필터에서 용어를 일치시킬 수 있습니다. OR에는 물음표(예: `?term`)를 사용합니다.

아래의 세 가지 로그 이벤트 사례를 살펴보십시오. `ERROR` 실시예 1 및 2 와 일치. `?ERROR ?WARN` 는 예제 1, 2 및 3과 일치합니다. 이 모두가 "ERROR" 또는 "WARN"이라는 단어를 포함합니다. `ERROR WARN` 은(는) 이 두 단어가 모두 포함된 유일한 항목이므로 예 1에 해당합니다. ERROR -WARN은 예시 2와 일치하는데, 예시 2는 ERROR를 포함하고 WARN을 포함하지 않는 문자열과 일치하기 때문입니다.

1. `ERROR WARN message`
2. `ERROR message`
3. `WARN message`

공백으로 구분되는 필터에서 OR 패턴 일치를 사용하여 용어를 일치시킬 수 있습니다. 공백으로 구분되는 필터를 사용하는 경우 `w1`은 로그 이벤트의 첫 번째 단어를, `w2`는 두 번째 단어를 나타내는 식입니다. 아래의 패턴 예제에서, ERROR가 첫 번째 단어이기 때문에 [w1=ERROR, w2]는 패턴 2와 일치하고, [w1=ERROR || w1=WARN, w2]는 패턴 2 및 3과 일치하고, [w1!=ERROR&&w1!=WARN, w2]는 ERROR와 WARN이 모두 있는 줄(패턴 1)과 일치합니다.

1. ERROR WARN 메시지
2. ERROR 메시지
3. WARN 메시지

공백으로 JSON 필터에서 OR 패턴 일치를 사용하여 용어를 일치시킬 수 있습니다. 아래의 패턴 예제에서, {$.foo = bar}는 패턴 1과 일치하고, {$.foo = baz }는 패턴 2와 일치하고, {$.foo = bar || $.foo = baz }는 패턴 1 및 2와 일치합니다.

1. {"foo": "bar"}
2. {"foo": "baz"}

JSON 로그 이벤트에서 일치하는 단어 검색

JSON 로그 이벤트에서 값을 추출할 수 있습니다. JSON 로그 이벤트에서 값을 추출하려면 문자열 기반의 지표 필터를 생성해야 합니다. 유효숫자 표기법이 포함된 문자열은 지원되지 않습니다. JSON 로그 이벤트 데이터의 항목들은 지표 필터와 정확하게 일치해야 합니다. 다음을 나타내기 위해 JSON 로그 이벤트에 지표 필터를 생성하고 싶을 수 있습니다.

- 특정 이벤트가 발생합니다. 예를 들어 eventName은 "UpdateTrail"입니다.
- IP는 알려진 서브넷 밖에 있습니다. 예를 들어 sourceIPAddress는 알려진 서브넷 범위에 있지 않습니다.
- 기타 조건들이 두 개 이상 결합되면 true입니다. 예를 들어 eventName은 "UpdateTrail"이고 recipientAccountId는 123456789012입니다.

지표 필터를 사용하여 JSON 로그 이벤트에서 값을 추출

지표 필터를 사용하여 JSON 로그 이벤트에서 값을 추출할 수 있습니다. 지표 필터는 수신 로그를 확인하여 로그 데이터에서 일치가 발견되면 숫자 값을 수정합니다. 지표 필터를 생성할 때 로그에서 일치하는 텍스트가 발견될 때마다 개수만 증가시킬 수도 있고 로그로부터 숫자 값을 추출하여 지표 값을 이 값만큼 증가시킬 수도 있습니다.

지표 필터를 사용한 JSON 단어 일치

JSON 로그 이벤트에 대한 지표 필터 구문은 다음 형식을 사용합니다.

```
{ SELECTOR EQUALITY_OPERATOR STRING }
```

지표 필터를 중괄호({ })로 묶어서 이것이 JSON 표현식임을 나타내야 합니다. 지표 필터에는 다음 요소들이 포함되어 있습니다.

SELECTOR

확인할 JSON 속성을 지정합니다. 속성 선택기는 JSON의 루트를 나타내기 위해 항상 달러 문자($)로 시작됩니다. 속성 선택기는 '-' 및 '_' 문자도 지원하는 영숫자 문자열입니다. 어레이 요소들은 [NUMBER] 구문으로 표시가 되며 하나의 속성을 따라야 합니다. 예를 들면 $.eventId, $.users[0], $.users[0].id, $.requestParameters.instanceId와 같습니다.

EQUALITY_OPERATOR

또는 입니다.

STRING

따옴표가 있거나 없는 문자열. '*' 와일드카드 문자를 사용하여 검색 단어의 텍스트나 검색 단어 앞뒤의 텍스트가 일치하는 것을 검색할 수 있습니다. 예를 들어 *Event는 PutEvent 및 GetEvent와 일치하게 됩니다. Event*는 EventId 및 EventName과 일치하게 됩니다. Ev*ent는 실제 문자열 Ev*ent하고만 일치하게 됩니다. 영숫자 문자로만 이루어진 문자열에는 따옴표를 붙일 필요가 없습니다. 유니코드와 '@,' '$,' '\,' 같은 다른 문자들을 가진 문자열이 유효하려면 큰 따옴표로 묶어야 합니다.

JSON 지표 필터 예제

다음은 JSON의 한 예입니다.

```
{
  "eventType": "UpdateTrail",
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {
         "name": "a",
         "id": 1
       },
       {
         "name": "b",
         "id": 2
       }
  ],
  "SomeObject": null,
  "ThisFlag": true
}
```

다음 필터들이 일치합니다.

```
{ $.eventType = "UpdateTrail" }
```

이벤트 유형이 UpdateTrail인 경우에 대한 필터.

```
{ $.sourceIPAddress != 123.123.* }
```

IP 주소가 서브넷 123.123 접두사 밖에 있는 경우에 대한 필터.

```
{ $.arrayKey[0] = "value" }
```

arrayKey의 첫 항목이 "value"인 경우에 대한 필터. arrayKey가 어레이가 아니면 false가 됩니다.

```
{ $.objectList[1].id = 2 }
```

objectList의 두 번째 항목이 id = 2라는 속성을 가지고 있는 경우에 대한 필터. objectList가 어레이가 아니면 false가 됩니다. objectList의 항목들이 개체가 아니거나 id 속성을 가지고 있지 않으면 false가 됩니다.

```
{ $.SomeObject IS NULL }
```

SomeObject가 null로 설정 중인 경우에 대한 필터. 지정된 개체가 null로 설정된 경우에만 true가 됩니다.

```
{ $.SomeOtherObject NOT EXISTS }
```

SomeOtherObject가 존재하지 않는 경우에 대한 필터. 지정된 개체가 로그 데이터에 존재하지 않는 경우에만 true가 됩니다.

```
{ $.ThisFlag IS TRUE }
```

ThisFlag가 TRUE인 경우에 대한 필터. FALSE 값을 확인하는 부울 필터에도 적용됩니다.

JSON 복합 조건식

OR (||) 및 AND (&&)를 사용하여 여러 조건식을 하나의 복합 표현식으로 결합할 수 있습니다. 괄호가 허용되며 구문은 () > && > ||라는 연산의 표준 순서를 따릅니다.

```
{
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
         "email": "Jane.Doe@example.com"
        }
    ],
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Examples

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

위의 JSON과 일치합니다.

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

위의 JSON과 일치하지 않습니다.

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = nonmatch &&
 $.actions[2] = nomatch }
```

위의 JSON과 일치합니다.

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = nonmatch) &&
 $.actions[2] = nomatch }
```

위의 JSON과 일치하지 않습니다.

JSON 특별 고려 사항

SELECTOR는 JSON에서 값 노드(문자열이나 숫자)를 가리켜야 합니다. 어레이나 개체를 가리킬 경우에는 로그 형식이 해당 필터와 일치하지 않기 때문에 필터가 적용되지 않습니다. 예를 들어 .{$.users = 1} 및 {$.users != 1} 모두 사용자가 어레이인 로그 이벤트와 일치하지 않게 됩니다.

```
{
  "users": [1, 2, 3]
}
```

숫자 비교식

지표 필터 구문은 숫자 비교식에서 정밀 매칭을 지원합니다. <, >, >=, <=, =, != 등의 숫자 비교식이 지원됩니다.

숫자 필터는 다음에 대한 구문을 가집니다.

```
{ SELECTOR NUMERIC_OPERATOR NUMBER }
```

지표 필터를 중괄호({ })로 묶어서 이것이 JSON 표현식임을 나타내야 합니다. 지표 필터에는 다음 요소들이 포함되어 있습니다.

SELECTOR

확인할 JSON 속성을 지정합니다. 속성 선택기는 JSON의 루트를 나타내기 위해 항상 달러 문자($)로 시작됩니다. 속성 선택기는 '-' 및 '_' 문자도 지원하는 영숫자 문자열입니다. 어레이 요소들은 [NUMBER] 구문으로 표시가 되며 하나의 속성을 따라야 합니다. $.latency, $.numbers[0], $.errorCode, $.processes[4].averageRuntime 등이 그 예입니다.

NUMERIC_OPERATOR

=, !=, <, >, <= 또는 >= 중 하나일 수 있습니다.

숫자

+ 또는 - 기호 옵션이 있는 정수, + 또는 - 기호 옵션이 있는 10진수, 유효숫자 표기법을 따르는 숫자(+ 또는 - 기호 옵션이 있는 정수 또는 10진수, 'e'가 뒤에 따라 나오는 숫자, + 또는 - 기호 옵션이 있는 정수가 뒤에 따라 나오는 숫자).

예제:

```
{ $.latency >= 500 }
{ $.numbers[0] < 10e3 }
{ $.numbers[0] < 10e-3 }
{ $.processes[4].averageRuntime <= 55.5 }
{ $.errorCode = 400 }
{ $.errorCode != 500 }
{ $.latency > +1000 }
```

지표 필터를 사용하여 공백으로 구분된 로그 이벤트에서 값을 추출

지표 필터를 사용하여 공백으로 구분된 로그 이벤트에서 값을 추출할 수 있습니다. 대괄호 [] 또는 큰 따옴표 ("")로 묶인 문자들은 단일 필드로 처리가 됩니다. 예: .

```
127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] "GET /apache_pb.gif HTTP/1.0" 200 1534
127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] "GET /apache_pb.gif HTTP/1.0" 500 5324
127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] "GET /apache_pb.gif HTTP/1.0" 200 4355
```

공백으로 구분된 이벤트를 구문 분석하는 지표 필터 패턴을 지정하려면 이름이 쉼표로 구분이 되고 전체 패턴이 대괄호로 묶여 있도록 필드를 지정해야 합니다. 예를 들면 [ip, user, username, timestamp, request, status_code, bytes]와 같이 말입니다.

필드 수를 모르는 경우에는 줄임표(...)를 이용해 간편 알림을 사용할 수 있습니다. 예: .

```
[..., status_code, bytes]
[ip, user, ..., status_code, bytes]
[ip, user, ...]
```

모든 조건이 일치하는 로그 이벤트만 필터와 일치가 되도록 하기 위해 필드에 조건을 추가할 수도 있습니다. 예: .

```
[ip, user, username, timestamp, request, status_code, bytes > 1000]
[ip, user, username, timestamp, request, status_code = 200, bytes]
[ip, user, username, timestamp, request, status_code = 4*, bytes]
[ip, user, username, timestamp, request = *html*, status_code = 4*, bytes]
```

다음 예제에 표시된 대로 &&를 논리 AND 연산자로 사용하고 ||를 논리 OR 연산자로 사용할 수 있습니다.

```
[ip, user, username, timestamp, request, status_code = 4* && bytes > 1000]
[ip, user, username, timestamp, request, status_code = 403 || status_code = 404, bytes]
```

CloudWatch Logs는 문자열과 숫자 조건 필드를 모두 지원합니다. 문자열 필드에서는 = 또는 != 연산자를 별표(*)와 함께 사용할 수 있습니다.

숫자 필드에서는 >, <, >=, <=, = 및 != 연산자를 사용할 수 있습니다.

공백으로 구분된 필터를 사용하는 경우에는 추출된 필드가 공백으로 구분된 필드의 이름(필터에 표현)과 각 필드의 값에 매핑됩니다. 공백으로 구분된 필터를 사용하지 않는 경우에는 필드가 비게 됩니다.

필터 예제:

```
[..., request=*.html*, status_code=4*,]
```

필터를 위한 로그 이벤트 예제:

```
127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /index.html HTTP/1.0\" 404 1534
```

로그 이벤트 및 필터 패턴을 위해 추출된 필드:

```
{
   "$status_code": "404",
   "$request": "GET /products/index.html HTTP/1.0",
   "$7": "1534",
   "$4": "10/Oct/2000:13:25:15 -0700",
   "$3": "frank",
   "$2": "-",
   "$1": "127.0.0.1"
}
```

일치가 발견될 경우 지표 값 변경 방법 설정

지표 필터는 로그 이벤트에서 일치하는 단어, 구문 또는 값 중 하나를 발견할 경우 지표 값에 지정된 값만큼 CloudWatch 지표에서 개수를 증가시킵니다. 지표 값은 집계되며 1분마다 보고됩니다.

1분 기간 동안 로그가 수집되었지만 일치가 발견되지 않은 경우 기본값(있는 경우)으로 지정된 값이 보고됩니다. 하지만 1분 기간 동안 로그 이벤트가 수집되지 않은 경우에는 아무 값도 보고되지 않습니다.

기본값을 지정할 경우 값이 0이라 해도 데이터가 보다 자주 보고되도록 하므로, 일차가 발견되지 않을 경우 지표가 불규칙해지는 것을 방지할 수 있습니다.

예를 들어 1분마다 두 개의 레코드를 게시하는 로그 그룹이 있고, 지표 값은 1, 기본값은 0이라고 가정해 보겠습니다. 처음 1분 동안 두 로그 레코드 모두에서 일치가 발견될 경우 이 기간의 지표 값은 2입니다. 두 번째 1분 동안 게시된 로그 레코드에서 일치가 발견되지 않을 경우 두 로그 레코드에 기본값 0이 사용되며 이 기간의 지표 값은 0입니다.

기본값을 지정하지 않을 경우 패턴 일치가 발견되지 않은 기간 동안에는 데이터가 보고되지 않습니다.

로그 항목에서 발견된 수치 값 게시

단순히 로그에서 발견된 일치 항목의 수를 계산하는 대신 지표 필터를 사용하여 로그에서 발견된 수치 값을 기반으로 값을 게시할 수 있습니다. 다음 절차는 JSON 요청 `metricFilter: { $.latency = * }` `metricValue: $.latency`에서 발견된 지연 시간이 포함된 지표를 게시하는 방법을 보여줍니다.

JSON 요청에서 지연 시간이 포함된 지표를 게시하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. `Actions`, 지표 필터 생성을 선택합니다.
4. 필터 패턴에 **`{ $.latency = * }`**를 입력하고 다음을 선택합니다.
5. 지표 이름에 myMetric을 입력합니다.
6. 지표 값에 **`$.latency`**를 입력합니다.
7. 기본값에 0을 입력하고 다음을 선택합니다. 기본값을 지정하면 필터와 일치하는 로그 이벤트가 없는 기간에도 데이터가 보고됩니다. 이렇게 하면 로그가 수집되지만 필터와 일치하지 않을 때 지표가 불규칙해지거나 누락되는 것을 방지할 수 있습니다.
8. 지표 필터 생성을 선택합니다.

아래 로그 이벤트는 필터 생성에 뒤이어 myMetric 지표에 50이라는 값을 게시하게 됩니다.

```
{
```

```
"latency": 50,
"requestType": "GET"
}
```

지표 필터 생성

다음 예제에서는 지표 필터를 생성하는 방법을 보여 줍니다.

예제:

- 예: 로그 이벤트 수 (p. 73)
- 예: 용어의 카운트 발생 (p. 74)
- 예: HTTP 404 코드 계산 (p. 75)
- 예: HTTP 4xx 코드 개수 (p. 76)
- 예: Apache Log에서 필드 추출 (p. 77)

예: 로그 이벤트 수

가장 간단한 유형의 로그 이벤트 모니터링은 발생하는 로그 이벤트의 수를 계산하는 것입니다. 모든 이벤트의 수를 유지하거나 "하트비트" 스타일 모니터를 생성하거나 단순히 지표 필터 생성을 연습하기 위해 계산을 원할 수 있습니다.

다음 CLI 예제에서는 MyNamespace라는 CloudWatch 네임스페이스에서 EventCount라는 지표를 생성하기 위해 MyAppAccessCount라는 지표 필터가 MyApp/access.log라는 로그 그룹에 적용됩니다. 이 필터는 모든 로그 이벤트 콘텐츠와 일치하며 지표를 "1"씩 늘리도록 구성되어 있습니다.

CloudWatch 콘솔을 사용하여 지표 필터를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 로그 그룹의 이름을 선택합니다.
4. `Actions`, 지표 필터 생성을 선택합니다.
5. 필터 패턴 및 테스트할 로그 데이터 선택을 비워 둡니다.
6. 다음을 선택한 후 필터 이름에 **`EventCount`**를 입력합니다.
7. 지표 세부 정보의 지표 네임스페이스에 **`MyNameSpace`**를 입력합니다.
8. 지표 이름에 **`MyAppEventCount`**를 입력합니다.
9. 지표 값이 1인지 확인합니다. 이는 모든 로그 이벤트에 대해 개수가 1씩 증가하도록 지정합니다.
10. 기본값에 0을 입력하고 다음을 선택합니다. 기본값을 지정할 경우 로그 이벤트가 발생하지 않는 기간에도 데이터가 보고되므로 때때로 데이터가 존재하지 않아 지표가 불규칙해지는 것을 방지할 수 있습니다.
11. 지표 필터 생성을 선택합니다.

AWS CLI를 사용하여 지표 필터를 생성하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
  --filter-pattern "" \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

모든 이벤트 데이터를 게재하여 이 새로운 정책을 테스트할 수 있습니다. 데이터 요소가 MyAppAccessEventCount 지표에 게시되어 있어야 합니다.

AWS CLI를 사용하여 이벤트 데이터를 게시하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

예: 용어의 카운트 발생

로그 이벤트에는 연산의 성공 또는 실패 횟수 같이 계산이 필요한 중요한 메시지가 종종 포함됩니다. 예를 들어 지정된 연산이 실패하면 오류가 발생하고 로그 파일에 오류가 기록될 수 있습니다. 이들 항목을 모니터링하여 오류의 트렌드를 이해하고 싶을 수 있습니다.

아래 예제에서는 Error라는 단어를 모니터링하기 위한 지표 필터가 생성됩니다. 정책이 생성되어 MyApp/message.log라는 로그 그룹에 추가되었습니다. CloudWatch Logs는 Error가 포함된 모든 이벤트가 "1"이라는 값을 갖는 MyApp/message.log에서 CloudWatch의 사용자 지정 지표인 ErrorCount에 데이터 요소를 게시합니다. Error라는 단어가 포함된 이벤트가 없으면 값 0이 게시됩니다. CloudWatch 콘솔에서 이 데이터를 그래픽 처리할 때는 반드시 Sum 통계를 사용해야 합니다.

지표 필터를 생성한 후 CloudWatch 콘솔에서 지표를 볼 수 있습니다. 보려는 지표를 선택할 때 로그 그룹 이름과 일치하는 지표 네임스페이스를 선택합니다. 자세한 내용은 사용 가능한 지표 보기를 참조하십시오.

CloudWatch 콘솔을 사용하여 지표 필터를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 로그 그룹의 이름을 선택합니다.
4. 작업, 지표 필터 생성을 선택합니다.
5. 필터 패턴에 **Error**를 입력합니다.

   Note

   필터 패턴의 모든 항목들은 대소문자를 구분합니다.
6. 필터 패턴을 테스트하려면 패턴 테스트를 선택합니다.
7. 다음을 선택한 후 지표 할당 페이지에서 필터 이름에 **MyAppErrorCount**를 입력합니다.
8. 지표 세부 정보의 지표 네임스페이스에 MyNameSpace를 입력합니다.
9. 지표 이름에 ErrorCount를 입력합니다.
10. 지표 값이 1인지 확인합니다. 이는 "Error"를 포함하는 모든 로그 이벤트에 대해 개수가 1씩 증가하도록 지정합니다.
11. 기본값에 0을 입력하고 다음을 선택합니다.
12. 지표 필터 생성을 선택합니다.

AWS CLI를 사용하여 지표 필터를 생성하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
```

```
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

메시지에 "Error"라는 단어가 포함된 이벤트를 게재하여 이 새로운 정책을 테스트할 수 있습니다.

AWS CLI를 사용하여 이벤트를 게재하려면

명령 프롬프트에서 다음 명령을 실행합니다. 패턴은 대소문자를 구분한다는 점을 유의하십시오.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
    timestamp=1394793528000,message="This message also contains an Error"
```

예: HTTP 404 코드 계산

CloudWatch Logs를 사용하여 Apache 서버가 발견되지 않은 페이지에 대한 응답 코드인 HTTP 404 응답을 반환하는 횟수를 모니터링할 수 있습니다. 사이트 방문자들이 원하는 리소스를 찾지 못하는 빈도를 파악하기 위해 모니터링을 원할 수 있습니다. 로그 레코드는 각 로그 이벤트(사이트 방문)에 대해 다음 정보를 포함하도록 구성되었다고 가정합니다.

- 요청자 IP 주소
- RFC 1413 ID
- Username
- Timestamp
- 요청된 리소스와 프로토콜이 포함된 요청 메소드
- 요청할 HTTP 응답 코드
- 요청 시 전송되는 바이트

예를 들어 다음과 같은 형태일 수 있습니다.

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

다음 예제에서 알 수 있듯이, HTTP 404 오류에서 해당 구조의 이벤트와 매칭을 시도하는 규칙을 지정할 수 있습니다.

CloudWatch 콘솔을 사용하여 지표 필터를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. `Actions`, 지표 필터 생성을 선택합니다.
4. 필터 패턴에 **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**를 입력합니다.
5. 필터 패턴을 테스트하려면 패턴 테스트를 선택합니다.
6. 다음을 선택하고 필터 이름에 HTTP404Errors를 입력합니다.
7. 지표 세부 정보의 지표 네임스페이스에 **`MyNameSpace`**를 입력합니다.
8. 지표 이름에 **`ApacheNotFoundErrorCount`**를 입력합니다.
9. 지표 값이 1인지 확인합니다. 이는 모든 404 오류 이벤트에 대해 개수가 1씩 증가하도록 지정합니다.
10. 기본값에 0을 입력하고 다음을 선택합니다.

11. 지표 필터 생성을 선택합니다.

AWS CLI를 사용하여 지표 필터를 생성하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
      metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

이 예제에서는 왼쪽/오른쪽 대괄호, 큰 따옴표, 문자열 404 같은 리터럴 문자가 사용되었습니다. 패턴은 모니터링하려는 로그 이벤트의 전체 로그 이벤트 메시지와 일치해야 합니다.

describe-metric-filters 명령을 사용하여 지표 필터가 생성되었는지 확인할 수 있습니다. 다음과 유사한 출력 화면이 표시되어야 합니다.

```
aws logs describe-metric-filters --log-group-name MyApp/access.log

{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404, size]"
        }
    ]
}
```

수동으로 몇 가지 이벤트를 게재할 수 있습니다.

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

이러한 샘플 로그 이벤트를 게재하고 나면 그 즉시 CloudWatch 콘솔에서 ApacheNotFoundErrorCount로 명명된 지표를 검색할 수 있습니다.

예: HTTP 4xx 코드 개수

이전 예제에서와 마찬가지로 사용자는 웹 서비스 액세스 로그를 모니터링하고 HTTP 응답 코드 수준을 모니터링하기 원할 수 있습니다. 예를 들어 HTTP 400 수준 오류를 모두 모니터링하고 싶을 수 있습니다. 그러나 모든 반환 코드에 새로운 지표 필터를 지정하고 싶지 않을 수 있습니다.

다음 예제는 예: HTTP 404 코드 계산 (p. 75) 예제에서 Apache 액세스 로그 형식을 사용하여 액세스 로그로부터 400 수준의 모든 HTTP 코드 응답을 포함하는 지표를 생성하는 방법을 보여줍니다.

CloudWatch 콘솔을 사용하여 지표 필터를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. Apache 서버의 로그 그룹 이름을 선택합니다.
4. `Actions`, 지표 필터 생성을 선택합니다.
5. 대상 필터 이름, 입력 **`HTTP4xxErrors`**.
6. 대상 필터 패턴, 입력 **`[ip, id, user, timestamp, request, status_code=4*, size]`**.
7. 필터 패턴을 테스트하려면 패턴 테스트를 선택합니다.
8. 선택 다음, 그리고 필터 이름, 유형 **`HTTP4xxErrors`**.
9. 아래 메트릭 세부 정보, 메트릭 네임스페이스, 입력 **`MyNameSpace`**.
10. 대상 메트릭 이름, 입력 HTTP4xx오류.
11. 대상 메트릭 값, 1을 입력합니다. 이는 4xx 오류를 포함하는 모든 로그 이벤트에 대해 개수가 1씩 증가하도록 지정합니다.
12. 대상 기본값 0을 입력한 다음 다음.
13. 지표 필터 생성을 선택합니다.

AWS CLI를 사용하여 지표 필터를 생성하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
  metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

PUT 이벤트 호출에서 다음 데이터를 사용하여 이 규칙을 테스트할 수 있습니다. 이전 예제에서 모니터링 규칙을 제거하지 않았다면 서로 다른 두 개의 지표가 생성됩니다.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

예: Apache Log에서 필드 추출

때로는 수를 계산하는 대신 지표 값으로 개별 로그 이벤트 내의 값을 사용하는 것이 도움이 됩니다. 이 예제는 Apache 웹 서버에서 전송된 바이트를 측정하는 지표를 생성하기 위해 추출 규칙을 만드는 방법을 보여줍니다

이 추출 규칙은 로그 이벤트의 필드 7개와 일치합니다. 7번째 일치된 토큰의 값이 지표 값이 됩니다. 추출 규칙의 `metricValue` 필드에 토큰에 대한 참조가 "$7"로 표시됩니다.

CloudWatch 콘솔을 사용하여 지표 필터를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. Apache 서버의 로그 그룹 이름을 선택합니다.
4. `Actions`, 지표 필터 생성을 선택합니다.

5. 대상 필터 패턴, 입력 **[ip, id, user, timestamp, request, status_code, size]**.
6. 필터 패턴을 테스트하려면 패턴 테스트를 선택합니다.
7. 선택 다음, 그리고 필터 이름, 유형 **size**.
8. 아래 메트릭 세부 정보, 메트릭 네임스페이스, 입력 **MyNameSpace**. 이는 새로운 네임스페이스입니다. 새 생성 을(를) 선택합니다.
9. 대상 메트릭 이름, 입력 **BytesTransferred**
10. 대상 메트릭 값, 입력 **$size**.
11. 대상 기본값 0을 입력한 다음 다음.
12. 지표 필터 생성을 선택합니다.

AWS CLI를 사용하여 지표 필터를 생성하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue=$size,defaultValue=0
```

PUT 로그 이벤트 호출에서 다음 데이터를 사용하여 이 규칙을 테스트할 수 있습니다. 이전 예제에서 모니터링 규칙을 제거하지 않았다면 서로 다른 두 개의 지표가 생성됩니다.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

지표 필터 나열

로그 그룹에 속한 모든 지표 필터를 나열할 수 있습니다.

CloudWatch 콘솔을 사용하여 지표 필터를 나열하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 콘텐츠 창에서 로그 그룹 목록의 지표 필터 열에서 필터 수를 선택합니다.

   로그 그룹 > Filters for(필터 대상) 화면에 해당 로그 그룹과 연관된 모든 지표 필터들이 나열됩니다.

AWS CLI를 사용하여 지표 필터를 나열하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

다음은 예제 출력입니다.

```
{
```

```
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404, size]"
        }
    ]
}
```

지표 필터 삭제

이름과 소속 로그 그룹으로 정책을 식별할 수 있습니다.

CloudWatch 콘솔을 사용하여 지표 필터를 삭제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 콘텐츠 창의 지표 필터 열에서 해당되는 지표 필터를 선택합니다.
4. 로그 지표 필터 정의 화면에서 지표 필터로 필터 삭제를 선택합니다.
5. 확인 메시지가 나타나면 Yes, Delete(예, 삭제)를 선택합니다.

AWS CLI를 사용하여 지표 필터를 삭제하려면

명령 프롬프트에서 다음 명령을 실행합니다.

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
 --filter-name MyFilterName
```

구독을 통한 로그 데이터 실시간 처리

구독을 사용하여 에서 로그 이벤트의 실시간 피드에 액세스할 수 있습니다. CloudWatch Logs 다른 서비스에 전달되도록 Amazon Kinesis 스트림, Amazon Kinesis Data Firehose 스트림, 또는 AWS Lambda 사용자 지정 처리, 분석 또는 다른 시스템으로 로드하기 위한. 로그 이벤트가 수신 서비스로 전송되면 Gzip 형식으로 Base64 인코딩 및 압축됩니다.

로그 이벤트 구독을 시작하려면 Kinesis 스트림에서 이벤트를 전달합니다. 구독 필터는 AWS 리소스에 제공되는 로그 이벤트와 일치하는 로그 이벤트를 전송할 대상에 대한 정보를 필터링하는 데 사용할 필터 패턴을 정의합니다.

각 로그 그룹에는 최대 2개의 구독 필터가 연결되어 있을 수 있습니다.

> **Note**
>
> 대상 서비스가 제한 예외 또는 재시도 가능 서비스 예외(예: HTTP 5xx)와 같은 재시도 가능 오류를 반환하는 경우 CloudWatch Logs 은(는) 최대 24시간 동안 배송을 재시도합니다. CloudWatch Logs 은(는) 오류가 다음과 같이 재시도할 수 없는 오류인 경우 재배송을 시도하지 않습니다. AccessDeniedException 또는 ResourceNotFoundException.

또한 CloudWatch Logs는 구독 서비스로의 로그 이벤트 전달에 대한 CloudWatch 지표를 생성합니다. 자세한 내용은 Amazon CloudWatch Logs 지표 및 차원을 참조하십시오.

내용

- Concepts (p. 80)
- CloudWatch Logs 구독 필터 사용 (p. 81)
- 구독과 교차 계정 로그 데이터 공유 (p. 91)

Concepts

각 구독 필터는 다음과 같은 키 요소들로 이루어져 있습니다.

log_group_name

구독 필터에 연결되는 로그 그룹입니다. 이 로그 그룹에 업로드된 모든 로그 이벤트에는 구독 필터가 적용되고, 필터와 일치하는 로그 이벤트는 일치하는 로그 이벤트를 받는 대상 서비스로 전달됩니다.

필터 패턴

CloudWatch Logs가 각 로그 이벤트의 데이터를 해석하는 방법과 함께 대상 AWS 리소스에 제공되는 이벤트를 제한하는 필터링 표현식을 심볼로 설명한 것입니다. 필터 패턴 구문에 대한 자세한 내용은 필터 및 패턴 구문 (p. 66) 단원을 참조하십시오.

destination arn

구독 피드 대상으로 사용하고자 하는 Kinesis 스트림이나 Kinesis Data Firehose 스트림, Lambda 함수의 Amazon 리소스 이름(ARN)입니다.

역할 ARN

안 IAM 지원한 역할 CloudWatch Logs 선택한 대상 에 데이터를 입력하는 데 필요한 권한을 지정합니다. 이 역할은 Lambda 대상에는 필요하지 않습니다. CloudWatch Logs가 Lambda 함수 자체에 대한 액세스 제어 설정에 필요한 권한을 얻을 수 있기 때문입니다.

배포

대상이 Amazon Kinesis 스트림일 때 해당 대상으로 로그 데이터를 배포하는 데 사용되는 방법입니다. 기본적으로 로그 스트림에 따라 로그 데이터가 그룹화됩니다. 보다 균등한 배포를 위해 로그 데이터를 무작위로 그룹화할 수 있습니다.

CloudWatch Logs 구독 필터 사용

Kinesis, Lambda 또는 Kinesis Data Firehose에서 구독 필터를 사용할 수 있습니다. 구독 필터를 통해 수신 서비스로 전송되는 로그는 Gzip 형식으로 Base64 인코딩 및 압축됩니다.

예제:

- 예 1: 구독 필터 Kinesis (p. 81)
- 예 2: 구독 필터 AWS Lambda (p. 84)
- 실시예 3: 구독 필터 Amazon Kinesis Data Firehose (p. 87)

예 1: 구독 필터 Kinesis

다음 예제는 기록된 모든 활동이 "RootAccess"라는 Kinesis 스트림에 제공된 "Root" AWS 자격 증명을 통해 이루어지도록 AWS CloudTrail 이벤트가 포함된 로그 그룹에 구독 필터를 연결합니다. 보내는 방법에 대한 자세한 내용은 AWS CloudTrail 이벤트 대상 CloudWatch Logs, 참조 전송 중 CloudTrail 이벤트 대상 CloudWatch Logs 에서 AWS CloudTrail User Guide.

Note

Kinesis 스트림을 생성하기 전에 생성할 로그 데이터의 볼륨을 계산합니다. 이 볼륨을 처리하기에 충분한 샤드로 Kinesis 스트림을 생성해야 합니다. 스트림에 샤드가 충분하지 않을 경우에는 로그 스트림에서 병목 현상이 발생합니다. Kinesis 스트림 볼륨 제한에 대한 자세한 내용은 Amazon Kinesis 데이터 스트림 제한을 참조하십시오.

Kinesis에 대한 구독 필터를 생성하려면

1. 다음 명령을 사용하여 대상 Kinesis 스트림을 생성합니다.

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. Kinesis 스트림이 활성 상태가 될 때까지 기다립니다(1~2분 정도 소요). 다음을 사용할 수 있습니다. Kinesis 설명-스트림 명령을 사용하여 StreamDescription.스트림상태 호텔. 또한 StreamDescription.스트레아mARN 다음 단계에서 필요할 수 있습니다.

```
aws kinesis describe-stream --stream-name "RootAccess"
```

다음은 예제 출력입니다.

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
```

```
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "49551135218688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Kinesis 스트림으로 데이터를 입력하기 위해 필요한 CloudWatch Logs 권한을 부여하는 IAM 역할을 생성합니다. 먼저 신뢰 정책을 파일로 생성해야 합니다(예: `~/TrustPolicyForCWL.json`). 텍스트 편집기를 사용하여 이 정책을 생성하십시오. IAM 콘솔을 사용하여 정책을 생성하지 마십시오.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

4. create-role 명령을 사용하여 신뢰 정책 파일을 지정하는 IAM 역할을 생성합니다. 이후 단계에서 필요할 수 있기 때문에 반환된 Role.Arn 값도 적어 둡니다.

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document file://
~/TrustPolicyForCWL.json
```

다음은 출력 의 예입니다.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. CloudWatch Logs가 계정에서 수행할 수 있는 작업을 정의하는 권한 정책을 생성합니다. 먼저 권한 정책을 파일로 생성합니다(예: `~/PermissionsForCWL.json`). 텍스트 편집기를 사용하여 이 정책을 생성하십시오. IAM 콘솔을 사용하여 정책을 생성하지 마십시오.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
  ]
```

```
}
```

6. 다음 put-role-policy 명령을 사용하여 권한 정책을 역할에 연결합니다.

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL.json
```

7. Kinesis 스트림이 Active 상태에 있고 IAM 역할을 생성하고 나면 CloudWatch Logs 구독 필터를 생성할 수 있습니다. 그 즉시 구독 필터는 실시간으로 선택한 로그 그룹에서 Kinesis 스트림으로 로그 데이터를 이동시키기 시작합니다.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. 구독 필터를 설정하고 나면 CloudWatch Logs는 들어오는 모든 로그 이벤트 중에서 필터 패턴과 일치하는 이벤트를 Kinesis 스트림으로 전달합니다. Kinesis 샤드 반복자를 확보하고 Kinesis get-records 명령을 사용해 몇몇 Kinesis 레코드를 가져와서 이러한 작업이 수행되고 있는지 확인할 수 있습니다.

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id shardId-000000000000
 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiK2OSh0uP"
```

이 호출을 몇 차례 반복해야 Kinesis가 데이터 반환을 시작할 수 있습니다.

레코드 어레이에서 응답을 확인할 수 있습니다. Kinesis 레코드의 데이터 속성은 gzip 형식으로 인코딩 및 압축된 Base64입니다. 다음 Unix 명령을 사용하여 명령줄에서 원시 데이터를 검토할 수 있습니다.

```
echo -n "<Content of Data>" | base64 -d | zcat
```

디코딩 및 압축 해제된 Base64 데이터는 다음 구조를 가진 JSON으로 포맷됩니다.

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail",
    "logStream": "111111111111_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
```

```
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

위의 데이터 구조에서 키 요소는 다음과 같습니다.

owner

원본 로그 데이터의 AWS 계정 ID입니다.

logGroup

원본 로그 데이터의 로그 그룹 이름입니다.

logStream

원본 로그 데이터의 로그 스트림 이름입니다.

subscriptionFilters

원본 로그 데이터과 일치한 구독 필터 이름 목록입니다.

messageType

데이터 메시지는 "DATA_MESSAGE" 유형을 사용합니다. 때로 CloudWatch Logs는 주로 대상이 도달 가능한지 확인하기 위한 목적으로 "CONTROL_MESSAGE" 유형을 가진 Kinesis 레코드를 출력할 수 있습니다.

logEvents

로그 이벤트 레코드 어레이 형태로 표현되는 실제 로그 데이터입니다. "id" 속성은 모든 로그 이벤터의 고유 식별자입니다.

예 2: 구독 필터 AWS Lambda

이 예제에서는 AWS Lambda 함수에 로그 데이터를 전송하는 CloudWatch Logs 구독 필터를 생성합니다.

Note

Lambda 함수를 생성하기 전에 생성할 로그 데이터의 볼륨을 계산합니다. 이 볼륨을 처리할 수 있는 함수를 생성해야 합니다. 함수에 볼륨이 충분하지 않을 경우에는 로그 스트림에서 병목 현상이 발생합니다. Lambda 제한에 대한 자세한 내용은 AWS Lambda 제한을 참조하십시오.

Lambda에 대한 구독 필터를 생성하려면

1. AWS Lambda 함수를 생성합니다.

   Lambda 실행 역할이 설정되었는지 확인합니다. 자세한 내용은 을 참조하십시오. 2.2단계: IAM 역할(실행 역할) 만들기 에서 AWS Lambda Developer Guide.

2. 다음 콘텐츠로 텍스트 편집기를 열고 helloWorld.js라는 파일을 생성합니다.

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString('ascii'));
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. 파일 압축 helloWorld.js하고 이름과 함께 저장합니다. helloWorld.zip.
4. 역할이 첫 단계에서 설정한 Lambda 실행 역할인 경우에는 다음 명령을 사용하십시오.

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. CloudWatch Logs에 함수를 실행할 권한을 부여합니다. 다음 명령을 사용하여 자리 표시자 계정을 자체 계정으로, 자리 표시자 그룹을 처리할 로그 그룹으로 바꿉니다.

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.region.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. 다음 명령을 사용하여 구독 필터를 생성하여 자리 표시자 계정을 자체 계정으로, 자리 표시자 그룹을 처리할 로그 그룹으로 바꿉니다.

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. (선택 사항) 샘플 로그 이벤트를 사용하여 테스트합니다. 명령 프롬프트에서 다음 명령을 실행하여 구독된 스트림으로 간단한 로그 메시지를 보냅니다.

   Lambda 함수의 출력을 보려면 /aws/lambda/helloworld에 출력이 표시되는 Lambda 함수를 탐색합니다.

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --log-
events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple Lambda
 Test\"}]"
```

   Lambda; 어레이에서 응답을 확인할 수 있습니다. Lambda 레코드의 Data 속성은 gzip 형식으로 인코딩 및 압축된 Base64입니다. Lambda가 수신하는 실제 페이로드는 { "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} } 형식을 따릅니다. 다음 Unix 명령을 사용하여 명령 줄에서 원시 데이터를 검토할 수 있습니다.

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

디코딩 및 압축 해제된 Base64 데이터는 다음 구조를 가진 JSON으로 포맷됩니다.

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

위의 데이터 구조에서 키 요소는 다음과 같습니다.

owner

원본 로그 데이터의 AWS 계정 ID입니다.

logGroup

원본 로그 데이터의 로그 그룹 이름입니다.

logStream

원본 로그 데이터의 로그 스트림 이름입니다.

subscriptionFilters

원본 로그 데이터과 일치한 구독 필터 이름 목록입니다.

messageType

데이터 메시지는 "DATA_MESSAGE" 유형을 사용합니다. 때로 CloudWatch Logs는 주로 대상이 도달 가능한지 확인하기 위한 목적으로 "CONTROL_MESSAGE" 유형을 가진 Lambda 레코드를 출력할 수 있습니다.

logEvents

로그 이벤트 레코드 어레이 형태로 표현되는 실제 로그 데이터입니다. "id" 속성은 모든 로그 이벤터의 고유 식별자입니다.

실시예 3: 구독 필터 Amazon Kinesis Data Firehose

이 예제에서는 들어오는 로그 이벤트 중에서 정의된 필터와 일치하는 이벤트를 Amazon Kinesis Data Firehose 전송 스트림에 전송하는 CloudWatch Logs 구독을 생성합니다. CloudWatch Logs에서 Amazon Kinesis Data Firehose으로 전송된 데이터는 이미 gzip 6 수준의 압축이 되었기 때문에 Kinesis Data Firehose 전송 스트림 내에서 압축을 사용할 필요가 없습니다.

> **Note**
>
> Kinesis Data Firehose 스트림을 생성하기 전에 생성할 로그 데이터의 볼륨을 계산합니다. 이 볼륨을 처리할 수 있는 Kinesis Data Firehose 스트림을 생성해야 합니다. 스트림이 볼륨을 처리할 수 없는 경우에는 로그 스트림에서 병목 현상이 발생합니다. Kinesis Data Firehose 스트림 볼륨 제한에 대한 자세한 내용은 Amazon Kinesis Data Firehose 데이터 제한을 참조하십시오.

Kinesis Data Firehose에 대한 구독 필터를 생성하려면

1. Amazon Simple Storage Service(Amazon S3) 버킷을 만듭니다. CloudWatch Logs를 위해 특별히 생성한 버킷을 사용하는 것이 좋습니다. 그러나 기존 버킷을 사용하고 싶으면 2단계로 건너뛸 수 있습니다.

   다음 명령을 실행하여 자리 표시자 리전을 사용하고자 하는 리전으로 바꿉니다.

   ```
   aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
    LocationConstraint=region
   ```

   다음은 예제 출력입니다.

   ```
   {
       "Location": "/my-bucket"
   }
   ```

2. Amazon S3 버킷으로 데이터를 입력하기 위해 필요한 Amazon Kinesis Data Firehose 권한을 부여하는 IAM 역할을 생성합니다.

   자세한 내용은 을 참조하십시오. 액세스 제어 Amazon Kinesis Data Firehose 에서 Amazon Kinesis Data Firehose 개발자 안내서.

   먼저 텍스트 편집기를 사용하여 다음과 같이 `~/TrustPolicyForFirehose.json` 파일로 신뢰 정책을 생성하여 account-id를 AWS 계정 ID로 바꿉니다.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "firehose.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": { "StringEquals": { "sts:ExternalId":"account-id" } }
     }
   }
   ```

3. create-role 명령을 사용하여 신뢰 정책 파일을 지정하는 IAM 역할을 생성합니다. 이후 단계에서 필요할 수 있기 때문에 반환된 Role.Arn 값을 적어 둡니다.

   ```
   aws iam create-role \
         --role-name FirehosetoS3Role \
         --assume-role-policy-document file://~/TrustPolicyForFirehose.json

   {
       "Role": {
           "AssumeRolePolicyDocument": {
   ```

```
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Kinesis Data Firehose가 계정에서 수행할 수 있는 작업을 정의하는 권한 정책을 생성합니다. 먼저 텍스트 편집기를 사용하여 권한 정책을 `~/PermissionsForFirehose.json` 파일로 생성합니다.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. 다음 put-role-policy 명령을 사용하여 권한 정책을 역할에 연결합니다.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. 대상 생성 Kinesis Data Firehose 다음과 같이 배달 스트림을 제공하고, 다음에 대한 자리 표시자 값을 대체합니다. RoleARN 및 BucketARN 역할 및 버킷과 함께 ARNs 사용자가 만든 항목:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

Kinesis Data Firehose는 제공된 Amazon S3 객체에서 YYYY/MM/DD/HH UTC 시간 형식의 접두사를 자동으로 사용합니다. 시간 형식 접두사 앞에 추가할 또 다른 접두사를 지정할 수 있습니다. 슬래시(/)로 끝난 접두사는 Amazon S3 버킷에서 자리 표시자로 표시됩니다.

7. 스트림이 활성 상태가 될 때까지 기다립니다(몇 분 소요). 다음을 사용할 수 있습니다. Kinesis Data Firehose 설명-전달-스트림 명령을 사용하여 DeliveryStreamDescription.배달 스트림 상태 호텔. 또한 DeliveryStreamDescription.배달스트레아mARN 다음 단계에서 필요할 수 있습니다.

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
```

```
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Kinesis Data Firehose 전송 스트림에 데이터를 입력하기 위해 필요한 CloudWatch Logs 권한을 부여하는 IAM 역할을 생성합니다. 먼저 텍스트 편집기를 사용하여 신뢰 정책을 `~/TrustPolicyForCWL.json` 파일로 생성합니다.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

9. create-role 명령을 사용하여 신뢰 정책 파일을 지정하는 IAM 역할을 생성합니다. 이후 단계에서 필요할 수 있기 때문에 반환된 Role.Arn 값을 적어 둡니다.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
```

```
}
```

10. CloudWatch Logs가 계정에서 수행할 수 있는 작업을 정의하는 권한 정책을 생성합니다. 먼저 텍스트 편집기를 사용하여 권한 정책을 파일로 생성합니다(예: `~/PermissionsForCWL.json`).

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:123456789012:*"]
      }
    ]
}
```

11. put-role-policy 명령을 사용하여 권한 정책을 역할에 연결합니다.

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name Permissions-
Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Amazon Kinesis Data Firehose 전송 스트림이 활성 상태에 있고 IAM 역할을 생성하고 나면 CloudWatch Logs 구독 필터를 생성할 수 있습니다. 그 즉시 구독 필터는 실시간으로 선택한 로그 그룹에서 Amazon Kinesis Data Firehose 전송 스트림으로 로그 데이터를 이동시키기 시작합니다.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-delivery-
stream" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. 구독 필터를 설정하고 나면 CloudWatch Logs는 들어오는 모든 로그 이벤트 중에서 필터 패턴과 일치하는 이벤트를 Amazon Kinesis Data Firehose 전송 스트림으로 전달합니다. Amazon Kinesis Data Firehose 전송 스트림에 설정된 시간 버퍼 간격에 따라 Amazon S3에 데이터가 나타나기 시작합니다. 충분한 시간이 지나고 나면 Amazon S3 버킷을 확인하여 데이터를 확인할 수 있습니다.

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
```

```
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250' testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Amazon S3 객체의 데이터는 gzip 형식으로 압축됩니다. 다음 Unix 명령을 사용하여 명령줄에서 원시 데이터를 검토할 수 있습니다.

```
zcat testfile.gz
```

구독과 교차 계정 로그 데이터 공유

다른 AWS 계정의 소유자와 협력해서 Amazon Kinesis 스트림 같은 AWS 리소스에서 로그 이벤트를 수신할 수 있습니다(이를 교차 계정 데이터 공유라고 함). 예를 들어 중앙의 Amazon Kinesis 스트림에서 이 로그 이벤트 데이터를 읽어 사용자 지정 처리 및 분석을 수행할 수 있습니다. 사용자 지정 처리는 특히 다수의 계정에서 데이터를 공동 사용 및 분석할 때 유용합니다. 예를 들어 기업의 정보 보안 그룹은 실시간 침입 탐지나 이상 행동에 대한 데이터를 분석하고자 중앙 처리를 위해 연동된 프로덕션 로그를 수집하여 모든 부서의 계정에 대해 감사를 실시할 수 있습니다. 이러한 계정에서 실시간 이벤트 스트림을 수집해서 정보 보안 그룹에 제공하면, 보안 그룹은 Kinesis를 사용해 기존의 보안 분석 시스템에 데이터를 연결할 수 있습니다.

Kinesis 스트림은 현재 교차 계정 구독용 대상으로 지원되는 유일한 리소스입니다.

계정에서 로그 데이터를 공유하려면 로그 데이터 발신자 및 수신자를 설정해야 합니다.

- 로그 데이터 발신자 - 수신자로부터 대상 정보를 얻고 지정된 대상으로 로그 이벤트를 전송할 준비가 되었음을 CloudWatch Logs에게 알립니다. — 이 단원의 나머지 부분에서는 로그 데이터 발신자가 가상의 AWS 계정 번호 111111111111로 표시됩니다.
- 로그 데이터 수신자—는 Kinesis 스트림을 캡슐화하는 대상을 설정하고 CloudWatch Logs에게 로그 데이터 수신을 원한다는 것을 알립니다. 그런 다음 수신자는 대상에 대한 정보를 발신자와 공유합니다. 이 단원의 나머지 부분에서는 로그 데이터 수신자가 가상의 AWS 계정 번호 999999999999로 표시됩니다.

교차 계정 사용자로부터 로그 이벤트 수신을 시작하기 위해 로그 데이터 수신자는 먼저 CloudWatch Logs 대상을 생성합니다. 각 대상은 다음과 같은 키 요소로 이루어져 있습니다.

대상 이름

생성하고자 하는 대상의 이름입니다.

대상 ARN

구독 피드의 대상으로 사용하고자 하는 AWS 함수의 Amazon 리소스 이름(ARN)입니다.

역할 ARN

선택한 Kinesis 스트림으로 데이터를 입력하기 위해 필요한 권한을 CloudWatch Logs에 부여하는 AWS Identity and Access Management(IAM) 역할입니다.

액세스 정책

대상에 쓰기 권한이 허용된 사용자들에게 적용되는 IAM 정책 문서(JSON 형식, IAM 정책 문법을 사용해 작성)입니다.

로그 그룹과 대상은 동일한 AWS 리전에 있어야 합니다. 하지만 대상이 가리키는 AWS 리소스는 다른 리전에 위치할 수 있습니다.

주제

- 대상 생성 (p. 92)
- 구독 필터 생성 (p. 95)
- 로그 이벤트 이동 검사 (p. 95)
- 런타임 시 대상 멤버십 수정 (p. 96)

대상 생성

Important

이 절차의 모든 단계는 로그 데이터 수신자 계정에서 수행해야 합니다.

이 예제에서 로그 데이터 수신자 계정은 AWS 계정 ID가 999999999999이고, 로그 데이터 발신자 AWS 계정은 ID가 111111111111입니다.

이 예에서는 Kinesis 스트림 호출됨 RecipientStream, 그리고 이 역할을 통해 CloudWatch Logs 를 클릭합니다().

대상을 생성하려면

1. Kinesis에서 대상 스트림을 생성합니다. 명령 프롬프트에서 다음과 같이 입력합니다.

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Kinesis 스트림이 활성 상태가 될 때까지 기다립니다. 다음을 사용할 수 있습니다. aws 키네시스 설명-스트림 명령을 사용하여 StreamDescription.스트림상태 호텔. 또한, StreamDescription.스트레아mARN 값이 전달되기 때문에 CloudWatch Logs 이후:

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702l8EXAMPLE"
        }
      }
    ]
  }
}
```

스트림이 활성 상태가 될 때까지 1~2분 정도 기다려야 할 수 있습니다.

3. Kinesis 스트림으로 데이터를 입력하기 위해 필요한 CloudWatch Logs 권한을 부여하는 IAM 역할을 생성합니다. 먼저 신뢰 정책을 ~/TrustPolicyForCWL.json 파일로 생성해야 합니다. 텍스트 편집기를 사용하여 이 정책 파일을 생성하고 IAM 콘솔은 사용하지 마십시오.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

4. aws iam create-role 명령을 사용하여 신뢰 정책 파일을 지정하는 IAM 역할을 생성합니다. CloudWatch Logs의 이후 단계에서 필요할 수 있기 때문에 반환된 Role.Arn 값을 적어둡니다.

```
aws iam create-role \
    --role-name CWLtoKinesisRole \
    --assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. CloudWatch Logs가 계정에서 수행할 수 있는 작업을 정의하는 권한 정책을 생성합니다. 먼저 텍스트 편집기를 사용하여 권한 정책을 ~/PermissionsForCWL.json 파일로 생성합니다.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. aws iam put-role-policy 명령을 사용하여 권한 정책을 역할에 연결합니다.

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Kinesis 스트림이 활성 상태이고 IAM 역할을 생성하고 나면 CloudWatch Logs 대상을 생성할 수 있습니다.

a. 이 단계를 수행했다고 액세스 정책이 대상에 연결되는 것은 아니며, 대상 생성을 완료하기 위한 두 단계 중 첫 번째 단계일 뿐입니다. 다음 사항을 기록해 두십시오. DestinationArn 페이로드에 반환되는

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

b. 7a 단계를 완료한 후 로그 데이터 수신자 계정에서 액세스 정책을 대상과 연결합니다. 이 정책은 로그 데이터 발신자 계정(111111111111)이 로그 데이터 수신자 계정(999999999999)의 대상에 액세스하도록 허용합니다. 텍스트 편집기를 사용하여 이 정책을 ~/AccessPolicy.json 파일에 저장할 수 있습니다.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

Note

다중 계정이 이 대상에 로그를 보내는 경우, 각 발신자 계정은 정책에 별도로 나열되어야 합니다. 이 정책은 `*`를 `Principal`로 지정하는 것을 지원하지 않거나 `aws:PrincipalOrgId` 전역 키 사용을 지원하지 않습니다.

c. 이렇게 하면 대상에 대해 쓰기 액세스 권한을 가진 사람을 정의하는 정책이 생성됩니다. 이 정책은 대상 액세스를 위한 logs:PutSubscriptionFilter 작업을 지정하게 됩니다. 교차 계정 사용자는 PutSubscriptionFilter 로그 이벤트를 대상으로 보내는 작업:

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

이 액세스 정책은 ID 111111111111인 AWS 계정의 사용자가 PutSubscriptionFilter ARN:aws:logs를 사용하여 대상에 대해 다음을 수행합니다.*region*:999999999999:목적지:테스트목적지. 다른 사용자의 통화 시도 PutSubscriptionFilter 이 목적지에 대한 은(는) 거부됩니다.

액세스 정책에 대한 사용자 권한의 유효성을 검사하려면 다음을 참조하십시오. 정책 유효성 검사기 사용 에서 IAM 사용 설명서.

구독 필터 생성

대상을 생성하고 나면 로그 데이터 수신자 계정에서 다른 AWS 계정이 로그 이벤트를 동일한 대상으로 전송할 수 있도록 대상 ARN(arn:aws:logs:us-east-1:999999999999:destination:testDestination)을 이들과 공유할 수 있습니다. 그러면 이러한 다른 전송 계정 사용자는 이 대상에 해당되는 로그 그룹에 대한 구독 필터를 생성합니다. 그 즉시 구독 필터는 실시간으로 선택한 로그 그룹에서 지정된 스트림으로 로그 데이터를 이동시키기 시작합니다.

다음 예제에서는 구독 필터가 보내는 계정에서 생성됩니다. 이 필터는 AWS CloudTrail 이벤트가 포함된 로그 그룹과 연결되는데, 이는 "Root" AWS 자격 증명으로 이루어진 모든 기록된 활동을 앞서 생성한 대상에 제공하기 위한 것입니다. 이 대상에서는 "RecipientStream"이라는 Kinesis 스트림을 캡슐화합니다. 보내는 방법에 대한 자세한 내용은 AWS CloudTrail 이벤트 대상 CloudWatch Logs, 참조 전송 중 CloudTrail 이벤트 대상 CloudWatch Logs 에서 AWS CloudTrail User Guide.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

로그 그룹과 대상은 동일한 AWS 리전에 있어야 합니다. 하지만 대상은 다른 리전에 있는 Kinesis 스트림과 같은 AWS 리소스를 가리킬 수 있습니다.

로그 이벤트 이동 검사

구독 필터를 만든 후 CloudWatch Logs 필터 패턴과 일치하는 모든 수신 로그 이벤트를 Kinesis 대상 스트림 내에 캡슐화된 "RecipientStream". 대상 소유자는 aws kinesis get-shard-iterator 명령을 사용해 Kinesis 샤드를 확보하고, aws kinesis get-records 명령을 사용해 몇몇 Kinesis 레코드를 가져와서 이러한 작업이 수행되고 있는지 확인할 수 있습니다.

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> get-records 명령어 반환을 몇 차례 반복해야 Kinesis가 데이터 반환을 시작할 수 있습니다

Kinesis 레코드 어레이에서 응답을 확인할 수 있습니다. Kinesis 레코드의 데이터 속성은 gzip 형식으로 압축된 다음 Base64로 인코딩됩니다. 다음 Unix 명령을 사용하여 명령줄에서 원시 데이터를 검토할 수 있습니다.

```
echo -n "<Content of Data>" | base64 -d | zcat
```

디코딩 및 압축 해제된 Base64 데이터는 다음 구조를 가진 JSON으로 포맷됩니다.

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail",
    "logStream": "111111111111_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        }
    ]
}
```

이 데이터 구조에서 키 요소는 다음과 같습니다.

owner

원본 로그 데이터의 AWS 계정 ID입니다.

logGroup

원본 로그 데이터의 로그 그룹 이름입니다.

logStream

원본 로그 데이터의 로그 스트림 이름입니다.

subscriptionFilters

원본 로그 데이터과 일치한 구독 필터 이름 목록입니다.

messageType

데이터 메시지는 "DATA_MESSAGE" 유형을 사용합니다. 때로 CloudWatch Logs는 주로 대상이 도달 가능한지 확인하기 위한 목적으로 "CONTROL_MESSAGE" 유형을 가진 Kinesis 레코드를 출력할 수 있습니다.

logEvents

로그 이벤트 레코드 어레이 형태로 표현되는 실제 로그 데이터입니다. ID 속성은 모든 로그 이벤터의 고유 식별자입니다.

런타임 시 대상 멤버십 수정

소유한 대상에서 몇몇 사용자의 멤버십을 추가 또는 제거해야 하는 상황에 직면할 수 있습니다. 다음을 사용할 수 있습니다. PutDestinationPolicy 새 액세스 정책을 사용하여 대상에 대한 작업을 수행합니다. 다음 예제에서는 이전에 추가한 계정인 111111111111이 추가적인 로그 데이터 전송을 중단하고 계정 222222222222가 활성화됩니다.

1. 현재 대상과 연결된 정책을 가져옵니다. testDestination 그리고 다음 사항을 메모하십시오 AccessPolicy:

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisRole",
     "DestinationArn": "arn:aws:logs:region:222222222222:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:222222222222:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:123456789012:destination:testDestination\"}] }"
   }
 ]
}
```

2. 계정 111111111111이 중단되고 계정 222222222222가 활성화되었음을 반영하도록 이 정책을 업데이트합니다. ~/NewAccessPolicy.json 파일에 이 정책을 저장합니다.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:222222222222:destination:testDestination"
    }
  ]
}
```

3. 통화 PutDestinationPolicy 에 정의된 정책을 연결하려면 NewAccessPolicy.j손 다음 대상에 대한 파일:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

이렇게 하면 계정 ID 111111111111에서 로그 이벤트가 비활성화됩니다. 계정 ID에서 이벤트 기록 222222222222번으로 전화하십시오. 계좌 소유자와 함께 목적지까지 이동하기 시작함 222222222222 번으로 전화하십시오. 를 사용하여 구독 필터 만들기 PutSubscriptionFilter.

로그 직접 보내기 Amazon S3 또는 Kinesis Data Firehose

일부 AWS 서비스에서 직접 로그를 Amazon S3 또는 Kinesis Data Firehose. 이렇게 하면 로그에 대한 주요 요구 사항이 이러한 서비스 중 하나에서 저장 또는 처리되는 경우 로그를 생성하는 서비스를 쉽게 사용하여 해당 로그를 로 직접 보낼 수 있습니다. Amazon S3 또는 Kinesis Data Firehose 추가 인프라를 설정하지 않아도 됩니다.

Amazon S3에 게시되는 로그는 사용자가 지정한 기존 버킷에 게시됩니다. 지정된 버킷에 5분마다 하나 이상의 로그 파일이 생성됩니다.

로그가 에 직접 게시되는 경우에도 Amazon S3 또는 Kinesis Data Firehose, CloudWatch Logs 요금이 부과됩니다. 자세한 내용은 을 참조하십시오. Vended 로그 에서 로그 탭 Amazon CloudWatch 가격 책정.

다음 로그는 Amazon S3에 직접 게시할 수 있습니다.

- VPC 흐름 로그 자세한 내용은 을 참조하십시오. 플로우 로그 게시 대상 Amazon S3 에서 Amazon VPC 사용 설명서.
- AWS 글로벌 액셀러레이터 흐름 로그. 자세한 내용은 을 참조하십시오. 플로우 로그 게시 대상 Amazon S3 에서 AWS 글로벌 액셀러레이터 개발자 가이드.

다음 로그는 Kinesis Data Firehose에 직접 게시할 수 있습니다.

- Amazon Managed Streaming for Apache Kafka 로그 자세한 내용은 을 참조하십시오. 로깅 에서 Amazon Managed Streaming for Apache Kafka 개발자 안내서.

Amazon S3로 로그 데이터 내보내기

로그 그룹에서 Amazon S3 버킷으로 로그 데이터를 내보내서 이 데이터를 사용자 지정 처리 및 분석에 사용하거나 다른 시스템에 로드할 수 있습니다.

로그 데이터 내보내기 Amazon S3 암호화되는 버킷 AWS KMS 은(는) 지원되지 않습니다.

내보내기 프로세스를 시작하려면 S3 버킷을 생성해서 내보낸 로그 데이터를 저장해야 합니다. Amazon S3 버킷에 내보낸 파일을 저장하고 Amazon S3 수명 주기 규칙을 정의하여 내보낸 파일을 자동으로 보관하거나 삭제할 수 있습니다.

AES-256으로 암호화된 S3 버킷으로의 내보내기가 지원됩니다. SSE-KMS로 암호화된 S3 버킷으로의 내보내기는 지원되지 않습니다. 자세한 내용은 S3 버킷의 기본 암호화를 활성화하려면 어떻게 해야 합니까?를 참조하십시오.

여러 로그 그룹이나 시간 범위에서 내보낸 로그를 동일한 S3 버킷으로 내보낼 수 있습니다. 각 내보내기 작업에 대해 로그 데이터를 분리하려면 내보낸 모든 객체에서 Amazon S3 키 접두사로 사용될 접두사를 지정할 수 있습니다.

로그 데이터가 내보내기가 가능한 상태가 되는 데는 최대 12시간이 걸릴 수 있습니다. 로그 데이터를 실시간에 가깝게 분석하려면 대신에 CloudWatch Logs Insights로 로그 데이터 분석 (p. 33) 또는 구독을 통한 로그 데이터 실시간 처리 (p. 80) 단원을 참조하십시오.

> **Note**
>
> 2019년 2월 15일부터 Amazon S3으로 내보내기 기능을 사용하려면 호출자가 대상 버킷에 대한 `s3:PutObject` 액세스 권한을 보유해야 합니다.

내용

- Concepts (p. 99)
- 콘솔을 사용하여 Amazon S3으로 로그 데이터 내보내기 (p. 100)
- AWS CLI를 사용하여 Amazon S3으로 로그 데이터 내보내기 (p. 102)

Concepts

내보내기를 시작하기 전에 다음 개념을 익힙니다.

log_group_name

내보내기 작업과 연관된 로그 그룹의 이름. 이 로그 그룹의 로그 데이터는 지정된 Amazon S3 버킷으로 내보내집니다.

(타임스탬프)부터

1970년 1월 1일 1:1970 00:00:00 UTC 이후 경과된 시간(밀리초)로 표현되는 필수 타임스탬프. 이 시간이 내보내기 된 이후에 수집된 로그 그룹의 모든 로그 이벤트.

(타임스탬프)까지

1970년 1월 1일 1:1970 00:00:00 UTC 이후 경과된 시간(밀리초)로 표현되는 필수 타임스탬프. 이 시간이 내보내기 되기 전에 수집된 로그 그룹의 모든 로그 이벤트.

대상 버킷

내보내기 작업과 연관된 Amazon S3 버킷의 이름. 이 버킷은 지정된 로그 그룹에서 로그 데이터를 내보내는 데 사용됩니다.

대상 접두사

내보낸 모든 개체에 대한 S3 키 접두사로 사용되는 속성 옵션. 이 옵션은 버킷에 폴더 같은 조직을 생성하는 데 도움이 됩니다.

콘솔을 사용하여 Amazon S3으로 로그 데이터 내보내기

다음 예제에서는 Amazon CloudWatch 콘솔을 사용하여 이름이 `my-log-group`인 Amazon CloudWatch Logs 로그 그룹의 모든 데이터를 이름이 `my-exported-logs`인 Amazon S3 버킷으로 내보냅니다.

로그 데이터 내보내기 Amazon S3 암호화되는 버킷 AWS KMS 은(는) 지원되지 않습니다.

단계 1. Amazon S3 버킷 생성

CloudWatch Logs를 위해 특별히 생성한 버킷을 사용하는 것이 좋습니다. 그러나 기존 버킷을 사용하고 싶으면 2단계로 건너뛸 수 있습니다.

Note

Amazon S3 버킷은 내보내려는 로그 데이터와 동일한 리전에 상주해야 합니다. CloudWatch Logs는 다른 리전에 있는 Amazon S3 버킷으로의 데이터 내보내기를 지원하지 않습니다.

Amazon S3 버킷을 생성하려면

1. https://console.aws.amazon.com/s3/에서 Amazon S3 콘솔을 엽니다.
2. 필요한 경우 리전을 변경합니다. 탐색 모음에서 CloudWatch Logs가 상주하는 리전을 선택합니다.
3. [Create Bucket]을 선택합니다.
4. 버킷 이름에서 버킷의 이름을 입력합니다.
5. 리전에서 CloudWatch Logs 데이터가 상주하는 리전을 선택합니다.
6. 생성을 선택합니다.

단계 2. 생성 IAM 에 대한 전체 액세스 권한이 있는 사용자 Amazon S3 및 CloudWatch Logs

다음 단계에서는 필수 권한으로 IAM 사용자를 생성합니다.

필수 IAM 사용자를 생성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. Users(사용자), Add user(사용자 추가)를 선택합니다.
3. 다음과 같은 사용자 이름을 입력합니다.*`CWLExportUser`*.
4. Programmatic access(프로그래밍 방식 액세스)와 AWS Management Console 액세스를 모두 선택합니다.
5. Autogenerated password(자동 생성된 암호) 또는 Custom password(사용자 지정 암호)를 선택합니다.
6. 다음을 선택합니다. 권한
7. Attach existing policies directly(기존 정책 직접 연결)를 선택하고 `AmazonS3FullAccess` 및 `CloudWatchLogsFullAccess` 정책을 사용자에게 연결합니다. 검색 상자를 사용하여 정책을 찾을 수 있습니다.
8. 다음을 선택합니다. 태그, 다음: 검토, 그리고 사용자 생성.

단계 3. 다음에 대한 권한 설정 Amazon S3 버킷

기본적으로 모든 Amazon S3 버킷 및 객체는 비공개입니다. 리소스 소유자(버킷을 생성한 AWS 계정)만 해당 버킷과 버킷에 포함된 객체에 액세스할 수 있습니다. 그러나 리소스 소유자는 액세스 정책을 작성하여 다른 리소스 및 사용자에게 액세스 권한을 부여할 수 있습니다.

정책을 설정할 때 의도한 로그 스트림만 버킷으로 내보내도록 임의로 생성되는 문자열을 버킷의 접두사로 포함시킬 것을 권장합니다.

Amazon S3 버킷에 대한 권한을 설정하려면

1. Amazon S3 콘솔에서 1단계에서 생성한 버킷을 선택합니다.
2. 권한, 버킷 정책을 선택합니다.
3. 버킷 정책 편집기에서 다음 정책 중 하나를 추가합니다. `my-exported-logs`를 S3 버킷의 이름으로 변경하고 `random-string`을 임의로 생성된 문자열로 변경합니다. 보안 주체에 올바른 리전 엔드포인트를 지정하십시오.

   - 버킷이 계정에 있으면 다음 정책을 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      }
    ]
}
```

   - 버킷이 다른 계정에 있으면 대신 다음 정책을 사용합니다. 이전 단계에서 생성한 IAM 사용자를 사용하는 추가 명령문을 포함합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "AWS": "arn:aws:iam::SendingAccountID:user/CWLExportUser" }
        }
    ]
}
```

4. 저장을 선택하여 버킷에서 액세스 정책으로 방금 추가한 정책을 설정합니다. 이 정책은 CloudWatch Logs가 Amazon S3 버킷으로 로그 데이터를 내보낼 수 있도록 합니다. 버킷 소유자는 내보낸 모든 개체에 대해 모든 권한을 가집니다.

   Warning

   기존 버킷에 이미 하나 이상의 정책이 연결되어 있는 경우 CloudWatch Logs 액세스용 명령문을 해당 정책이나 정책들에 추가합니다. 발생한 권한 집합이 버킷에 액세스하는 사용자에게 적절한지를 여부를 평가하는 것이 좋습니다.

단계 4. 내보내기 작업 생성

이 단계에서는 로그 그룹에서 로그를 내보낼 수 있도록 내보내기 작업을 생성합니다.

CloudWatch 콘솔을 사용하여 Amazon S3에 데이터를 내보내려면

1. (으)로 로그인 IAM 사용자가 에서 만든 사용자 2단계: 생성 IAM 에 대한 전체 액세스 권한이 있는 사용자 Amazon S3 및 CloudWatch Logs.
2. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
3. 탐색 창에서 로그 그룹을 선택합니다.
4. 로그 그룹 화면에서 로그 그룹의 이름을 선택합니다.
5. 작업, Amazon S3로 데이터 내보내기를 선택합니다.
6. Amazon S3로 데이터 내보내기 화면의 내보낼 데이터 정의에서 From(부터) 및 To(까지)를 사용하여 내보낼 데이터에 대한 시간 범위를 설정합니다.
7. 로그 그룹에 여러 개의 로그 스트림이 있는 경우에는 로그 스트림 접두사를 제공하여 로그 그룹 데이터를 특정 스트림으로 제한할 수 있습니다. 고급을 선택하고 스트림 접두사에 로그 스트림 접두사를 입력합니다.
8. S3 버킷 선택에서 Amazon S3 버킷과 연관된 계정을 선택합니다.
9. S3 버킷 이름에서 Amazon S3 버킷을 선택합니다.
10. S3 버킷 접두사에 버킷 정책에서 지정한 임의로 생성된 문자열을 입력합니다.
11. 내보내기를 선택하여 Amazon S3로 로그 데이터를 내보냅니다.
12. Amazon S3으로 내보낸 로그 데이터의 상태를 보려면 작업, Amazon S3에 대한 모든 내보내기 보기를 선택합니다.

AWS CLI를 사용하여 Amazon S3으로 로그 데이터 내보내기

다음 예제에서는 내보내기 작업을 사용하여 CloudWatch Logs 로그 그룹 이름 `my-log-group` 에게 Amazon S3 버킷 이름 `my-exported-logs`. 이 예에서는 다음과 같은 로그 그룹을 이미 만들었다고 가정합니다. `my-log-group`.

로그 데이터 내보내기 Amazon S3 암호화되는 버킷 AWS KMS 은(는) 지원되지 않습니다.

단계 1. Amazon S3 버킷 생성

CloudWatch Logs를 위해 특별히 생성한 버킷을 사용하는 것이 좋습니다. 그러나 기존 버킷을 사용하고 싶으면 2단계로 건너뛸 수 있습니다.

> Note
>
> Amazon S3 버킷은 내보내려는 로그 데이터와 동일한 리전에 상주해야 합니다. CloudWatch Logs는 다른 리전에 있는 Amazon S3 버킷으로의 데이터 내보내기를 지원하지 않습니다.

AWS CLI를 사용하여 Amazon S3 버킷을 생성하려면

명령 프롬프트에서 다음 create-bucket 명령을 입력합니다. 여기서 `LocationConstraint`는 로그 데이터를 내보내는 리전입니다.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

다음은 예제 출력입니다.

```
{
    "Location": "/my-exported-logs"
}
```

단계 2. 생성 IAM 에 대한 전체 액세스 권한이 있는 사용자 Amazon S3 및 CloudWatch Logs

다음 단계에서는 필수 권한으로 IAM 사용자를 생성합니다.

사용자를 생성하고 권한을 할당하려면

1. 다음 명령을 입력하여 IAM 사용자를 생성합니다.

   ```
   aws iam create-user --user-name CWLExportUser
   ```

2. IAM 관리형 정책을 지금 생성한 IAM 사용자에게 연결합니다.

   ```
   export S3POLICYARN=$(aws iam list-policies --query 'Policies[?
   PolicyName==`AmazonS3FullAccess`].{ARN:Arn}' --output text)
   ```

   ```
   export CWLPOLICYARN=$( aws iam list-policies --query 'Policies[?
   PolicyName==`CloudWatchLogsFullAccess`].{ARN:Arn}' --output text)
   ```

   ```
   aws iam attach-user-policy --user-name CWLExportUser --policy-arn $S3POLICYARN
   ```

   ```
   aws iam attach-user-policy --user-name CWLExportUser --policy-arn $CWLPOLICYARN
   ```

3. 두 개의 관리형 정책이 연결되어 있는지 확인합니다.

   ```
   aws iam list-attached-user-policies --user-name CWLExportUser
   ```

4. 구성 AWS CLI 포함시키기 위해 IAM 자격 증명 *CWLExportUser* IAM user 자세한 내용은 AWS CLI 구성을 참조하십시오.

단계 3. 다음에 대한 권한 설정 Amazon S3 버킷

기본적으로 모든 Amazon S3 버킷 및 객체는 비공개입니다. 리소스 소유자(버킷을 생성한 계정)만 해당 버킷과 버킷에 포함된 객체에 액세스할 수 있습니다. 그러나 리소스 소유자는 액세스 정책을 작성하여 다른 리소스 및 사용자에게 액세스 권한을 부여할 수 있습니다.

Amazon S3 버킷에 대한 권한을 설정하려면

1. 이름이 `policy.json`인 파일을 생성하고 다음 액세스 정책을 추가합니다. `Resource`를 S3 버킷의 이름으로 변경하고 `Principal`을 로그 데이터를 내보내는 리전의 엔드포인트로 변경합니다. 텍스트 편집기를 사용하여 이 정책 파일을 생성합니다. IAM 콘솔을 사용하지 마십시오.

   - 버킷이 계정에 있으면 다음 정책을 사용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "s3:GetBucketAcl",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        }
    ]
}
```

   - 버킷이 다른 계정에 있으면 대신 다음 정책을 사용합니다. 이전 단계에서 생성한 IAM 사용자를 사용하는 추가 명령문을 포함합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "AWS": "arn:aws:iam::SendingAccountID:user/CWLExportUser" }
      }
```

```
    ]
}
```

- 버킷이 다른 계정에 있고 IAM 사용자 대신 IAM 역할을 사용하고 있으면 대신 다음 정책을 사용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "AWS": "arn:aws:iam::SendingAccountID:role/CWLExportUser" }
      }
    ]
}
```

2. put-bucket-policy 명령을 사용하여 버킷에 액세스 정책으로 방금 추가한 정책을 설정합니다. 이 정책은 CloudWatch Logs가 Amazon S3 버킷으로 로그 데이터를 내보낼 수 있도록 합니다. 버킷 소유자는 내보낸 모든 개체에 대해 모든 권한을 갖게 됩니다.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

Warning

기존 버킷에 이미 하나 이상의 정책이 연결되어 있는 경우 CloudWatch Logs 액세스용 명령문을 해당 정책이나 정책들에 추가합니다. 발생한 권한 집합이 버킷에 액세스하는 사용자에게 적절한지를 여부를 평가하는 것이 좋습니다.

단계 4. 내보내기 작업 생성

로그 그룹에서 로그를 내보내기 위해 내보내기 작업을 생성하고 나면 내보낼 데이터의 크기에 따라 내보내기 작업에 몇 초부터 몇 시간까지 소요될 수 있습니다.

AWS CLI를 사용하여 내보내기 작업을 생성하려면

명령 프롬프트에서 다음 create-export-task 명령을 사용하여 내보내기 작업을 생성합니다.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-group-09-10-2015"
 --log-group-name "my-log-group" --from 1441490400000 --to 1441494000000 --destination "my-
exported-logs" --destination-prefix "export-task-output"
```

다음은 예제 출력입니다.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

단계 5. 내보내기 작업 설명

내보내기 작업을 생성하고 나면 작업의 현재 상태를 파악할 수 있습니다.

AWS CLI를 사용하여 내보내기 작업을 설명하려면

명령 프롬프트에서 다음 describe-export-tasks 명령을 사용합니다.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

다음은 예제 출력입니다.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "tTo": 1441494000000
   }]
}
```

세 가지 방법으로 `describe-export-tasks` 명령을 사용할 수 있습니다.

- 필터 없음: 모든 내보내기 작업을 생성 순서와 반대로 나열합니다.
- 태스크 ID 필터링: 지정된 ID를 가진 내보내기 작업이 있는 경우 이를 나열합니다.
- 태스크 상태 필터링: 지정된 상태의 내보내기 작업을 나열합니다.

예를 들어, 다음 명령을 사용하여 `FAILED` 상태를 필터링합니다.

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

다음은 예제 출력입니다.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
```

```
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

단계 6. 내보내기 작업 취소

PENDING 또는 RUNNING 상태에 있을 경우 내보내기 작업을 취소할 수 있습니다.

AWS CLI를 사용하여 내보내기 작업을 취소하려면

명령 프롬프트에서 다음 cancel-export-task 명령을 사용합니다.

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

describe-export-tasks 명령을 사용하여 작업이 성공적으로 취소되었는지 확인할 수 있습니다.

CloudWatch Logs 데이터를 Amazon Elasticsearch Service로 스트리밍

CloudWatch Logs 구독을 통해 실시간에 가깝게 Amazon Elasticsearch Service(Amazon ES) 클러스터로 수신한 데이터를 스트리밍하도록 CloudWatch Logs 로그 그룹을 구성할 수 있습니다. 자세한 내용은 구독을 통한 로그 데이터 실시간 처리 (p. 80) 단원을 참조하십시오.

스트리밍되는 로그 데이터의 양에 따라 함수에 함수 수준의 동시 실행 제한을 설정할 수 있습니다. 자세한 내용은 함수 수준 동시 실행 한도를 참조하십시오.

> Note
>
> Amazon ES로 대량의 CloudWatch Logs 데이터를 스트리밍하면 사용 요금이 증가할 수 있습니다. 따라서 Billing and Cost Management 콘솔에서 예산을 생성하는 것이 좋습니다. 자세한 내용은 예산을 통해 비용 관리 단원을 참조하십시오.

사전 요구 사항

시작에 앞서 Amazon ES 도메인을 생성합니다. Amazon ES 도메인은 퍼블릭 액세스 또는 VPC 액세스 중 하나를 가질 수 있지만 도메인을 생성한 후에는 액세스 유형을 수정할 수 없습니다. Amazon ES 도메인 설정을 나중에 검토해서 클러스터가 처리할 데이터 양에 따라 클러스터 구성을 수정하고 싶을 수 있습니다.

Amazon ES에 대한 자세한 내용은 Amazon Elasticsearch Service 개발자 안내서 단원을 참조하십시오.

Amazon ES 도메인을 생성하려면

명령 프롬프트에서 아래 create-elasticsearch-domain 명령을 사용하십시오.

```
aws es create-elasticsearch-domain --domain-name my-domain
```

Amazon ES에 대한 로그 그룹 구독

CloudWatch 콘솔을 사용하여 Amazon ES에 대한 로그 그룹을 구독할 수 있습니다.

Amazon ES에 대한 로그 그룹을 구독하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. 로그 그룹의 이름을 선택합니다.
4. 작업, Create Elastisearch subscription filter(Elastisearch 구독 필터 생성)을 선택합니다.
5. 이 계정의 클러스터로 스트리밍할지 아니면 다른 계정의 클러스터로 스트리밍할지 선택합니다.
6. Amazon ES 클러스터의 경우 이전 단계에서 생성한 클러스터를 선택합니다.
7. Lambda 함수의 Lambda IAM 실행 역할에서 Amazon ES에 대한 호출을 실행할 때 Lambda가 사용해야 하는 IAM 역할을 선택하고 다음을 선택합니다.

   선택한 IAM 역할은 다음 요구 사항을 충족해야 합니다.

- 신뢰 관계를 맺고 있는 `lambda.amazonaws.com`이 있어야 합니다.
- 다음 정책이 포함되어야 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:region:account-id:domain/target-domain-name/*"
        }
    ]
}
```

- 대상 Amazon ES 도메인에서 VPC 액세스를 사용하는 경우 역할에 AWSLambdaVPCAccessExecutionRole 정책이 연결되어 있어야 합니다. Amazon에서 관리하는 이 정책은 고객의 VPC에 Lambda 액세스 권한을 부여하여 Lambda가 VPC의 Amazon ES 엔드포인트에 기록할 수 있게 합니다.

8. 로그 형식에서 로그 형식을 선택합니다.
9. 구독 필터 패턴에 로그 이벤트에서 찾을 용어나 패턴을 입력합니다. 이렇게 하면 Amazon ES 클러스터로 원하는 데이터만 전송할 수 있습니다. 자세한 내용은 필터를 사용하여 로그 이벤트에서 지표 생성 (p. 65) 단원을 참조하십시오.
10. (선택 사항) 테스트할 로그 데이터 선택에서 로그 스트림을 선택한 다음 패턴 테스트를 선택해서 검색 필터가 예상한 결과를 반환하고 있는지 확인합니다.
11. 스트리밍 시작을 선택합니다.

CloudWatch 로그에 로그를 게시하는 AWS 서비스

다음 AWS 서비스는 로그를 CloudWatch Logs에 게시합니다. 이러한 서비스에서 보내는 로그에 대한 자세한 내용은 연결된 설명서를 참조하십시오.

서비스	설명서:
Amazon API Gateway	설정 CloudWatch API 로그인 API 게이트웨이
Amazon Aurora MySQL	출판 Amazon Aurora MySQL 다음에 대한 로그 Amazon CloudWatch Logs
AWS CloudHSM	모니터링 AWS CloudHSM 감사 로그인 Amazon CloudWatch Logs
AWS CloudTrail	모니터링 CloudTrail 로그 파일 Amazon CloudWatch Logs
Amazon Cognito	생성 CloudWatch Logs IAM 역할
Amazon Connect	로깅 및 모니터링 Amazon Connect
AWS DataSync	허용 중 DataSync Amazon에 로그를 업로드하려면 CloudWatch 그룹 로깅
AWS Elastic Beanstalk	사용 Elastic Beanstalk 및 Amazon CloudWatch Logs
Amazon Elastic Container Service	컨테이너 인스턴스에 CloudWatch Logs 사용
Amazon Elastic Kubernetes Service	아마존 Amazon Elastic Kubernetes Service 제어 평면 로깅
AWS Fargate	awslogs 로그 드라이버 사용
AWS Glue	AWS Glue 작업에 대한 지속 로깅
AWS IoT	모니터링 CloudWatch Logs
AWS Lambda	액세스 중 Amazon CloudWatch Logs 에 대해 AWS Lambda
Amazon MQ	구성 Amazon MQ 일반 및 감사 로그 게시 대상 Amazon CloudWatch Logs
AWS OpsWorks	AWS OpsWorks Stacks에 Amazon CloudWatch Logs 사용
Amazon Relational Database Service	PostgreSQL 로그 게시 대상 CloudWatch Logs
AWS RoboMaker	오프라인 지원을 제공하는 AWS RoboMaker CloudWatch ROS 노드
Amazon Route 53	Amazon 경로 53의 로깅 및 모니터링

서비스	설명서:
Amazon SageMaker	로그 Amazon SageMaker 이벤트 Amazon CloudWatch
Amazon Simple Notification Service	보기 CloudWatch Logs
Amazon VPC	VPC 흐름 로그

Amazon CloudWatch Logs의 보안

AWS에서는 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 매우 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 귀하의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 – AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Amazon WorkSpaces에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하십시오.
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon CloudWatch Logs를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 보안 및 규정 준수 목적에 맞게 Amazon CloudWatch Logs를 구성하는 방법을 보여줍니다. 또한 CloudWatch Logs 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배웁니다.

목차

- Amazon CloudWatch Logs에서 데이터 보호 (p. 112)
- Amazon CloudWatch Logs의 ID 및 액세스 관리 (p. 113)
- Amazon CloudWatch Logs에 대한 규정 준수 확인 (p. 128)
- Amazon CloudWatch Logs의 복원성 (p. 128)
- Amazon CloudWatch Logs의 인프라 보안 (p. 128)
- 인터페이스 VPC 엔드포인트와 함께 CloudWatch Logs 사용 (p. 129)

Amazon CloudWatch Logs에서 데이터 보호

AWS 공동 책임 모델은 Amazon CloudWatch Logs의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호해야 합니다. 이 인프라에서 호스팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 이 콘텐츠에는 사용하는 AWS 서비스에 대한 보안 구성 및 관리 작업이 포함됩니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하십시오. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델과 GDPR 블로그 게시물을 참조하십시오.

데이터를 보호하려면 AWS 계정 자격 증명을 보호하고 AWS Identity and Access Management(IAM)을 사용해 개별 사용자 계정을 설정하는 것이 좋습니다. 이러한 방식에서는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 멀티 팩터 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2 이상을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.

- Amazon S3에 저장된 개인 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2를 참조하십시오.

이름 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마십시오. 여기에는 CloudWatch Logs 또는 기타 AWS 제품에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. CloudWatch Logs 또는 기타 서비스에 입력하는 모든 데이터는 진단 로그에 포함하기 위해 선택될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시키지 마십시오.

저장 데이터 암호화

CloudWatch Logs는 암호화를 사용하여 저장 시 데이터를 보호합니다. 모든 로그 그룹이 암호화됩니다. 기본적으로 CloudWatch Logs 서비스는 서버 측 암호화 키를 관리합니다.

로그를 암호화하고 해독하는 데 사용되는 키를 관리하려면 AWS Key Management Service의 고객 마스터 키(CMK)를 사용합니다. 자세한 내용은 AWS KMS를 사용하여 CloudWatch Logs에서 로그 데이터를 암호화 (p. 59) 항목을 참조하십시오.

전송 중 데이터 암호화

CloudWatch Logs는 전송 중인 데이터의 종단 간 암호화를 사용합니다. CloudWatch Logs 서비스는 서버 측 암호화 키를 관리합니다.

Amazon CloudWatch Logs의 ID 및 액세스 관리

Amazon CloudWatch Logs에 액세스하려면 AWS가 요청을 인증하는 데 사용할 수 있는 자격 증명이 필요합니다. 이 자격 증명에는 클라우드 리소스에 대한 CloudWatch Logs 데이터 검색과 같이 AWS 리소스에 액세스할 수 있는 권한이 포함되어야 합니다. 다음 단원에서는 AWS Identity and Access Management(IAM) 및 CloudWatch Logs을 사용하여 리소스에 액세스할 수 있는 대상을 제어함으로써 리소스를 보호하는 방법에 대해 자세히 설명합니다.

- Authentication (p. 113)
- 액세스 제어 (p. 114)

Authentication

다음과 같은 자격 증명 유형으로 AWS에 액세스할 수 있습니다.

- AWS 계정 루트 사용자 – AWS에 가입할 때 AWS 계정과 연결된 이메일 주소 및 암호를 입력합니다. 이 두 가지가 루트 자격 증명으로, 모든 AWS 리소스에 대한 전체 액세스를 제공합니다.

  Important

  보안상 관리자 사용자, 즉 AWS 계정에 대한 전체 권한이 있는 IAM 사용자를 만들 때에만 루트 자격 증명을 사용하는 것이 좋습니다. 그런 다음 이 관리자를 사용하여 제한된 권한이 있는 다른 IAM 사용자 및 역할을 만들 수 있습니다. 자세한 내용은 을 참조하십시오. IAM 모범 사례 및 관리 사용자 및 그룹 생성 에서 IAM 사용 설명서.
- IAM 사용자 – IAM 사용자는 특정 사용자 지정 권한(예: CloudWatch Logs에서 지표를 볼 수 있는 권한)이 있는 AWS 계정의 자격 증명입니다. IAM 사용자 이름과 암호를 사용하여 AWS Management 콘솔, AWS 토론 포럼 또는 AWS Support Center 같은 보안 AWS 웹 페이지에 로그인할 수 있습니다.

사용자 이름과 암호 외에도 각 사용자에 대해 액세스 키를 생성할 수 있습니다. 여러 SDK 중 하나를 통해 또는 AWS Command Line Interface(AWS CLI)를 사용하여 AWS 서비스에 프로그래밍 방식으로 액세스할 때 이러한 키를 사용할 수 있습니다. SDK 및 CLI 도구는 액세스 키를 사용하여 암호화 방식으로 요청에 서명합니다. AWS 도구를 사용하지 않는 경우, 직접 요청에 서명해야 합니다. CloudWatch Logs supports 서명 버전 4인바운드 API 요청을 인증하기 위한 프로토콜입니다. 요청 인증에 대한 자세한 내용은 AWS General Reference의 서명 버전 4 서명 프로세스를 참조하십시오.

- IAM 역할 – IAM 역할은 계정에 만들 수 있는 특정 권한이 있는 또 다른 IAM 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. IAM 역할을 사용하면 AWS 서비스와 리소스에 액세스하는 데 사용할 수 있는 임시 액세스 키를 얻을 수 있습니다. 임시 자격 증명을 가진 IAM 역할은 다음과 같은 상황에서 유용합니다.

  - 연합된 사용자 액세스 – IAM 사용자를 만드는 대신 AWS Directory Service, 엔터프라이즈 사용자 디렉터리 또는 웹 자격 증명 공급자의 기존 사용자 자격 증명을 사용할 수 있습니다. 이러한 사용자를 연합된 사용자라고 합니다. 자격 증명 공급자를 통해 액세스를 요청하면 AWS가 연합된 사용자에게 역할을 할당합니다. 통합 사용자에 대한 자세한 내용은 를 참조하십시오. 통합 사용자 및 역할 에서 IAM 사용 설명서.

  - 교차 계정 액세스 – 계정의 IAM 역할을 사용하여 다른 AWS 계정에 계정 리소스에 액세스할 권한을 부여할 수 있습니다. 예는 다음을 참조하십시오. 튜토리얼: IAM 역할을 사용하여 AWS 계정에 대한 액세스 위임 에서 IAM 사용 설명서.

  - AWS 서비스 액세스 – 계정의 IAM 역할을 사용하여 AWS 서비스에 계정의 리소스에 액세스할 권한을 부여할 수 있습니다. 예를 들어 Amazon Redshift에서 자동으로 Amazon S3 버킷에 액세스하도록 허용하는 역할을 만든 후 버킷에 저장된 데이터를 Amazon Redshift 클러스터에 로드할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스에 권한을 위임하기 위한 역할 생성을 참조하십시오.

  - Amazon EC2에서 실행되는 애플리케이션 – 인스턴스에서 실행되고 AWS API 요청을 하는 애플리케이션에서 사용할 수 있도록 EC2 인스턴스 내에 액세스 키를 저장하는 대신에, IAM 역할을 사용하여 이러한 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 인스턴스에 연결된 인스턴스 프로파일을 만들 수 있습니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 자격 증명을 얻을 수 있습니다. 자세한 내용은 을 참조하십시오. Amazon EC2에서 애플리케이션에 대한 역할 사용 에서 IAM 사용 설명서.

액세스 제어

요청을 인증할 수 있는 유효한 자격 증명이 있더라도 권한이 없으면 CloudWatch Logs 리소스를 생성하거나 액세스할 수 없습니다. 예를 들어 로그 스트림과 로그 그룹을 생성할 권한이 있어야 합니다.

다음 단원에서는 CloudWatch Logs에 대한 권한을 관리하는 방법을 설명합니다. 먼저 개요를 읽어 보면 도움이 됩니다.

- CloudWatch Logs 리소스에 대한 액세스 권한 관리 개요 (p. 115)
- CloudWatch Logs에 대한 자격 증명 기반 정책(IAM 정책) 사용 (p. 118)
- CloudWatch Logs 권한 참조 문서 (p. 123)

CloudWatch Logs 리소스에 대한 액세스 권한 관리 개요

모든 AWS 리소스는 AWS 계정의 소유이고, 리소스 생성 또는 리소스 액세스 권한은 권한 정책에 따라 결정됩니다. 계정 관리자는 IAM 자격 증명(즉, 사용자, 그룹, 역할)에 권한 정책을 연결할 수 있고, 일부 서비스(예: AWS Lambda)에서는 리소스에 대한 권한 정책 연결도 지원합니다.

> **Note**
>
> 계정 관리자 또는 관리자 IAM 사용자는 관리자 권한이 있는 사용자입니다. 자세한 내용은 IAM 사용 설명서에서 IAM 모범 사례를 참조하십시오.

권한을 부여하려면 권한을 부여 받을 사용자, 권한 대상이 되는 리소스, 해당 리소스에 허용되는 특정 작업을 결정합니다.

주제

- CloudWatch Logs 리소스와 작업 (p. 115)
- 리소스 소유권 이해 (p. 116)
- 리소스 액세스 관리 (p. 116)
- 정책 요소 지정: 조치, 효과 및 원칙 (p. 118)
- 정책에서 조건 지정 (p. 118)

CloudWatch Logs 리소스와 작업

CloudWatch Logs에서 로그 그룹, 로그 스트림 및 대상이 기본 리소스입니다. CloudWatch Logs는 하위 리소스(기본 리소스에서 사용되는 다른 리소스)를 지원하지 않습니다.

다음 표에 나와 있는 것처럼 이러한 리소스와 하위 리소스에는 고유한 Amazon 리소스 이름(ARN)이 연결되어 있습니다.

리소스 유형	ARN 형식
로그 그룹	arn:aws:로그:*region*:*account-id*:로그 그룹:*log_group_name*
로그 스트림	arn:aws:로그:*region*:*account-id*:로그 그룹:*log_group_name*:로그 스트림:*log-stream-name*
대상	arn:aws:로그:*region*:*account-id*대상*destination_name*

ARN에 대한 자세한 내용은 다음을 참조하십시오. ARN 에서 IAM 사용 설명서. 에 대한 정보 CloudWatch Logs ARN, 참조 Amazon 리소스 이름(ARN) 및 AWS 서비스 네임스페이스 에서 Amazon Web Services 일반 참조. CloudWatch Logs에 적용되는 정책의 예제는 CloudWatch Logs에 대한 자격 증명 기반 정책(IAM 정책) 사용 (p. 118) 단원을 참조하십시오.

CloudWatch Logs은(는) CloudWatch Logs 리소스를 처리하기 위한 작업을 제공합니다. 사용 가능한 작업 목록은 CloudWatch Logs 권한 참조 문서 (p. 123) 단원을 참조하십시오.

리소스 소유권 이해

AWS 계정은 리소스를 누가 생성했든 상관없이 계정에서 생성된 리소스를 소유합니다. 특히, 리소스 소유자는 리소스 생성 요청을 인증하는 보안 주체 엔터티(즉, 루트 계정, IAM 사용자 또는 IAM 역할)의 AWS 계정입니다. 다음 예에서는 이 계정의 작동 방식을 설명합니다.

- AWS 계정의 루트 계정 자격 증명을 사용하여 로그 그룹을 생성하면, AWS 계정이 CloudWatch Logs 리소스의 소유자가 됩니다.
- AWS 계정에서 IAM 사용자를 생성하고 CloudWatch Logs 리소스를 생성할 수 있는 권한을 해당 사용자에게 부여하면 해당 사용자는 CloudWatch Logs 리소스를 생성할 수 있습니다. 하지만 해당 사용자가 속한 AWS 계정이 CloudWatch Logs 리소스를 소유합니다.
- AWS 계정에서 CloudWatch Logs 리소스를 생성할 권한이 있는 IAM 역할을 만드는 경우, 해당 역할을 담당할 수 있는 사람은 누구나 CloudWatch Logs 리소스를 생성할 수 있습니다. 이 경우 역할이 속한 AWS 계정이 CloudWatch Logs 리소스를 소유합니다.

리소스 액세스 관리

권한 정책은 누가 무엇에 액세스 할 수 있는지를 나타냅니다. 다음 단원에서는 권한 정책을 만드는 데 사용 가능한 옵션에 대해 설명합니다.

> **Note**
>
> 이 단원에서는 CloudWatch Logs의 맥락에서 IAM을 사용하는 방법에 대해 설명하며, IAM 서비스에 대한 자세한 정보는 다루지 않습니다. 전체 IAM 설명서는 IAM의 IAM 사용 설명서이란 무엇입니까?를 참조하십시오. IAM 정책 구문과 설명에 대한 자세한 내용은 IAM 사용 설명서의 IAM 정책 참조를 참조하십시오.

IAM 자격 증명에 연결된 정책을 자격 증명 기반 정책(IAM 정책)이라고 하고, 리소스에 연결된 정책을 리소스 기반 정책이라고 합니다. CloudWatch Logs는 대상에 자격 증명 기반 정책과 리소스 기반 정책을 지원하는데, 이들 정책을 사용하여 계정 구독에서 활성화를 수행할 수 있습니다. 자세한 정보는 구독과 교차 계정 로그 데이터 공유 (p. 91) 단원을 참조하십시오.

주제

- 그룹 권한 및 기여자 인사이트 로깅 (p. 116)
- 자격 증명 기반 정책(IAM 정책) (p. 116)
- 리소스 기반 정책 (p. 117)

그룹 권한 및 기여자 인사이트 로깅

출품자 인사이트는 CloudWatch 로그 그룹의 데이터를 분석하고 기여자 데이터를 표시하는 시계열을 생성할 수 있습니다. 상위 N개의 기고자, 총 고유 기고자 수 및 사용량에 대한 지표를 볼 수 있습니다. 자세한 내용은 Contributor Insights를 사용하여 높은 카디널리티 데이터 분석을 참조하십시오.

사용자에게 권한을 부여할 때 `cloudwatch:PutInsightRule` 및 `cloudwatch:GetInsightRuleReport` 사용자는 모든 로그 그룹을 평가하는 규칙을 만들 수 있습니다. CloudWatch Logs 결과를 확인합니다. 결과는 해당 로그 그룹에 대한 기여자 데이터를 포함할 수 있습니다. 이 데이터를 볼 수 있어야 하는 사용자에게만 이러한 권한을 부여해야 합니다.

자격 증명 기반 정책(IAM 정책)

정책을 IAM 자격 증명에 연결할 수 있습니다. 예를 들면,

- 계정 내 사용자 또는 그룹에 권한 정책 연결 – 사용자에게 CloudWatch Logs 콘솔에서 로그를 볼 수 있는 권한을 부여하려면 권한 정책을 사용자 또는 해당 사용자가 속한 그룹에 연결하면 됩니다.

- 역할에 권한 정책 연결(교차 계정 권한 부여) – 자격 증명 기반 권한 정책을 IAM 역할에 연결하여 교차 계정 권한을 부여할 수 있습니다. 예를 들어, 계정 A의 관리자는 다음과 같이 다른 AWS 계정(예: 계정 B) 또는 AWS 서비스에 교차 계정 권한을 부여할 역할을 생성할 수 있습니다.
  1. 계정 A 관리자는 IAM 역할을 생성하고 계정 A의 리소스에 대한 권한을 부여하는 역할에 권한 정책을 연결합니다.
  2. 계정 A 관리자는 계정 B를 역할을 수임할 보안 주체로 식별하는 역할에 신뢰 정책을 연결합니다.
  3. 계정 B 관리자는 계정 B의 사용자에게 역할을 수임할 권한을 위임할 수 있습니다. 그러면 계정 B의 사용자가 계정 A에서 리소스를 생성하거나 액세스할 수 있습니다. AWS 서비스에 역할 수임 권한을 부여할 경우 신뢰 정책의 보안 주체가 AWS 서비스 보안 주체이기도 합니다.

  IAM을 사용하여 권한을 위임하는 방법에 대한 자세한 내용은 https://docs.aws.amazon.com/IAM/latest/UserGuide/access.html의 IAM 사용 설명서액세스 관리를 참조하십시오.

다음은 us-east-1의 모든 리소스에 대해 `logs:PutLogEvents`, `logs:CreateLogGroup` 및 `logs:CreateLogStream` 작업 권한을 부여하는 정책의 예시입니다. CloudWatch Logs는 로그 그룹을 위해 일부 API 작업에서 리소스 ARN(리소스 수준 권한이라고도 함)을 사용하여 특정 리소스를 식별할 수 있도록 지원합니다. 로그 그룹을 모두 포함시키고 싶으면 와일드카드 문자 (*)를 지정해야 합니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Action":[
            "logs:PutLogEvents",
            "logs:CreateLogGroup",
            "logs:CreateLogStream"
         ],
         "Resource":"arn:aws:logs:us-east-1:*:*"
      }
   ]
}
```

CloudWatch Logs에서 자격 증명 기반 정책을 사용하는 방법에 대한 자세한 내용은 CloudWatch Logs에 대한 자격 증명 기반 정책(IAM 정책) 사용 (p. 118) 단원을 참조하십시오. 사용자, 그룹, 역할 및 권한에 대한 자세한 내용은 https://docs.aws.amazon.com/IAM/latest/UserGuide/id.html의 IAM 사용 설명서자격 증명(사용자, 그룹 및 역할)을(를) 참조하십시오.

리소스 기반 정책

CloudWatch Logs는 대상에 리소스 기반 정책을 지원하는 데, 이 정책을 사용하여 계정 구독에서 활성화를 수행할 수 있습니다. 자세한 정보는 대상 생성 (p. 92) 단원을 참조하십시오. PutDestination API를 사용하여 대상을 생성하고, PutDestination API를 사용하여 대상에 리소스 정책을 추가할 수 있습니다. 다음 예제는 계정 ID가 111122223333인 또 다른 AWS 계정이 대상 `arn:aws:logs:us-east-1:123456789012:destination:testDestination`에 대한 로그 그룹을 구독할 수 있도록 허용합니다.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-east-1:123456789012:destination:testDestination"
```

```
    }
  ]
}
```

정책 요소 지정: 조치, 효과 및 원칙

각 CloudWatch Logs 리소스에 대해 서비스는 일련의 API 작업을 정의합니다. 이러한 API 작업에 대한 권한을 부여하기 위해 CloudWatch Logs에서는 정책에서 지정할 수 있는 작업을 정의합니다. 일부 API 작업에서는 API 작업을 수행하기 위해 복수의 작업에 대한 권한이 필요할 수 있습니다. 리소스 및 API 작업에 대한 자세한 내용은 CloudWatch Logs 리소스와 작업 (p. 115) 및 CloudWatch Logs 권한 참조 문서 (p. 123) 단원을 참조하십시오.

다음은 기본 정책 요소입니다.

- 리소스 – Amazon 리소스 이름(ARN)을 사용하여 정책을 적용할 리소스를 식별합니다. 자세한 정보는 CloudWatch Logs 리소스와 작업 (p. 115) 단원을 참조하십시오.
- 작업 – 작업 키워드를 사용하여 허용 또는 거부할 리소스 작업을 식별합니다. 예를 들어, `logs.DescribeLogGroups` 권한은 사용자에게 `DescribeLogGroups` 작업 수행 권한을 허용합니다.
- 효과 – 사용자가 특정 작업을 요청하는 경우 허용할지 아니면 거부할지 그 결과를 지정합니다. 명시적으로 리소스에 대한 액세스 권한을 부여(허용)하지 않는 경우, 액세스는 묵시적으로 거부됩니다. 다른 정책에서는 액세스 권한을 부여하더라도 리소스에 대한 액세스를 명시적으로 거부하여 사용자가 해당 리소스에 액세스하지 못하게 할 수도 있습니다.
- 보안 주체 – 자격 증명 기반 정책(IAM 정책)에서 정책이 연결되는 사용자는 암시적인 보안 주체입니다. 리소스 기반 정책의 경우 사용자, 계정, 서비스 또는 권한의 수신자인 기타 개체를 지정합니다(리소스 기반 정책에만 해당). CloudWatch Logs는 대상에 대한 리소스 기반 정책을 지원합니다.

에 대해 자세히 알아보기 IAM 정책 구문 및 설명, 참조 아슬란드 IAM 정책 참조 에서 IAM 사용 설명서.

모든 CloudWatch Logs API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 CloudWatch Logs 권한 참조 문서 (p. 123) 단원을 참조하십시오.

정책에서 조건 지정

권한을 부여할 때 액세스 정책 언어를 사용하여 조건이 적용되는 조건을 지정할 수 있습니다. 예를 들어, 특정 날짜 이후에만 정책을 적용할 수 있습니다. 정책 언어에서의 조건 지정에 관한 자세한 내용은 IAM 사용 설명서의 조건 단원을 참조하십시오.

조건을 표시하려면 미리 정의된 조건 키를 사용합니다. 각 가 지원하는 컨텍스트 키 목록은 AWS 서비스 및 목록 AWS-전체 정책 키, 참조 AWS 서비스 작업 및 상태 컨텍스트 키 및 글로벌 및 IAM 조건 컨텍스트 키 에서 IAM 사용 설명서.

CloudWatch Logs에 대한 자격 증명 기반 정책(IAM 정책) 사용

이 섹션에서는 계정 관리자가 IAM 자격 증명(즉 사용자, 그룹, 역할)에 권한 정책을 연결할 수 있는 자격 증명 기반 정책의 예를 제시합니다.

> **Important**
>
> CloudWatch Logs 리소스에 대한 액세스 관리를 위해 제공되는 기본 개념과 옵션 설명에 대한 소개 주제 부분을 우선 읽어 보는 것이 좋습니다. 자세한 정보는 CloudWatch Logs 리소스에 대한 액세스 권한 관리 개요 (p. 115) 단원을 참조하십시오.

이 주제는 다음을 다룹니다.

- CloudWatch 콘솔 사용에 필요한 권한 (p. 119)
- CloudWatch Logs에 대한 AWS 관리형(미리 정의된) 정책 (p. 121)

- 고객 관리형 정책 예 (p. 121)

다음은 권한 정책의 예제입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

이 정책에는 로그 그룹 및 로그 스트림을 생성하고 로그 스트림에 로그 이벤트를 업로드하며 로그 스트림에 대한 세부 사항을 나열할 수 있는 권한을 부여하는 명령문이 하나 포함되어 있습니다.

`Resource` 값 끝에 와일드카드 문자(*)가 있다는 것은 이 정책 명령문이 어떤 로그 그룹에서든 `logs:CreateLogGroup`, `logs:CreateLogStream`, `logs:PutLogEvents` 및 `logs:DescribeLogStreams` 작업에 대한 권한을 허용한다는 의미입니다. 이러한 권한을 특정 로그 그룹으로 제한하려면 리소스 ARN에 있는 와일드카드 문자(*)를 특정 로그 그룹 ARN으로 대체합니다. 그룹 내의 섹션에 대한 자세한 내용은 IAM 정책문, 참조 IAM 정책 요소 참조 에서 IAM 사용 설명서. 모든 CloudWatch Logs 작업들을 보여주는 목록은 CloudWatch Logs 권한 참조 문서 (p. 123) 단원을 참조하십시오.

CloudWatch 콘솔 사용에 필요한 권한

사용자가 CloudWatch 콘솔에서 CloudWatch Logs를 사용하려면 AWS 계정에서 다른 AWS 리소스를 설명할 수 있는 최소한의 권한 세트가 사용자에게 필요합니다. CloudWatch 콘솔에서 CloudWatch Logs을 사용하려면 다음 서비스에서도 권한이 있어야 합니다.

- CloudWatch
- CloudWatch Logs
- Amazon ES
- IAM
- Kinesis
- Lambda
- Amazon S3

최소 필수 권한보다 더 제한적인 IAM 정책을 만들면 콘솔에서는 해당 IAM 정책에 연결된 사용자에 대해 의도대로 작동하지 않습니다. 이 사용자가 CloudWatch 콘솔을 사용할 수 있도록 하려면 `CloudWatchReadOnlyAccess` 관리형 정책을 사용자에게 연결합니다(CloudWatch Logs에 대한 AWS 관리형(미리 정의된) 정책 (p. 121) 참조).

AWS CLI 또는 CloudWatch Logs API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다.

CloudWatch 콘솔을 사용하여 로그 구독을 관리하지 않는 사용자가 콘솔에서 작업하는 데 필요한 권한의 전체 집합은 다음과 같습니다.

- cloudwatch:getMetricData

- cloudwatch:listMetrics
- logs:cancelExportTask
- logs:createExportTask
- logs:createLogGroup
- logs:createLogStream
- logs:deleteLogGroup
- logs:deleteLogStream
- logs:deleteMetricFilter
- logs:deleteQueryDefinition
- logs:deleteRetentionPolicy
- logs:deleteSubscriptionFilter
- logs:describeExportTasks
- logs:describeLogGroups
- logs:describeLogStreams
- logs:describeMetricFilters
- logs:describeQueryDefinitions
- logs:describeSubscriptionFilters
- logs:filterLogEvents
- logs:getLogEvents
- logs:putMetricFilter
- logs:putQueryDefinition
- logs:putRetentionPolicy
- logs:putSubscriptionFilter
- logs:testMetricFilter

사용자가 콘솔을 사용하여 로그 구독도 관리할 경우 다음 권한도 필요합니다.

- es:describeElasticsearchDomain
- es:listDomainNames
- iam:attachRolePolicy
- iam:createRole
- iam:getPolicy
- iam:getPolicyVersion
- iam:getRole
- iam:listAttachedRolePolicies
- iam:listRoles
- kinesis:describeStreams
- kinesis:listStreams
- lambda:addPermission
- lambda:createFunction
- lambda:getFunctionConfiguration
- lambda:listAliases
- lambda:listFunctions
- lambda:listVersionsByFunction
- lambda:removePermission
- s3:listBuckets

CloudWatch Logs에 대한 AWS 관리형(미리 정의된) 정책

AWS는 AWS에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반 사용 사례를 처리합니다. 관리형 정책은 사용자가 필요한 권한을 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 단원을 참조하십시오.https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#aws-managed-policies

계정의 사용자에게 연결할 수 있는 다음 AWS 관리형 정책은 CloudWatch Logs에 대해 고유합니다.

- CloudWatchLogsFullAccess – CloudWatch Logs에 대한 전체 액세스 권한을 부여합니다.
- CloudWatchLogsReadOnlyAccess – CloudWatch Logs에 대한 읽기 전용 액세스 권한을 부여합니다.

> Note
>
> IAM 콘솔에 로그인하고 이 콘솔에서 특정 정책을 검색하여 이러한 권한 정책을 검토할 수 있습니다.

CloudWatch Logs 작업 및 리소스에 대한 권한을 허용하는 고유한 사용자 지정 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

고객 관리형 정책 예

이 단원에서는 다양한 CloudWatch Logs 작업에 대한 권한을 부여하는 사용자 정책의 예를 제공합니다. 이러한 정책은 CloudWatch Logs API, AWS SDK 또는 AWS CLI를 사용하는 경우에 적용됩니다.

예제:

- 예 1 전체 액세스 허용 CloudWatch Logs (p. 121)
- 예 2 읽기 전용 액세스 허용 CloudWatch Logs (p. 121)
- 예 3 하나의 로그 그룹에 대한 액세스 허용 (p. 122)

예 1 전체 액세스 허용 CloudWatch Logs

다음 정책은 사용자가 모든 CloudWatch Logs 작업을 호출할 수 있도록 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

예 2 읽기 전용 액세스 허용 CloudWatch Logs

AWS는 CloudWatch Logs 데이터에 대한 읽기 전용 액세스를 허용하는 CloudWatchLogsReadOnlyAccess 정책을 제공합니다. 이 정책에는 다음 권한이 포함되어 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
```

```
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

예 3 하나의 로그 그룹에 대한 액세스 허용

다음 정책에서는 사용자가 지정된 로그 그룹 하나에서 로그 이벤트를 읽고 쓸 수 있게 허용합니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

로그 그룹 수준의 제어를 위한 태깅 및 IAM 정책 사용

사용자에게 특정 로그 그룹에 대한 액세스 권한을 부여함과 동시에 다른 로그 그룹에 대한 액세스를 방지할 수 있습니다. 이렇게 하려면 로그 그룹에 태그를 지정하고 해당 태그를 참조하는 IAM 정책을 사용하십시오.

로그 그룹의 태깅에 대한 자세한 내용은 Amazon CloudWatch Logs에서 로그 그룹에 태그 지정 (p. 57) 단원을 참조하십시오.

로그 그룹에 태그를 지정할 때 특정 태그가 있는 로그 그룹에만 액세스를 허용할 수 있도록 사용자에게 IAM 정책에 대한 권한을 부여할 수 있습니다. 예를 들어, 다음 정책 명령문은 태그 키 `Team`에 대한 값이 `Green`인 로그 그룹에만 액세스하도록 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "logs:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

사용 방법에 대한 자세한 내용은 IAM 정책문, 참조 정책을 사용하여 액세스 제어 에서 IAM 사용 설명서.

CloudWatch Logs 권한 참조 문서

IAM 자격 증명에 연결할 수 있는 액세스 제어 (p. 114) 및 쓰기 권한 정책(자격 증명 기반 정책)을 설정할 때 다음 표를 참조로 사용할 수 있습니다. 표에는 각 CloudWatch Logs API 작업과 이 작업을 수행할 수 있는 권한을 부여할 수 있는 작업이 나와 있습니다. 정책의 `Action` 필드에 작업을 지정합니다. `Resource` 필드에서는 로그 그룹 또는 로그 스트림의 ARN을 지정하거나 `*`를 지정하여 모든 CloudWatch Logs 리소스를 대표할 수 있습니다.

CloudWatch Logs 정책에서 AWS 차원 조건 키를 사용하면 제반 조건을 표시할 수 있습니다. AWS 전체 키의 전체 목록은 다음을 참조하십시오. AWS 글로벌 및 IAM 조건 컨텍스트 키 에서 IAM 사용 설명서.

> **Note**
>
> 작업을 지정하려면 `logs:` 접두사 다음에 API 작업 이름을 사용합니다. 예: . `logs:CreateLogGroup`, `logs:CreateLogStream`, 또는 `logs:*` (전체 CloudWatch Logs 작업).

CloudWatch Logs API 작업 및 작업에 대한 필수 권한

CloudWatch Logs API 작업	필요한 권한(API 작업)
CancelExportTask	`logs:CancelExportTask` 보류 또는 실행 중인 내보내기 작업을 취소하는 데 필요합니다.
CreateExportTask	`logs:CreateExportTask` 로그 그룹에서 Amazon S3 버킷으로 데이터를 내보내는 데 필요합니다.
CreateLogGroup	`logs:CreateLogGroup` 새 보안 그룹을 생성하는 데 필요합니다.
CreateLogStream	`logs:CreateLogStream` 로그 그룹에서 로그 스트림을 새로 생성하는 데 필요합니다.
DeleteDestination	`logs:DeleteDestination` 로그 대상을 삭제하고 이에 대한 모든 구독 필터를 비활성화하는 데 필요합니다.
DeleteLogGroup	`logs:DeleteLogGroup` 로그 그룹과 보관되는 모든 연관 로그 이벤트를 삭제하는 데 필요합니다.
DeleteLogStream	`logs:DeleteLogStream` 로그 스트림과 보관되는 모든 연관 로그 이벤트를 삭제하는 데 필요합니다.
DeleteMetricFilter	`logs:DeleteMetricFilter` 로그 그룹과 연관된 지표 필터를 삭제하는 데 필요합니다.

CloudWatch Logs API 작업	필요한 권한(API 작업)
DeleteQueryDefinition	`logs:DeleteQueryDefinition` CloudWatch Logs Insights에서 저장된 쿼리 정의를 삭제하는 데 필요합니다.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` CloudWatch Logs 리소스 정책을 삭제하는 데 필요합니다.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` 로그 그룹의 보존 정책을 삭제하는 데 필요합니다.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` 로그 그룹과 연관된 구독 필터를 삭제하는 데 필요합니다.
DescribeDestinations	`logs:DescribeDestinations` 계정과 연결된 모든 대상들을 보는 데 필요합니다.
DescribeExportTasks	`logs:DescribeExportTasks` 계정과 연관된 모든 내보내기 작업들을 보는 데 필요합니다.
DescribeLogGroups	`logs:DescribeLogGroups` 계정과 연관된 모든 로그 그룹들을 보는 데 필요합니다.
DescribeLogStreams	`logs:DescribeLogStreams` 로그 그룹과 연관된 모든 로그 스트림을 보는 데 필요합니다.
DescribeMetricFilters	`logs:DescribeMetricFilters` 로그 그룹과 연관된 모든 지표를 보는 데 필요합니다.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` CloudWatch Logs Insights에서 저장된 쿼리 정의의 목록을 보는 데 필요합니다.
DescribeQueries	`logs:DescribeQueries` 예약되었거나, 실행 중이거나, 최근에 중단된 CloudWatch Logs Insights 쿼리의 목록을 보는 데 필요합니다.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` CloudWatch Logs 리소스 정책의 목록을 보는 데 필요합니다.

CloudWatch Logs API 작업	필요한 권한(API 작업)
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` 로그 그룹과 연관된 모든 구독 필터를 보는 데 필요합니다.
FilterLogEvents	`logs:FilterLogEvents` 로그 그룹 필터 패턴에 따라 로그 이벤트를 정렬하는 데 필요합니다.
GetLogEvents	`logs:GetLogEvents` 로그 스트림에서 로그 이벤트를 검색하는 데 필요합니다.
GetLogGroupFields	`logs:GetLogGroupFields` 로그 그룹의 로그 이벤트에 포함된 필드 목록을 검색하는 데 필요합니다.
GetLogRecord	`logs:GetLogRecord` 단일 로그 이벤트에서 세부 정보를 검색하는 데 필요합니다.
GetQueryResults	`logs:GetQueryResults` CloudWatch Logs Insights 쿼리의 결과를 검색하는 데 필요합니다.
ListTagsLogGroup	`logs:ListTagsLogGroup` 로그 그룹과 연결된 태그를 나열하는 데 필요합니다.
PutDestination	`logs:PutDestination` 대상 로그 스트림(예: Kinesis 스트림)을 생성 또는 업데이트하는 데 필요합니다.
PutDestinationPolicy	`logs:PutDestinationPolicy` 기존 로그 대상과 연관된 액세스 정책을 생성 또는 업데이트하는 데 필요합니다.
PutLogEvents	`logs:PutLogEvents` 로그 스트림에서 로그 이벤트 배치를 업로드하는 데 필요합니다.
PutMetricFilter	`logs:PutMetricFilter` 지표 필터를 생성 또는 업데이트하고 이를 로그 그룹과 연관시키는 데 필요합니다.
PutQueryDefinition	`logs:PutQueryDefinition` CloudWatch Logs Insights에 쿼리를 저장하는 데 필요합니다.

CloudWatch Logs API 작업	필요한 권한(API 작업)
PutResourcePolicy	`logs:PutResourcePolicy` CloudWatch Logs 리소스 정책을 만드는 데 필요합니다.
PutRetentionPolicy	`logs:PutRetentionPolicy` 로그 그룹에 로그 이벤트를 유지하는 일수(보존 일수)를 설정하는 데 필요합니다.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` 구독 필터를 생성 또는 업데이트하고 이를 로그 그룹과 연관시키는 데 필요합니다.
StartQuery	`logs:StartQuery` CloudWatch Logs Insights 쿼리를 시작하는 데 필요합니다.
StopQuery	`logs:StopQuery` 진행 중인 CloudWatch Logs Insights 쿼리를 중지하는 데 필요합니다.
TagLogGroup	`logs:TagLogGroup` 로그 그룹 태그를 추가하거나 업데이트하는 데 필요합니다.
TestMetricFilter	`logs:TestMetricFilter` 로그 이벤트 메시지 샘플을 기준으로 필터 패턴을 테스트하는 데 필요합니다.

CloudWatch Logs에 서비스 연결 역할 사용

Amazon CloudWatch Logs에서는 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 CloudWatch Logs에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 CloudWatch Logs에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다.

서비스 연결 역할이 있으면 CloudWatch Logs 필요한 권한을 수동으로 추가할 필요가 없기 때문에 효율성이 더 높습니다. CloudWatch Logs 서비스 연결 역할의 권한을 정의하고 달리 정의되지 않은 경우 CloudWatch Logs 는 이러한 역할을 맡을 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함됩니다. 해당 권한 정책은 다른 IAM 법인.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 IAM으로 작업하는 AWS 서비스 단원을 참조하십시오. 다음과 같은 서비스를 찾아보십시오. 예 에서 서비스 연결 역할 열. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

CloudWatch Logs에 대한 서비스 연결 역할 권한

CloudWatch Logs 는 서비스 연결 역할을 사용합니다. AWS서비스역할로그배달. CloudWatch Logs 는 이 서비스 연결 역할을 사용하여 로그를 직접 Kinesis Data Firehose. 자세한 정보는 로그 직접 보내기 Amazon S3 또는 Kinesis Data Firehose (p. 98) 단원을 참조하십시오.

더 AWS서비스역할로그배달 서비스 연결 역할은 다음 서비스를 신뢰하여 역할을 맡습니다.

- `CloudWatch Logs`

역할 권한 정책은 CloudWatch Logs가 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업 `firehose:PutRecord` 및 `firehose:PutRecordBatch` 모든 Kinesis Data Firehose 태그가 있는 스트림 `LogDeliveryEnabled` 키(값: `True`. 이 태그는 Kinesis Data Firehose 스트림을 스트리밍할 수 있습니다. Kinesis Data Firehose.

다음을 허용하려면 권한을 구성해야 합니다. IAM 엔티티를 사용하여 서비스 연결 역할 을 생성, 편집 또는 삭제할 수 있습니다. 이 엔티티는 사용자, 그룹 또는 역할일 수 있습니다. 자세한 내용은 을 참조하십시오. 서비스 링크된 역할 권한 에서 IAM 사용 설명서.

CloudWatch Logs에 대한 서비스 연결 역할 생성

서비스 연결 역할 을 수동으로 생성할 필요는 없습니다. 로그가 로 직접 전송되도록 설정할 때 Kinesis Data Firehose 스트림 AWS Management 콘솔, AWS CLI또는 AWS API, CloudWatch Logs 는 에서 서비스 연결 역할을 만듭니다.

이 서비스 연결 역할을 삭제한 다음 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. 다시 설정한 로그는 Kinesis Data Firehose 스트림, CloudWatch Logs 은(는) 사용자를 위한 서비스 연결 역할을 다시 생성합니다.

CloudWatch Logs에 대한 서비스 연결 역할 편집

CloudWatch Logs 은(는) 편집을 허용하지 않습니다. AWS서비스역할로그배달또는 기타 서비스 연결 역할을 생성할 수 없습니다. 다양한 엔터티가 역할을 참조할 수 있으므로 역할의 이름을 변경할 수 없습니다. 그러나 IAM를 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하십시오.

CloudWatch Logs에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 이렇게 하면 적극적으로 모니터링하거나 유지 관리하지 않은 미사용 엔터티가 없습니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

> Note
>
> 리소스를 삭제하려 할 때 CloudWatch Logs 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하십시오.

삭제하려면 CloudWatch Logs 리소스 사용 AWSServiceRoleForLogDelivery 서비스 연결 역할

- 로그 직접 보내기 중지 Kinesis Data Firehose 스트림.

를 사용하여 서비스 연결 역할을 수동으로 삭제하려면 IAM

사용 방법 IAM 콘솔, AWS CLI또는 AWS API를 사용하여 AWS서비스역할로그배달 서비스 연결 역할. 자세한 내용은 을 참조하십시오. 서비스 연결 역할 삭제

CloudWatch Logs 서비스 연결 역할을 지원하는 리전

CloudWatch Logs에서는 서비스를 사용할 수 있는 모든 AWS 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 CloudWatch Logs 리전 및 엔드포인트 단원을 참조하십시오.

Amazon CloudWatch Logs에 대한 규정 준수 확인

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon CloudWatch Logs의 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램의 범위 내에 있는 AWS 서비스 목록은 규정 준수 프로그램 제공 범위 내 AWS 서비스를 참조하십시오. 일반적인 내용은 AWS 규정 준수 프로그램을 참조하십시오.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하십시오.

Amazon CloudWatch Logs 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 – 이 배포 가이드에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 AWS 환경을 배포하기 위한 단계를 제공합니다.
- HIPAA 보안 및 규정 준수 기술 백서 설계 – 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 만드는 방법을 설명합니다.
- AWS 규정 준수 리소스 – 이 워크북 및 안내서 모음은 귀사가 속한 업계 및 위치에 적용될 수 있습니다.
- AWS Config Developer Guide의 규칙을 사용하여 리소스 평가 – AWS Config는 리소스 구성이 내부 사례, 업계 지침, 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub – 이 AWS 제품으로 보안 업계 표준 및 모범 사례 규정 준수 여부를 확인하는 데 도움이 되는 AWS 내 보안 상태에 대한 포괄적인 관점을 제공합니다.

Amazon CloudWatch Logs의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하십시오.

Amazon CloudWatch Logs의 인프라 보안

관리형 서비스인 Amazon CloudWatch Logs는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon CloudWatch Logs에 액세스합니다. 클라이언트가 TLS(전송 계층 보안) 1.0 이상을 지원해야 합니다. TLS 1.2 이상을 권장합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 PFS(전달 완전 보안, Perfect Forward Secrecy)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 연결된 보안 액세스 키를 사용해 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

인터페이스 VPC 엔드포인트와 함께 CloudWatch Logs 사용

Amazon Virtual Private Cloud(Amazon VPC)를 사용하여 AWS 리소스를 호스팅하는 경우, VPC와 CloudWatch Logs 간에 프라이빗 연결을 설정할 수 있습니다. 이 연결을 사용하여 인터넷을 통하지 않고 CloudWatch Logs에 로그를 보낼 수 있습니다.

Amazon VPC란 사용자가 정의한 가상 네트워크에서 AWS 리소스를 시작할 때 사용할 수 있는 AWS 서비스입니다. VPC가 있으면 IP 주소 범위, 서브넷, 라우팅 테이블, 네트워크 게이트웨이 등 네트워크 설정을 제어할 수 있습니다. VPC를 CloudWatch Logs에 연결하려면 CloudWatch Logs에 대해 인터페이스 VPC 엔드포인트를 정의하십시오. 이 유형의 엔드포인트를 사용하여 VPC를 AWS 서비스에 연결할 수 있습니다. 이 엔드포인트를 이용하면 인터넷 게이트웨이나 NAT(네트워크 주소 변환) 인스턴스 또는 VPN 연결 없이도 CloudWatch Logs에 안정적이고 확장 가능하게 연결됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC란 무엇입니까? 단원을 참조하십시오.

인터페이스 VPC 엔드포인트는 프라이빗 IP 주소와 함께 탄력적 네트워크 인터페이스를 사용하여 AWS 서비스 간 프라이빗 통신을 사용할 수 있는 AWS 기술인 AWS PrivateLink에 의해 구동됩니다. 자세한 내용은 새 기능 - AWS 서비스를 위한 AWS PrivateLink 단원을 참조하십시오.

다음은 Amazon VPC 사용자를 위한 단계들입니다. 자세한 내용은 시작하기(출처: Amazon VPC 사용 설명서)를 참조하십시오.

가용성

현재 CloudWatch Logs가 VPC 엔드포인트를 지원하는 리전은 다음과 같습니다.

- 미국 동부(오하이오)
- 미국 동부(버지니아 북부)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 아시아 태평양(홍콩)
- 아시아 태평양(뭄바이)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 남아메리카(상파울루)
- AWS GovCloud(US-East)
- AWS GovCloud (US-West)

CloudWatch Logs용 VPC 엔드포인트 생성

VPC에서 CloudWatch Logs를 사용하기 시작하려면 CloudWatch Logs에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 선택할 서비스 이름은 com.amazonaws.***Region***.logs입니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 생성을 참조하십시오.

CloudWatch Logs에 대해 설정을 변경할 필요가 없습니다. CloudWatch Logs는 퍼블릭 엔드포인트 또는 프라이빗 인터페이스 VPC 엔드포인트 중 사용 중인 엔드포인트를 사용하여 다른 AWS 서비스를 호출합니다. 예를 들어, CloudWatch Logs용 인터페이스 VPC 엔드포인트를 생성할 때, Kinesis Data Streams용 CloudWatch Logs 구독 필터와 Kinesis Data Streams용 인터페이스 VPC 엔드포인트가 이미 있는 경우에는 CloudWatch Logs과 Kinesis Data Streams 사이의 호출이 인터페이스 VPC 엔드포인트를 통해 흐르기 시작합니다.

VPC와 CloudWatch Logs 간의 연결 테스트

엔드포인트를 생성한 후에 연결을 테스트 할 수 있습니다.

VPC와 CloudWatch Logs 엔드포인트 간 연결을 테스트하려면

1. VPC에 있는 Amazon EC2 인스턴스에 연결합니다. 연결에 대한 자세한 내용은 Amazon EC2 설명서의 Linux 인스턴스에 연결이나 Windows 인스턴스에 연결을 참조하십시오.
2. 인스턴스에서 AWS CLI를 사용하여 기존 로그 그룹 중 하나에 로그 항목을 생성합니다.

   먼저 로그 이벤트 JSON 파일을 생성합니다. 타임스탬프는 1970년 1월 1일 1:1970 00:00:00 UTC 이후 경과된 시간(밀리초)으로 지정해야 합니다.

   ```
   [
     {
       "timestamp": 1533854071310,
       "message": "VPC Connection Test"
     }
   ]
   ```

   그런 후 `put-log-events` 명령을 사용하여 로그 항목을 생성합니다.

   ```
   aws logs put-log-events --log-group-name LogGroupName --log-stream-name LogStreamName
    --log-events file://JSONFileName
   ```

   명령이 성공해 VPC 엔드포인트를 사용할 수 있는 상태가 되었다면 명령에 대한 응답에 `nextSequenceToken` 명령이 포함됩니다.

CloudWatch Logs VPC 엔드포인트에 대한 액세스 제어

VPC 엔드포인트 정책은 엔드포인트를 만들거나 수정 시 엔드포인트에 연결하는 IAM 리소스 정책입니다. 엔드포인트를 만들 때 정책을 추가하지 않으면 서비스에 대한 모든 액세스를 허용하는 기본 정책이 추가됩니다. 엔드포인트 정책은 IAM 사용자 정책 또는 서비스별 정책을 무시하거나 교체하지 않습니다. 이는 엔드포인트에서 지정된 서비스로의 액세스를 제어하기 위한 별도의 정책입니다.

엔드포인트 정책은 JSON 형식으로 작성해야 합니다.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 통해 서비스에 대한 액세스 제어 단원을 참조하십시오.

다음은 CloudWatch Logs에 대한 엔드포인트 정책 예제입니다. 이 정책에서는 사용자가 VPC를 통해 CloudWatch Logs에 연결하여 로그 스트림을 생성하고 CloudWatch Logs에 로그를 전송할 수 있도록 허용하고, 다른 CloudWatch Logs 작업을 수행하지 못하게 금지합니다.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
```

```
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

CloudWatch Logs에 대한 VPC 엔드포인트 정책을 수정하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 엔드포인트를 선택합니다.
3. CloudWatch Logs에 대한 엔드포인트를 아직 생성하지 않았다면 엔드포인트 생성을 선택합니다. 그런 다음 com.amazonaws.***Region***.logs를 선택하고 엔드포인트 생성을 선택합니다.
4. com.amazonaws.***Region***.logs 엔드포인트를 선택하고 화면 하단의 정책 탭을 선택합니다.
5. 정책 편집을 선택하고 정책을 변경합니다.

VPC 컨텍스트 키에 대한 지원

CloudWatch Logs는 특정 VPC 또는 특정 VPC 엔드포인트에 대한 액세스를 제한할 수 있는 `aws:SourceVpc` 및 `aws:SourceVpce` 컨텍스트 키를 지원합니다. 이러한 키는 사용자가 VPC 엔드포인트를 사용하고 있을 때만 작동합니다. 자세한 내용은 IAM 사용 설명서의 일부 서비스에 사용 가능한 키를 참조하십시오.

AWS CloudTrail에서 Amazon CloudWatch Logs API 호출 로깅

Amazon CloudWatch Logs은 CloudWatch Logs에서 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 AWS 계정에 의해 실행되거나 AWS 계정을 대신하여 실행되는 API 호출을 기록합니다. 캡처되는 호출에는 CloudWatch 콘솔로부터의 호출과 CloudWatch Logs API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 CloudWatch Logs에 대한 이벤트를 비롯하여 CloudTrail 이벤트를 Amazon S3 버킷으로 지속적으로 배포할 수 있습니다. 추적을 구성하지 않은 경우 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. CloudTrail에서 수집하는 정보를 사용하여 CloudWatch Logs에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

그 구성 및 활성화 방법을 포함하여 CloudTrail에 대한 자세한 내용은 AWS CloudTrail User Guide를 참조하십시오.

주제

- CloudTrail의 CloudWatch Logs 정보 (p. 132)
- 로그 파일 항목 이해 (p. 133)

CloudTrail의 CloudWatch Logs 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. 지원되는 이벤트 활동이 CloudWatch Logs에서 이루어지면 해당 활동이 이벤트 이력의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록에서 이벤트 보기를 참조하십시오.

CloudWatch Logs 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려는 경우 추적을 생성합니다. 추적은 CloudTrail이 Amazon S3 버킷으로 로그 파일을 전송할 수 있도록 합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 정보는 다음을 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

CloudWatch Logs는 CloudTrail 로그 파일의 이벤트로 다음 작업의 로깅을 지원합니다.

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream

- DeleteMetricFilter
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

다음 CloudWatch Logs API 작업에서는 요청 요소만 CloudTrail에 저장됩니다.

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeSubscriptionFilters
- GetLogGroupFields
- GetLogRecord

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 AWS Identity and Access Management(IAM) 사용자 자격 증명으로 했는지 여부
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하십시오.

로그 파일 항목 이해

추적은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 제공할 수 있도록 해 주는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 어떤 소스로부터의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 포함되어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 주문 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

아래 로그 파일 항목은 사용자가 CloudWatch Logs CreateExportTask 작업을 호출했음을 보여줍니다.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
```

```
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

CloudWatch Logs 에이전트 참조

Important

이 참조는 이전 CloudWatch Logs agent(사용 중단 경로)를 클릭합니다(). 통합 CloudWatch 대신 에이전트입니다. 해당 에이전트에 대한 자세한 내용은 다음을 참조하십시오. 에서 메트릭 및 로그 수집 Amazon EC2 인스턴스 및 온-프레미스 서버를 CloudWatch 에이전트.

CloudWatch Logs 에이전트는 Amazon EC2 인스턴스에서 CloudWatch Logs로 로그 데이터를 자동 전송할 수 있게 해줍니다. 에이전트에는 다음 구성 요소가 포함되어 있습니다.

- CloudWatch Logs에 로그 데이터를 푸시하기 위한 AWS CLI 플러그인 기능입니다.
- CloudWatch Logs로 데이터를 푸시하는 프로세스를 시작하는 스크립트(데몬)입니다.
- 데몬이 항상 실행되도록 하는 cron 작업입니다.

에이전트 구성 파일

CloudWatch Logs 에이전트 구성 파일은 CloudWatch Logs 에이전트에 필요한 정보를 설명합니다. 에이전트 구성 파일의 [general] 섹션은 모든 로그 스트림에 적용되는 공통 구성을 정의합니다. [logstream] 섹션은 원격 로그 스트림에 로컬 파일을 전송하는 데 필요한 정보를 정의합니다. [logstream] 섹션은 하나 이상 정의가 가능하지만, 각 섹션은 구성 파일 내에 고유한 이름을 가져야 합니다(예: [logstream1], [logstream2] 등). 로그 파일에 있는 데이터의 첫 줄과 함께 [logstream] 값은 로그 파일의 ID를 정의합니다.

```
[general]
state_file = value
logging_config_file = value
use_gzip_http_content_encoding = [true | false]

[logstream1]
log_group_name = value
log_stream_name = value
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

상태 파일

상태 파일이 저장되는 장소를 지정합니다.

logging_config_file

(선택 사항) 에이전트 로깅 구성 파일의 위치를 지정합니다. 여기에서 에이전트 로깅 구성 파일을 지정하지 않으면 기본 파일인 awslogs.conf가 사용됩니다. 스크립트를 통해 에이전트를 설치한 경우 기

본 파일 위치는 `/var/awslogs/etc/awslogs.conf`이고, rpm을 통해 에이전트를 설치한 경우 `/etc/awslogs/awslogs.conf`입니다. 이 파일은 Python 구성 파일 형식을 가지고 있습니다(https://docs.python.org/2/library/logging.config.html#logging-config-fileformat). 다음 이름을 가진 로거를 사용자가 지정할 수 있습니다.

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

아래 샘플은 기본 값이 INFO인 경우 독자와 게시자의 수준을 WARNING으로 수준을 변경합니다.

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

true(기본)로 설정되어 있으면 gzip http 콘텐츠 인코딩을 활성화하여 CloudWatch Logs로 압축된 페이로드를 전송합니다. 이렇게 하면 CPU 사용량을 줄이고 NetworkOut을 낮추며 PUT 레코드 지연 시간을 줄일 수 있습니다. 이 기능을 비활성화하려면 사용_gzip_http_콘텐츠_인코딩 = 거짓 을(를) [일반] 섹션 CloudWatch Logs 에이전트 구성 파일을 클릭한 다음 에이전트를 다시 시작합니다.

Note

이 설정은 awscli-cwlogs 버전 1.3.3 이상에서만 사용할 수 있습니다.

log_group_name

대상 로그 그룹을 지정합니다. 로그 그룹이 없는 경우 이 설정에 따라 로그 그룹이 자동으로 생성됩니다. 로그 그룹 이름에 포함되는 문자 길이는 1~512자입니다. 허용되는 문자: a-z, A-Z, 0-9, '_'(밑줄), '- (하이픈), '/'(슬래시) 및 '.'(마침표)

log_stream_name

대상 로그 스트림을 지정합니다. 리터럴 문자열이나 미리 정의된 변수({instance_id}, {hostname}, {ip_address}) 또는 이 둘의 조합을 사용하여 로그 스트림 이름을 정의할 수 있습니다. 로그 스트림이 없는 경우 이 설정에 따라 로그 스트림이 자동으로 생성됩니다.

datetime_format

타임스탬프가 로그에서 추출되는 방법을 지정합니다. 타임스탬프는 로그 이벤트를 검색하고 지표를 생성하는 데 사용됩니다. datetime_format이 제공되지 않는 경우에는 각 로그 이벤트에서 현재 시간이 사용됩니다. 제공된 datetime_format 값이 해당 로그 메시지에서 유효하지 않으면 타임스탬프가 성공적으로 구문 분석된 마지막 로그 이벤트에서 나온 타임스탬프가 사용됩니다. 이전 로그 이벤트가 존재하지 않는 경우 현재 시간이 사용됩니다.

공통적인 datetime_format 코드가 아래에 나열되어 있습니다. Python, datetime.strptime()에서 지원되는 datetime_format 코드면 무엇이든 사용 가능합니다. 시간대 오프셋(%z)도 지원되기는 하지만, 콜론(:)이 없는 python 3.2, [+-]HHMM이 나올 때까지는 지원되지 않습니다. 자세한 내용은 strftime() 및 strptime() 동작을 참조하십시오.

y 제로 패딩된 10진수 형태의 세기를 제외한 연도 00, 01, ..., 99

전년 대비 %: 100년을 십진수로 표기.1970, 1988, 2001, 2013

b 로케일의 단축명 로 월. Jan, Feb, ..., Dec (en_US);

%억: 로케일의 전체 이름으로 월 January, February, ..., December (en_US);

%분: 제로 패딩된 10진수 형태의 월 01, 02, ..., 12

%d개: 제로 패딩된 10진수 형태의 월 날짜 01, 02, ..., 31

%시간: 제로 패디드 십진수 로서 시간(24시간 시계). 00, 01, ..., 23

%I개: 제로 패디드 십진수 로서 시간(12시간제). 01, 02, ..., 12

<p> 로케일은 AM 또는 PM과 동일합니다.

%백만: 0으로 채워진 소수점 숫자 의 분. 00, 01, ..., 59

%S(%S): 두 번째는 제로 패디드 소수입니다. 00, 01, ..., 59

f 마이크로초를 소수점 숫자로 표시하고, 좌측에 제로 패디드. 000000, ..., 999999

%z개: +HHMM 또는 -HHMM 형식의 UTC 오프셋. +0000, -0400, +1030

형식의 예:

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

로그 이벤트 타임스탬프의 시간대를 지정합니다. UTC 및 LOCAL 등 2개의 값이 지원됩니다. 기본값인 LOCAL은 datetime_format에 따라 시간대를 추정할 수 없을 때 사용됩니다.

--file

CloudWatch Logs에 푸시하고 싶은 로그 파일을 지정합니다. 파일은 특정 파일을 가리키거나 여러 개의 파일을 가리킬 수 있습니다(/var/log/system.log* 같은 와일드카드를 사용). 파일 수정 시간에 따라 최신 파일만 CloudWatch Logs로 푸시됩니다. 와일드카드는 여러 종류의 파일(예: access_log_80 및 access_log_443)이 아니라 종류가 같은 일련의 파일(예: access_log.2014-06-01-01, access_log.2014-06-01-02 등)을 지정할 때 사용하는 것이 좋습니다. 여러 종류의 파일을 지정하려면 로그 파일의 종류에 따라 다른 로그 스트림에 들어가도록 구성 파일에 또 다른 로그 스트림 항목을 추가합니다. 압축 파일은 지원되지 않습니다.

file_fingerprint_lines

파일을 식별하기 위한 줄의 범위를 지정합니다. 유효한 값은 하나의 숫자(예: '1')나 대시로 구분된 두 개의 숫자(예: '2-5')입니다. 기본값은 '1'이기 때문에 지문 산출을 위해 첫 번째 줄이 사용됩니다. 지정된 모든 줄이 사용 가능한 상태가 되지 않는 한, 지문 줄은 CloudWatch Logs로 전송되지 않습니다.

multi_line_start_pattern

로그 메시지의 시작을 식별하기 위해 패턴을 지정합니다. 로그 메시지는 패턴과 일치하는 하나의 줄과 패턴과 일치하지 않는 나머지 줄들로 이루어져 있습니다. 유효한 값은 정규식 또는 {datetime_format}입니다. {datetime_format}을 사용할 때는 반드시 datetime_format 옵션이 지정되어 있어야 합니다. 기본값은 '^[^\s]'이기 때문에 공백이 아닌 문자로 시작되는 줄이 있으면 이전의 로그 메시지가 종료되고 새로운 로그 메시지가 시작됩니다.

initial_position

데이터 읽기를 시작할 지점(start_of_file 또는 end_of_file)을 지정합니다. 기본값은 파일의 시작 지점입니다. 해당 로그 스트림에서 지속되는 상태가 없을 때만 사용됩니다.

encoding

파일을 정확하게 읽을 수 있도록 로그 파일의 인코딩을 설정합니다. 기본값은 utf_8입니다. Python codecs.decode()에서 지원되는 인코딩을 여기에서 사용할 수 있습니다.

Warning

인코딩을 잘못 지정하면 디코딩이 불가능한 문자를 다른 문자들이 대체하면서 데이터 손실이 야기될 수 있습니다.

다음은 몇 가지 일반적인 인코딩입니다.

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737, cp775,
cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862, cp863, cp864,
cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950, cp1006, cp1026,
cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255, cp1256, cp1257,
cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr, gb2312, gbk, gb18030,
hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2, iso2022_jp_2004, iso2022_jp_3,
iso2022_jp_ext, iso2022_kr, latin_1, iso8859_2, iso8859_3, iso8859_4,
iso8859_5, iso8859_6, iso8859_7, iso8859_8, iso8859_9, iso8859_10,
iso8859_13, iso8859_14, iso8859_15, iso8859_16, johab, koi8_r, koi8_u,
mac_cyrillic, mac_greek, mac_iceland, mac_latin2, mac_roman, mac_turkish,
ptcp154, shift_jis, shift_jis_2004, shift_jisx0213, utf_32, utf_32_be,
utf_32_le, utf_16, utf_16_be, utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

로그 이벤트를 일괄 처리하는 기간을 지정합니다. 최소값은 5000ms이고, 기본값은 5000ms입니다.

batch_count

일괄 처리할 로그 이벤트의 최대 수를 지정합니다(10000까지 가능). 기본값은 10000입니다.

batch_size

일괄 처리할 로그 이벤트의 최대 크기를 바이트로 지정합니다(1048576바이트까지 가능). 기본값은 1048576바이트입니다. 이 크기는 UTF-8에서 모든 이벤트 메시지를 합한 값에 각 로그 이벤트마다 26바이트를 추가하여 계산한 값입니다.

HTTP 프록시와 함께 CloudWatch Logs 에이전트 사용

CloudWatch Logs 에이전트를 HTTP 프록시와 함께 사용할 수 있습니다.

Note

HTTP 프록시는 awslogs-agent-setup.py 버전 1.3.8 이상에서 지원됩니다.

HTTP 프록시와 함께 CloudWatch Logs 에이전트를 사용하려면

1. 다음 중 하나를 수행합니다.
   a. CloudWatch Logs 에이전트를 새로 설치하려면 아래 명령을 실행합니다.

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
```

   EC2 인스턴스에서 Amazon EC2 메타데이터 서비스에 대한 액세스를 유지하려면 --no-proxy 169.254.169.254를 사용하십시오(권장 사항). 자세한 내용은 을 참조하십시오. 인스턴스 메타데이터 및 사용자 데이터 에서 Linux 인스턴스용 Amazon EC2 사용 설명서.

   `http-proxy` 및 `https-proxy` 값에서 전체 URL을 지정합니다.

   b. 기존에 CloudWatch Logs 에이전트가 설치되어 있는 경우에는 /var/awslogs/etc/proxy.conf를 편집해서 프록시를 추가하십시오.

```
HTTP_PROXY=
HTTPS_PROXY=
NO_PROXY=
```

2. 에이전트를 다시 시작하면 변경 사항이 적용됩니다.

```
sudo service awslogs restart
```

   Amazon Linux 2를 사용 중인 경우 다음 명령과 함께 에이전트를 다시 시작합니다.

```
sudo service awslogsd restart
```

CloudWatch Logs 에이전트 구성 파일 분류

awslogs-agent-setup.py 버전 1.3.8 이상을 awscli-cwlogs 1.3.3 이상과 함께 사용하면 /var/awslogs/etc/config/ 디렉터리에서 추가 구성 파일을 생성하여 다양한 구성 요소들에 대해 서로 다른 스트림 구성을 독립

적으로 가져올 수 있습니다. CloudWatch Logs 에이전트가 시작될 때 이러한 추가 구성 파일에 스트림 구성이 포함됩니다. [general] 섹션의 구성 속성들은 반드시 기본 구성 파일(/var/awslogs/etc/awslogs.conf)에 정의되어 있어야 하며, /var/awslogs/etc/config/에 있는 추가 구성 파일에서는 무시가 됩니다.

rpm을 통해 에이전트를 설치하지 않았기 때문에 /var/awslogs/etc/config/ 디렉터리가 없는 경우 /etc/awslogs/config/ 디렉터리를 대신 사용할 수 있습니다.

에이전트를 다시 시작하면 변경 사항이 적용됩니다.

```
sudo service awslogs restart
```

Amazon Linux 2를 사용 중인 경우 다음 명령과 함께 에이전트를 다시 시작합니다.

```
sudo service awslogsd restart
```

CloudWatch Logs 에이전트 FAQ

어떤 종류의 파일 로테이션이 지원됩니까?

다음과 같은 파일 로테이션 메커니즘이 지원되고 있습니다.

- 숫자 접미사를 붙여서 기존 로그 파일의 이름을 바꾸고 원래 빈 로그 파일을 다시 생성하는 방법입니다. 예를 들어 /var/log/syslog.log는 /var/log/syslog.log.1로 이름이 바뀝니다. 이전 로테이션에서 /var/log/syslog.log.1이 이미 존재하는 경우에는 /var/log/syslog.log.2로 이름이 바뀝니다.
- 복사본을 생성한 후에 원래 로그 파일을 잘라냅니다. 예를 들어 /var/log/syslog.log는 /var/log/syslog.log.1에 복사가 된 후 잘립니다. 이 경우 데이터 손실이 발생할 수 있기 때문에 이 파일 로테이션 메커니즘을 사용할 때는 조심해야 합니다.
- 기존 파일과 같은 공통 패턴을 따르는 파일을 새로 생성하는 방법입니다. 예를 들어 /var/log/syslog.log.2014-01-01이 그대로 남아 있는 상태에서 /var/log/syslog.log.2014-01-02가 생성됩니다.

파일의 지문(소스 ID)은 로그 스트림 키와 파일 콘텐츠의 첫 줄을 해싱하여 산출됩니다. 이 동작을 재정의하기 위해 file_fingerprint_lines 옵션을 사용할 수 있습니다. 파일 로테이션이 일어나면 새 파일은 새 콘텐츠를 가진 것으로 간주되지만, 기존 파일은 콘텐츠가 추가된 것으로 간주되지 않습니다. 따라서 에이전트는 기존 파일에 대한 읽기를 마치고 나면 새 파일을 푸시합니다.

사용 중인 에이전트의 버전을 어떻게 확인할 수 있습니까?

설정 스크립트를 사용해 CloudWatch Logs 에이전트를 설치했다면 /var/awslogs/bin/awslogs-version.sh를 사용해 사용 중인 에이전트의 버전을 확인할 수 있습니다. 에이전트의 버전과 중요한 플러그인들이 출력됩니다. yum을 사용해 CloudWatch Logs 에이전트를 설치했다면 "yum info awslogs" 및 "yum info aws-cli-plugin-cloudwatch-logs"를 사용해 CloudWatch Logs 에이전트의 버전과 플러그인을 확인할 수 있습니다.

로그 항목들은 어떻게 로그 이벤트로 변환됩니까?

로그 이벤트에는 이벤트가 발생한 시점에 대한 타임스탬프와 원시 로그 메시지 등 두 개의 속성이 포함되어 있습니다. 기본적으로 공백이 아닌 문자로 시작되는 줄이 있으면 이전의 로그 메시지가 종료되고 새로운 로그 메시지가 시작됩니다. 이 동작을 재정의하기 위해 multi_line_start_pattern을 사용할 수 있으면 패턴과 일치하는 모든 줄에서 새로운 로그 메시지가 시작됩니다. 어떤 regex 또는 {datetime_format}이든 패턴이 될 수 있습니다. 예를 들어 모든 로그 메시지의 첫 줄에 '2014-01-02T13:13:01Z' 같은 타임스탬프가 포함되어 있으면 multi_line_start_pattern을 '\d\{4\}-\d\{2\}-\d\{2\}T\d\{2\}:\d\{2\}:\d\{2\}Z'로 설정할 수 있습니다. 간편한 구성을 위해 datetime_format option이 지정되어 있는 경우에 '{datetime_format}' 변수를 사용할 수 있습니다. 같은 예에서 datetime_format이 '%Y-%m-%dT%H:%M:%S%z'로 설정되어 있으면 multi_line_start_pattern이 간단히 '{datetime_format}'이 될 수 있습니다.

datetime_format이 제공되지 않는 경우에는 각 로그 이벤트에서 현재 시간이 사용됩니다. 제공된 datetime_format 값이 해당 로그 메시지에서 유효하지 않으면 타임스탬프가 성공적으로 구문 분석된 마

지막 로그 이벤트에서 나온 타임스탬프가 사용됩니다. 이전 로그 이벤트가 존재하지 않는 경우 현재 시간이 사용됩니다. 로그 이벤트가 현재 시간이나 이전 로그 이벤트 시간으로 돌아가면 경고 메시지가 기록됩니다.

타임스탬프는 로그 이벤트를 검색하고 지표를 생성하는 데 사용되기 때문에 형식을 잘못 지정하면 로그 이벤트 검색이 불가능해지고 잘못된 지표가 생성될 수 있습니다.

로그 이벤트는 어떻게 일괄 처리됩니까?

아래 조건 중 어떤 것이든 충족이 되면 배치(batch)가 가득 차면서 게시가 됩니다.

1. 첫 번째 로그 이벤트가 추가되었기 때문에 buffer_duration 시간이 경과되었습니다.
2. batch_size 보다 작은 로그 이벤트들이 누적되었지만 새로운 로그 이벤트를 추가하면 batch_size를 초과하게 됩니다.
3. 로그 이벤트의 수가 batch_count에 도달했습니다.
4. 배치에서 나온 로그 이벤트들이 24시간 이상 지속되지 않지만, 새 로그 이벤트를 추가하면 24시간이라는 제약 조건을 넘어서게 됩니다.

로그 항목, 로그 이벤트 또는 배치는 어떤 이유로 건너 뛰기가 되거나 잘라집니까?

`PutLogEvents` 작업의 제약 조건을 따르다보면 다음과 같은 문제들로 인해 로그 이벤트나 배치를 건너뛰는 상황이 발생할 수 있습니다.

> **Note**
>
> 데이터를 건너뛰면 CloudWatch Logs 에이전트가 로그에 경고를 기록합니다.

1. 로그 이벤트의 크기가 256KB를 초과하면 로그 이벤트가 완전히 건너 뛰기 됩니다.
2. 로그 이벤트의 타임스탬프가 미래에 2시간 이상이면 해당 로그 이벤트가 건너 뛰기 됩니다.
3. 로그 이벤트의 타임스탬프가 과거에 14일 이상이었으면 해당 로그 이벤트가 건너 뛰기 됩니다.
4. 로그 그룹의 보존 기간을 지난 로그 이벤트가 있으면 전체 배치가 건너 뛰기 됩니다.
5. 단일 `PutLogEvents` 요청에서 로그 이벤트에 대한 배치가 24시간 이상 지속된 경우에는 `PutLogEvents` 작업이 실패합니다.

에이전트 중지로 인해 데이터 손실/중복이 발생하고 있습니까?

상태 파일이 사용 가능하고 마지막 실행 이후로 파일 로테이션이 발생하지 않은 한 그렇지 않습니다. CloudWatch Logs 에이전트는 중지된 지점부터 시작해서 로그 데이터를 계속해서 푸시할 수 있습니다.

같은 호스트나 다른 호스트에서 나온 서로 다른 로그 파일이 동일한 로그 스트림으로 가리키도록 할 수 있습니까?

단일 로그 스트림으로 데이터를 전송하기 위해 여러 개의 로그 소스를 구성하는 것이 불가능합니다.

에이전트가 어떤 API를 호출합니까? (또는 IAM 정책에 어떤 작업을 추가해야 합니까?)

더 CloudWatch Logs 에이전트는 `CreateLogGroup`, `CreateLogStream`, `DescribeLogStreams`, 및 `PutLogEvents` 작업. 최신 에이전트를 사용하고 있는 경우에는 `DescribeLogStreams`가 필요하지 않습니다. IAM 정책 샘플은 아래를 참조하십시오.

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
```

```
  }
 ]
}
```

CloudWatch Logs 에이전트에서 로그 그룹이나 로그 스트림이 자동 생성되도록 하고 싶지 않습니다. 어떻게 하면 에이전트가 로그 그룹 또는 로그 스트림을 다시 생성하지 않게 할 수 있습니까?

귀하의 IAM 에이전트를 다음 작업으로만 제한할 수 있습니다. `DescribeLogStreams`, `PutLogEvents`.

에이전트에서 `CreateLogGroup` 및 `CreateLogStream` 권한을 취소하기 전에 에이전트에서 사용할 로그 그룹과 로그 스트림을 모두 생성해야 합니다. 로그 에이전트는 `CreateLogGroup` 및 `CreateLogStream` 권한이 모두 없는 한 사용자가 생성한 로그 그룹에 로그 스트림을 생성할 수 없습니다.

문제 해결 시 어떤 로그를 살펴봐야 합니까?

에이전트 설치 로그는 `/var/log/awslogs-agent-setup.log`에, 에이전트 로그는 `/var/log/awslogs.log`에 있습니다.

CloudWatch 지표를 통한 사용량 모니터링

CloudWatch Logs는 1분마다 Amazon CloudWatch로 지표를 전송합니다.

CloudWatch Logs 지표

`AWS/Logs` 네임스페이스에는 다음 지표가 포함되어 있습니다.

지표	설명
`IncomingBytes`	CloudWatch Logs로 업로드되는 비압축 바이트 단위의 로그 이벤트 볼륨. `LogGroupName` 차원과 함께 사용하면 로그 그룹으로 업로드되는 비압축 바이트 단위의 로그 이벤트 볼륨이 됩니다. 유효한 차원: LogGroupName 유효한 통계: Sum 단위: 바이트
`IncomingLogEvents`	CloudWatch Logs로 업로드되는 로그 이벤트 수. `LogGroupName` 차원과 함께 사용하면 로그 그룹으로 업로드되는 로그 이벤트 수가 됩니다. 유효한 차원: LogGroupName 유효한 통계: Sum 단위: 없음
`ForwardedBytes`	구독 대상으로 전송되는 압축 바이트 단위의 로그 이벤트 볼륨 유효한 차원: LogGroupName, DestinationType, FilterName 유효한 통계: Sum 단위: 바이트
`ForwardedLogEvents`	구독 대상으로 전송되는 로그 이벤트 수 유효한 차원: LogGroupName, DestinationType, FilterName 유효한 통계: Sum 단위: 없음
`DeliveryErrors`	데이터를 구독 대상으로 전송할 때 CloudWatch Logs로 오류가 수신된 로그 이벤트 수 유효한 차원: LogGroupName, DestinationType, FilterName 유효한 통계: Sum 단위: 없음

지표	설명
`DeliveryThrottling`	데이터를 구독 대상으로 전송할 때 CloudWatch Logs에 병목 현상이 발생한 로그 이벤트 수 유효한 차원: LogGroupName, DestinationType, FilterName 유효한 통계: Sum 단위: 없음

CloudWatch Logs 지표의 차원

다음은 CloudWatch Logs 지표에서 사용할 수 있는 차원입니다.

차원	설명
`LogGroupName`	지표를 표시할 CloudWatch Logs 로그 그룹 이름
`DestinationType`	CloudWatch Logs 데이터의 구독 대상으로서 AWS Lambda, Amazon Kinesis Data Streams 또는 Amazon Kinesis Data Firehose가 될 수 있습니다.
`FilterName`	데이터를 로그 그룹에서 대상으로 전송하는 구독 필터 이름. 구독 필터 이름은 CloudWatch에서 ASCII로 자동 변환되며, 지원되지 않는 문자는 모두 물음표(?)로 바뀝니다.

CloudWatch Logs 할당량

CloudWatch Logs에는 다음과 같은 할당량이 있습니다.

Resource	기본 할당량
배치 크기	1MB(최대). 이 할당량은 변경할 수 없습니다.
데이터 보관	최대 5GB까지 데이터를 무료 보관할 수 있습니다. 이 할당량은 변경할 수 없습니다.
CreateLogStream	50건의 초당 트랜잭션(TPS/계정/리전). 이후 트랜잭션에 병목 현상이 발생합니다. 할당량 증가를 요청할 수 있습니다.
DescribeLogGroups	5건의 초당 트랜잭션(TPS/계정/리전). 할당량 증가를 요청할 수 있습니다.
DescribeLogStreams	5건의 초당 트랜잭션(TPS/계정/리전). 할당량 증가를 요청할 수 있습니다.
검색된 로그 필드	CloudWatch Logs Insights는 로그 그룹에서 최대 1000개의 로그 이벤트 필드를 검색할 수 있습니다. 이 할당량은 변경할 수 없습니다. 자세한 내용은 지원되는 로그 및 검색되는 필드 (p. 33) 항목을 참조하십시오.
JSON 로그에서 추출된 로그 필드	CloudWatch Logs Insights는 JSON 로그에서 최대 100개의 로그 이벤트 필드를 추출할 수 있습니다. 이 할당량은 변경할 수 없습니다. 자세한 내용은 지원되는 로그 및 검색되는 필드 (p. 33) 항목을 참조하십시오.
이벤트 크기	256KB(최대). 이 할당량은 변경할 수 없습니다.
내보내기 작업	계정당 한 번에 하나씩 (실행 중이거나 보류 중) 내보내기 작업이 활성화됩니다. 이 할당량은 변경할 수 없습니다.
FilterLogEvents	5건의 초당 트랜잭션(TPS)/계정/리전. 이 할당량은 변경할 수 없습니다.
GetLogEvents	리전별 계정마다 초당 10개의 요청. 이 할당량은 변경할 수 없습니다. 새로운 데이터를 지속적으로 처리하는 경우 구독을 사용하는 것이 좋습니다. 기록 데이터가 필요한 경우 데이터를 Amazon S3로 내보내는 것이 좋습니다.
수신 데이터	최대 5GB까지 데이터를 무료로 수신할 수 있습니다. 이 할당량은 변경할 수 없습니다.
로그 그룹	리전별 계정당 1,000,000개의 로그 그룹. 할당량 증가를 요청할 수 있습니다. 하나의 로그 그룹에서 포함할 수 있는 로그 스트림의 수에는 할당량이 없습니다.

Resource	기본 할당량
지표 필터	로그 그룹당 100개. 이 할당량은 변경할 수 없습니다.
임베디드 지표 형식 지표	로그 이벤트당 100개의 지표 및 지표당 9개의 차원. 임베디드 지표 형식에 대한 자세한 내용은 사양:을 참조하십시오. 의 임베디드 지표 형식Amazon CloudWatch 사용 설명서.
PutLogEvents	로그 스트림당 초당 5개의 요청. 추가 요청은 제한됩니다. 이 할당량은 변경할 수 없습니다. 요청의 최대 배치 크기는 1MB입니다.PutLogEvents 초당, 계정당, 리전당 1500건의 트랜잭션. 할당량이 초당, 계정당, 리전당 800건의 트랜잭션인 미국 동부(버지니아 북부), 미국 서부(오레곤) 및 유럽(아일랜드) 리전은 제외됩니다.. 할당량 증가를 요청할 수 있습니다.
쿼리 실행 제한 시간	CloudWatch Logs Insights의 쿼리는 15분 후에 시간 초과됩니다. 이 시간 제한은 변경할 수 없습니다.
쿼리된 로그 그룹	하나의 CloudWatch Logs Insights 쿼리에서 최대 20개의 로그 그룹을 쿼리할 수 있습니다. 이 할당량은 변경할 수 없습니다.
쿼리 동시성	동시에 실행 가능한 CloudWatch Logs Insights 쿼리는 최대 10개인데, 여기에는 대시보드에 추가한 쿼리도 포함됩니다. 할당량 증가를 요청할 수 있습니다.
쿼리 결과 가용 시간	쿼리 결과는 7일 동안 검색할 수 있습니다. 이 가용 시간은 변경할 수 없습니다.
콘솔에 표시되는 쿼리 결과	기본적으로 최대 1,000행의 쿼리 결과가 콘솔에 표시됩니다. 쿼리에서 **`limit`** 명령을 사용하여 이 제한을 10,000개 행으로 늘릴 수 있습니다. 자세한 내용은 CloudWatch Logs Insights 쿼리 구문 (p. 39) 항목을 참조하십시오.
저장된 쿼리	계정별로 리전당 최대 1,000개의 CloudWatch Logs Insights 쿼리를 저장할 수 있습니다. 이 할당량은 변경할 수 없습니다.
구독 필터	로그 그룹당 2개. 이 할당량은 변경할 수 없습니다.

문서 기록

다음 표에서는 2018년 6월부터 적용되는 CloudWatch Logs 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

update-history-change	update-history-description	update-history-date
CloudWatch Logs Insights 출시 (p. 147)	CloudWatch Logs Insights를 사용하면 로그 데이터를 대화식으로 검색해 분석할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs User Guide의 CloudWatch Logs Insights로 로그 데이터 분석 단원을 참조하십시오.	November 27, 2018
Amazon VPC 엔드포인트에 대한 지원 (p. 147)	이제 VPC와 CloudWatch Logs 간에 프라이빗 연결을 설정할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs User Guide의 인터페이스 VPC 엔드포인트와 함께 CloudWatch Logs 사용을 참조하십시오.	June 28, 2018

아래 표에 Amazon CloudWatch Logs 사용 설명서의 주요 변경 사항이 설명되어 있습니다.

변경 사항	설명	릴리스 날짜
인터페이스 VPC 엔드포인트	일부 리전에서는 인터페이스 VPC 엔드포인트를 사용하여 Amazon 네트워크에서 Amazon VPC 및 CloudWatch Logs 간의 트래픽을 유지할 수 있습니다. 자세한 내용은 인터페이스 VPC 엔드포인트와 함께 CloudWatch Logs 사용 (p. 129) 단원을 참조하십시오.	2018년 3월 7일
Route 53 DNS 쿼리 로그	CloudWatch Logs를 사용하여 Route 53가 수신하는 DNS 쿼리에 대한 로그를 저장할 수 있습니다. 자세한 정보는 Amazon Route 53 개발자 안내서의 Amazon CloudWatch Logs이란 무엇입니까? (p. 1) 또는 DNS 쿼리 로깅을 참조하십시오.	2017년 9월 7일
로그 그룹 태그 지정	태그를 사용하여 로그 그룹을 분류할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs에서 로그 그룹에 태그 지정 (p. 57) 단원을 참조하십시오.	2016년 12월 13일
콘솔 개선	지표 그래프부터 관련 로그 그룹까지 검색할 수 있습니다. 자세한 내용은 지표에서 로그로 피벗 적용 (p. 56) 단원을 참조하십시오.	2016년 11월 7일
콘솔 활용도 개선	보다 손쉬운 검색, 필터링 및 문제 해결을 위해 경험을 개선했습니다. 예를 들어 날짜 및 시간 범위로 로드 데이터를 필터링할 수 있습니다. 자세한 정보는 CloudWatch Logs에 전송된 로그 데이터 보기 (p. 54) 단원을 참조하십시오.	2016년 8월 29일

변경 사항	설명	릴리스 날짜
Amazon CloudWatch Logs 및 새로운 CloudWatch Logs 지표에 대한 AWS CloudTrail 지원을 추가했습니다.	CloudWatch Logs에 대한 AWS CloudTrail 지원이 추가되었습니다. 자세한 내용은 AWS CloudTrail에서 Amazon CloudWatch Logs API 호출 로깅 (p. 132) 단원을 참조하십시오.	2016년 3월 10일
Amazon S3로의 CloudWatch Logs 내보내기에 대한 지원이 추가되었습니다.	Amazon S3로의 CloudWatch Logs 데이터 내보내기에 대한 지원이 추가되었습니다. 자세한 내용은 Amazon S3로 로그 데이터 내보내기 (p. 99) 단원을 참조하십시오.	2015년 12월 7일
Amazon CloudWatch Logs에서 AWS CloudTrail 기록 이벤트에 대한 지원이 추가되었습니다.	CloudWatch에서 경보를 생성하고, CloudTrail에서 포착된 특정 API 활동에 대한 알림을 수신하며, 이러한 알림을 사용하여 문제를 해결할 수 있습니다.	2014년 11월 10일
Amazon CloudWatch Logs에 대한 지원 추가	Amazon CloudWatch Logs을 사용하여 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스나 다른 원본에서 시스템, 애플리케이션 및 사용자 지정 로그 파일을 모니터링, 저장 및 액세스할 수 있습니다. Amazon CloudWatch 콘솔, AWS CLI의 CloudWatch Logs 명령 또는 CloudWatch Logs SDK를 사용하여 CloudWatch Logs에서 관련 로그 데이터를 가져올 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs이란 무엇입니까? (p. 1) 단원을 참조하십시오.	2014년 7월 10일

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.

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