Amazon CloudWatch

사용 설명서

Amazon CloudWatch: 사용 설명서

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Amazon CloudWatch란 무엇입니까? 1
CloudWatch에 액세스 1
관련 AWS 서비스 1
CloudWatch 작동 방식 2
개념 2
네임스페이스 3
지표 3
차원 4
통계 5
백분위수 7
경보 7
리소스 7
설정 9
Amazon Web Services(AWS)에 가입 9
Amazon CloudWatch 콘솔에 로그인 9
AWS CLI 설정 9
시작하기 10
모든 AWS 서비스의 주요 지표 살펴보기 12
교차 서비스 대시보드에 나타나지 않도록 서비스 제거 13
단일 서비스에 집중 13
리소스 그룹에 집중 14
대시보드 사용 16
대시보드 생성 17
교차 계정 교차 리전 대시보드 17
AWS Management 콘솔을 사용하여 교차 계정 교차 리전 대시보드 생성 및 사용 18
프로그래밍 방식으로 교차 계정 교차 리전 대시보드 생성 18
그래프 추가 또는 제거 20
그래프 이동 또는 크기 조정 22
그래프 편집 22
CloudWatch 대시보드에서 수동으로 지표 그래프 생성 24
그래프 이름 변경 25
텍스트 위젯 추가 또는 제거 25
경보 추가 또는 제거 26
그래프 링크 및 링크 해제 26
즐겨찾기 목록에 대시보드 추가 27
기간 재정의 설정 또는 새로 고침 간격 변경 27
시간 범위 또는 시간대 형식 변경 28
지표 사용 29
얻을 수 있는 지표 보기 29
사용 가능한 지표 검색 32
지표에 대한 통계 얻기 33
특정 리소스에 대한 통계 얻기 33
리소스에서 통계 집계 36
Auto Scaling 그룹별 통계 집계 38
AMI별 통계 집계 39
지표 그래프 작성 40
지표 그래프 작성 40
동적 레이블 사용 43
그래프의 시간 범위 또는 시간대 형식 수정 44
그래프의 Y축 수정 45
그래프의 지표에서 경보 생성 46
사용자 지정 메트릭 게시 47
고분해능 지표 47
차원 사용 47

단일 데이터 요소 게시 48
통계 세트 게시 49
0 값 게시 49
지표 수식 사용 49
CloudWatch 그래프에 수학 표현식 추가 50
지표 수학 구문 및 함수 50
IF 표현식 사용 57
그래프에서 검색 표현식 사용 59
검색 표현식 구문 59
검색 표현식 예제 63
검색 표현식이 있는 그래프 생성 65
경보 사용 67
지표 경보 상태 67
경보 평가 67
경보가 누락 데이터를 처리하는 방법 구성 68
데이터가 누락되었을 때 경보 상태 평가 방법 69
고분해능 경보 71
수학 표현식에 대한 경보 71
백분위수 기반 경보 및 데이터 샘플 부족 71
CloudWatch 경보의 일반적인 기능 71
SNS 주제 설정 72
AWS Management 콘솔을 사용하여 Amazon SNS 주제 설정 72
AWS CLI를 사용하여 SNS 주제 설정 73
정적 임계값을 기반으로 경보 생성 74
이상 탐지에 기반한 경보 생성 76
이상 탐지 모델 수정 78
이상 탐지 모델 삭제 78
지표 수학 표현식을 기반으로 경보 생성 78
복합 경보 생성 81
CloudWatch 경보 편집 83
CPU 사용률 경보 생성 84
AWS Management 콘솔을 사용하여 CPU 사용률 경보 설정 84
AWS CLI를 사용하여 CPU 사용률 경보 설정 86
로드 밸런서 지연 시간 경보 생성 86
AWS Management 콘솔을 사용하여 지연 시간 경보 설정 87
AWS CLI를 사용하여 지연 시간 경보 설정 87
스토리지 처리량 경보 생성 88
AWS Management 콘솔을 사용하여 스토리지 처리량 경보 설정 88
AWS CLI를 사용하여 스토리지 처리량 경보 설정 88
인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 89
Amazon CloudWatch 경보에 중지 작업 추가 90
Amazon CloudWatch 경보에 종료 작업 추가 91
Amazon CloudWatch 경보에 재부팅 작업 추가 92
Amazon CloudWatch 경보에 복구 작업 추가 93
트리거된 경보 및 작업 기록 보기 94
결제 경보 생성 94
결제 알림 활성화 95
결제 경보 생성 95
결제 경보 삭제 96
Amazon EC2 Auto Scaling 경보 숨기기 96
카나리아 사용 97
필요한 역할 및 권한 98
카나리아 생성 101
카나리아에 대해 생성되는 리소스 103
카나리아 라이브러리 및 구성 정보 103
카나리아 통계 및 세부 정보 보기 103
카나리아 삭제 104

ServiceLens를 사용하여 애플리케이션의 상태 모니터링 106
ServiceLens 배포 107
AWS X-Ray 배포 107
CloudWatch 에이전트 및 X-Ray 데몬 배포 109
서비스 맵 사용 115
추적 보기 사용 116
ServiceLens 문제 해결 117
내 로그가 모두 표시되지 않음 117
서비스 맵에 내 경보가 모두 표시되지 않음 118
서비스 맵에 일부 AWS 리소스가 표시되지 않음 118
내 서비스 맵에 너무 많은 노드가 있음 118
교차 계정 교차 리전 CloudWatch 콘솔 119
교차 계정 교차 리전 기능 활성화 119
(선택 사항) AWS Organizations와 통합 121
문제 해결 121
이상 탐지 사용 123
이상 탐지 작동 방식 124
Contributor Insights를 사용하여 높은 카디널리티 데이터 분석 126
Contributor Insights 규칙 생성 126
Contributor Insights 규칙 구문 128
규칙 예제 131
Contributor Insights 보고서 보기 133
규칙에 의해 생성된 지표 그래프 작성 133
Contributor Insights 지표 데이터에 대한 경보 설정 134
Contributor Insights 기본 제공 규칙 사용 135
Container Insights 사용 136
Container Insights 설정 137
Amazon ECS에서 Container Insights 설정 138
Amazon EKS 및 Kubernetes에서 Container Insights 설정 145
Container Insights 지표 보기 159
CloudWatch Logs Insights을 사용하여 Container Insights 데이터 보기 159
Container Insights에서 수집된 지표 161
Amazon ECS Container Insights 지표 161
Amazon EKS 및 and Kubernetes Container Insights 지표 165
성능 로그 참조 167
Amazon ECS에 대한 성능 로그 이벤트 167
Amazon EKS 및 Kubernetes에 대한 성능 로그 이벤트 170
Amazon EKS 및 Kubernetes에 대한 성능 로그 이벤트의 관련 필드 181
Container Insights 문제 해결 187
Amazon EKS 또는 Kubernetes에서 실패한 배포 187
권한이 없는 패닉: Kubelet에서 cadvisor 데이터를 검색할 수 없음 187
삭제했다가 다시 생성한 클러스터에 Container Insights 배포 188
콘솔에 지표가 나타나지 않음 188
CloudWatch 에이전트의 CrashLoopBackoff 오류 188
보류 상태에서 멈춘 CloudWatch 에이전트 또는 FluentD Pod 188
자체 CloudWatch 에이전트 도커 이미지 빌드 188
CloudWatch 에이전트로 지표 및 로그 수집 189
CloudWatch 에이전트 설치 190
명령줄을 사용하여 CloudWatch 에이전트 설치 190
시스템 관리자를 사용하여 CloudWatch 에이전트 설치 201
AWS CloudFormation을 사용하여 CloudWatch 에이전트 설치 213
CloudWatch 에이전트 패키지의 서명 확인 217
CloudWatch 에이전트 구성 파일 생성 222
마법사로 CloudWatch 에이전트 구성 파일 만들기 222
CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 227
CloudWatch 에이전트가 수집하는 지표 255
Windows 서버 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 255

Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 255
CloudWatch 에이전트를 사용하는 일반적인 시나리오 263
다른 사용자로 CloudWatch 에이전트 실행 263
CloudWatch 에이전트에 의해 수집되는 지표에 사용자 지정 차원 추가 264
여러 CloudWatch 에이전트 구성 파일 265
CloudWatch 에이전트에 의해 수집되는 지표 집계 또는 롤업 266
CloudWatch 에이전트로 고분해능 지표 수집 267
다른 계정에 지표 및 로그 전송 268
통합 CloudWatch 에이전트와 이전 CloudWatch Logs 에이전트 간 타임스탬프 차이 269
CloudWatch 에이전트의 문제 해결 270
CloudWatch 에이전트 명령줄 파라미터 270
Run Command를 사용한 CloudWatch 에이전트 설치 실패 270
CloudWatch 에이전트가 시작되지 않음 270
CloudWatch 에이전트가 실행 중인지 확인 271
지표가 저장되는 위치 271
CloudWatch 에이전트가 시작되지 않고 오류에 Amazon EC2 리전이 언급되어 있음 272
Windows Server에서 자격 증명을 찾을 수 없음 272
CloudWatch 에이전트 파일 및 위치 272
CloudWatch 에이전트에 의해 생성되는 로그 273
CloudWatch 에이전트 중지 및 다시 시작 273
CloudWatch Application Insights for .NET and SQL Server를 사용하여 일반적인 애플리케이션 문제 감지 275
CloudWatch Application Insights란 무엇입니까? 275
개념 276
요금 276
관련 AWS 제품 276
지원되는 애플리케이션 구성 요소 277
지원되는 기술 스택 278
시작하기 278
CloudWatch Application Insights for .NET and SQL Server에 액세스 278
사전 조건 278
IAM 정책 279
애플리케이션 설정 280
Application Insights 보기 및 문제 해결 289
구성 오류 290
감지된 문제에 대한 알림 및 작업 설정 291
CloudWatch Application Insights 작동 방식 292
데이터 보존 293
구성 요소 구성 293
할당량 300
로그 및 지표 300
Amazon Elastic Compute Cloud(EC2) 301
Elastic Load Balancer(ELB) 307
애플리케이션 ELB 307
Amazon EC2 Auto Scaling 그룹 307
Amazon Simple Queue Server(SQS) 308
Amazon Relational Database Service(RDS) 308
데이터 포인트 요구 사항이 있는 지표 309
권장 지표 313
성능 카운터 지표 315
지표를 게시하는 서비스 319
경보 이벤트 및 EventBridge 324
CloudWatch의 샘플 이벤트 324
높은 카디널리티 로그 수집 및 CloudWatch 임베디드 지표 형식으로 지표 생성 328
임베디드 지표 형식을 사용하여 로그 생성 328
클라이언트 라이브러리 사용 329
수동으로 임베디드 지표 형식 로그 생성 329
콘솔에서 지표 및 로그 보기 340

AWS SDK 지표를 사용하여 애플리케이션 모니터링 342
SDK 지표 for Enterprise Support에서 수집한 지표와 데이터 343
SDK 지표 설정 345
SDK 지표에 대해 CloudWatch 에이전트 구성 346
SDK 지표에 대한 IAM 권한 설정 347
서비스 할당량 통합 및 사용량 지표 348
서비스 할당량 시각화 및 경보 설정 348
CloudWatch 사용량 지표 349
CloudWatch 자습서 351
시나리오: 예상 요금 모니터링 351
1단계: 결제 경보 활성화 351
2단계: 결제 경보 만들기 352
3단계: 경보 상태 확인 353
4단계: 결제 경보 편집 353
5단계: 결제 경보 삭제 353
시나리오: 지표 게시 353
1단계: 데이터 구성 정의 354
2단계: CloudWatch에 지표 추가 354
3단계: CloudWatch에서 통계 얻기 355
4단계: 콘솔을 사용하여 그래프 보기 355
보안 356
데이터 보호 356
전송 중 데이터 암호화 357
ID 및 액세스 관리 357
인증 357
액세스 제어 358
CloudWatch 대시보드 권한 업데이트 358
액세스 관리 개요 359
자격 증명 기반 정책(IAM 정책) 사용 362
조건 키를 사용하여 CloudWatch 네임스페이스에 대한 액세스 제한 368
서비스 연결 역할 사용 369
Application Insights에 서비스 연결 역할 사용 372
Amazon CloudWatch 권한 참조 문서 376
규정 준수 확인 381
복원성 382
인프라 보안 382
네트워크 격리 382
인터페이스 VPC 엔드포인트 382
가용성 383
CloudWatch용 VPC 종단점 생성 383
CloudWatch VPC 종단점에 대한 액세스 제어 384
Synthetics 카나리아에 대한 보안 고려 사항 384
보안 연결 사용 385
카나리아 이름 지정 고려 사항 385
카나리아 코드의 암호 385
사용 권한 고려 사항 385
스택 추적 및 예외 메시지 386
IAM 역할 범위를 좁게 지정 386
헤더 로깅 386
API 호출 로깅 387
CloudTrail의 CloudWatch 정보 387
예제: CloudWatch 로그 파일 항목 388
Grafana 통합 391
Service Quotas 392
문서 기록 395

Amazon CloudWatch란 무엇입니까?

Amazon CloudWatch는 Amazon Web Services(AWS) 리소스와 AWS에서 실시간으로 실행 중인 애플리케이션을 모니터링합니다. CloudWatch를 사용하여 리소스 및 애플리케이션에 대해 측정할 수 있는 변수인 지표를 수집하고 추적할 수 있습니다.

CloudWatch 홈 페이지에는 사용 중인 모든 AWS 서비스에 대한 지표가 자동으로 표시됩니다. 사용자 지정 대시보드를 추가로 생성해 사용자 지정 애플리케이션에 대한 지표를 표시하고, 선택한 지표의 사용자 지정 집합을 표시할 수 있습니다.

지표를 감시해 알림을 보내거나 임계값을 위반한 경우 모니터링 중인 리소스를 자동으로 변경하는 경보를 생성할 수 있습니다. 예를 들어 Amazon EC2 인스턴스의 CPU 사용량과 디스크 읽기 및 쓰기를 모니터링한 다음, 이러한 데이터를 사용하여 증가된 로드를 처리하기 위해 추가 인스턴스를 시작해야 할지 결정할 수 있습니다. 또한 이러한 데이터를 사용하여 잘 사용되지 않는 인스턴스를 중지할 수도 있습니다.

CloudWatch를 사용하면 시스템 전체의 리소스 사용률, 애플리케이션 성능 및 운영 상태를 파악할 수 있습니다.

CloudWatch에 액세스

다음 방법 중 하나를 사용하여 CloudWatch에 액세스할 수 있습니다.

- Amazon CloudWatch콘솔–https://console.aws.amazon.com/cloudwatch/
- AWS CLI – 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS Command Line Interface 설정하기를 참조하십시오.
- CloudWatch API – 자세한 내용은 Amazon CloudWatch API 참조를 참조하십시오.
- AWSSDK– 자세한 내용은 Amazon Web Services용 도구를 참조하십시오.

관련 AWS 서비스

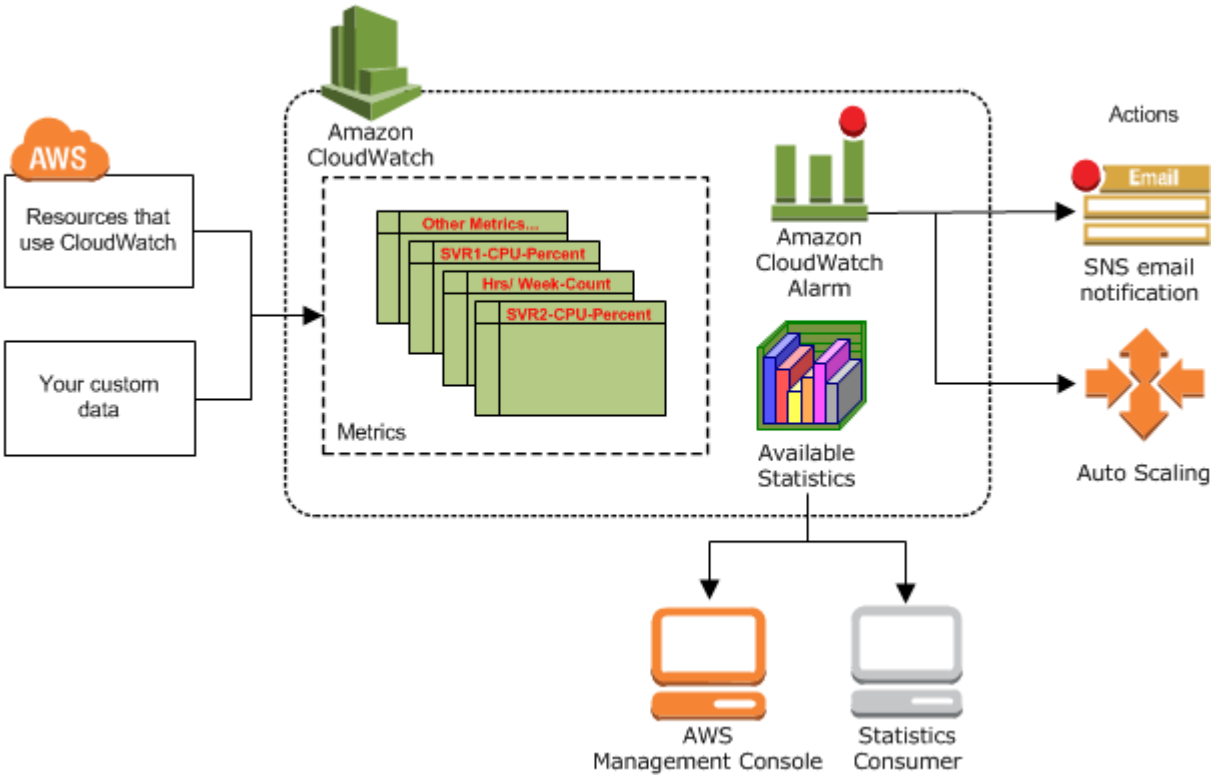
다음 서비스가 Amazon CloudWatch과 함께 사용됩니다.

- Amazon Simple Notification Service(Amazon SNS)는 구독 엔드포인트나 클라이언트로의 메시지 배달 또는 전송을 조정하고 관리합니다. CloudWatch와 함께 Amazon SNS를 사용하여 경보 임계값에 도달한 경우 메시지를 보냅니다. 자세한 내용은 Amazon SNS 알림 설정 (p. 72) 단원을 참조하십시오.
- Amazon EC2 Auto Scaling은 사용자 정의 정책, 상태 확인 및 예약 일정에 따라 Amazon EC2 인스턴스를 자동으로 시작 또는 종료할 수 있습니다. Amazon EC2 Auto Scaling과 CloudWatch 경보를 함께 사용하여 필요에 따라 EC2 인스턴스를 확장할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 동적 조정을 참조하십시오.
- AWS CloudTrail은 AWS Management 콘솔, AWS CLI 및 기타 서비스를 통해 이루어진 호출을 포함하여 계정에서 Amazon CloudWatch API에 대한 호출을 모니터링할 수 있도록 합니다. CloudTrail 기록이 활성화되어 있으면 CloudWatch는 CloudTrail을 구성할 때 지정한 Amazon S3 버킷에 로그 파일을 기록합니다. 자세한 내용은 AWS CloudTrail 사용을 통한 Amazon CloudWatch API 호출 로깅 (p. 387) 단원을 참조하십시오.
- AWS Identity and Access Management(IAM)은 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 Amazon CloudWatch에 대한 ID 및 액세스 관리 (p. 357) 단원을 참조하십시오.

Amazon CloudWatch 작동 방식

Amazon CloudWatch는 기본적으로 측정치 리포지토리입니다. AWS 서비스(예: —Amazon EC2—)는 지표를 리포지토리에 저장하므로 이러한 지표를 기반으로 통계를 검색할 수 있습니다. 사용자 지정 지표를 리포지토리에 저장하면 해당 지표에 대한 통계도 검색할 수 있습니다.



지표를 사용하여 통계를 계산한 다음 CloudWatch 콘솔에서 데이터를 그래픽으로 나타낼 수 있습니다. 지표를 생성하여 CloudWatch에 전송하는 다른 AWS 리소스에 대한 자세한 내용은 CloudWatch 지표를 게시하는 AWS 서비스 (p. 319) 단원을 참조하십시오.

특정 기준을 충족하는 경우 Amazon EC2 인스턴스를 중지, 시작 또는 종료하도록 경보 작업을 구성할 수 있습니다. 또한 Amazon EC2 Auto Scaling 및 Amazon Simple Notification Service(Amazon SNS) 작업을 대신 시작하는 경보를 만들 수 있습니다. CloudWatch 경보를 만드는 방법에 대한 자세한 내용은 경보 (p. 7) 단원을 참조하십시오.

AWS 클라우드 컴퓨팅 리소스는 가용성이 매우 높은 데이터 설비에 있습니다. 확장성 및 안정성을 더욱 높이기 위해 각 데이터 센터 설비는 지역이라고 하는 특정 지리적 영역에 있습니다. 장애 격리 및 안정성을 최대한 높이기 위해 리전이 서로 완전히 분리되도록 설계되었습니다. 지표는 리전에 별도로 저장되지만 CloudWatch 교차 리전 기능을 사용하여 다른 리전의 통계를 집계할 수 있습니다. 자세한 내용은 교차 계정 교차 리전 CloudWatch 콘솔 (p. 119) 및 Amazon Web Services 일반 참조의 리전 및 엔드포인트를 참조하십시오.

Amazon CloudWatch 개념

다음 용어와 개념은 Amazon CloudWatch의 이해와 사용에 매우 중요합니다.

- 네임스페이스 (p. 3)
- 지표 (p. 3)
- 차원 (p. 4)
- 통계 (p. 5)
- 백분위수 (p. 7)
- 경보 (p. 7)

네임스페이스

네임스페이스는 CloudWatch 지표용 컨테이너입니다. 다른 네임스페이스의 지표는 서로 격리되어 있으므로 다른 애플리케이션의 지표가 실수로 동일한 통계로 집계되는 일은 없습니다.

기본 네임스페이스는 없습니다. CloudWatch에 게시하는 각 데이터 요소의 네임스페이스를 지정해야 합니다. 사용자는 지표를 생성할 때 네임스페이스 이름을 지정할 수 있습니다. 이 이름은 유효한 XML 문자를 포함하고 있어야 하고 길이가 256자 미만이어야 합니다. 가능한 문자로는 영숫자 문자(0-9A-Za-z), 마침표(.), 하이픈(-), 밑줄(_), 슬래시(/), 해시(#), 콜론(:)이 있습니다.

AWS 네임스페이스는 일반적으로 `AWS/`*`service`*라는 명명 규칙을 사용합니다. 예를 들어 Amazon EC2는 `AWS/EC2` 네임스페이스를 사용합니다. AWS 네임스페이스 목록은 CloudWatch 지표를 게시하는 AWS 서비스 (p. 319) 단원을 참조하십시오.

지표

지표는 CloudWatch의 기본 개념입니다. 지표는 CloudWatch에 게시된 시간 순서별 데이터 요소 세트를 나타냅니다. 지표는 모니터링할 변수로, 데이터 요소는 시간에 따른 변수의 값을 나타내는 것으로 간주합니다. 예를 들어 특정 EC2 인스턴스의 CPU 사용량은 Amazon EC2가 제공하는 하나의 지표입니다. 데이터 요소 그 자체는 데이터를 수집하는 애플리케이션이나 비즈니스 활동에서 나올 수 있습니다.

AWS 서비스는 CloudWatch에 지표를 전송하기 때문에 사용자가 자체적으로 지정한 지표를 CloudWatch에 전송할 수 있습니다. 데이터 요소는 순서에 상관 없이 추가할 수 있습니다. 이러한 데이터 요소에 대한 통계를 정렬된 시계열 집합으로 검색할 수 있습니다.

지표는 생성된 리전에만 존재합니다. 지표는 삭제가 불가능하지만, 지표에 새 데이터가 게시되지 않을 경우 15개월 후에 자동으로 만료됩니다. 15개월이 지난 데이터 요소는 순서대로 만료됩니다. 새로운 데이터 요소가 들어오면 15개월이 지난 데이터가 삭제됩니다.

지표는 이름, 네임스페이스 및 0개 이상의 차원으로 고유하게 정의됩니다. 지표의 각 데이터 요소에는 타임스탬프와 측정 단위(선택 사항)가 있습니다. 모든 지표에 대해 CloudWatch에서 통계를 검색할 수 있습니다.

자세한 내용은 얻을 수 있는 지표 보기 (p. 29) 및 사용자 지정 메트릭 게시 (p. 47) 단원을 참조하십시오.

타임스탬프

각 지표 데이터 요소에는 타임스탬프가 연결되어 있어야 합니다. 타임스탬프는 최대 2주 전이고 최대 2시간 빠를 수 있습니다. 타임스탬프를 제공하지 않으면 CloudWatch는 데이터 요소를 받은 시간을 기준으로 타임스탬프를 생성합니다.

타임스탬프는 시간, 분, 초(예: 2016-10-31T23:59:59Z)와 더불어 완전한 날짜가 있는 `dateTime` 개체입니다. 자세한 내용은 dateTime을 참조하십시오. 필수 요건은 아니지만 UTC(협정 세계시)를 사용하는 것이 좋습니다. CloudWatch에서 통계를 검색할 때 모든 시간은 UTC를 따릅니다.

CloudWatch 경보는 UTC로 된 현재 시간을 기준으로 지표를 확인합니다. 사용자 지정 지표가 현재 UTC 시간이 아닌 타임스탬프로 CloudWatch로 전송되면 경보에 데이터 부족 상태가 표시되나 경보가 지연되는 결과가 초래될 수 있습니다.

지표 보존 기간

CloudWatch는 다음과 같이 지표 데이터를 보존합니다.

- 기간이 60초 미만으로 설정된 데이터 요소들은 3시간 동안 사용이 가능합니다. 이러한 데이터 요소는 고분해능 사용자 지정 지표입니다.
- 기간이 60초(1분)로 설정된 데이터 요소들은 15일 동안 사용이 가능
- 기간이 300초(5분)로 설정된 데이터 요소들은 63일 동안 사용이 가능
- 기간이 3600초(1시간)로 설정된 데이터 요소들은 455일(15개월) 동안 사용이 가능

원래 더 짧은 기간으로 게시된 데이터 요소는 장기 보관을 위해 집계됩니다. 예를 들어 데이터를 1분 기간으로 수집할 경우 15일 동안 1분 분해능으로 데이터를 사용할 수 있습니다. 15일 이후에는 이 데이터를 계속 사용할 수 있지만 데이터가 5분 분해능으로 집계됩니다. 63일 이후에는 이 데이터가 추가로 집계되어 1시간 분해능으로 제공됩니다.

CloudWatch는 2016년 7월 9일부터 5분 및 1시간 지표 데이터를 보존하기 시작했습니다.

차원

차원은 지표의 자격 증명에 속하는 이름/값 페어입니다. 각 지표에 차원을 최대 10개까지 할당할 수 있습니다.

모든 지표에는 자신을 설명하는 고유한 특징이 있고 차원을 이러한 특징에 대한 범주로 생각할 수 있습니다. 차원을 사용하면 통계 계획을 위한 구조를 설계할 수 있습니다. 차원은 지표에 대한 고유한 식별자의 일부이므로 지표 중 하나에 이름/값 쌍을 추가할 때마다 해당 지표의 새로운 변형이 생성되는 것입니다.

CloudWatch로 데이터를 전송하는 AWS 서비스는 각 지표에 차원을 연결합니다. 차원을 사용하여 CloudWatch가 반환하는 결과를 필터링할 수 있습니다. 예를 들어 지표를 검색할 때 `InstanceId` 차원을 지정하여 특정 EC2 인스턴스에 대한 통계를 얻을 수 있습니다.

Amazon EC2와 같은 특정 AWS 서비스에서 생성된 지표의 경우 CloudWatch에서는 차원 간에 데이터를 집계할 수 있습니다. 예를 들어 `AWS/EC2` 네임스페이스의 지표를 검색하되 어떤 차원도 지정하지 않으면 CloudWatch가 지정된 지표에 대한 모든 데이터를 집계해서 요청한 통계를 산출합니다. CloudWatch는 사용자 지정 지표에서 차원 간 데이터를 집계하지 않습니다.

차원 조합

지표가 동일한 지표 이름을 가지고 있다 하더라도 CloudWatch는 각각의 고유한 차원의 조합을 별도의 지표로 처리합니다. 사용자가 게시한 차원의 조합만 사용해서 통계를 검색할 수 있습니다. 통계를 검색할 때 지표 생성 시 사용했던 네임스페이스, 지표 이름 및 차원 파라미터에 동일한 값을 지정합니다. 또한 CloudWatch에서 집계에 사용할 시작 시간 및 종료 시간을 지정할 수도 있습니다.

예를 들어 다음 속성을 가진 DataCenterMetric 네임스페이스에 ServerStats라는 이름을 가진 서로 다른 4개의 지표를 게시한다고 가정합니다.

```
Dimensions: Server=Prod, Domain=Frankfurt, Unit: Count, Timestamp: 2016-10-31T12:30:00Z,
 Value: 105
Dimensions: Server=Beta, Domain=Frankfurt, Unit: Count, Timestamp: 2016-10-31T12:31:00Z,
 Value: 115
Dimensions: Server=Prod, Domain=Rio,       Unit: Count, Timestamp: 2016-10-31T12:32:00Z,
 Value: 95
Dimensions: Server=Beta, Domain=Rio,       Unit: Count, Timestamp: 2016-10-31T12:33:00Z,
 Value: 97
```

이러한 4개의 지표만 게시할 경우, 차원의 조합에 대한 통계를 검색할 수 있습니다.

- `Server=Prod,Domain=Frankfurt`
- `Server=Prod,Domain=Rio`
- `Server=Beta,Domain=Frankfurt`
- `Server=Beta,Domain=Rio`

다음 차원이 사용되거나 차원을 지정하지 않은 경우에는 통계를 검색할 수 없습니다.

- `Server=Prod`
- `Server=Beta`
- `Domain=Frankfurt`
- `Domain=Rio`

통계

통계는 지정한 기간에 걸친 지표 데이터 집계입니다. CloudWatch에서는 사용자 지정 데이터를 통해 제공되었거나 다른 AWS 서비스에서 CloudWatch에 제공한 지표 데이터 요소를 기반으로 통계를 제공합니다. 집계는 네임스페이스, 지표 이름, 차원 및 데이터 요소 측정 단위를 사용하여 지정한 기간에 대해 수행됩니다. 다음 표에서는 사용 가능한 통계에 대해 설명합니다.

통계	설명
`Minimum`	지정된 기간 중 관찰된 가장 낮은 값입니다. 이 값을 사용하여 애플리케이션에 대한 낮은 볼륨의 활동을 확인할 수 있습니다.
`Maximum`	지정된 기간 중 관찰된 가장 높은 값입니다. 이 값을 사용하여 애플리케이션에 대한 높은 볼륨의 활동을 확인할 수 있습니다.
`Sum`	일치하는 지표에 대해 제출된 모든 값이 서로 더해진 값입니다. 이 통계는 지표의 총 볼륨을 확인할 때 유용할 수 있습니다.
`Average`	지정된 기간 중 `Sum`/`SampleCount`의 값입니다. 이 통계를 `Minimum` 및 `Maximum`과 비교하면 지표의 전체 범위와 평균 사용량이 `Minimum` 및 `Maximum`에 얼마나 근접했는지 확인할 수 있습니다. 이와 같은 비교를 통해 필요에 따라 리소스를 늘리거나 줄어야 하는 시점을 파악할 수 있습니다.
`SampleCount`	통계 계산에 사용된 데이터 요소의 수(숫자)입니다.
`pNN.NN`	지정된 백분위 수의 값. 소수점 두 자리까지 사용하여 백분위 수를 지정할 수 있습니다 (예: p95.45). 음수 값이 포함된 지표에서는 백분위수 통계를 사용할 수 없습니다. 자세한 내용은 백분위수 (p. 7) 단원을 참조하십시오.

사전 산출된 통계값을 추가할 수 있습니다. 데이터 요소 값 대신 `SampleCount`, `Minimum`, `Maximum` 및 `Sum`에 대한 값을 지정합니다. CloudWatch에서는 대신 평균을 계산합니다. 이러한 방식으로 추가된 값은 일치하는 지표와 연결된 다른 모든 값과 함께 집계됩니다.

단위

각각의 통계는 측정 단위를 가지고 있습니다. 단위로는 `Bytes`, `Seconds`, `Count` 및 `Percent`가 있습니다. CloudWatch에서 지원하는 전체 단위 목록은 Amazon CloudWatch API Reference의 MetricDatum 데이터 형식을 참조하십시오.

사용자 지정 지표를 만들 때 단위를 지정할 수도 있습니다. 단위를 지정하지 않을 경우 CloudWatch는 `None`을 단위로 사용합니다. 단위를 사용하면 데이터에 개념적 의미를 더할 수 있습니다. CloudWatch에서는

단위가 내부적으로 크게 중요하지 않지만 기타 애플리케이션에서는 선택한 단위를 기반으로 의미 있는 정보를 추출할 수 있습니다.

측정 단위를 지정하는 지표 데이터 요소들은 개별적으로 집계됩니다. 단위를 지정하지 않고 통계를 얻으려고 하면 CloudWatch에서는 단위가 동일한 데이터 요소를 함께 집계합니다. 단위만 다른 동일한 지표가 두 개 있는 경우에는 각 단위에 대해 하나씩, 개별 데이터 스트림 두 개가 반환됩니다.

기간

기간은 특정 Amazon CloudWatch 통계와 연관된 시간의 길이입니다. 각 통계는 지정한 기간에 대해 수집된 지표 데이터의 집계를 나타냅니다. 기간은 초 단위로 정의되며, 유효한 기간 값은 1, 5, 10, 30 또는 60의 배수입니다. 예를 들어 6분의 기간을 지정하려면 기간 값으로 360을 사용합니다. 기간을 변경하여 데이터가 집계되는 방식을 조정할 수 있습니다. 기간은 최소 1초에서 최대 1일(86,400초)일 수 있습니다. 기본 값은 60초입니다.

저장 분해능 1초를 사용하여 정의한 사용자 지정 지표만 1분 미만 기간을 지원합니다. 콘솔에서는 항상 60 미만의 기간을 설정할 수 있지만 지표가 저장되는 방식과 일치하도록 기간을 선택해야 합니다. 1분 미만 기간을 지원하는 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.

통계를 검색할 때 기간, 시작 시간, 종료 시간을 지정할 수 있습니다. 이들 파라미터는 통계와 연관된 전체 기간을 결정합니다. 시작 시간과 종료 시간은 지난 1시간 동안의 통계를 얻을 수 있도록 기본 설정되어 있습니다. 시작 시간과 종료 시간에 값을 지정하여 CloudWatch에서 값이 반환되는 기간의 수를 결정할 수 있습니다. 예를 들어 기간, 시작 시간 및 종료 시간에 대한 기본값을 사용해 통계를 검색하면 전 시간 동안 1분마다 집계된 통계값들이 반환됩니다. 10분 단위로 집계된 통계를 선호할 경우에는 기간을 600으로 지정합니다. 전체 시간 동안 통계를 집계하고 싶은 경우에는 기간을 3600으로 지정합니다.

특정 시간 동안 통계가 집계되는 경우, 통계가 그 기간이 시작하는 시간으로 타임 스탬프가 추가됩니다. 예를 들어, 7:00pm에서 8:00pm에 집계된 데이터는 타임 스탬프가 7:00pm로 추가됩니다. 또한, 7:00pm와 8:00pm 사이에 집계된 데이터는 7:00pm에 표시되기 시작하며, 그렇게 집계된 데이터의 값은 CloudWatch가 해당 기간 동안 더 많은 샘플을 수집하면서 변경될 수 있습니다.

기간은 CloudWatch 경보에도 중요합니다. 특정 지표를 모니터링하도록 경보를 생성하면 CloudWatch가 해당 지표를 지정된 임계값과 비교하게 됩니다. CloudWatch에서 비교하는 방식을 광범위하게 제어할 수 있습니다. 비교 작업이 수행되는 기간을 지정할 수 있을 뿐 아니라, 결론에 도달하기까지 사용되는 평가 기간의 수를 지정할 수 있습니다. 예를 들어 세 평가 기간을 지정하면 CloudWatch가 세 가지 데이터 요소의 기간을 비교합니다. CloudWatch는 가장 오래된 데이터 요소가 임계값을 위반하고 나머지 데이터 요소들이 임계값을 위반하거나 데이터를 누락하는 경우에만 이를 알려줍니다. 지표를 연속해서 내보낸 경우 CloudWatch는 세 번 실패가 발견될 때까지 이를 알리지 않습니다.

집계

Amazon CloudWatch에서는 통계 검색 시 지정한 기간에 따라 통계를 집계합니다. 동일하거나 유사한 타임스탬프를 사용하여 데이터 요소를 원하는 만큼 게시할 수 있습니다. CloudWatch는 지정된 기간에 따라 이를 집계합니다. CloudWatch는 리전 간 데이터를 집계하지 않습니다.

동일한 타임스탬프 뿐만 아니라 동일한 네임스페이스 및 차원을 공유하는 지표에 대한 데이터 요소를 게시할 수 있습니다. CloudWatch는 이러한 데이터 요소들에 대해 통계를 집계해 반환합니다. 또한 타임스탬프에 상관 없이 동일하거나 다른 지표에 대한 여러 데이터 요소를 게시할 수도 있습니다.

대량의 데이터 세트에서는 통계 세트라는 사전 집계된 데이터 세트를 삽입할 수 있습니다. 통계 세트를 사용하면 CloudWatch에 데이터 요소 수의 Min, Max, Sum 및 SampleCount 값을 제공할 수 있습니다. 통계 세트는 1분에 여러 번 데이터를 수집해야 하는 경우 일반적으로 사용됩니다. 예를 들어 웹 페이지의 요청 지연 시간에 대한 지표가 있다고 가정해 보겠습니다. 웹 페이지 방문 시 데이터를 게시하는 것은 적절하지 않습니다. 해당 웹 페이지에 대한 모든 방문의 지연 시간을 수집한 다음, 1분에 한 번 이러한 데이터를 집계하여 CloudWatch에 통계 세트를 전송하는 것이 좋습니다.

Amazon CloudWatch에서는 지표 소스를 구분하지 않습니다. 다른 소스에서 네임스페이스 및 차원이 동일한 지표를 게시하면 CloudWatch에서는 이 지표를 단일 지표로 처리합니다. 이는 확장된 분산형 시스템의 서비스 지표에 유용할 수 있습니다. 예를 들어 웹 서버 애플리케이션의 모든 호스트는 처리 중인 요청의 지연 시

간을 나타내는 동일한 지표를 게시할 수 있습니다. CloudWatch에서는 이러한 지표를 단일 지표로 처리하므로 애플리케이션 간 모든 요청에 대한 최소값, 최대값, 평균 및 합계 통계를 얻을 수 있습니다.

백분위수

백분위수는 데이터 세트에서 값의 상대적 위치를 나타냅니다. 예를 들어 95 백분위는 데이터의 95%가 이 값보다 아래에 있고 5%가 이 값보다 위에 있다는 것을 의미합니다. 백분위수는 지표 데이터의 분포를 정확하게 이해하는 데 도움이 됩니다.

백분위 수는 종종 이상치를 격리하는 데 사용됩니다. 일반적인 분포에서 데이터의 95%는 평균값으로부터 2 표준 편차 내에 있으며, 데이터의 99.7%는 평균값으로부터 3 표준 편차 내에 있습니다. 3 표준 편차 밖에 있는 데이터는 평균값에서 크게 벗어나 있다는 점에서 종종 이상치로 간주됩니다. 예를 들어 뛰어난 고객 경험을 보장하기 위해 EC2 인스턴스의 CPU 사용률을 모니터링하고 있다고 가정합니다. 평균값을 모니터링하면 이상치가 감춰질 수 있습니다. 최대값을 모니터링하면 단 하나의 이상치로도 결과가 잘못될 수 있습니다. 백분위수를 사용하면 CPU 사용률에 대한 95 백분위를 모니터링하여 비정상적으로 부하가 많은 인스턴스를 확인할 수 있습니다.

일부 CloudWatch 지표는 백분위수를 통계로 지원합니다. 이러한 지표의 경우, 다른 CloudWatch 통계(평균, 최소값, 최대값, 합계)를 사용할 때와 마찬가지로 백분위수를 사용해 시스템 및 애플리케이션을 모니터링할 수 있습니다. 예를 들어 경보를 생성할 때 통계 함수로 백분위수를 사용할 수 있습니다. 소수점 두 자리까지 사용하여 백분위수를 지정할 수 있습니다(예: p95.45).

사용자 지정 지표에 대해 요약되지 않은 원시 데이터 요소를 게시하는 경우 사용자 지정 지표에 백분위수 통계를 사용할 수 있습니다. 지표 값에 음수 값이 포함된 지표에서는 백분위수 통계를 사용할 수 없습니다.

CloudWatch가 백분위수를 계산하려면 원시 데이터 요소가 필요합니다. 대신 통계 세트를 사용해 데이터를 게시하면 아래 조건 중 하나가 true인 경우에만 이 데이터에 대한 백분위수 통계를 검색할 수 있습니다.

- 통계 세트의 SampleCount 값은 1이고 Min, Max 및 Sum은 모두 같습니다.
- Min과 Max는 같고 Sum은 Min에 SampleCount를 곱한 값과 같습니다.

경보

경보를 사용하여 작업을 자동으로 시작할 수 있습니다. 경보는 지정한 기간에 단일 지표를 감시하고 시간에 따른 임계값에 대한 지표 값을 기준으로 지정된 작업을 하나 이상 수행합니다. 이 작업은 Amazon SNS 주제나 Auto Scaling 정책으로 전송되는 알림입니다. 대시보드에 경보를 추가할 수도 있습니다.

경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 단순히 특정 상태에 있다고 해서 작업을 호출하지 않습니다. 상태가 변경되어 지정된 기간 수 동안 유지되어야 합니다.

경보를 만들 때 지표를 모니터링할 빈도와 동일하거나 더 긴 기간을 선택합니다. 예를 들어 Amazon EC2에 대한 기본 모니터링은 5분마다 인스턴스에 지표를 제공합니다. 기본 모니터링 지표에 대한 경보 설정 시 기간을 300초(5분) 이상으로 선택합니다. Amazon EC2에 대한 세부 모니터링은 1분마다 인스턴스에 지표를 제공합니다. 세부 모니터링 지표에 대한 경보 설정 시 기간을 60초(1분) 이상으로 선택합니다.

고분해능 지표에 대해 경보를 설정할 경우 고분해능 경보를 10초 또는 30초 기간으로 지정하거나 60초의 배수 기간으로 정기 경보를 설정할 수 있습니다. 고분해능 경보는 요금이 더 비쌉니다. 고분해능 지표에 대한 자세한 내용은 사용자 지정 메트릭 게시 (p. 47) 단원을 참조하십시오.

자세한 내용은 Amazon CloudWatch 경보 사용 (p. 67) 및 그래프의 지표에서 경보 생성 (p. 46) 단원을 참조하십시오.

Amazon CloudWatch 리소스

다음 표에는 이 서비스를 이용할 때 참조할 수 있는 관련 리소스가 나와 있습니다.

리소스	설명
Amazon CloudWatch FAQ	FAQ는 이 제품에 대해 개발자들이 가장 많이 질문한 상위 개 내용을 소개합니다.
릴리스 정보	릴리스 정보에는 현재 릴리스에 대한 고급 수준의 개요를 제공합니다. 특히 새로운 기능, 상관 관계 및 알려진 문제점에 대해 다룹니다.
AWS Developer Resource Center	설명서, 코드 예제, 릴리스 정보 및 AWS를 사용하여 혁신적인 응용 프로그램을 구현하는 데 도움이 되는 기타 정보를 이 한 곳에서 찾을 수 있습니다.
AWS Management 콘솔	이 콘솔에서는 Amazon CloudWatch 및 다양한 다른 AWS 제품 기능의 대부분을 프로그래밍 작업 없이 수행할 수 있습니다.
Amazon CloudWatch Discussion Forums	Amazon CloudWatch와 관련된 기술적 질문에 대해 토론할 수 있는 개발자를 위한 커뮤니티 기반 포럼입니다.
AWS Support	AWS Support 사례를 생성 및 관리하는 곳입니다. 또한 포럼, 기술 FAQ, 서비스 상태 및 AWSAWS Trusted Advisor 등의 기타 유용한 자료에 대한 링크가 있습니다.
Amazon CloudWatch 제품 정보	Amazon CloudWatch에 대한 정보를 제공하는 기본 웹 페이지입니다.
문의처	AWS 결제, 계정, 이벤트, 침해 등에 대해 문의할 수 있는 중앙 연락 지점입니다.

설정

Amazon CloudWatch를 사용하려면 AWS 계정이 있어야 합니다. AWS 계정이 있어야 서비스(예: Amazon EC2)를 사용해 포인트 앤 클릭 방식의 웹 기반 인터페이스인 CloudWatch 콘솔에서 확인 가능한 지표를 생성할 수 있습니다. 뿐만 아니라 AWS 명령줄 인터페이스(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. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요할 경우 탐색 표시줄을 사용해 AWS 리소스가 있는 리전으로 리전을 변경하십시오.
3. CloudWatch 콘솔 사용이 처음이라 하더라도 Your Metrics(사용자 지표)에 이미 지표가 보고 되었을 수 있습니다. 왜냐하면 Amazon CloudWatch로 지표를 자동 푸시하는 AWS 제품을 무료로 사용하였기 때문입니다. 다른 AWS 제품들에서는 지표를 활성화해야 합니다.

   경보가 없으면 Your Alarms(사용자 경보) 섹션에 경보 생성 버튼이 표시됩니다.

AWS CLI 설정

AWS CLI 또는 Amazon CloudWatch CLI를 사용하여 CloudWatch 명령을 수행할 수 있습니다. AWS CLI에만 새로운 CloudWatch 기능을 포함시킨 경우에 AWS CLI가 CloudWatch CLI를 대체합니다.

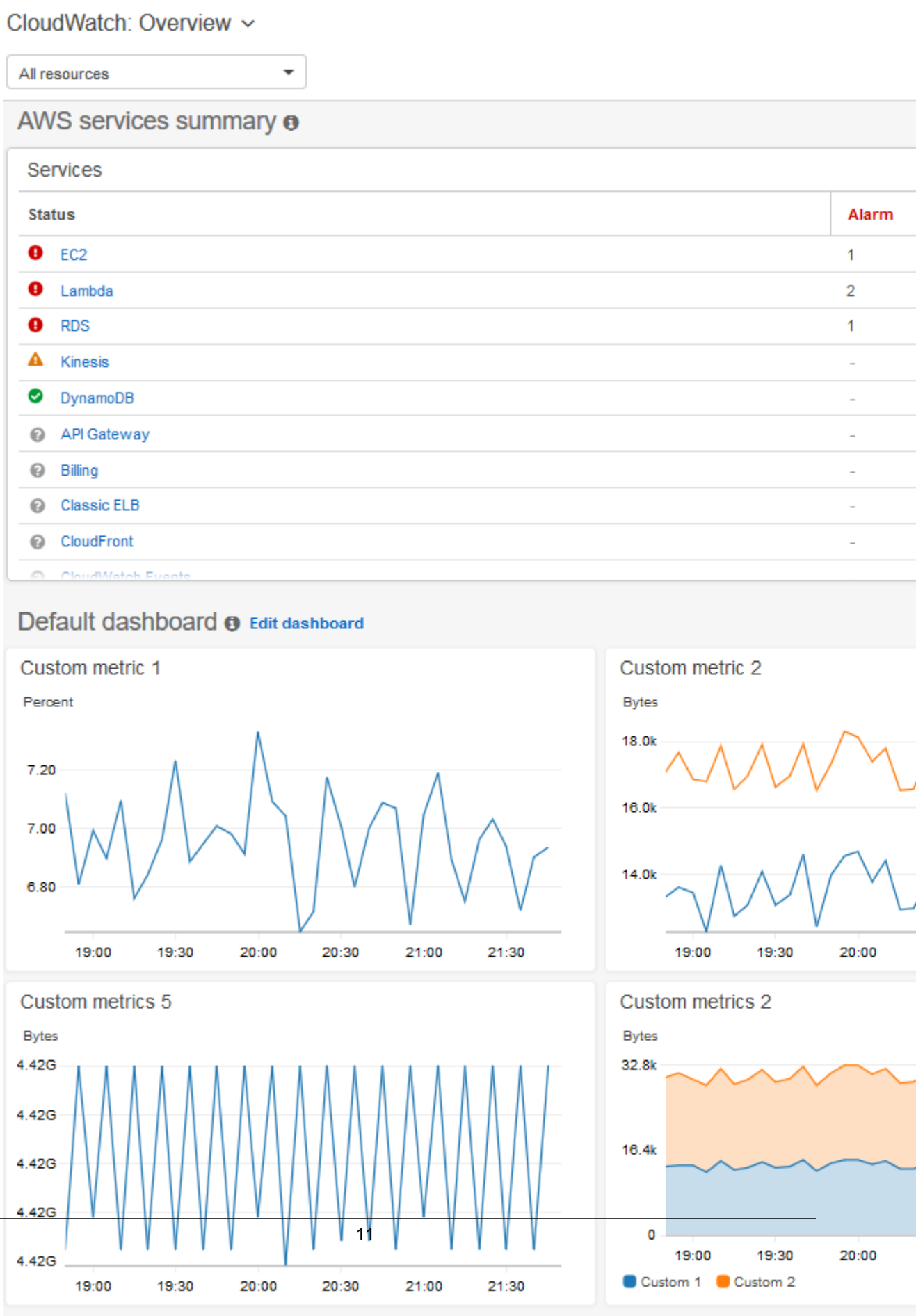
AWS CLI의 설치 및 구성 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 AWS 명령줄 인터페이스를 사용한 설정을 참조하십시오.

Amazon CloudWatch CLI의 설치 및 구성 방법에 대한 자세한 내용은 Amazon CloudWatch CLI Reference의 명령줄 인터페이스 설정을 참조하십시오.

Amazon CloudWatch 시작하기

https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.

CloudWatch 개요 홈 페이지가 표시됩니다.


CloudWatch: Overview
All resources
AWS services summary
Services
Status
Alarm
EC2
1
Lambda
2
RDS
1
Kinesis
-
DynamoDB
-
API Gateway
-
Billing
-
Classic ELB
-
CloudFront
-
Default dashboard
Edit dashboard
Custom metric 1
Percent
7.20
7.00
6.80
19:00
19:30
20:00
20:30
21:00
21:30
Custom metric 2
Bytes
18.0k
16.0k
14.0k
19:00
19:30
20:00
Custom metrics 5
Bytes
4.42G
4.42G
4.42G
4.42G
4.42G
19:00
19:30
20:00
20:30
21:00
21:30
Custom metrics 2
Bytes
32.8k
16.4k
0
19:00
19:30
20:00
Custom 1
Custom 2

개요에는 다음 항목이 표시되며, 이러한 항목은 자동으로 새로고침됩니다.

- 왼쪽 위에는 계정에서 사용하고 있는 AWS 서비스 목록과 해당 서비스의 경보 상태가 함께 표시됩니다. 오른쪽 위에는 사용 중인 AWS 서비스 수에 따라 계정의 경보 2개 또는 4개가 표시됩니다. 표시된 경보는 ALARM 상태의 경보 또는 가장 최근에 상태가 변경된 경보입니다.

  이러한 상단 영역에서는 모든 서비스의 경보 상태와 최근에 상태가 변경된 경보를 보고 AWS 서비스의 상태를 평가할 수 있습니다. 따라서 모니터링해 문제를 빠르게 진단할 수 있습니다.
- 이 영역 아래에는 기본 대시보드가 있습니다(있는 경우). 기본 대시보드는 사용자가 생성하여 CloudWatch-Default라고 이름을 지정한 사용자 지정 대시보드입니다. 대시보드에서는 편리하게, 사용자 지정 서비스 또는 애플리케이션에 대한 지표를 개요 페이지에 추가하거나 가장 모니터링하고 싶은 AWS 서비스에서 주요 지표를 추가로 가져와 표시할 수 있습니다.
- AWS 서비스를 6개 이상 사용하는 경우 기본 대시보드 아래에는 자동 교차 서비스 대시보드에 대한 링크가 있습니다. 교차 서비스 대시보드에는 사용 중인 모든 AWS 서비스의 주요 지표가 자동으로 표시되므로 모니터링할 지표를 선택하거나 사용자 지정 대시보드를 생성할 필요가 없습니다. 또한 대시보드를 사용해 모든 AWS 서비스를 드릴다운하고 해당 서비스의 주요 지표를 추가로 확인할 수 있습니다.

  AWS 서비스를 6개 미만으로 사용하는 경우 이 페이지에는 교차 서비스 대시보드가 자동으로 표시됩니다.

이러한 개요에서는 특정 리소스 그룹 또는 특정 AWS 서비스를 집중적으로 살펴볼 수 있습니다. 따라서 관심 있는 리소스의 일부분으로 시야를 좁힐 수 있습니다. 리소스 그룹을 사용하면 태그를 사용해 프로젝트를 정리하거나, 아키텍처의 일부분을 집중적으로 살펴보거나, 프로덕션 환경과 개발 환경을 구분할 수 있습니다. 자세한 내용은 AWS 리소스 그룹이란 무엇입니까?단원을 참조하십시오.

주제

- 모든 AWS 서비스의 주요 지표 살펴보기 (p. 12)
- 단일 AWS 서비스의 지표 및 경보에 집중 (p. 13)
- 리소스 그룹의 지표 및 경보에 집중 (p. 14)

모든 AWS 서비스의 주요 지표 살펴보기

AWS 서비스를 6개 이상 사용하는 경우 개요 페이지에는 교차 서비스 대시보드가 표시되지 않습니다. 이 대시보드로 전환해 사용 중인 모든 AWS 서비스의 주요 지표를 확인할 수 있습니다.

교차 서비스 대시보드를 열려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.

   개요가 나타납니다.
2. 페이지 하단 근처에서 교차 서비스 대시보드 보기를 선택합니다.

   교차 서비스 대시보드가 나타나고, 여기에는 사용 중인 각 AWS 서비스가 알파벳 순서로 표시되어 있습니다. 서비스마다 주요 지표가 한두 개 표시되어 있습니다.
3. 다음 두 가지 방식으로 특정 서비스를 집중적으로 살펴볼 수 있습니다.

   a. 서비스에 대한 주요 지표를 추가로 보려면 화면 상단에 있는 목록에서 지표의 이름을 선택합니다. 현재 화면 상단에는 교차 서비스 대시보드가 표시되어 있습니다. 또는 서비스 이름 옆에서 Service dashboard(서비스 대시보드 보기)를 선택할 수 있습니다.

      해당 서비스에 대한 자동 대시보드가 나타나 이 서비스에 대한 지표를 추가로 보여줍니다. 또한 일부 서비스의 경우 서비스 대시보드 하단에 해당 서비스와 관련된 리소스가 표시됩니다. 서비스 콘솔에 대한 리소스 중 하나를 선택해 해당 리소스를 집중적으로 살펴볼 수 있습니다.

b. 서비스와 관련된 경보를 모두 보려면 화면 오른쪽에서 해당 서비스 이름 옆에 있는 버튼을 선택합니다. 이러한 버튼에 표시된 텍스트는 해당 서비스에서 생성한 경보 수와 ALARM 상태인 경보가 있는지 여부를 나타냅니다.

   경보가 표시되면 설정(예: 차원, 임계값 또는 기간)이 유사한 여러 경보는 단일 그래프에 표시될 수 있습니다.

   그런 다음 경보에 대한 세부 정보를 보고 경보 기록을 볼 수 있습니다. 이렇게 하려면 경보 그래프 위에 마우스를 올려 놓고 작업 아이콘, 경보에서 보기를 선택합니다.

   새 브라우저 탭에 경보 보기가 나타나는데, 여기에는 선택한 경보에 대한 세부 정보와 함께 경보 목록이 표시됩니다. 경보 기록을 보려면 기록 탭을 선택합니다.

4. 특정 리소스 그룹의 리소스를 집중적으로 살펴볼 수 있습니다. 이렇게 하려면 모든 리소스가 표시된 페이지 상단의 목록에서 리소스 그룹을 선택합니다.

   자세한 내용은 리소스 그룹의 지표 및 경보에 집중 (p. 14) 단원을 참조하십시오.

5. 현재 표시된 모든 그래프 및 경고에 나타나는 시간 범위를 변경하려면 화면 상단에서 시간 범위 옆에서 원하는 범위를 선택합니다. 사용자 지정을 선택하여 기본적으로 표시된 시간 범위보다 더 많은 시간 범위 옵션 중에서 선택합니다.

6. 경보는 항상 1분마다 새로 고침됩니다. 보기를 새로 고치려면 화면 오른쪽 상단에서 새로고침 아이콘(구부러진 화살표 2개)을 선택합니다. 화면에서 경보 이외의 항목에 대한 자동 새로 고침 속도를 변경하려면 새로 고침 아이콘 옆에 있는 아래쪽 화살표를 선택하고 원하는 새로 고침 속도를 선택합니다. 또한 자동 새로 고침을 끄도록 선택할 수도 있습니다.

교차 서비스 대시보드에 나타나지 않도록 서비스 제거

서비스 지표가 교차 서비스 대시보드에 나타나지 않도록 지정할 수 있습니다. 이렇게 하면 모니터링하려는 서비스에 대한 교차 서비스 대시보드를 집중적으로 살펴볼 수 있습니다.

교차 서비스 대시보드에서 서비스를 제거하더라도 해당 서비스에 대한 경보는 경보 보기에 그대로 나타납니다.

교차 서비스 대시보드에 나타나지 않도록 서비스 지표를 제거하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.

   홈 페이지가 나타납니다.

2. 페이지 상단의 개요에서 제거하려는 서비스를 선택합니다.

   해당 서비스에 대한 지표만 표시하도록 보기가 바뀝니다.

3. 작업을 선택하고 교차 서비스 대시보드에 표시 옆에 있는 확인란을 선택 취소합니다.

단일 AWS 서비스의 지표 및 경보에 집중

CloudWatch 홈 페이지에서 단일 AWS 서비스에 대한 보기를 집중적으로 살펴볼 수 있습니다. 단일 AWS 서비스와 리소스 그룹 둘 다를 동시에 집중적으로 살펴보면 좀 더 자세히 드릴다운할 수 있습니다. 다음 절차는 AWS 서비스를 집중적으로 살펴보는 방법만 보여줍니다.

단일 서비스를 집중적으로 살펴보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.

   홈 페이지가 나타납니다.

2. 현재 개요가 표시되어 있는 화면 상단의 목록에서 서비스 이름을 선택합니다.

선택한 서비스의 주요 지표 그래프를 표시하도록 보기가 바뀝니다.

3. 서비스에 대한 경보를 보도록 전환하려면 현재 서비스 대시보드가 표시된 화면 상단에서 경보 대시보드를 선택합니다.
4. 지표를 볼 때 다음과 같은 여러 가지 방법으로 특정 지표를 집중적으로 살펴볼 수 있습니다.

   a. 임의의 그래프에서 지표를 보다 자세히 살펴보려면 그래프 위에 마우스를 놓고 작업 아이콘 지표에서 보기를 선택합니다.

      새 탭에 그래프가 나타나는데, 해당 그래프 아래에는 관련 지표가 나열되어 있습니다. 그래프 보기를 사용자 지정해 표시되는 지표 및 리소스, 통계, 기간 및 기타 요소를 변경하여 현재 상황을 더욱 정확하게 파악하도록 할 수 있습니다.

   b. 그래프에 표시된 시간 범위에 발생한 로그 이벤트를 볼 수 있습니다. 이렇게 하면 인프라에서 발생해 지표에 예기치 않은 변화를 가져온 이벤트를 찾을 수 있습니다.

      로그 이벤트를 보려면 그래프 위에 마우스를 올려 놓고 작업 아이콘, 로그에서 보기를 선택합니다.

      CloudWatch Logs 보기가 새 탭에 나타나 로그 그룹의 목록을 표시합니다. 로그 그룹 중 하나에서 원래 그래프에 표시된 시간 범위에서 발생한 로그 이벤트를 보려면 로그 그룹을 선택합니다.
5. 경보를 볼 때 다음과 같은 여러 가지 방법으로 특정 경보를 집중적으로 살펴볼 수 있습니다.

   - 경보를 보다 자세히 살펴보려면 경보 위에 마우스를 놓고 작업 아이콘 경보에서 보기를 선택합니다.

     새 탭에 경보 보기가 나타나는데, 여기에는 선택한 경보에 대한 세부 정보와 함께 경보 목록이 표시됩니다. 경보 기록을 보려면 기록 탭을 선택합니다.
6. 경보는 항상 1분마다 한 번씩 새로 고침됩니다. 보기를 새로 고치려면 화면 오른쪽 상단에서 새로고침 아이콘(구부러진 화살표 2개)을 선택합니다. 화면에서 경보 이외의 항목에 대한 자동 새로 고침 속도를 변경하려면 새로 고침 아이콘 옆에 있는 아래쪽 화살표를 선택하고 새로 고침 속도를 선택합니다. 또한 자동 새로 고침을 끄도록 선택할 수도 있습니다.
7. 현재 표시된 모든 그래프 및 경고에 나타나는 시간 범위를 변경하려면 화면 상단에서 시간 범위 옆에서 범위를 선택합니다. 기본적으로 표시된 시간 범위보다 더 많은 시간 범위 옵션 중에서 선택하려면 사용자 지정을 선택합니다.
8. 교차 서비스 대시보드로 돌아가려면 현재 집중적으로 살펴보고 있는 서비스가 표시된 화면 상단의 목록에서 개요를 선택합니다.

   또는 아무 보기에서나 화면 상단에서 CloudWatch를 선택하여 필터를 모두 지우고 개요 페이지로 돌아갈 수 있습니다.

리소스 그룹의 지표 및 경보에 집중

보기에 집중해 단일 리소스 그룹의 지표 및 경보를 표시할 수 있습니다. 리소스 그룹을 사용하면 태그를 사용해 프로젝트를 정리하거나, 아키텍처의 일부분을 집중적으로 살펴보거나, 프로덕션 환경과 개발 환경을 구분할 수 있습니다. 또한 CloudWatch 개요에서 이러한 각 리소스 그룹을 집중적으로 살펴볼 수 있습니다. 자세한 내용은 AWS 리소스 그룹이란 무엇입니까?단원을 참조하십시오.

리소스 그룹을 집중적으로 살펴볼 때 해당 리소스 그룹의 일부로 리소스에 태그를 지정한 서비스만 표시하도록 바뀝니다. 최신 경보 영역에는 해당 리소스 그룹과 관련된 경보만 표시됩니다. 또는 CloudWatch-Default-ResourceGroupName이라는 대시보드를 생성한 경우 해당 대시보드가 기본 대시보드 영역에 표시됩니다.

단일 AWS 서비스와 리소스 그룹 둘 다를 동시에 집중적으로 살펴보면 좀 더 자세히 드릴다운할 수 있습니다. 다음 절차는 리소스 그룹을 집중적으로 살펴보는 방법을 보여줍니다.

단일 리소스 그룹을 집중적으로 살펴보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.

2. 모든 리소스가 표시된 페이지 상단에서 리소스 그룹을 선택합니다.
3. 해당 리소스 그룹과 관련된 지표를 추가로 보려면 화면 하단 근처에서 교차 서비스 대시보드 보기를 선택합니다.

   교차 서비스 대시보드가 나타나면 리소스 그룹과 관련된 서비스만 표시됩니다. 서비스마다 주요 지표가 한두 개 표시되어 있습니다.
4. 현재 표시된 모든 그래프 및 경고에 나타나는 시간 범위를 변경하려면 화면 상단의 시간 범위에서 범위를 선택합니다. 기본적으로 표시된 시간 범위보다 더 많은 시간 범위 옵션 중에서 선택하려면 사용자 지정을 선택합니다.
5. 경보는 항상 1분마다 한 번씩 새로 고침됩니다. 보기를 새로 고치려면 화면 오른쪽 상단에서 새로고침 아이콘(구부러진 화살표 2개)을 선택합니다. 화면에서 경보 이외의 항목에 대한 자동 새로 고침 속도를 변경하려면 새로 고침 아이콘 옆에 있는 아래쪽 화살표를 선택하고 새로 고침 속도를 선택합니다. 또한 자동 새로 고침을 끄도록 선택할 수도 있습니다.
6. 계정의 모든 리소스에 대한 정보를 표시하는 화면으로 되돌아가려면 현재 리소스 그룹의 이름이 표시되어 있는 화면 상단 근처에서 모든 리소스를 선택합니다.

Amazon CloudWatch 대시보드 사용

Amazon CloudWatch 대시보드는 CloudWatch 콘솔에서 사용자 지정이 가능한 홈 페이지로, 다른 리전에 분산되어 있는 리소스들을 단일 뷰에서 모니터링하는 데 사용할 수 있습니다. CloudWatch 대시보드를 사용해 AWS 리소스에 대한 지표 및 경보를 보여주는 사용자 지정 뷰를 생성할 수 있습니다.

대시보드에서 다음을 생성할 수 있습니다.

- 선택한 지표 및 경보에 대한 단일 뷰를 생성하여 하나 이상의 리전에서 리소스 및 애플리케이션의 상태를 평가할 수 있습니다. 여러 그래프에서 동일한 지표를 손쉽게 추적할 수 있도록 각 그래프에서 지표 각각에 사용되는 색상을 선택할 수 있습니다.

  여러 AWS 계정 및 여러 리전에서 그래프 및 기타 위젯을 표시하는 대시보드를 생성할 수 있습니다. 자세한 내용은 교차 계정 교차 리전 CloudWatch 콘솔 (p. 119) 단원을 참조하십시오.
- 작동 지침서를 생성하여 운영 이벤트 동안 팀원들에게 특정 사고에 대한 대응 방법에 관해 지침을 제공할 수 있습니다.
- 중요한 리소스 및 애플리케이션 지표에 대한 공통 뷰를 생성하여 운영 이벤트 동안 신속하고 원활한 의사소통을 위해 팀원들이 이를 공유하도록 할 수 있습니다.

콘솔, AWS CLI 또는 `PutDashboard` API를 사용하여 대시보드를 생성할 수 있습니다.

CloudWatch 대시보드에 액세스하려면 다음 중 하나가 필요합니다.

- AdministratorAccess 정책
- CloudWatchFullAccess 정책
- 다음과 같은 특정 권한 중 하나 이상을 포함하는 사용자 지정 정책:
  - 대시보드를 볼 수 있는 `cloudwatch:GetDashboard` 및 `cloudwatch:ListDashboards`
  - 대시보드를 생성하거나 수정할 수 있는 `cloudwatch:PutDashboard`
  - 대시보드를 삭제할 수 있는 `cloudwatch:DeleteDashboards`

목차

- CloudWatch 대시보드 생성 (p. 17)
- 교차 계정 교차 리전 대시보드 (p. 17)
- CloudWatch 대시보드에서 그래프를 추가 또는 제거 (p. 20)
- CloudWatch 대시보드에서 그래프 이동 또는 크기 조정 (p. 22)
- CloudWatch 대시보드에서 그래프를 편집하려면 (p. 22)
- CloudWatch 대시보드에서 수동으로 지표 그래프 생성 (p. 24)
- CloudWatch 대시보드에서 그래프 이름 변경 (p. 25)
- CloudWatch 대시보드에서 텍스트 위젯을 추가 또는 제거 (p. 25)
- CloudWatch 대시보드에서 경보를 추가 또는 제거 (p. 26)
- CloudWatch 대시보드에서 그래프 링크 또는 링크 해제 (p. 26)
- 즐겨찾기 목록에 대시보드 추가 (p. 27)
- CloudWatch 대시보드에 대한 기간 재정의 설정 또는 새로 고침 간격 변경 (p. 27)
- CloudWatch 대시보드의 시간 범위 또는 시간대 형식 변경 (p. 28)

CloudWatch 대시보드 생성

CloudWatch 대시보드를 시작하려면 먼저 대시보드를 만들어야 합니다. 대시보드는 여러 개 생성할 수 있습니다. AWS 계정에 있는 CloudWatch 대시보드의 수에는 제한이 없습니다. 모든 대시보드는 전역 대시보드이며 리전별로 구분되지 않습니다.

이 단원에서는 콘솔을 사용하여 대시보드를 생성합니다. `PutDashboard` API를 사용하여 대시보드를 생성할 수도 있습니다. 그러면 이 API가 JSON 문자열을 사용하여 대시보드 내용을 정의합니다. `PutDashboard`를 사용하여 대시보드를 생성하고 기존 대시보드를 기반으로 이 대시보드를 구성하려면 작업, View/edit source(소스 보기/편집)를 차례로 선택하여 새 대시보드에 사용할 현재 대시보드의 JSON 문자열을 표시 및 복사합니다.

API를 사용한 대시보드 생성에 대한 자세한 내용은 Amazon CloudWatch API Reference의 PutDashboard를 참조하십시오.

콘솔을 사용하여 대시보드를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드, 대시보드 생성을 선택합니다.
3. 새 대시보드 생성 대화 상자에서 대시보드 이름을 입력하고 대시보드 생성을 선택합니다.

   CloudWatch-Default라는 이름을 사용하는 경우 대시보드가 CloudWatch 홈 페이지의 개요에 나타납니다. 자세한 내용은 Amazon CloudWatch 시작하기 (p. 10) 단원을 참조하십시오.

   리소스 그룹을 사용하고 대시보드의 이름을 CloudWatch-Default-ResourceGroupName으로 지정하면 해당 리소스 그룹에 마우스를 가져가면 CloudWatch 홈페이지에 대시보드가 나타납니다.
4. 이 대시보드에 추가 대화 상자에서 다음 중 하나를 수행합니다.
   - 대시보드에 그래프를 추가하려면 행 또는 누적 면적을 선택하고 구성을 선택합니다. 지표 그래프 추가 대화 상자에서 그래프로 표시할 지표를 선택하고 위젯 생성을 선택합니다. 특정 지표가 14일 이상 데이터를 게시하지 않아 대화 상자에 나타나지 않는 경우 수동으로 추가할 수 있습니다. 자세한 내용은 CloudWatch 대시보드에서 수동으로 지표 그래프 생성 (p. 24) 단원을 참조하십시오.
   - 대시보드에 지표를 표시하는 숫자를 추가하려면 번호, 구성을 선택합니다. 지표 그래프 추가 대화 상자에서 그래프로 표시할 지표를 선택하고 위젯 생성을 선택합니다.
   - 대시보드에 텍스트 블록을 추가하려면 텍스트, 구성을 선택합니다. 새 텍스트 위젯 대화 상자의 마크다운에서 마크다운을 사용하여 텍스트를 추가하고 포맷합니다. 위젯 생성을 선택합니다.
5. 선택적으로, 위젯 추가를 선택하고 4단계를 반복하여 대시보드에 다른 위젯을 추가합니다. 이 단계를 여러 번 반복할 수 있습니다.
6. 대시보드 저장을 선택합니다.

교차 계정 교차 리전 대시보드

여러 AWS 계정 및 여러 리전의 CloudWatch 데이터를 하나의 대시보드로 요약하는 교차 계정 교차 리전 대시보드를 생성할 수 있습니다. 이 개략적인 대시보드에서 전체 애플리케이션을 볼 수 있으며 계정에서 로그인 및 로그아웃하거나 리전을 전환할 필요 없이 보다 구체적인 대시보드로 드릴다운할 수 있습니다.

AWS Management 콘솔에서 프로그래밍 방식으로 교차 계정 교차 리전 대시보드를 생성할 수 있습니다.

필수 조건

교차 계정 교차 리전 대시보드를 생성하려면 먼저 하나 이상의 공유 계정과 하나 이상의 모니터링 계정을 활성화해야 합니다. 또한 AWS Management 콘솔을 사용하여 교차 계정 대시보드를 생성할 수 있으려면 교차 계정 기능에 대한 콘솔을 활성화해야 합니다. 자세한 내용은 교차 계정 교차 리전 CloudWatch 콘솔 (p. 119) 단원을 참조하십시오.

AWS Management 콘솔을 사용하여 교차 계정 교차 리전 대시보드 생성 및 사용

AWS Management 콘솔을 사용하여 교차 계정 교차 리전 대시보드를 생성할 수 있습니다.

교차 계정 교차 리전 대시보드를 생성하려면

1. 모니터링 계정에 로그인합니다.
2. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
3. 탐색 창에서 대시보드를 선택합니다.
4. 대시보드를 선택하거나 새 대시보드를 생성합니다.
5. 화면 상단에서 데이터 보기 옆의 상자를 사용하여 계정과 리전 간에 전환합니다.

   대시보드를 생성할 때 여러 계정 및 리전의 위젯을 포함할 수 있습니다. 위젯에는 그래프, 경보 및 CloudWatch Logs 인사이트 위젯이 포함됩니다.

다른 계정 및 리전의 지표로 그래프 생성

1. 모니터링 계정에 로그인합니다.
2. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
3. 탐색 창에서 지표를 선택합니다.
4. 모든 지표에서 데이터 보기 옆의 상자를 사용하여 지표를 추가할 계정과 리전을 선택합니다.
5. 그래프에 원하는 지표를 추가합니다. 자세한 내용은 지표 그래프 작성 (p. 40) 단원을 참조하십시오.
6. 4-5단계를 반복하여 다른 계정 및 리전의 지표를 추가합니다.
7. (선택 사항) 그래프로 표시된 지표 탭을 선택하고 선택한 지표를 사용하는 지표 수학 함수를 추가합니다. 자세한 내용은 지표 수식 사용 (p. 49) 단원을 참조하십시오.

   여러 `SEARCH` 함수를 포함하도록 단일 그래프를 설정할 수도 있습니다. 검색마다 서로 다른 계정 또는 리전을 참조할 수 있습니다.
8. 그래프 사용을 마치면 작업, 대시보드에 추를를 선택합니다.

   교차 계정 대시보드를 선택하고 대시보드에 추가를 선택합니다.

교차 계정 대시보드에 다른 계정의 경보 추가

1. 모니터링 계정에 로그인합니다.
2. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
3. 데이터 보기 옆의 상자를 사용하여 경보가 있는 계정을 선택합니다.
4. 탐색 창에서 Alarms를 선택합니다.
5. 추가할 경보 옆의 확인란을 선택하고 대시보드에 추가를 선택합니다.
6. 추가할 교차 계정 대시보드를 선택하고 대시보드에 추가를 선택합니다.

프로그래밍 방식으로 교차 계정 교차 리전 대시보드 생성

AWS API 및 SDK를 사용하여 프로그래밍 방식으로 대시보드를 생성할 수 있습니다. 자세한 내용은 PutDashboard를 참조하십시오.

교차 계정 교차 리전 대시보드를 활성화하기 위해 다음 표와 예제와 같이 대시보드 본문 구조에 새 파라미터를 추가했습니다. 전체 대시보드 본문 구조에 대한 자세한 내용은 대시보드 본문 구조 및 구문을 참조하십시오.

파라미터	--set-visible-to-all-users	Scope	기본값
`accountId`	위젯 또는 지표가 있는 계정의 ID를 지정합니다.	위젯 또는 지표	현재 로그인되어 있는 계정
`region`	지표의 리전을 지정합니다.	위젯 또는 지표	콘솔에서 선택된 현재 리전

다음 예제에서는 교차 계정 교차 리전 대시보드에서 위젯의 JSON 소스를 보여줍니다.

이 예제에서는 `accountId` 필드를 위젯 수준에서 공유 계정의 ID로 설정합니다. 이 위젯의 모든 지표를 해당 공유 계정 및 리전에서 가져오도록 지정합니다.

```
{
    "widgets": [
        {
            ...
            "properties": {
              "metrics": [
                     …..
                     ],
                    "accountId": "111122223333",
                    "region": "us-east-1"
           }
       }
  ]
}
```

이 예제에서는 각 지표 레벨에서 `accountId` 필드를 다르게 설정합니다. 이 예제에서는 이 지표 수학 표현식의 다른 지표를 서로 다른 공유 계정과 서로 다른 리전에서 가져옵니다.

```
{
    "widgets": [
        {
            ...
            "properties": {
                "metrics": [
                    [ { "expression": "SUM(METRICS())", "label": "[avg: ${AVG}]
 Expression1", "id": "e1", "stat": "Sum" } ],
                    [ "AWS/EC2", "CPUUtilization", { "id": "m2", "accountId":
 "5555666677778888", "region": "us-east-1", "label": "[avg: ${AVG}] ApplicationALabel
 " } ],
                    [ ".", ".", { "id": "m1", "accountId": "9999000011112222", "region":
 "eu-west-1", "label": "[avg: ${AVG}] ApplicationBLabel" } ]
                ],
                "view": "timeSeries",
                "stacked": false,
                "stat": "Sum",
                "period": 300,
                "title": "Cross account example"
            }
        }
    ]
}
```

경보 위젯을 보여주는 예제입니다.

```
{
    "type": "metric",
    "x": 6,
    "y": 0,
    "width": 6,
    "height": 6,
    "properties": {
        "accountID": "111122223333",
        "title": "over50",
        "annotations": {
            "alarms": [
                "arn:aws:cloudwatch:us-east-1:379642911888:alarm:over50"
            ]
        },
        "view": "timeSeries",
        "stacked": false
    }
}
```

이 예제는 CloudWatch Logs 인사이트 위젯에 대한 것입니다.

```
{
    "type": "log",
    "x": 0,
    "y": 6,
    "width": 24,
    "height": 6,
    "properties": {
        "query": "SOURCE 'route53test' | fields @timestamp, @message\n| sort @timestamp
 desc\n| limit 20",
        "accountId": "111122223333",
        "region": "us-east-1",
        "stacked": false,
        "view": "table"
    }
}
```

프로그래밍 방식으로 대시보드를 생성하는 또 다른 방법은 먼저 AWS Management 콘솔에서 대시보드를 생성한 다음 이 대시보드의 JSON 소스를 복사하는 것입니다. 이렇게 하려면 대시보드를 로드하고 작업, 소스 보기/편집을 선택합니다. 그런 다음 이 대시보드 JSON을 복사해 템플릿으로 사용하여 유사한 대시보드를 생성할 수 있습니다.

CloudWatch 대시보드에서 그래프를 추가 또는 제거

하나 이상의 지표를 포함하고 있는 그래프를 모니터링 중인 리소스에 대한 대시보드에 추가할 수 있습니다. 더 이상 필요가 없는 그래프는 제거할 수 있습니다.

대시보드에 그래프를 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 위젯 추가를 선택합니다.
4. 행 또는 누적 면적을 선택하고 구성을 선택합니다.
5. 모든 지표 탭에서 그래프로 표시할 지표를 선택합니다. 특정 지표가 14일 이상 데이터를 게시하지 않아 대화 상자에 나타나지 않는 경우 수동으로 추가할 수 있습니다. 자세한 내용은 CloudWatch 대시보드에서 수동으로 지표 그래프 생성 (p. 24) 단원을 참조하십시오.
6. (선택 사항) 그래프로 표시할 지표를 선택할 때 그래프 범례에서 각 지표에 대해 표시할 동적 레이블을 지정할 수 있습니다. 동적 레이블은 지표에 대한 통계를 표시하고 대시보드 또는 그래프를 새로 고칠 때

자동으로 업데이트됩니다. 동적 레이블을 추가하려면 그래프로 표시된 지표를 선택한 다음 Dynamic labels(동적 레이블)를 선택합니다.

기본적으로 레이블에 추가하는 동적 값은 레이블 시작 부분에 나타납니다. 그런 다음 지표에 대한 레이블 값을 선택하여 레이블을 편집할 수 있습니다. 자세한 내용은 동적 레이블 사용 (p. 43) 단원을 참조하십시오.

7. (선택 사항) 그래프 처리할 지표를 선택했으면 그래프에서 색상을 변경할 수 있습니다. 이렇게 하려면 그래프로 표시된 지표를 선택하고 지표 옆의 색상 정사각형을 선택하여 Color Picker 상자를 표시합니다. Color Picker에서 또 다른 색상 정사각형을 선택합니다. Color Picker 밖을 클릭하면 그래프에 새로운 색상이 나타납니다. 또는 Color Picker에서 원하는 색상에 대한 6자리 표준 HTML 16진수 색상 코드를 입력하고 Enter 키를 누릅니다.
8. (선택 사항) 그래프 처리하려는 지표에 대한 자세한 정보를 보려면 범례에 마우스 포인터를 둡니다.
9. (선택 사항) 위젯 유형을 변경하려면 그래프 제목 영역에 마우스 포인터를 놓고 위젯 작업과 위젯 유형을 선택합니다.
10. (선택 사항) 지표에 사용된 통계를 변경하려면 그래프로 표시된 지표, 통계를 선택한 후 사용하기 원하는 통계를 선택합니다. 자세한 내용은 통계 (p. 5) 단원을 참조하십시오.
11. (선택 사항) 그래프에 표시된 시간 범위를 변경하려면 그래프 맨 위의 사용자 지정이나 사용자 지정 왼쪽에 있는 기간 중 하나를 선택합니다.
12. (선택 사항) 가로 주석을 사용하면 대시보드 사용자는 지표가 특정 레벨까지 급상승하는지 아니면 지표가 사전 정의된 범위 내에 있는지 여부를 빠르게 확인할 수 있습니다. 가로 주석을 추가하려면 그래프 옵션, 가로 주석 추가를 선택합니다.

    a. 레이블에 주석의 레이블을 입력합니다.
    b. 값에 가로 주석이 표시될 지표 값을 입력합니다.
    c. Fill(채우기)에서 이 주석에 채우기 셰이딩을 사용할지 여부를 지정합니다. 예를 들어 채울 영역에 대해 `Above` 또는 `Below`를 선택합니다. `Between`을 지정할 경우 다른 `Value` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.
    d. 그래프에 여러 지표가 포함된 경우 축에서 `Value`의 숫자가 왼쪽 Y축과 연결된 지표를 참조하는지 아니면 오른쪽 Y축과 연결된 지표를 참조하는지를 지정합니다.

    주석의 왼쪽 열에서 색상 정사각형을 선택하여 주석의 채우기 색상을 변경할 수 있습니다.

    동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

    주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

    주석을 삭제하려면 작업 열에서 x를 선택합니다.
13. (선택 사항) 세로 주석을 사용하면 그래프에 운영 이벤트나 배포의 시작과 끝과 같은 마일스톤을 표시하는 데 도움이 됩니다. 세로 주석을 추가하려면 그래프 옵션, Add vertical annotation(세로 주석 추가)을 선택합니다.

    a. 레이블에 주석의 레이블을 입력합니다. 주석에 날짜와 시간만 표시하려면 레이블 필드를 비워둡니다.
    b. 날짜에서, 세로 주석이 표시되는 날짜와 시간을 지정합니다.
    c. Fill(채우기)에서 채우기 셰이딩을 세로 주석 앞에 또는 뒤에 사용할지 아니면 두 개의 세로 주석 사이에 사용할지 여부를 지정합니다. 예를 들어 채울 영역에 대해 `Before` 또는 `After`를 선택합니다. `Between`을 지정할 경우 다른 `Date` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.

    동일한 그래프에 여러 세로 주석을 추가하려면 이들 단계를 반복합니다.

    주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

    주석을 삭제하려면 작업 열에서 x를 선택합니다.
14. 위젯 생성을 선택합니다.

15. 대시보드 저장을 선택합니다.

대시보드에서 그래프를 제거하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 그래프 제목 위로 마우스를 이동하고 위젯 작업과 삭제를 선택합니다.
4. 대시보드 저장을 선택합니다. 변경 사항을 저장하기 전에 대시보드에서 다른 곳으로 이동하려고 시도하면 변경 사항을 저장하거나 삭제하라는 메시지가 나타납니다.

CloudWatch 대시보드에서 그래프 이동 또는 크기 조정

CloudWatch 대시보드에서 그래프를 정렬하고 크기를 조정할 수 있습니다.

대시보드에서 그래프를 이동시키려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 선택 아이콘이 나타날 때까지 그래프 제목에 마우스 포인터를 둡니다. 원하는 그래프를 선택하고 대시보드의 새로운 위치로 드래그합니다.
4. 대시보드 저장을 선택합니다.

그래프 크기를 조정하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 크기를 늘리거나 줄이려면 그래프에 마우스 포인터를 두고 그래프의 오른쪽 하단 모서리를 드래그합니다.
4. 대시보드 저장을 선택합니다.

그래프를 일시적으로 확대하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 그래프를 선택합니다. 또는 그래프 제목에 마우스 포인터를 놓고 위젯 작업과 확대를 선택합니다.

CloudWatch 대시보드에서 그래프를 편집하려면

그래프를 편집하여 제목, 통계 또는 기간을 변경하거나 지표를 추가 또는 제거할 수 있습니다. 하나의 그래프에 여러 개의 지표를 표시한 경우에는 필요 없는 지표를 일시적으로 숨겨서 간결하게 표현할 수 있습니다.

대시보드에서 그래프를 편집하려면

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

3. 그래프 제목에 마우스 포인터를 놓고 위젯 작업, 편집을 선택합니다.
4. 그래프의 제목을 변경하려면 해당 제목을 선택하고 새로운 제목을 입력한 다음, Enter 키를 누릅니다.
5. 그래프에 표시된 시간 범위를 변경하려면 그래프 맨 위의 사용자 지정이나 사용자 지정 왼쪽에 있는 기간 중 하나를 선택합니다.
6. 그래프의 별개 줄, 누적 줄, 숫자 등 위젯 유형을 바꾸려면 사용자 지정의 오른쪽 옆 상자를 선택한 다음 행, 누적 면적, 번호 중 하나를 선택합니다.
7. 화면 하단에 있는 그래프로 표시된 지표 탭에서 지표에 해당하는 동적 레이블, 색상, 통계 또는 기간을 변경할 수 있습니다.

   a. (선택 사항) 지표에 동적 레이블을 지정하려면 Dynamic labels(동적 레이블)를 선택합니다. 동적 레이블은 지표에 대한 통계를 표시하고 대시보드 또는 그래프를 새로 고칠 때 자동으로 업데이트됩니다.

      기본적으로 레이블에 추가하는 동적 값은 레이블 시작 부분에 나타납니다. 그런 다음 지표에 대한 레이블 값을 선택하여 레이블을 편집할 수 있습니다. 자세한 내용은 동적 레이블 사용 (p. 43) 단원을 참조하십시오.

   b. 줄 가운데 하나의 색상을 바꾸려면 지표 옆의 색상 정사각형을 선택하여 Color Picker 상자를 표시합니다. Color Picker에서 또 다른 색상을 선택하고 Color Picker 밖을 클릭하면 그래프에 새로운 색상이 나타납니다. 또는 Color Picker에서 원하는 색상에 대한 6자리 HTML 16진수 색상 코드를 입력하고 Enter 키를 누릅니다.

   c. 통계를 변경하려면 창 하단에서 통계를 선택하고 원하는 새 통계를 선택합니다. 자세한 내용은 통계 (p. 5) 단원을 참조하십시오.

   d. 창 하단의 통계 옆에 있는 기간을 바꾸려면 기간을 선택하고 다른 값을 선택합니다. 대시보드 자체에 대한 기간이 `Auto`로 설정되어 있는 경우에만 대시보드에서 이러한 새 설정값이 사용됩니다. 그렇지 않으면 대시보드에 대한 기간 설정값이 개별 위젯에 대한 기간 설정값을 덮어쓰게 됩니다.
8. 가로 주석을 추가 또는 편집하려면 그래프 옵션을 선택합니다.

   a. 가로 주석을 추가하려면 가로 주석 추가를 선택합니다.

   b. 레이블에 주석의 레이블을 입력합니다.

   c. 값에 가로 주석이 표시될 지표 값을 입력합니다.

   d. Fill(채우기)에서 어떻게 이 주석에서 채우기 셰이딩을 사용할지 지정합니다. 예를 들어 채울 영역에 대해 `Above` 또는 `Below`를 선택합니다. `Between`을 지정할 경우 다른 `Value` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.

      주석의 왼쪽 열에서 색상 정사각형을 선택하여 주석의 채우기 색상을 변경할 수 있습니다.

   e. 그래프에 여러 지표가 포함된 경우 축에서 `Value`의 숫자가 왼쪽 Y축과 연결된 지표를 참조하는지 아니면 오른쪽 Y축과 연결된 지표를 참조하는지를 지정합니다.

   동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

   주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

   주석을 삭제하려면 작업 열에서 x를 선택합니다.
9. 세로 주석을 추가하거나 편집하려면 그래프 옵션: Add vertical annotation(세로 주석 추가)을 선택합니다.

   a. 세로 주석을 추가하려면 Add vertical annotation(세로 주석 추가)을 선택합니다.

   b. 레이블에 주석의 레이블을 입력합니다. 주석에 날짜와 시간만 표시하려면 레이블 필드를 비워둡니다.

   c. 날짜에서, 세로 주석이 표시되는 날짜와 시간을 지정합니다.

   d. Fill(채우기)에서 채우기 셰이딩을 세로 주석 앞에 또는 뒤에 사용할지 아니면 두 개의 세로 주석 사이에 사용할지 여부를 지정합니다. 예를 들어 채울 영역에 대해 `Before` 또는 `After`를 선택합니다. `Between`을 지정할 경우 다른 `Date` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.

동일한 그래프에 여러 세로 주석을 추가하려면 이들 단계를 반복합니다.

주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

주석을 삭제하려면 작업 열에서 x를 선택합니다.

10. 그래프 범례의 위치를 숨기거나 변경하려면 그래프 제목에 마우스 포인터를 놓고 위젯 작업, 편집을 선택합니다. 범례에 마우스 포인터를 놓고 Hidden(숨김), Bottom(하단), 또는 오른쪽을 선택합니다.
11. Y축을 사용자 지정하려면 그래프 옵션을 선택합니다. 왼쪽 Y축 밑에 있는 레이블에 사용자 지정 레이블을 입력할 수 있습니다. 그래프가 오른쪽 Y축의 값도 표시하는 경우 이 레이블도 사용자 지정할 수 있습니다. Y축 값에 최소값과 최대값도 설정할 수 있으며 그래프는 지정한 값의 범위만 표시합니다.
12. 변경을 마쳤으면 위젯 업데이트를 선택합니다.

대시보드의 그래프에서 지표를 일시적으로 숨기려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 그래프의 바닥글에서 범례의 색상 정사각형에 마우스 포인터를 가져갑니다. 정사각형이 X로 바뀌면 선택합니다.
4. 지표를 복원하려면 회색으로 표시된 정사각형과 지표 이름을 선택합니다.

CloudWatch 대시보드에서 수동으로 지표 그래프 생성

지표가 지난 14일간 데이터를 게시하지 않은 경우 CloudWatch 대시보드에서 그래프에 추가할 지표를 검색할 때 이 지표를 찾을 수 없습니다. 다음 단계에 따라 기존 그래프에 지표를 수동으로 추가할 수 있습니다.

그래프에 검색에서 찾을 수 없는 지표를 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 대시보드에 지표를 추가하고자 하는 그래프가 이미 있어야 합니다. 아직 없는 경우 그래프를 생성한 다음 지표를 추가합니다. 자세한 내용은 CloudWatch 대시보드에서 그래프를 추가 또는 제거 (p. 20) 단원을 참조하십시오.
4. 작업, 소스 보기/편집을 선택합니다.

   JSON 블록이 표시됩니다. 블록은 대시보드 및 해당 콘텐츠의 위젯을 지정합니다. 다음은 그래프 하나를 정의하는 이 블록 일부의 예제입니다.

```
{
            "type": "metric",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "properties": {
                "view": "singleValue",
                "metrics": [
                    [ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ]
                ],
                "region": "us-west-1"
            }
```

```
        },
```

이 예제에서 다음 섹션은 이 그래프에 표시되는 지표를 정의합니다.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ]
```

5. 아직 없는 경우 닫는 괄호 뒤에 쉼표를 추가한 후, 쉼표 뒤에 비슷한 괄호로 묶은 섹션을 추가합니다. 이 새 섹션에서 네임스페이스, 지표 이름 및 그래프에 추가하려는 지표의 필요한 모든 차원을 지정합니다. 다음은 예제입니다.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ],
[ "MyNamespace", "MyMetricName", "DimensionName", "DimensionValue" ]
```

JSON으로 지표 형식 지정에 대한 자세한 정보는 지표 위젯 객체의 속성을 참조하십시오.

6. 업데이트를 선택합니다.

CloudWatch 대시보드에서 그래프 이름 변경

CloudWatch가 대시보드에서 그래프에 할당한 기본 이름을 변경할 수 있습니다.

대시보드에서 그래프 이름을 변경하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 그래프 제목 위로 마우스를 이동하고 위젯 작업, 편집을 선택합니다.
4. 그래프 편집 화면의 위쪽에서 그래프의 제목을 선택합니다.
5. 제목에 새 이름을 입력하고 확인(체크 표시)을 선택합니다. 그래프 편집 화면의 우측 하단 모서리에서 위젯 업데이트를 선택합니다.

CloudWatch 대시보드에서 텍스트 위젯을 추가 또는 제거

텍스트 위젯에는 마크다운 형식으로 된 텍스트 블록이 포함되어 있습니다. CloudWatch 대시보드에서 텍스트 위젯을 추가, 편집 또는 제거할 수 있습니다.

대시보드에 텍스트 위젯을 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 위젯 추가를 선택합니다.
4. 텍스트와 구성을 차례로 선택합니다.
5. 마크다운에서 텍스트를 추가하고 마크다운을 사용하여 서식을 지정하고 위젯 생성을 선택합니다.
6. 대시보드 저장을 선택합니다.

대시보드에서 텍스트 위젯을 편집하려면

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

3. 텍스트 블록의 오른쪽 상단 모서리 위로 마우스를 이동하고 위젯 작업과 편집을 차례로 선택합니다.
4. 필요에 따라 텍스트를 업데이트하고 위젯 업데이트를 선택합니다.
5. 대시보드 저장을 선택합니다.

대시보드에서 텍스트 위젯을 제거하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 텍스트 블록의 오른쪽 상단 모서리 위로 마우스를 이동하고 위젯 작업과 삭제를 차례로 선택합니다.
4. 대시보드 저장을 선택합니다.

CloudWatch 대시보드에서 경보를 추가 또는 제거

경보를 생성해서 대시보드에 추가할 수 있습니다. 대시보드에서 경보가 발행되면 빨간색으로 `ALARM` 상태라고 표시됩니다.

대시보드에 경보를 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보를 선택하고 추가할 경보를 선택한 다음 대시보드에 추가를 선택합니다.
3. 대시보드를 선택하고, 위젯 유형(행, 누적 면적 또는 번호)을 선택한 다음 대시보드에 추가를 선택합니다.
4. 대시보드에 경보를 표시하려면 탐색 창에서 대시보드를 선택하고 해당 대시보드를 선택합니다.
5. (선택 사항) 경보 그래프를 일시적으로 확대하려면 해당 그래프를 선택합니다.
6. (선택 사항) 위젯 유형을 변경하려면 그래프 제목 위로 마우스를 이동하고 위젯 작업과 위젯 유형을 선택합니다.

대시보드에서 경보를 제거하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 그래프 제목 위로 마우스를 이동하고 위젯 작업과 삭제를 선택합니다.
4. 대시보드 저장을 선택합니다. 변경 사항을 저장하기 전에 대시보드에서 다른 곳으로 이동하려고 시도하면 변경 사항을 저장하거나 삭제하라는 메시지가 나타납니다.

CloudWatch 대시보드에서 그래프 링크 또는 링크 해제

하나의 그래프를 확대 또는 축소하면 다른 그래프들도 동시에 확대 또는 축소가 되도록 대시보드의 그래프들을 하나로 링크할 수 있습니다. 그래프 링크를 해제하면 하나의 그래프로만 확대/축소를 제한할 수 있습니다.

대시보드에서 그래프들을 링크하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 작업을 선택한 후 그래프 연결을 선택합니다.

대시보드에서 그래프들의 링크를 해제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 작업과 그래프 연결의 선택을 차례로 취소합니다.

즐겨찾기 목록에 대시보드 추가

즐겨찾는 대시보드 목록에 CloudWatch 대시보드를 추가하면 빠르게 찾을 수 있습니다. 즐겨찾기 목록은 탐색 창의 맨 아래에 나타납니다.

즐겨찾기 목록에 대시보드를 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택합니다.
3. 대시보드 옆에 있는 별표 기호를 선택하여 추가합니다.

CloudWatch 대시보드에 대한 기간 재정의 설정 또는 새로 고침 간격 변경

이 대시보드에 추가된 그래프의 기간 설정을 유지하거나 수정하는 방법을 지정할 수 있습니다.

기간 재정의 옵션을 변경하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 작업을 선택합니다.
3. 기간 재정의에서 다음 중 하나를 선택합니다.

   - 각 그래프에서 지표 기간을 대시보드의 시간 범위에 맞게 자동으로 조정하려면 자동을 선택합니다.
   - 각 그래프의 기간 설정을 항상 준수하려면 재정의하지 않습니다를 선택합니다.
   - 대시보드에 추가된 그래프의 기간 설정을, 선택된 해당 기간 설정으로 항상 조정하려면 다른 옵션 중 하나를 선택합니다.

   기간 재정의는 대시보드를 종료하거나 브라우저를 새로 고치면 항상 자동으로 재설정됩니다. 기간 재정의에 대해 설정을 여러 개 저장할 수는 없습니다.

CloudWatch 대시보드의 데이터가 새로 고침되는 간격을 변경하고 자동 새로 고침이 되도록 설정할 수 있습니다.

대시보드 새로 고침 간격을 변경하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 새로 고침 옵션 메뉴(오른쪽 상단 모서리)에서 10초, 1분, 2분, 5분 또는 15분을 선택합니다.

대시보드를 자동으로 새로 고치려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.

2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 새로 고침 옵션, 자동 새로 고침을 선택합니다.

CloudWatch 대시보드의 시간 범위 또는 시간대 형식 변경

대시보드 데이터가 표시되는 시간 범위를 분, 시간, 일 또는 주 단위로 변경할 수 있습니다. 대시보드 데이터가 표시되는 시간 형식도 UTC 또는 현지 시간으로 변경할 수 있습니다.

Note

100개 이상의 고분해능 지표를 포함하는 그래프를 사용하여 대시보드를 생성하는 경우 양호한 대시보드 성능이 유지되도록 시간 범위를 1시간 이내로 설정하는 것이 좋습니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.

대시보드 시간 범위를 변경하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 다음 중 하나를 수행하십시오.
   - 1h, 3h, 12h, 1d, 3d, 1w와 같이 1시간부터 1주까지 사전 정의된 범위 중 하나를 선택합니다.
   - 사용자 지정을 선택하고 상대를 선택합니다. 1분부터 15개월까지 사전 정의된 범위 중 하나를 선택합니다.
   - 사용자 지정을 선택하고 절대를 선택합니다. 캘린더 선택기나 텍스트 필드를 사용하여 시간 범위를 지정합니다.

Note

집계 기간이 자동으로 설정된 상태에서 그래프의 시간 범위를 변경할 경우 CloudWatch가 기간을 변경할 수 있습니다. 수동으로 기간을 설정하려면 작업을 선택하고 기간:에 새 값을 선택합니다.

대시보드 시간 형식을 변경하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 사용자 지정을 선택합니다.
4. 상단 모서리에서 UTC 또는 현지 시간대를 선택합니다.

Amazon CloudWatch 지표 사용

지표는 시스템 성능에 대한 데이터입니다. 기본적으로 몇몇 서비스는 리소스에 대한 무료 지표(예: Amazon EC2 인스턴스, Amazon EBS 볼륨 및 Amazon RDS DB 인스턴스)를 제공합니다. Amazon EC2 인스턴스 같은 일부 리소스에 대한 세부 모니터링을 활성화하거나 자체 애플리케이션 지표를 게시할 수도 있습니다. Amazon CloudWatch는 검색, 그래프 작성 및 경보 발행을 위해 계정에 모든 지표(제공된 AWS 리소스 지표 및 애플리케이션 지표 둘 다)를 로드할 수 있습니다.

지표 데이터는 15개월 동안 보관되기 때문에 최신 데이터와 이력 데이터를 모두 볼 수 있습니다.

목차

- 얻을 수 있는 지표 보기 (p. 29)
- 사용 가능한 지표 검색 (p. 32)
- 지표에 대한 통계 얻기 (p. 33)
- 지표 그래프 작성 (p. 40)
- 사용자 지정 메트릭 게시 (p. 47)
- 지표 수식 사용 (p. 49)
- 그래프에서 검색 표현식 사용 (p. 59)

얻을 수 있는 지표 보기

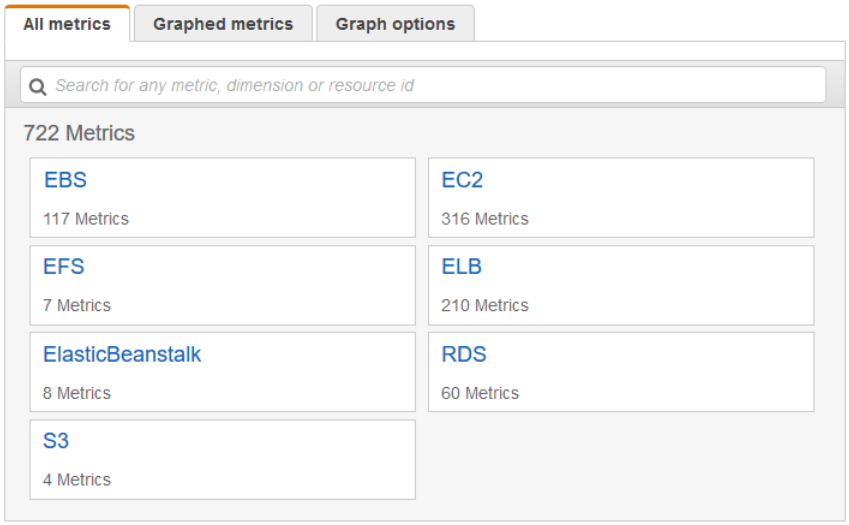
지표는 먼저 네임스페이스별로 그룹화된 다음, 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다. 예를 들어 모든 EC2 지표, 인스턴스별로 그룹화된 EC2 지표, Auto Scaling 그룹별로 그룹화된 EC2 지표를 모두 볼 수 있습니다.

사용 중인 AWS 서비스만 Amazon CloudWatch로 지표를 보냅니다.

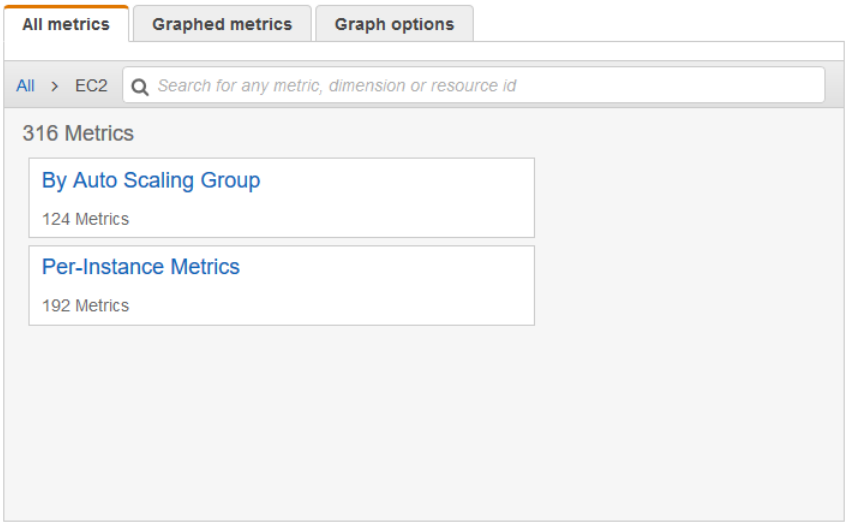
CloudWatch에 지표를 전송하는 AWS 서비스 목록은 CloudWatch 지표를 게시하는 AWS 서비스 (p. 319) 단원을 참조하십시오. 이 페이지에서는 각 서비스에 의해 게시된 지표 및 차원도 볼 수 있습니다.

콘솔을 사용하여 네임스페이스와 차원별로 사용 가능한 지표를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 지표 네임스페이스(예: EC2)를 선택합니다.



4. 지표 차원(예: 인스턴스별 지표)을 선택합니다.



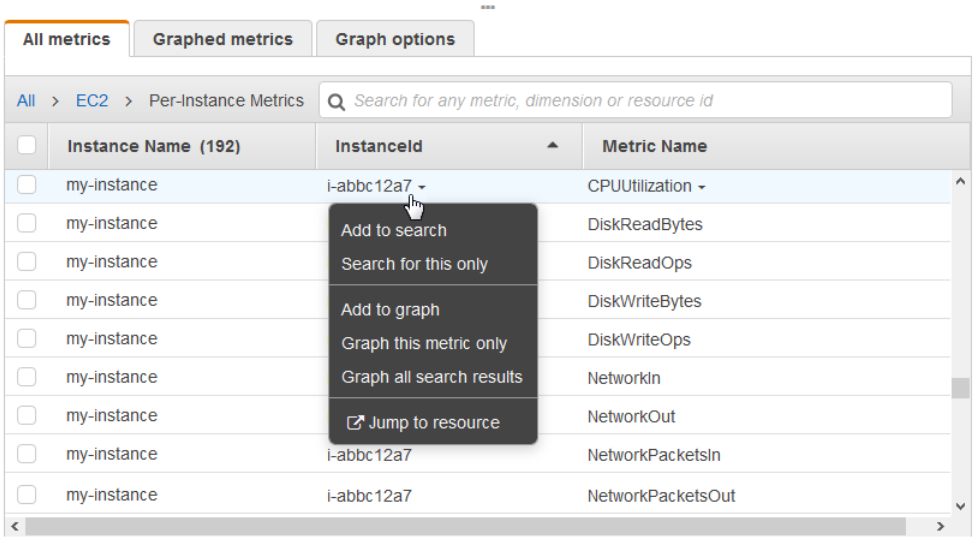
5. 모든 지표 탭에 네임스페이스의 해당 차원에 대한 모든 지표가 표시됩니다. 다음을 수행할 수 있습니다.

   a. 테이블을 정렬하려면 열 머리글을 사용합니다.

   b. 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다. 모든 지표를 선택하려면 테이블의 머리글 행에 있는 확인란을 선택합니다.

   c. 리소스로 필터링하려면 리소스 ID를 선택한 후 검색에 추가를 선택합니다.

   d. 측정치로 필터링하려면 측정치 이름을 선택한 후 검색에 추가를 선택합니다.



AWS CLI를 사용하여 네임스페이스, 차원 또는 지표별로 사용 가능한 지표를 보려면

list-metrics 명령을 사용하여 CloudWatch 지표를 나열합니다. 지표를 게시하는 모든 서비스의 네임스페이스, 지표 및 차원 목록을 보려면 CloudWatch 지표를 게시하는 AWS 서비스 (p. 319) 단원을 참조하십시오.

다음 예제는 Amazon EC2에 대한 모든 지표를 볼 수 있도록 `AWS/EC2` 네임스페이스를 지정합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

다음은 예제 출력입니다.

```
{
  "Metrics" : [
    ...
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkOut"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
```

```
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkIn"
    },
    ...
  ]
}
```

지정된 리소스에서 사용 가능한 모든 지표를 나열하려면

다음 예제는 지정한 인스턴스의 결과만 보도록 `AWS/EC2` 네임스페이스와 `InstanceId` 차원을 지정합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
 Name=InstanceId,Value=i-1234567890abcdef0
```

모든 리소스에 대한 지표를 나열하려면

다음 예제는 지정한 지표의 결과만 보도록 `AWS/EC2` 네임스페이스와 지표 이름을 지정합니다.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

사용 가능한 지표 검색

대상으로 지정된 검색어를 사용하여 계정의 모든 지표 중에서 검색할 수 있습니다. 네임스페이스, 지표 이름 또는 차원 내에서 결과가 일치하는 지표가 반환됩니다.

CloudWatch에서 사용 가능한 지표를 검색하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 모든 지표 탭의 검색 필드에 검색어(예: 지표 이름, 네임스페이스, 차원 이름이나 값 또는 리소스 이름)를 입력합니다. 이렇게 하면 이 검색어가 포함된 지표가 있는 모든 네임스페이스가 표시됩니다.

   예를 들어 **volume**을 검색하면 이름에 이 단어가 있는 지표가 포함된 네임스페이스가 표시됩니다.

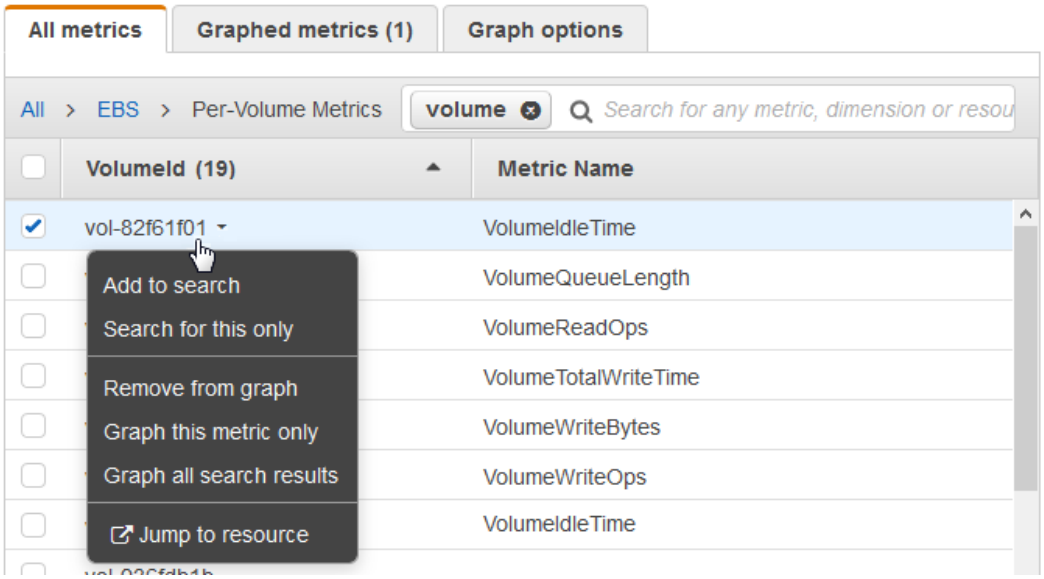
   검색에 대한 자세한 내용은 그래프에서 검색 표현식 사용 (p. 59) 단원을 참조하십시오.
4. 모든 검색 결과를 그래프로 작성하려면 Graph search(그래프 검색)를 선택합니다.

   또는

   해당 네임스페이스에서 지표를 볼 네임스페이스를 선택합니다. 그러면 다음 작업을 수행할 수 있습니다.

   a. 하나 이상의 지표를 그래프 처리하려면 각 지표 옆에 있는 확인란을 선택합니다. 모든 지표를 선택하려면 테이블의 머리글 행에 있는 확인란을 선택합니다.
   b. 검색을 구체화하려면 지표 이름 위로 마우스를 이동하고 검색에 추가 또는 이 항목만 검색을 선택합니다.
   c. 콘솔에서 리소스 중 하나를 보려면 리소스 ID를 선택하고 리소스로 이동을 선택합니다.
   d. 지표에 대한 도움말을 보려면 지표 이름을 선택하고 이것은 무엇입니까?를 선택합니다.

   선택한 지표가 그래프에 표시됩니다.



5. (선택 사항) 검색어의 해당 부분을 편집하려면 검색 표시줄에 있는 버튼 중 하나를 선택합니다.

지표에 대한 통계 얻기

다음 예제는 EC2 인스턴스 같은 리소스에서 CloudWatch 지표에 대한 통계를 얻을 수 있는 방법을 보여줍니다.

예제

- 특정 리소스에 대한 통계 얻기 (p. 33)
- 리소스에서 통계 집계 (p. 36)
- Auto Scaling 그룹별 통계 집계 (p. 38)
- Amazon Machine Image(AMI)별 통계 집계 (p. 39)

특정 리소스에 대한 통계 얻기

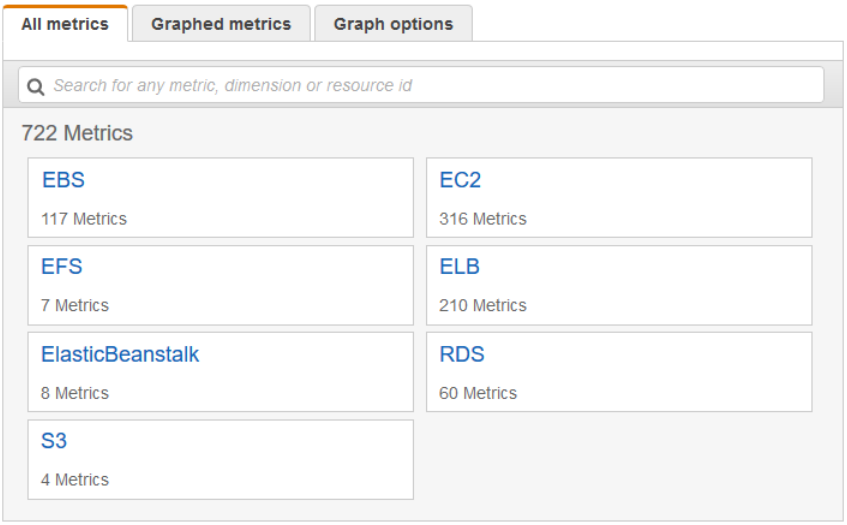
다음 예제는 특정 EC2 인스턴스의 최대 CPU 사용률을 확인하는 방법을 보여 줍니다.

요구 사항

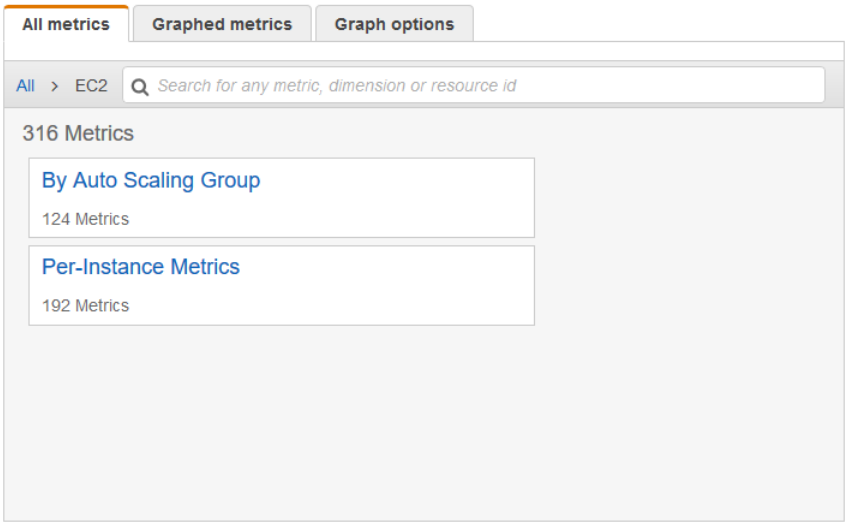
- 인스턴스의 ID가 필요합니다. 인스턴스 ID는 Amazon EC2 콘솔이나 describe-instances 명령을 사용하여 확인할 수 있습니다.
- 기본적으로 기본 모니터링이 사용되지만 세부 모니터링을 사용하도록 설정할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 인스턴스에 대한 세부 모니터링 활성화 또는 비활성화를 참조하십시오.

콘솔을 사용하여 특정 인스턴스에 대한 평균 CPU 사용률을 표시하려면

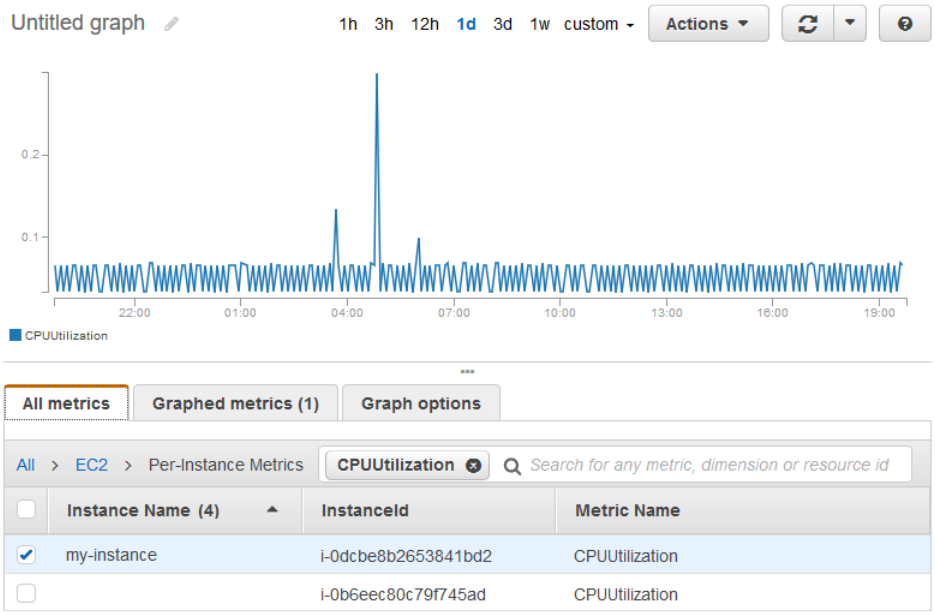
1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 지표 네임스페이스를 선택합니다.



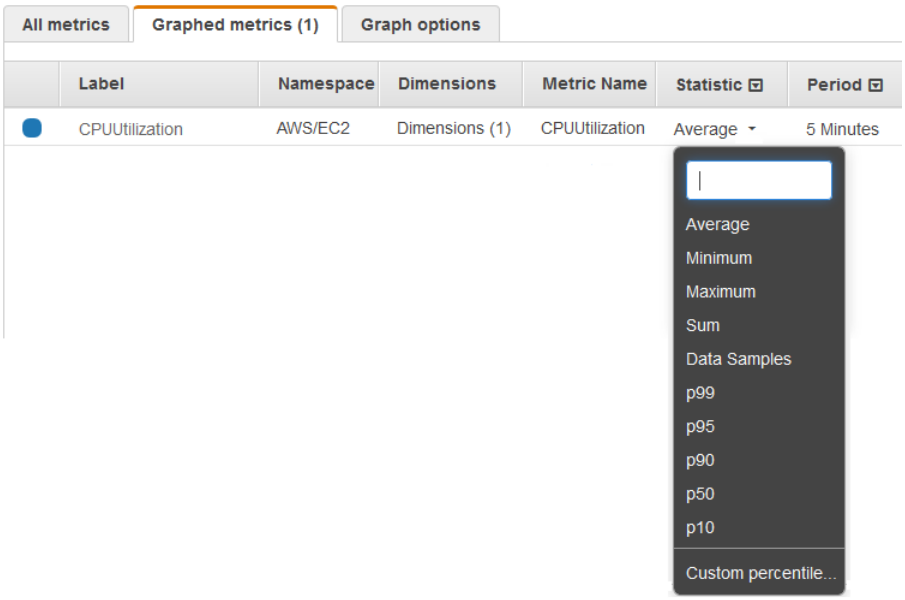
4. 인스턴스별 지표 차원을 선택합니다.



5. 검색 필드에 **`CPUUtilization`**을 입력하고 Enter를 누릅니다. 특정 인스턴스의 행을 선택합니다. 그러면 해당 인스턴스의 `CPUUtilization` 지표 그래프가 표시됩니다. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 사용자 지정을 선택합니다.



6. 통계를 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: **`p95.45`**)를 지정합니다.



7. 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

AWS CLI를 사용하여 EC2 인스턴스별 CPU 사용률을 얻으려면

다음과 같이 get-metric-statistics 명령을 사용하여 지정한 인스턴스의 `CPUUtilization` 지표를 확인합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=InstanceId,Value=i-1234567890abcdef0 --statistics Maximum \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00 --period 360
```

24시간이라는 요청된 시간 간격에서 6분마다 통계 값이 반환됩니다. 각 값은 6분이라는 특정 기간 동안 지정된 인스턴스의 최대 CPU 사용률을 나타냅니다. 데이터 요소는 시간 순서대로 반환되지 않습니다. 다음은 예제 출력의 시작 부분을 보여줍니다(전체 출력에는 24시간 동안 6분마다 반환된 데이터 요소가 포함됨).

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

리소스에서 통계 집계

여러 리소스에서 AWS 리소스에 대한 지표를 집계할 수 있습니다. Amazon CloudWatch는 리전 간에 데이터를 집계할 수 없습니다. 리전마다 지표가 완전히 분리되어 있습니다.

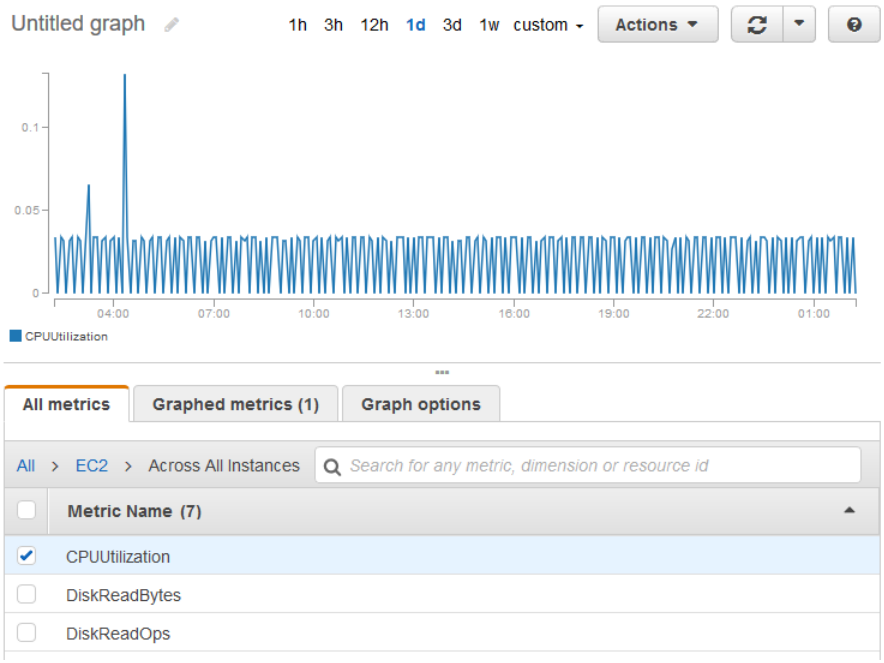
예를 들어 세부 모니터링이 활성화된 EC2 인스턴스에 대한 통계를 집계할 수 있습니다. 기본 모니터링을 사용하는 인스턴스는 포함되지 않습니다. 따라서 1분 기간으로 데이터를 제공하는 세부 모니터링(추가 요금 부과)을 활성화해야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 인스턴스에 대한 세부 모니터링 활성화 또는 비활성화를 참조하십시오.

이 예제는 EC2 인스턴스의 평균 CPU 사용률을 확인하는 방법을 보여줍니다. 지정된 차원이 없으므로 CloudWatch에서는 `AWS/EC2` 네임스페이스의 모든 차원에 대한 통계를 반환합니다. 다른 지표에 대한 통계를 얻으려면 CloudWatch 지표를 게시하는 AWS 서비스 (p. 319) 단원을 참조하십시오.

> **Important**
>
> AWS 네임스페이스에서 모든 차원을 검색하는 기능은 CloudWatch에 게시한 사용자 지정 네임스페이스에 대해서는 작동하지 않습니다. 사용자 지정 네임스페이스를 사용하는 경우 데이터 요소가 포함된 통계를 검색하려면 특정 데이터 요소와 연결된 전체 차원 세트를 지정해야 합니다.

EC2 인스턴스에 대한 평균 CPU 사용률을 표시하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 네임스페이스를 선택한 후 전체 인스턴스를 선택합니다.
4. `CPUUtilization`을 포함하는 행을 선택합니다. 그러면 모든 EC2 인스턴스에 대한 지표 그래프가 표시됩니다. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 사용자 지정을 선택합니다.



5. 통계를 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: **p95.45**)를 지정합니다.
6. 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

AWS CLI를 사용하여 EC2 인스턴스 간 평균 CPU 사용률을 얻으려면

다음과 같이 get-metric-statistics 명령을 사용합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --
statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00 --period 3600
```

다음은 예제 출력입니다.

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2016-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        ...
```

```
    ],
    "Label": "CPUUtilization"
}
```

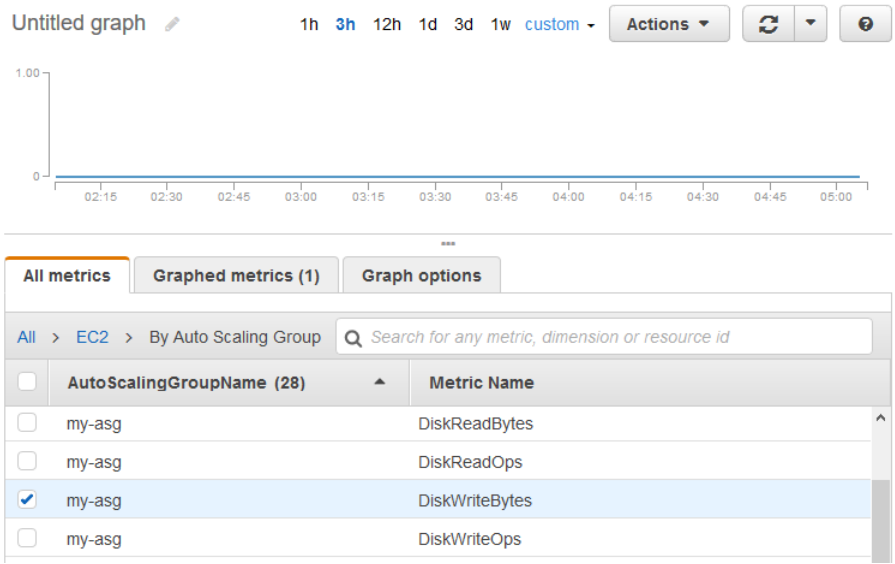
Auto Scaling 그룹별 통계 집계

EC2 인스턴스에 대한 통계를 한 Auto Scaling 그룹으로 집계할 수 있습니다. Amazon CloudWatch는 리전 간에 데이터를 집계할 수 없습니다. 리전마다 지표가 완전히 분리되어 있습니다.

이 예제는 하나의 Auto Scaling 그룹에 대해 디스크에 기록되는 총 바이트 수를 확인하는 방법을 보여줍니다. 이 총계는 지정한 Auto Scaling 그룹의 모든 EC2 인스턴스에서 24시간 간격으로 1분 기간에 대해 계산됩니다.

콘솔을 사용하여 한 Auto Scaling 그룹의 인스턴스에 대한 DiskWriteBytes를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 네임스페이스를 선택한 후 Auto Scaling 그룹별을 선택합니다.
4. DiskWriteBytes 측정치의 행과 특정 Auto Scaling 그룹을 선택합니다. 그러면 해당 Auto Scaling 그룹의 인스턴스에 대한 측정치 그래프가 표시됩니다. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 사용자 지정을 선택합니다.



5. 통계를 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: **p95.45**)를 지정합니다.
6. 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

AWS CLI를 사용하여 한 Auto Scaling 그룹의 인스턴스에 대한 DiskWriteBytes를 얻으려면

다음과 같이 get-metric-statistics 명령을 사용합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
--dimensions Name=AutoScalingGroupName,Value=my-asg --statistics "Sum" "SampleCount" \
--start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00 --period 360
```

다음은 예제 출력입니다.

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2016-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2016-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

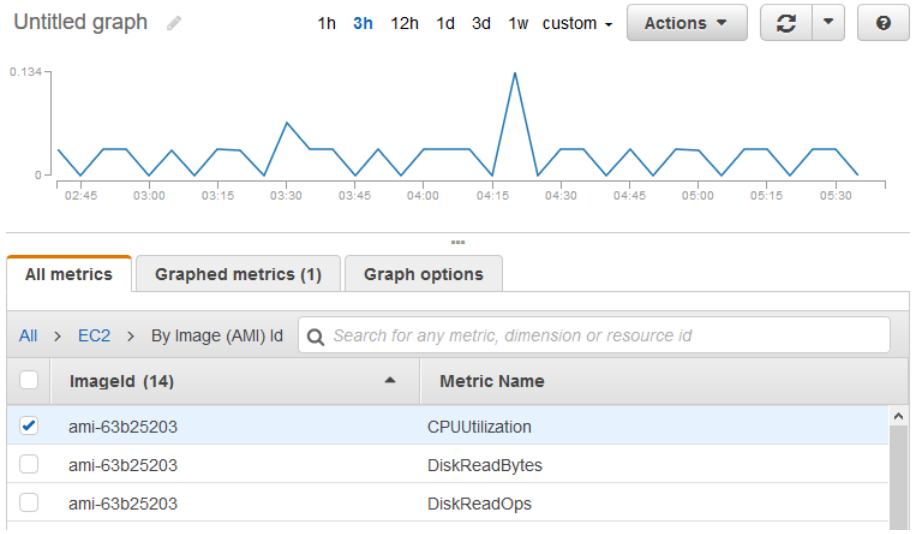
Amazon Machine Image(AMI)별 통계 집계

세부 모니터링이 활성화된 EC2 인스턴스에 대해 통계를 집계할 수 있습니다. 기본 모니터링을 사용하는 인스턴스는 포함되지 않습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 인스턴스에 대한 세부 모니터링 활성화 또는 비활성화를 참조하십시오.

이 예제는 지정된 AMI를 사용하는 모든 인스턴스의 평균 CPU 사용률을 확인하는 방법을 보여줍니다. 평균은 1일 기간의 60초 시간 간격에 대한 평균입니다.

콘솔을 사용하여 AMI의 평균 CPU 사용률을 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. EC2 네임스페이스를 선택한 후 이미지(AMI) ID별을 선택합니다.
4. `CPUUtilization` 지표 행과 특정 AMI를 선택합니다. 그러면 지정한 AMI의 지표 그래프가 표시됩니다. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 사용자 지정을 선택합니다.



5. 통계를 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: **`p95.45`**)를 지정합니다.
6. 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

AWS CLI를 사용하여 AMI당 평균 CPU 사용률을 얻으려면

다음과 같이 get-metric-statistics 명령을 사용합니다.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=ImageId,Value=ami-3c47a355 --statistics Average \
--start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00 --period 3600
```

이 작업은 1일 간격으로 1시간 값인 통계를 반환합니다. 각 값은 지정된 AMI를 실행 중인 EC2 인스턴스의 평균 CPU 사용률을 나타냅니다. 다음은 예제 출력입니다.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

지표 그래프 작성

CloudWatch 콘솔을 사용하여 다른 AWS 서비스에서 생성된 지표 데이터를 그래프로 표시합니다. 이렇게 하면 서비스의 지표 활동을 보다 효율적으로 확인할 수 있습니다. 다음 절차에서는 CloudWatch에서 지표를 그래프로 표시하는 방법에 대해 설명합니다.

목차

- 지표 그래프 작성 (p. 40)
- 동적 레이블 사용 (p. 43)
- 그래프의 시간 범위 또는 시간대 형식 수정 (p. 44)
- 그래프의 Y축 수정 (p. 45)
- 그래프의 지표에서 경보 생성 (p. 46)

지표 그래프 작성

CloudWatch 콘솔을 사용하여 지표를 선택하고 지표 데이터의 그래프를 작성할 수 있습니다.

CloudWatch는 지표에 대해 `Average`, `Minimum`, `Maximum`, `Sum`, `SampleCount` 등의 통계를 지원합니다. 자세한 내용은 통계 (p. 5) 단원을 참조하십시오.

다양한 세부 수준에서 데이터를 볼 수 있습니다. 예를 들어, 문제 해결 시 유용할 수 있는 1분 보기를 선택할 수 있습니다. 또는 세부적이지 않은 1시간 보기를 선택할 수 있습니다. 이 보기는 시간 경과에 따른 추세를 볼

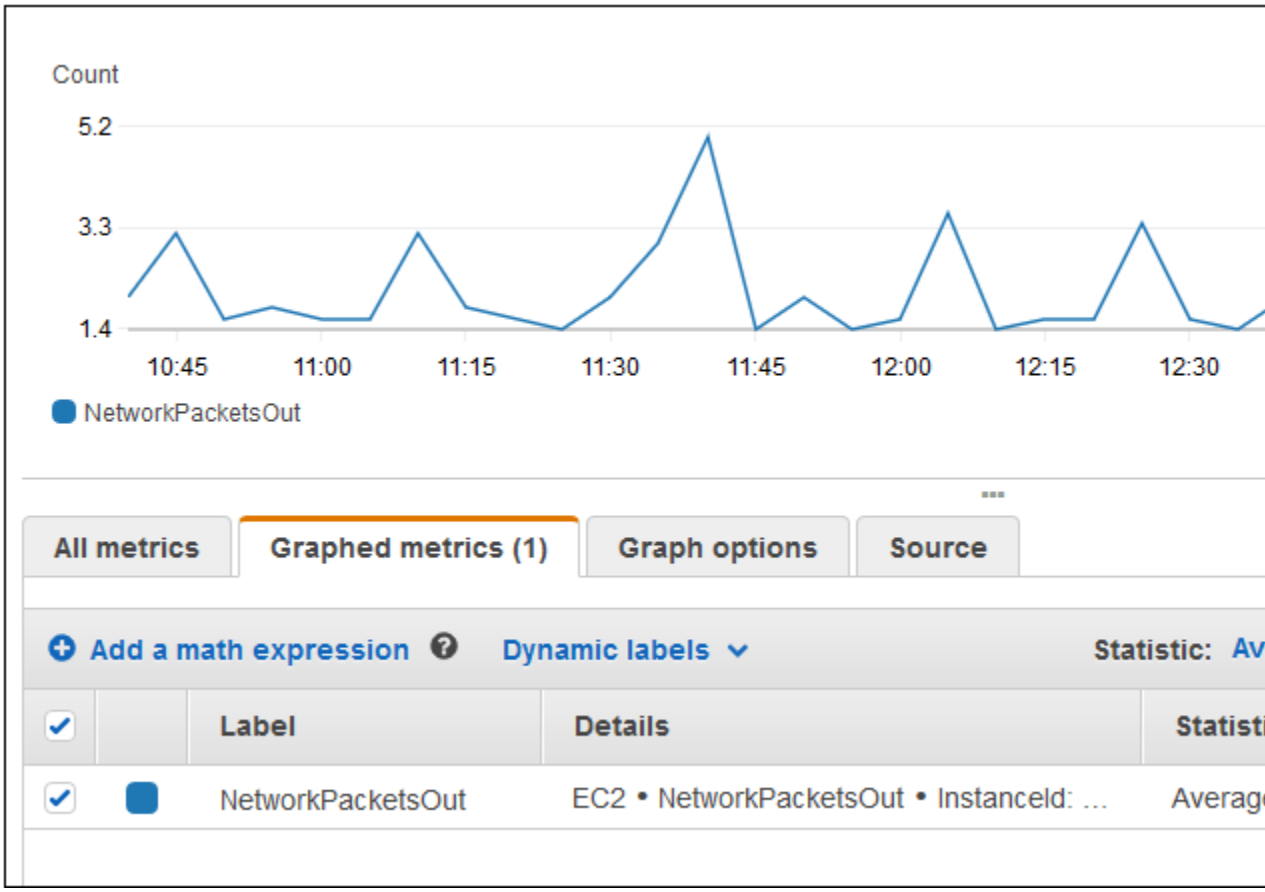
수 있도록 넓은 시간 범위(예: 3일)를 볼 때 유용할 수 있습니다. 자세한 내용은 기간 (p. 6) 단원을 참조하십시오.

그래프 작성

지표 그래프를 작성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 모든 지표 탭의 검색 필드에 검색어(예: 지표 이름 또는 리소스 이름)를 입력하고 Enter 키를 누릅니다.

   예를 들어, `CPUUtilization` 지표를 검색하면 이 지표와 함께 네임스페이스와 차원이 표시됩니다.
4. 지표 검색을 위해 결과 중 하나를 선택합니다.
5. 하나 이상의 지표를 그래프 처리하려면 각 지표 옆에 있는 확인란을 선택합니다. 모든 지표를 선택하려면 테이블의 머리글 행에 있는 확인란을 선택합니다.
6. (선택 사항) 지표에 대한 예상 값을 보여주는 이상 탐지 밴드를 추가하려면, 지표 옆에 있는 작업에서 이상 탐지 아이콘을 선택합니다.



CloudWatch에서는 지표의 최근 과거 데이터를 최대 2주분까지 사용하여 예상 값에 대한 모델을 계산합니다. 그런 다음 이 예상 값의 범위를 그래프에 밴드로 표시합니다. CloudWatch에서는 지표 아래에 새로운 행을 추가하여 ANOMALY_DETECTION_BAND로 레이블이 지정된 이상 탐지 밴드 수학 표현식을 표시합니다. 최근 과거 데이터가 있으면 모델에서 생성된 이상 탐지 밴드의 근사치인 미리 보기 이상 탐지 밴드가 즉시 표시됩니다. 실제 이상 탐지 밴드가 표시되기까지 최대 15분이 걸립니다.

기본적으로 CloudWatch에서는 밴드 임계값에 대한 2개의 기본값으로 예상 값 밴드의 상한과 하한을 생성합니다. 이 숫자를 변경하려면, 밴드의 세부정보에서 표현식 끝에 있는 값을 변경합니다.

- (선택 사항) 이상 탐지 모델 계산 방법을 변경하려면 Edit model(모델 편집)을 선택합니다. 모델 계산을 위해 교육에서 과거 및 미래 기간을 사용하지 않도록 제외할 수 있습니다. 시스템 중단, 배포 및 휴일과 같은 비정상적인 이벤트 시스템을 교육 데이터에서 제외하는 것이 중요합니다. 일광 절약 시간제 변경을 위해 모델에 사용할 시간대를 지정할 수도 있습니다.

자세한 내용은 CloudWatch 이상 탐지 사용 (p. 123) 단원을 참조하십시오.

그래프에서 모델을 숨기려면 `ANOMALY_DETECTION_BAND` 기능이 있는 선에서 체크 표시를 제거하거나 `x` 아이콘을 선택하십시오. 모델을 완전히 삭제하려면 Edit model(모델 편집), Delete model(모델 삭제)을 선택합니다.

7. (선택 사항) 그래프로 표시할 지표를 선택할 때 각 지표의 그래프 범례에 표시할 동적 레이블을 지정합니다. 동적 레이블은 지표에 대한 통계를 표시하고 대시보드 또는 그래프를 새로 고칠 때 자동으로 업데이트됩니다. 동적 레이블을 추가하려면 그래프로 표시된 지표, Dynamic labels(동적 레이블)를 선택합니다.

   기본적으로 레이블에 추가하는 동적 값은 레이블 시작 부분에 나타납니다. 그런 다음 지표에 대한 레이블 값을 선택하여 레이블을 편집할 수 있습니다. 자세한 내용은 동적 레이블 사용 (p. 43) 단원을 참조하십시오.
8. 그래프로 표시하려는 지표에 대한 자세한 정보를 보려면 범례에 마우스 포인터를 둡니다.
9. 가로 주석을 사용하면 그래프 사용자가 지표가 특정 레벨까지 급상승하는지 아니면 지표가 사전 정의된 범위 내에 있는지 여부를 보다 효율적으로 확인할 수 있습니다. 가로 주석을 추가하려면 그래프 옵션을 선택한 후 가로 주석 추가를 선택합니다.

   a. 레이블에 주석의 레이블을 입력합니다.
   b. 값에 가로 주석이 표시될 지표 값을 입력합니다.
   c. Fill(채우기)에서 이 주석에 채우기 셰이딩을 사용할지 여부를 지정합니다. 예를 들어 채울 영역에 대해 `Above` 또는 `Below`를 선택합니다. `Between`을 지정할 경우 다른 `Value` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.
   d. 그래프에 여러 지표가 포함된 경우 축에서 `Value`의 숫자가 왼쪽 Y축과 연결된 지표를 참조하는지 아니면 오른쪽 Y축과 연결된 지표를 참조하는지를 지정합니다.

      주석의 왼쪽 열에서 색상 정사각형을 선택하여 주석의 채우기 색상을 변경할 수 있습니다.

   동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

   주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

   주석을 삭제하려면 작업 열에서 x를 선택합니다.
10. 그래프에 대한 URL을 얻으려면 작업과 공유를 선택합니다. 저장하거나 공유할 URL을 복사합니다.
11. 대시보드에 그래프를 추가하려면 작업과 대시보드에 추가를 선택합니다.

그래프 업데이트

그래프를 업데이트하려면

1. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다.
2. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 사용자 지정을 선택합니다. 자세한 내용은 그래프의 시간 범위 또는 시간대 형식 수정 (p. 44) 단원을 참조하십시오.
3. 통계를 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: **`p95.45`**)를 지정합니다.
4. 기간을 변경하려면 그래프로 표시된 지표 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

5. 가로 주석을 추가하려면 그래프 옵션을 선택한 후 가로 주석 추가를 선택합니다.

   a. 레이블에 주석의 레이블을 입력합니다.

   b. 값에 가로 주석이 표시될 지표 값을 입력합니다.

   c. Fill(채우기)에서 이 주석에 채우기 셰이딩을 사용할지 여부를 지정합니다. 예를 들어 채울 영역에 대해 `Above` 또는 `Below`를 선택합니다. `Between`을 지정할 경우 다른 `Value` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.

   d. 그래프에 여러 지표가 포함된 경우 축에서 `Value`의 숫자가 왼쪽 Y축과 연결된 지표를 참조하는지 아니면 오른쪽 Y축과 연결된 지표를 참조하는지를 지정합니다.

   주석의 왼쪽 열에서 색상 정사각형을 선택하여 주석의 채우기 색상을 변경할 수 있습니다.

   동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

   주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

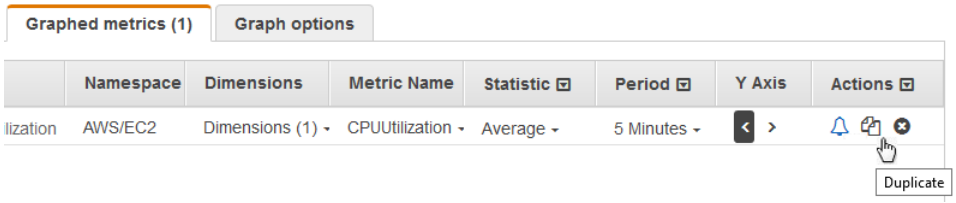
   주석을 삭제하려면 작업 열에서 x를 선택합니다.

6. 새로 고침 간격을 변경하려면 새로 고침 옵션을 선택한 후 자동 새로 고침을 선택하거나 1분, 2분, 5분 또는 15분을 선택합니다.

지표 복제

지표를 복제하려면

1. 그래프로 표시된 지표 탭을 선택합니다.
2. 작업에서 복제 아이콘을 선택합니다.



3. 필요에 따라 복제 지표를 업데이트합니다.

동적 레이블 사용

그래프에서 동적 레이블을 사용할 수 있습니다. 동적 레이블은 선택한 지표의 레이블에 동적으로 업데이트된 값을 추가합니다. 표시된 값에는 평균, 최대, 최소, 합계 또는 최근 데이터 포인트가 포함될 수 있습니다. 레이블에 표시된 동적 값은 현재 그래프에 표시된 시간 범위에서 파생됩니다. 대시보드 또는 그래프를 새로 고칠 때 레이블의 이 부분이 자동으로 업데이트됩니다.

검색 표현식에서 동적 레이블을 사용하면 검색에서 반환된 모든 지표에 동적 레이블이 적용됩니다.

CloudWatch 콘솔을 사용하면 레이블에 동적 값을 쉽게 추가할 수 있습니다. 그런 다음 레이블을 추가로 편집하고 레이블에서 동적 값의 위치를 변경하거나 다른 사용자 지정 작업을 수행할 수 있습니다.

동적 레이블 구문

동적 레이블 내에서 동적 값은 다음을 포함할 수 있습니다.

변수	설명
${AVG}	현재 그래프에 표시된 시간 범위의 평균 값
${LAST}	현재 그래프에 표시된 시간 범위에서 가장 최근 값
${LABEL}	지표의 기본 레이블
${MAX}	현재 그래프에 표시된 시간 범위의 최대값
${MIN}	현재 그래프에 표시된 시간 범위의 최소값
${SUM}	현재 그래프에 표시된 시간 범위 값의 합계

예를 들어 각 Lambda 함수에 대해 `Errors`를 찾는 검색 표현식 **`SEARCH(' {AWS/Lambda, FunctionName} Errors ', 'Sum', 300)`**가 있다고 가정합니다. 레이블을 `[max: ${MAX} Errors for Function Name ${LABEL}]`로 설정하면 각 지표의 레이블은 [max: ***`number`*** Errors for Function Name ***`Name`***]이 됩니다.

레이블에 최대 5개의 동적 통계 값을 추가할 수 있습니다. 각 레이블 내에서 `${LABEL}` 자리 표시자를 한 번만 사용할 수 있습니다.

그래프의 시간 범위 또는 시간대 형식 수정

그래프의 시간 범위 또는 시간대 형식을 변경할 수 있습니다.

상대적 시간 범위 설정

그래프의 상대적 시간 범위를 설정할 수 있습니다.

그래프의 상대적 시간 범위를 설정하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 페이지 상단에 나와 있는 사전 정의된 범위(1시간부터 1주까지) 중 하나를 선택합니다.
4. 더 많은 사전 정의된 범위를 보려면 사용자 지정 메뉴를 선택하고 상대를 선택합니다. 5분부터 15개월까지 사전 정의된 범위 중 하나를 선택합니다.

절대적 시간 범위 설정

그래프의 절대적 시간 범위를 설정할 수 있습니다.

그래프의 절대적 시간 범위를 설정하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 사용자 지정 메뉴에서 절대를 선택합니다. 캘린더 선택기나 텍스트 필드를 사용하여 시간 범위를 지정합니다.

시간대 형식 설정

그래프에서 UTC 시간을 사용할지 아니면 사용자의 현지 시간을 사용할지를 지정할 수 있습니다.

그래프의 시간 범위를 지정하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 사용자 지정 메뉴를 선택한 다음 UTC 또는 현지 시간대를 선택합니다.

그래프 확대

그래프의 세부 수준을 변경하여 잠깐 동안 데이터를 확대해서 볼 수 있습니다.

그래프를 확대하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 그래프 영역을 선택하여 드래그한 후 마우스 버튼을 놓습니다.
4. 확대된 그래프를 재설정하려면 확대/축소 재설정 아이콘을 선택합니다.

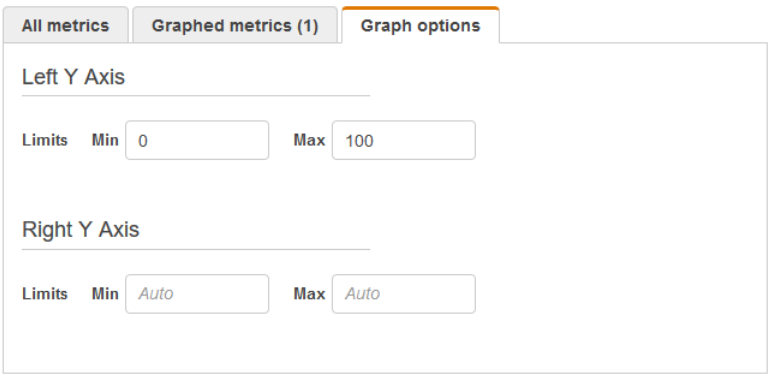
그래프의 Y축 수정

데이터 확인이 용이하도록 그래프의 Y축에 대해 사용자 지정 경계를 설정할 수 있습니다. 예를 들어, CPU 사용률이 낮은지(그려지는 선이 그래프 하단 부근에 있음) 또는 높은지(그려지는 선이 그래프 상단 부근에 있음)를 쉽게 확인할 수 있도록 `CPUUtilization` 그래프의 경계를 100%로 변경할 수 있습니다.

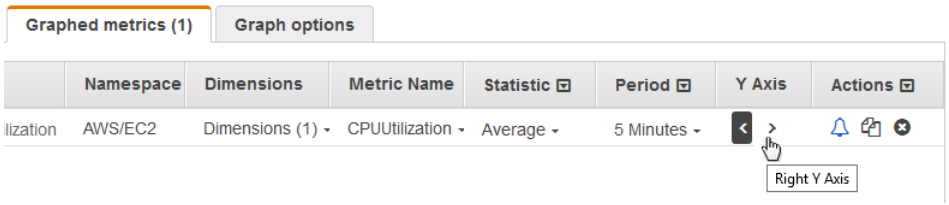
그래프에서 서로 다른 두 Y축 간을 전환할 수 있습니다. 이 기능은 그래프에 단위가 다르거나 값의 범위에 큰 차이가 있는 지표가 포함되어 있는 경우에 유용합니다.

그래프의 Y축을 수정하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 지표 네임스페이스(예: EC2)를 선택한 후 지표 차원(예: 인스턴스별 지표)을 선택합니다.
4. 모든 지표 탭에 네임스페이스의 해당 차원에 대한 모든 지표가 표시됩니다. 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다.
5. 그래프 옵션에서 왼쪽 Y축에 대한 최소 값과 최대 값을 지정합니다. 최소 값은 최대 값보다 클 수 없습니다.



6. 두 번째 Y축을 작성하려면 오른쪽 Y축에 대한 최소 값 및 최대 값을 지정합니다.
7. 두 Y축 간을 전환하려면 그래프로 표시된 지표 탭을 선택합니다. Y축에서 왼쪽 Y축 또는 오른쪽 Y축을 선택합니다.

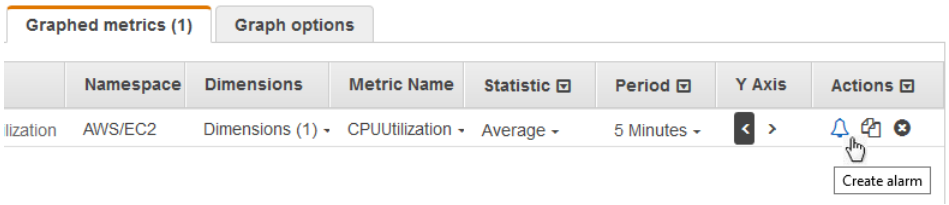


그래프의 지표에서 경보 생성

지표를 그래프 처리한 다음, 그래프의 지표에서 경보를 생성할 수 있습니다. 이 경우 다수의 경보 필드에서 값이 채워진다는 장점이 있습니다.

그래프의 지표에서 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 지표 네임스페이스(예: EC2)를 선택한 후 지표 차원(예: 인스턴스별 지표)을 선택합니다.
4. 모든 지표 탭에 네임스페이스의 해당 차원에 대한 모든 지표가 표시됩니다. 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다.
5. 지표에 대한 경보를 생성하려면 그래프로 표시된 지표 탭을 선택합니다. 작업에서 경보 아이콘을 선택합니다.



6. 조건에서 정적 또는 이상 탐지를 선택하여 경보에 대해 정적 임계값을 사용할지 아니면 이상 탐지 모델을 사용할지를 지정합니다.

   선택한 사항에 따라 경보 조건에 대한 데이터의 나머지 부분을 입력합니다.
7. 추가 구성을 선택합니다. Datapoints to alarm(경보에 대한 데이터 포인트)에서 경보를 트리거하기 위해 평가 기간(데이터 포인트)이 `ALARM` 상태로 유지해야 하는 기간을 지정합니다. 두 값이 일치하는 경우 다수의 연속 기간이 위반되면 `ALARM` 상태가 되는 경보가 생성됩니다.

   N 중 M 경보를 생성하려면 두 번째 값에 지정한 값보다 낮은 값을 첫 번째 값에 지정합니다. 자세한 내용은 경보 평가 (p. 67) 단원을 참조하십시오.
8. Missing data treatment(누락 데이터 처리)에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 내용은 CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 (p. 68) 단원을 참조하십시오.
9. 다음을 선택합니다.
10. 알림에서 경보가 `ALARM` 상태, `OK` 상태 또는 `INSUFFICIENT_DATA` 상태일 때 알릴 SNS 주제를 선택합니다.

    경보가 동일한 경보 상태 또는 다른 경보 상태에 대해 여러 개의 알림을 보내도록 설정하려면 알림 추가를 선택합니다.

    경보에서 알림을 보내지 않게 하려면 제거를 선택합니다.
11. 경보가 Auto Scaling 또는 EC2 작업을 수행하도록 하려면 해당 버튼을 선택하고 경보 상태 및 수행할 작업을 선택합니다.

12. 마친 후에는 다음을 선택합니다.
13. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다. 다음을 선택합니다.
14. Preview and create(미리 보기 및 생성)에서 정보 및 조건이 원하는 내용인지 확인한 다음 경보 생성을 선택합니다.

사용자 지정 메트릭 게시

AWS CLI 또는 API를 사용하여 CloudWatch에 자체 지표를 게시할 수 있습니다. AWS Management 콘솔을 사용하여 게시된 지표의 통계 그래프를 볼 수 있습니다.

CloudWatch에서는 지표에 대한 데이터를 일련의 데이터 요소로 저장합니다. 각 데이터 요소에는 연결된 시간스탬프가 있습니다. 통계 세트라는 집계된 데이터 요소 세트를 게시할 수도 있습니다.

주제

- 고분해능 지표 (p. 47)
- 차원 사용 (p. 47)
- 단일 데이터 요소 게시 (p. 48)
- 통계 세트 게시 (p. 49)
- 0 값 게시 (p. 49)

고분해능 지표

각 지표는 다음 중 하나입니다.

- 표준 분해능 - 1분 세분화 데이터
- 고분해능 - 1초 세분화 데이터

AWS 서비스에 의해 생성되는 지표는 기본적으로 표준 분해능입니다. 사용자 지정 지표를 게시할 때는 지표를 표준 분해능 또는 고분해능으로 정의할 수 있습니다. 고분해능 지표를 게시할 경우, CloudWatch는 1초 분해능으로 지표를 저장하고, 사용자는 1초, 5초, 10초, 30초 또는 60초의 배수 기간으로 읽고 검색할 수 있습니다.

고분해능 지표는 애플리케이션의 단기(1분 미만) 활동을 보다 즉각적으로 관찰할 수 있게 해줍니다. 사용자 지정 지표에 대해 `PutMetricData`를 호출할 때마다 요금이 부과되며, 따라서 고분해능 지표에 대해 `PutMetricData`를 자주 호출할수록 요금이 증가할 수 있다는 점에 유의하십시오. CloudWatch 요금에 대한 자세한 내용은 Amazon CloudWatch Pricing 단원을 참조하십시오.

고분해능 지표에 대해 경보를 설정할 경우 고분해능 경보를 10초 또는 30초 기간으로 지정하거나 60초의 배수 기간으로 정기 경보를 설정할 수 있습니다. 10초 또는 30초 기간의 고분해능 경보는 요금이 더 비쌉니다.

차원 사용

사용자 지정 지표에서 `--dimensions` 파라미터는 공통입니다. 차원은 지표가 무엇이고 어떤 데이터가 저장되었는지 훨씬 명확하게 보여줍니다. 지표 하나에 차원을 최대 10개까지 가질 수 있으며, 각 차원은 이름-값 페어로 정의됩니다.

서로 다른 명령을 사용하면 차원을 지정하는 방법도 달라집니다. put-metric-data에서는 각 차원을 *`MyName`*=*`MyValue`*로 지정하고, get-metric-statistics 또는 put-metric-alarm에서는 `Name=`*`MyName`*, `Value=`*`MyValue`* 형식을 사용합니다. 예를 들어, 다음 명령은 이름이 `InstanceId` 및 `InstanceType`인 두 차원으로 `Buffers` 지표를 게시합니다.

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace MyNameSpace --unit Bytes
 --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small
```

이 명령은 동일한 지표에 대한 통계를 검색합니다. 단일 차원의 Name 및 Value 부분은 쉼표로 구분하지만, 차원이 여러 개인 경우에는 하나의 차원과 그 다음 차원 사이에 공백을 둡니다.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace --
dimensions Name=InstanceId,Value=1-23456789 Name=InstanceType,Value=m1.small --start-time
 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average --period 60
```

단일 지표에 여러 개의 차원이 포함된 경우에는 get-metric-statistics를 사용할 때 정의된 모든 차원에 대해 값을 지정해야 합니다. 예를 들어 Amazon S3 지표의 경우 `BucketSizeBytes`에 `BucketName` 및 `StorageType` 차원이 포함되어 있기 때문에 get-metric-statistics에서 두 차원 모두를 지정해야 합니다.

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --start-time
 2017-01-23T14:23:00Z --end-time 2017-01-26T19:30:00Z --period 3600 --namespace
 AWS/S3 --statistics Maximum --dimensions Name=BucketName,Value=MyBucketName
 Name=StorageType,Value=StandardStorage --output table
```

지표에 정의되는 차원을 확인하려면 list-metrics 명령을 사용합니다.

단일 데이터 요소 게시

신규 또는 기존 지표에서 단일 데이터 요소를 게시하려면 하나의 값과 타임스탬프와 함께 put-metric-data 명령을 사용하십시오. 예를 들어, 다음 각 작업은 데이터 요소 하나를 게시합니다.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --value 2
 --timestamp 2016-10-20T12:00:00.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --value 4
 --timestamp 2016-10-20T12:00:01.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --value 5
 --timestamp 2016-10-20T12:00:02.000Z
```

새 지표 이름으로 이 명령을 호출하면 CloudWatch가 지표를 생성합니다. 그렇지 않으면 CloudWatch는 지정된 기존의 지표에 데이터를 연결합니다.

> **Note**
>
> 지표를 작성할 때 get-metric-statistics 명령을 사용하여 새 지표에 대한 통계를 검색할 수 있기까지 최대 2분이 소요될 수 있습니다. 그러나 list-metrics 명령을 사용하여 검색된 지표 목록에 새 지표가 나타나려면 최대 15분이 걸릴 수 있습니다.

1,000분의 1초만큼 세분화된 타임스탬프를 사용하여 데이터 요소를 게시하더라도 CloudWatch에서는 최소 1초로 세분화되도록 데이터를 집계합니다. CloudWatch는 각 기간으로 수신된 값의 평균(모든 항목의 합계를 항목 수로 나눈 값)을 비롯해 샘플 수, 동일 기간에 대한 최댓값 및 최솟값을 기록합니다. 예를 들어 이전 예의 `PageViewCount` 지표에는 타임스탬프가 몇 초 간격인 데이터 요소 3개가 들어 있습니다. 기간을 1분으로 설정하면 타임 스탬프가 모두 1분 기간 내에 있기 때문에 CloudWatch에서는 3개의 데이터 요소를 집계합니다.

get-metric-statistics 명령을 사용하여 게시된 데이터 요소에 따라 통계를 검색할 수 있습니다.

```
aws cloudwatch get-metric-statistics --namespace MyService --metric-name PageViewCount \
--statistics "Sum" "Maximum" "Minimum" "Average" "SampleCount" \
--start-time 2016-10-20T12:00:00.000Z --end-time 2016-10-20T12:05:00.000Z --period 60
```

다음은 예제 출력입니다.

```
{
    "Datapoints": [
        {
            "SampleCount": 3.0,
            "Timestamp": "2016-10-20T12:00:00Z",
            "Average": 3.6666666666666665,
            "Maximum": 5.0,
            "Minimum": 2.0,
            "Sum": 11.0,
            "Unit": "None"
        }
    ],
    "Label": "PageViewCount"
}
```

통계 세트 게시

CloudWatch에 게시하기 전에 데이터를 집계할 수 있습니다. 분당 여러 데이터 요소가 있는 경우 데이터를 집계하면 put-metric-data에 대한 호출 수가 최소화됩니다. 예를 들어, 서로 3초 내에 있는 데이터 요소 3개에 대해 put-metric-data를 여러 번 호출하는 대신 `--statistic-values` 파라미터를 사용하여 호출 한 번으로 게시한 통계 세트로 데이터를 집계할 수 있습니다.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService
 --statistic-values Sum=11,Minimum=2,Maximum=5,SampleCount=3 --
timestamp 2016-10-14T12:00:00.000Z
```

CloudWatch가 백분위수를 계산하려면 원시 데이터 요소가 필요합니다. 대신 통계 세트를 사용해 데이터를 게시하면 아래 조건 중 하나를 충족하기 전까지 이 데이터에 대한 백분위수 통계를 검색할 수 없습니다.

- 통계 세트의 `SampleCount`는 1입니다.
- 통계 세트의 `Minimum` 및 `Maximum`이 동일합니다.

0 값 게시

데이터가 훨씬 산발적이고 연결된 데이터가 없는 기간이 있으면 해당 기간에 대해 0 값(0)을 게시하거나 값을 전혀 게시하지 않도록 선택할 수 있습니다. `PutMetricData`에 대한 정기적인 호출을 통해 애플리케이션의 상태를 모니터링하는 경우 값을 게시하지 않는 대신 0을 게시하려고 할 수 있습니다. 예를 들어 애플리케이션에서 지표를 게시하는 데 실패한 경우 5분마다 이를 알리도록 CloudWatch 경보를 설정할 수 있습니다. 애플리케이션에서 연결된 데이터가 없는 기간에 대해 0을 게시하도록 하려고 합니다.

또한 데이터 요소의 총 수를 추적하거나 최소값 또는 평균과 같은 통계에 값이 0인 데이터 요소를 포함하려는 경우 0을 게시할 수도 있습니다.

지표 수식 사용

지표 수식을 사용하면 여러 CloudWatch 지표를 조회하고, 수학 표현식을 사용하여 이러한 지표에 기반한 새로운 시계열을 만들 수 있습니다. CloudWatch 콘솔에서 결과 시계열을 시각화하고 이를 대시보드에 추가할 수 있습니다. 예를 들어 AWS Lambda 지표를 사용하여 `Errors` 지표를 `Invocations` 지표로 나누어 오류 발생률을 얻을 수 있습니다. 그런 다음 결과 시계열을 CloudWatch 대시보드의 그래프에 추가합니다.

`GetMetricData` API 작업을 사용하여 지표 계산을 프로그래밍방식으로 실행할 수도 있습니다. 자세한 내용은 GetMetricData 단원을 참조하십시오.

CloudWatch 그래프에 수학 표현식 추가

CloudWatch 대시보드에서 그래프에 수학 표현식을 추가할 수 있습니다. 각 그래프의 지표와 표현식은 최대 100개까지 사용할 수 있도록 제한되므로 그래프의 지표가 99개 이하일 경우에만 수학 표현식을 추가할 수 있습니다. 이것은 그래프에 모든 지표가 표시되지 않는 경우에도 적용됩니다.

그래프에 수학 표현식을 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 그래프 또는 줄 위젯을 만들거나 편집합니다.
3. 그래프로 표시된 지표를 선택합니다.
4. Add a math expression(수학 표현식 추가)을 선택합니다. 표현식의 새 줄이 나타납니다.
5. 세부 정보 열에 수학 표현식을 입력합니다. 지표 수식 구문 및 함수 섹션의 표에는 표현식에 사용할 수 있는 함수가 나열됩니다.

   이 표현식을 위한 공식의 일부로 지표 또는 다른 표현식 결과를 사용하려면 Id 열에 표시된 값을 사용합니다(예: m1+m2 또는 e1-MIN(e1)).

   Id 값은 변경할 수 없습니다. 숫자, 문자, 밑줄이 포함될 수 있으며, 소문자로 시작해야 합니다. Id의 값을 좀 더 의미 있는 이름으로 변경하면 그래프를 더 쉽게 이해할 수 있습니다(예: m1 및 m2에서 errors 및 requests로 변경하는 경우).

   > Tip
   >
   > 수식 표현식 옆의 아래쪽 화살표를 선택하여 표현식을 만들 때 사용할 수 있는 지원되는 함수 목록을 확인합니다.

6. 표현식의 레이블 열에는 표현식으로 계산되는 사항을 설명하는 이름을 입력합니다.

   표현식의 결과가 시계열 배열인 경우, 그 각각의 시계열이 그래프에 각각의 행과 서로 다른 색상으로 표시됩니다. 그래프 바로 아래에 그래프 내 각 행의 범례가 표시됩니다. 하나의 표현식이 여러 개의 시계열을 생성하는 경우, 해당 시계열의 범례 캡션은 ***Expression-Label Metric-Label(###-### ##-###)*** 형식으로 표시됩니다. 예를 들어, 그래프에 Errors(오류)라는 레이블을 가진 지표와 Filled With 0:(0으로 채움:)이라는 레이블을 가진 FILL(METRICS(), 0) 표현식이 포함되어 있다면, 범례의 한 행은 Filled With 0: Errors(0으로 채움: 오류)가 될 것입니다. 범례에 원래의 지표 레이블만 표시하려면 ***Expression-Label(###-###)***을 비워둡니다.

   한 표현식이 그래프에 시계열 배열을 생성하면 해당 시계열 각각에 사용된 색상을 변경할 수 없습니다.
7. 원하는 표현식을 추가한 후에는 원래 지표 일부를 숨겨 그래프를 간소화할 수 있습니다. 지표 또는 표현식을 숨기려면 Id 필드 좌측의 확인란 선택을 지웁니다.

지표 수학 구문 및 함수

아래 단원에서는 지표 수식에서 사용되는 함수를 설명합니다. 모든 함수는 대문자로 작성해야 하며(예: AVG), 모든 지표와 수학 표현식의 Id 필드는 소문자로 시작해야 합니다.

수학 표현식의 최종 결과는 단일 시계열이거나 시계열 배열이어야 합니다. 일부 함수는 스칼라 수를 생성합니다. 최종적으로 하나의 시계열을 생성하는 더 큰 함수 안에서 이러한 함수를 사용할 수 있습니다. 예를 들어, 단일 시계열에서 AVG를 빼면 스칼라 수가 생성되므로 최종 표현식 결과가 될 수 없습니다. 그렇지만 이것을 함수 m1-AVG(m1)에서 사용하여 각 개별 데이터 요소와 해당 데이터 요소의 평균값 차이인 시계열을 표시할 수 있습니다.

데이터 형식 약어

일부 함수는 특정 형식의 데이터에만 유효합니다. 다음 목록에 나와 있는 약어는 각 함수에 지원되는 데이터 형식을 나타내는 함수 표에서 사용됩니다.

- S는 2, -5 또는 50.25 같은 스칼라 수를 나타냅니다.
- TS는 시계열(시간 경과에 따른 일련의 단일 CloudWatch 지표 값)입니다. 예를 들어, 지난 3일 동안 `i-1234567890abcdef0` 인스턴스에 대한 `CPUUtilization` 지표입니다.
- TS[]는 시계열 배열입니다(예: 여러 지표에 대한 시계열).

METRICS() 함수

METRICS() 함수는 요청에 모든 지표를 반환합니다. 수학 표현식은 포함되지 않습니다.

단일 시계열이나 시계열 배열을 생성하는 더 큰 표현식 안에 METRICS() 를 사용할 수 있습니다. 예를 들어, 표현식 SUM(METRICS())은 모든 그래프 지표 값의 합인 시계열(TS)을 반환합니다. METRICS()/100 은 시계열 배열을 반환하며 그 각각은 지표 중 하나의 각 데이터 요소를 100으로 나눈 값을 표시하는 시계열입니다.

METRICS() 함수를 문자열과 함께 사용하여 그 Id 필드에 해당 문자열이 있는 그래프 지표만 반환할 수 있습니다. 예를 들어, 표현식 SUM(METRICS("errors"))은 그 Id 필드에 '오류'가 있는 모든 그래프 지표 값의 합인 시계열을 반환합니다. SUM([METRICS(“4xx”), METRICS(“5xx”)])을 사용하여 여러 문자열을 일치시킬 수도 있습니다.

기본 산술 함수

다음 표에는 지원되는 기본 산술 함수가 나와 있습니다. 시계열의 누락 값은 0으로 처리됩니다. 데이터 요소의 값 때문에 함수에서 0으로 나누려고 시도할 경우 해당 데이터 요소가 누락됩니다.

연산	인수	예제
산술 연산자: + - * / ^	S, S S, TS TS, TS S, TS[] TS, TS[]	PERIOD(m1)/60 5 * m1 m1 - m2 SUM(100/[m1, m2]) AVG([m1,m2]/m3) METRICS()*100
빼기 기호 -	S TS TS[]	-5*m1 -m1 SUM(-[m1, m2])

비교 및 논리 연산자

비교 및 논리 연산자를 시계열 쌍이나 단일 스칼라 값 쌍과 함께 사용할 수 있습니다. 비교 연산자를 시계열 쌍과 함께 사용하면 연산자는 각 데이터 요소가 0(false) 또는 1(true)인 시계열을 반환합니다. 스칼라 값 쌍에 비교 연산자를 사용하면 0 또는 1 중 하나의 스칼라 값이 반환됩니다.

비교 연산자가 두 시계열 사이에 사용되고 시계열 중 하나에만 특정 타임스탬프에 대한 값이 있는 경우, 이 함수는 다른 시계열의 누락 값을 0으로 처리합니다.

논리 연산자를 비교 연산자와 함께 사용하여 보다 복잡한 함수를 만들 수 있습니다.

다음 표에는 지원되는 연산자가 나와 있습니다.

연산자 유형	지원되는 연산자
비교 연산자	== != <= >= < >
논리 연산자	AND 및 && OR 또는 ||

이 연산자가 사용되는 방법을 보기 위해 두 개의 시계열이 있다고 가정해 보겠습니다. metric1에는 `[30, 20, 0, 0]`의 값이 있고 metric2에는 `[20, -, 20, -]`의 값이 있습니다. 여기서 `-`는 해당 타임스탬프에 대한 값이 없음을 나타냅니다.

Expression	출력
(metric1 < metric2)	0, 0, 1, 0
(metric1 >= 30)	1, 0, 0, 0
(metric1 > 15 AND metric2 > 15)	1, 0, 0, 0

지표 수식에 지원되는 함수

다음 표에서는 수학 표현식에서 사용할 수 있는 함수를 설명합니다. 모든 함수를 대문자로 입력합니다.

수학 표현식의 최종 결과는 단일 시계열이거나 시계열 배열이어야 합니다. 아래 단원에 나오는 표의 일부 함수는 스칼라 수를 생성합니다. 최종적으로 하나의 시계열을 생성하는 더 큰 함수 안에서 이러한 함수를 사용할 수 있습니다. 예를 들어, 단일 시계열에서 AVG를 빼면 스칼라 수가 생성되므로 최종 표현식 결과가 될 수 없습니다. 그렇지만 이것을 함수 m1-AVG(m1)에서 사용하여 각 개별 데이터 요소와 해당 데이터 요소의 평균값 차이인 시계열을 표시할 수 있습니다.

아래의 표에서 예제 열의 모든 예제는 단일 시계열이나 시계열 배열을 생성하는 표현식입니다. 스칼라 수를 반환하는 함수를 단일 시계열을 생성하는 유효한 표현식의 일부로 사용하는 방법을 보여주는 예제입니다.

함수	인수	반환 유형*	설명	예제
ABS	TS TS[]	TS TS[]	각 데이터 요소의 절대값을 반환합니다.	ABS(m1-m2) MIN(ABS([m1, m2])) ABS(METRICS())
ANOMALY_DETECTION_BAND	TS TS, S	TS[]	지정된 지표에 대한 이상 탐지 밴드를 반환합니다. 밴드는 두 시계열로 구성되어 있으며 하나는 지표의 "정상" 기댓값에	ANOMALY_DETECTION_BAND(m1) ANOMALY_DETECTION_BAND(m1,4)

함수	인수	반환 유형*	설명	예제
			대한 상한 치수를 나타내며 다른 하나는 하한 치수를 나타냅니다. 함수는 두 개의 인수가 필요합니다. 첫 번째는 밴드를 생성할 지표의 ID입니다. 두 번째 인수는 밴드에 사용할 표준 편차의 수입니다. 이 인수를 지정하지 않으면, 두 개의 기본값이 사용됩니다. 자세한 내용은 CloudWatch 이상 탐지 사용 (p. 123) 단원을 참조하십시오.	
AVG	TS TS[]	S TS	단일 시계열의 AVG는 지표의 모든 데이터 요소 평균을 나타내는 스칼라를 반환합니다. 시계열 배열의 AVG는 단일 시계열을 반환합니다. 누락된 값은 0로 처리됩니다.	SUM([m1,m2])/AVG(m2) AVG(METRICS())
CEIL	TS TS[]	TS TS[]	각 지표의 상한을 반환합니다. 상한은 각 값보다 크거나 같은 가장 작은 정수입니다.	CEIL(m1) CEIL(METRICS()) SUM(CEIL(METRICS()))
FILL	TS, TS/S TS[], TS/S	TS TS[]	지표 값이 부족할 때 지정된 필터 값으로 지표의 누락 값을 채웁니다.	FILL(m1,10) FILL(METRICS(), 0) FILL(m1, MIN(m1))
FIRST LAST	TS[]	TS	시계열 배열에서 첫 번째 시계열 또는 마지막 시계열을 반환합니다. SORT 함수와 함께 사용할 때 유용합니다. 또한 ANOMALY_DETECTION_BAND 함수에서 높은 임계값과 낮은 임계값을 가져오는 데도 사용할 수 있습니다.	IF(FIRST(SORT(METRICS(), AVG, DESC))>100, 1, 0) 배열에서 AVG별로 정렬된 상위 지표를 확인합니다. 그런 다음 해당 데이터 요소 값이 100 이상인지 여부에 따라 각 데이터 요소에 대해 1 또는 0을 반환합니다. LAST(ANOMALY_DETECTION_BAND(m1)) 은 이상 예측 밴드의 하한을 반환합니다.
FLOOR	TS TS[]	TS TS[]	각 지표의 하한을 반환합니다. 하한은 각 값보다 작거나 같은 가장 큰 정수입니다.	FLOOR(m1) FLOOR(METRICS())
IF	IF 표현식	TS	IF를 비교 연산자와 함께 사용하여 시계열에서 데이터 요소를 필터링하거나 수집된 여러 시계열로 구성된 혼합 시계열을 생성합니다. 자세한 내용은 IF 표현식 사용 (p. 57) 단원을 참조하십시오.	

함수	인수	반환 유형*	설명	예제
INSIGHT_RULE_METRIC	INSIGHT_RULE_METRIC(ruleName, metricName)	TS	INSIGHT_RULE_METRIC을 사용하여 Contributor Insights의 규칙에서 통계를 추출합니다. 자세한 내용은 규칙에 의해 생성된 지표 그래프 작성 (p. 133) 단원을 참조하십시오.	
MAX	TS TS[]	S TS	단일 시계열의 MAX는 지표의 모든 데이터 요소의 최대값을 나타내는 스칼라를 반환합니다. 시계열 배열의 MAX 값은 단일 시계열을 반환합니다.	MAX(m1)/m1 MAX(METRICS())
METRIC_COUNT	TS[]	S	시계열 어레이의 지표 수를 반환합니다.	m1/METRIC_COUNT(METRICS())
METRICS()	null string	TS[]	METRICS() 함수는 요청에 모든 CloudWatch 지표를 반환합니다. 수학 표현식은 포함되지 않습니다. 단일 시계열이나 시계열 배열을 생성하는 더 큰 표현식 안에 METRICS() 를 사용할 수 있습니다. METRICS() 함수를 문자열과 함께 사용하여 그 Id 필드에 해당 문자열이 있는 그래프 지표만 반환할 수 있습니다. 예를 들어, 표현식 SUM(METRICS("errors"))은 그 Id 필드에 '오류'가 있는 모든 그래프 지표 값의 합인 시계열을 반환합니다. SUM([METRICS(“4xx”), METRICS(“5xx”)])을 사용하여 여러 문자열을 일치시킬 수도 있습니다.	AVG(METRICS()) SUM(METRICS("errors"))
MIN	TS TS[]	S TS	단일 시계열의 MIN은 지표의 모든 데이터 요소의 최소값을 나타내는 스칼라를 반환합니다. 시계열 배열의 MIN 은 단일 시계열을 반환합니다.	m1-MIN(m1) MIN(METRICS())
PERIOD	TS	S	지표의 기간(초)을 반환합니다. 유효한 입력은 지표이지 다른 표현식의 결과가 아닙니다.	m1/PERIOD(m1)

함수	인수	반환 유형*	설명	예제
속도	TS TS[]	TS TS[]	지표의 초당 변경 비율을 반환합니다. 이것은 마지막 데이터 요소 값과 그 이전의 데이터 요소 값의 차이를 두 값의 시간차(초)로 나눈 값으로 계산됩니다.	RATE(m1) RATE(METRICS())
REMOVE_EMPTY	TS[]	TS[]	시계열 배열에서 데이터 포인터가 없는 시계열을 제거합니다. 결과는 각 시계열에 적어도 하나의 데이터 포인터가 포함된 시계열 배열입니다.	REMOVE_EMPTY(METRICS())
SEARCH	검색 표현식	하나 이상의 TS	지정한 검색 기준과 일치하는 시계열을 하나 이상 반환합니다. SEARCH 함수를 사용하면 표현식 하나로 그래프에 관련된 시계열을 여러 개 추가할 수 있습니다. 나중에 추가되고 검색 기준과 일치하는 새 지표를 포함하도록 그래프가 동적으로 업데이트됩니다. 자세한 내용은 그래프에서 검색 표현식 사용 (p. 59) 단원을 참조하십시오.	
SERVICE_QUOTA	사용량 지표인 TS	TS	지정된 사용량 지표에 대한 서비스 할당량을 반환합니다. 이 함수를 사용하여 현재 사용량을 할당량과 비교하는 방법을 시각화하고 할당량에 접근할 때 경고하는 경보를 설정할 수 있습니다. 자세한 내용은 서비스 할당량 통합 및 사용량 지표 (p. 348) 단원을 참조하십시오.	

함수	인수	반환 유형*	설명	예제
SLICE	(TS[], S, S) 또는 (TS[], S)	TS[] TS	시계열 배열의 일부를 검색합니다. SORT와 결합할 때 특히 유용합니다. 예를 들어 시계열 배열에서 상위 결과를 제외할 수 있습니다. 2개의 스칼라 인수를 사용하여 반환하려는 시계열 집합을 정의할 수 있습니다. 2개의 스칼라는 반환할 배열의 시작(포함)과 끝(제외)을 정의합니다. 배열은 0을 기반으로 인덱싱되므로 배열의 첫 번째 시계열은 시계열 0입니다. 또는 값을 하나만 지정할 수 있고, CloudWatch는 해당 값으로 시작하는 모든 시계열을 반환합니다.	SLICE(SORT(METRICS(), SUM, DESC), 0, 10)는 요청의 지표 배열에서 SUM 값이 가장 높은 10개의 지표를 반환합니다. SLICE(SORT(METRICS(), AVG, ASC), 5)는 AVG 통계별로 지표 배열을 정렬한 다음 AVG가 가장 낮은 5를 제외한 모든 시계열을 반환합니다.
정렬	(TS[], FUNCTION, SORT_ORDER) (TS[], FUNCTION, SORT_ORDER, S)	TS[]	지정한 함수에 따라 시계열 배열을 정렬합니다. 사용하는 함수는 AVG, MIN, MAX 또는 SUM일 수 있습니다. 정렬 순서는 가장 낮은 값을 먼저 정렬하는 오름차순의 ASC 또는 더 높은 값을 먼저 정렬하는 DESC일 수 있습니다. 선택적으로 정렬 순서 뒤에 제한 역할을 하는 숫자를 지정할 수 있습니다. 예를 들어 제한을 5로 지정하면 정렬에서 상위 5개의 시계열만 반환됩니다. 이 수학 함수가 그래프에 표시되면 그래프의 각 지표에 대한 레이블도 정렬되고 번호가 매겨집니다.	SORT(METRICS(), AVG, DESC, 10)은 각 시계열의 평균값을 계산하고, 정렬의 시작 부분에서 가장 높은 값을 가진 시계열을 정렬하고, 평균이 가장 높은 10개의 시계열만 반환합니다. SORT(METRICS(), MAX, ASC)는 MAX 통계별로 지표 배열을 정렬한 다음 모두 오름차순으로 반환합니다.
STDDEV	TS TS[]	S TS	단일 시계열의 STDDEV는 지표의 모든 데이터 요소의 표준편차를 나타내는 스칼라를 반환합니다. 시계열 배열의 STDDEV 는 단일 시계열을 반환합니다.	m1/STDDEV(m1) STDDEV(METRICS())
SUM	TS TS[]	S TS	단일 시계열의 SUM은 지표의 모든 데이터 요소 값의 합을 나타내는 스칼라를 반환합니다. 시계열 배열의 SUM 은 단일 시계열을 반환합니다.	SUM(METRICS())/SUM(m1) SUM([m1,m2]) SUM(METRICS("errors"))/ SUM(METRICS("requests"))*100

*스칼라 수만 반환하는 함수만 사용하는 것은 유효하지 않습니다. 표현식의 모든 최종 결과가 단일 시계열 또는 시계열 배열이어야 하기 때문입니다. 이러한 함수는 시계열을 반환하는 더 큰 표현식의 일부로 사용하십시오.

IF 표현식 사용

IF를 비교 연산자와 함께 사용하여 시계열에서 데이터 요소를 필터링하거나 수집된 여러 시계열로 구성된 혼합 시계열을 생성합니다.

IF는 다음 인수를 사용합니다.

```
IF(condition, trueValue, falseValue)
```

조건은 조건 데이터 요소의 값이 0이면 FALSE로, 조건 값이 양수인지 음수인지 여부에 관계없이 다른 값이면 TRUE로 평가됩니다. 조건이 시계열이면 모든 타임스탬프에 대해 개별적으로 평가됩니다.

다음은 유효한 구문 목록입니다. 이러한 구문 각각에 대한 출력은 단일 시계열입니다.

- IF(TS ***Comparison Operator*** S, S | TS, S | TS)
- IF(TS, TS, TS)
- IF(TS, S, TS)
- IF(TS, TS, S)
- IF(TS, S, S)
- IF(S, TS, TS)

다음 단원에서는 이러한 구문에 대한 자세한 내용과 예제를 제공합니다.

IF(TS ***Comparison Operator*** S, scalar2 | metric2, scalar3 | metric3)

해당 출력 시계열 값:

- TS ***Comparison Operator*** S가 TRUE인 경우 값이 scalar2 또는 metric2입니다.
- TS ***Comparison Operator*** S가 FALSE인 경우 값이 scalar3 또는 metric3입니다.
- 해당 데이터 요소가 metric3에 없거나 scalar3 /metric3이 이 표현식에서 생략된 경우 빈 시계열입니다.

IF(metric1, metric2, metric3)

metric1의 각 데이터 요소의 경우, 해당 출력 시계열 값:

- metric1의 해당 데이터 요소가 TRUE인 경우 값이 metric2입니다.
- metric1의 해당 데이터 요소가 FALSE인 경우 값이 metric3입니다.
- metric1의 해당 데이터 요소가 TRUE이고 해당 데이터 요소가 metric2에 없는 경우 값이 0입니다.
- metric1의 해당 데이터 요소가 FALSE이고 해당 데이터 요소가 metric3에 없는 경우 또는 metric3이 표현식에서 생략된 경우 삭제됩니다.

다음 표에서는 이 구문의 예제를 보여줍니다.

지표 또는 함수	Values
(metric1)	[1, 1, 0, 0, -]

지표 또는 함수	Values
(metric2)	[30, -, 0, 0, 30]
(metric3)	[0, 0, 20, -, 20]
IF(metric1, metric2, metric3)	[30, 0, 20, -, -]

IF(metric1, scalar2, metric3)

metric1의 각 데이터 요소의 경우, 해당 출력 시계열 값:

- metric1의 해당 데이터 요소가 TRUE인 경우 값이 scalar2입니다.
- metric1의 해당 데이터 요소가 FALSE인 경우 값이 metric3입니다.
- metric1의 해당 데이터 요소가 FALSE이고 해당 데이터 요소가 metric3에 없는 경우 또는 metric3이 표현력에서 생략된 경우 삭제됩니다.

지표 또는 함수	Values
(metric1)	[1, 1, 0, 0, -]
scalar2	5
(metric3)	[0, 0, 20, -, 20]
IF(metric1, scalar2, metric3)	[5, 5, 20, -, -]

IF(metric1, metric2, scalar3)

metric1의 각 데이터 요소의 경우, 해당 출력 시계열 값:

- metric1의 해당 데이터 요소가 TRUE인 경우 값이 metric2입니다.
- metric1의 해당 데이터 요소가 FALSE인 경우 값이 scalar3입니다.
- metric1의 해당 데이터 요소가 TRUE이고 해당 데이터 요소가 metric2에 없는 경우 값이 0입니다.

지표 또는 함수	Values
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
scalar3	5
IF(metric1, metric2, scalar3)	[30, 0, 5, 5, 5]

IF(scalar1, metric2, metric3)

해당 출력 시계열 값:

- scalar1이 TRUE인 경우 값이 metric2입니다.
- scalar1이 FALSE인 경우 값이 metric3입니다.
- metric3이 표현식에서 생략된 경우 빈 시계열입니다.

IF 표현식에 대한 사용 사례 예제

다음 예제에서는 IF 함수의 가능한 용도를 보여줍니다.

- 지표의 하한값만 표시하려면:

  IF(metric1<400, metric1)
- 지표의 각 데이터 요소를 두 값 중 하나로 변경하고 원래 지표의 상대적 상한값 및 하한값을 표시하려면:

  IF(metric1<400, 10, 2)
- 지연 시간이 임계값을 초과하는 각 타임 스탬프에 대해 1을 표시하고 다른 모든 데이터 요소에 대해 0을 표시하려면:

  IF(latency>threshold, 1, 0)

GetMetricData API 작업과 함께 지표 수식 사용

`GetMetricData`를 사용하여 수학 표현식을 사용하는 계산을 수행할 수 있을 뿐만 아니라 하나의 API 호출에서 대규모 지표 데이터 배치를 검색할 수도 있습니다. 자세한 내용은 GetMetricData 단원을 참조하십시오.

그래프에서 검색 표현식 사용

검색 표현식은 CloudWatch 그래프에 추가할 수 있는 수학 표현식의 유형입니다. 검색 표현식을 사용하면 관련된 여러 개의 지표를 그래프에 빠르게 추가할 수 있습니다. 또한 처음 그래프를 생성할 때 지표가 없는 경우에도 적절한 지표를 표시에 자동으로 추가하는 동적 그래프를 생성할 수 있습니다.

예를 들어, 리전의 모든 인스턴스에 대해 `AWS/EC2 CPUUtilization` 지표를 표시하는 검색 표현식을 생성할 수 있습니다. 나중에 새 인스턴스를 시작하면 새 인스턴스의 `CPUUtilization`이 그래프에 자동으로 추가됩니다.

그래프에 검색 표현식을 사용하면 검색은 지표 이름, 네임스페이스, 차원 이름 및 차원 값에서 검색 표현식을 찾습니다. 더 복잡하고 강력한 검색을 위해 부울 연산자를 사용할 수 있습니다.

주제

- CloudWatch 검색 표현식 구문 (p. 59)
- CloudWatch 검색 표현식 예제 (p. 63)
- 검색 표현식이 있는 CloudWatch 그래프 생성 (p. 65)

CloudWatch 검색 표현식 구문

유효한 검색 표현식의 형식은 다음과 같습니다.

```
SEARCH(' {Namespace, DimensionName1, DimensionName2, ...} SearchTerm', 'Statistic', Period)
```

예:

```
SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average', 300)
```

- `SEARCH` 단어 뒤에 중괄호로 묶여 있는 쿼리의 첫 번째 부분은 검색할 지표 스키마입니다. 지표 스키마에는 하나의 지표 네임스페이스와 하나 이상의 차원 이름이 포함됩니다. 검색 쿼리에 지표 스키마를 포함하는 것은 선택 사항입니다. 지정된 경우 지표 스키마에는 네임스페이스가 있어야 하며, 해당 네임스페이스에서 유효한 하나 이상의 차원 이름을 선택적으로 포함할 수 있습니다.

네임스페이스 또는 차원 이름에 공백이나 영숫자가 아닌 문자가 포함되지 않는 한 지표 스키마 내에 따옴표를 사용할 필요가 없습니다. 따옴표를 사용해야 하는 경우 해당 문자가 포함된 이름을 큰따옴표로 묶어야 합니다.

- `SearchTerm`도 선택 사항이지만, 유효한 검색에는 지표 스키마, `SearchTerm` 또는 둘 다 포함되어야 합니다. 일반적으로 `SearchTerm`에는 하나 이상의 지표 이름이나 차원 값이 포함되어 있습니다. `SearchTerm`에는 부분 일치 및 정확한 일치로 검색할 용어가 여러 개 포함될 수 있습니다. 또한 부울 연산자도 포함될 수 있습니다.

  `SearchTerm`에는 하나 이상의 지정자가 포함될 수 있지만(이 예에서는 `MetricName=`), 지정자를 사용하는 것은 필수 사항이 아닙니다.

  지표 스키마 및 `SearchTerm`은 작은따옴표 쌍으로 함께 묶여야 합니다.
- `Statistic`은 유효한 CloudWatch 통계의 이름입니다. 이는 작은따옴표로 묶여야 합니다. 자세한 내용은 통계 (p. 5) 단원을 참조하십시오.
- `Period`는 집계 기간(초)입니다.

위의 예제에서는 차원 이름이 `InstanceId`인 지표에 대한 `AWS/EC2` 네임스페이스를 검색합니다. 집계 기간이 5분인 `Average` 통계를 보여주는 그래프와 함께 찾은 모든 `CPUUtilization` 지표를 반환합니다.

검색 표현식 제한

검색 표현식 쿼리의 최대 크기는 1024자입니다. 그래프 하나에는 최대 5개의 검색 표현식이 있을 수 있습니다. 그래프에는 최대 100개의 시계열이 표시될 수 있습니다.

CloudWatch 검색 표현식: 토큰화

`SearchTerm`을 지정하면 검색 기능은 CloudWatch가 전체 지표 이름, 차원 이름, 차원 값 및 네임스페이스에서 자동으로 생성하는 하위 문자열인 토큰을 검색합니다. CloudWatch는 원래 문자열에서 낙타 대문자로 구별된 토큰을 생성합니다. 또한 숫자 문자는 새 토큰의 시작 부분으로 사용되며, 영숫자가 아닌 문자는 구분 기호로 사용되며, 영숫자가 아닌 문자 앞/뒤에는 토큰을 생성합니다.

동일한 유형의 토큰 구분 기호로 이루어진 연속 문자열이 한 개의 토큰이 됩니다.

생성된 토큰은 모두 소문자입니다. 다음 표에는 생성된 토큰의 일부 예가 나와 있습니다.

원래 문자열	생성된 토큰
CustomCount1	`customcount1`, `custom`, `count`, `1`
SDBFailure	`sdbfailure`, `sdb`, `failure`
Project2-trial333	`project2trial333`, `project`, `2`, `trial`, `333`

CloudWatch 검색 표현식: 부분 일치

`SearchTerm`을 지정하면 검색어도 토큰화됩니다. CloudWatch는 검색어에서 생성된 단일 토큰이 지표 이름, 네임스페이스, 차원 이름 또는 차원 값에서 생성된 단일 토큰과 일치하는 부분 일치에 따라 지표를 찾습니다.

단일 토큰과 일치하는 부분 일치 검색은 대/소문자를 구별하지 않습니다. 예를 들어, 다음 검색어를 사용하면 `CustomCount1` 지표가 반환될 수 있습니다.

- `count`
- `Count`

- COUNT

그러나 couNT를 검색어로 사용하면 검색어 couNT의 대문자가 cou 및 NT로 토큰화되므로 CustomCount1을 찾지 않습니다.

검색 시 원래 이름에 연속으로 나타나는 여러 토큰인 복합 토큰과도 일치할 수 있습니다. 복합 토큰과 일치시키기 위해 검색 시 대/소문자를 구별합니다. 예를 들어, 원래 용어가 CustomCount1인 경우 CustomCount 또는 Count1에 대한 검색은 성공하지만 customcount 또는 count1에 대한 검색은 실패합니다.

CloudWatch 검색 표현식: 정확한 일치

정확하게 일치해야 하는 검색어 부분을 큰따옴표로 묶어 검색어와 정확하게 일치하는 용어만 찾도록 검색을 정의할 수 있습니다. 이러한 큰따옴표는 전체 검색어에 주위에 사용되는 작은따옴표로 묶여 있습니다. 예를 들어, **SEARCH(' {MyNamespace}, "CustomCount1" ', 'Maximum', 120)**는 이름이 MyNamespace인 네임스페이스에 지표 이름, 차원 이름 또는 차원 값으로 있는 경우 정확한 문자열 CustomCount1을 찾습니다. 그러나 **SEARCH(' {MyNamespace}, "customcount1" ', 'Maximum', 120)** 또는 **SEARCH(' {MyNamespace}, "Custom" ', 'Maximum', 120)** 검색은 이 문자열을 찾지 않습니다.

단일 검색 표현식에서 부분 일치 용어와 정확한 일치 용어를 결합할 수 있습니다. 예를 들어, **SEARCH(' {AWS/NetworkELB, LoadBalancer}, "ConsumedLCUs" OR flow ', 'Maximum', 120)**는 이름이 ConsumedLCUs인 Elastic Load Balancing 지표를 반환하며 flow 토큰이 포함된 Elastic Load Balancing 지표 또는 차원도 모두 반환합니다.

또한 다음 예제와 같이 영숫자가 아닌 문자 또는 공백 같은 특수 문자가 있는 이름을 찾을 때 정확한 일치를 사용하는 것이 좋습니다.

```
SEARCH(' {"My Namespace", "Dimension@Name"}, "Custom:Name[Special_Characters" ', 'Maximum',
 120)
```

CloudWatch 검색 표현식: 지표 스키마 제외

지금까지 살펴본 모든 예제의 지표 스키마는 중괄호 안에 포함되어 있습니다. 지표 스키마를 생략하는 검색도 유효합니다.

예를 들어, **SEARCH(' "CPUUtilization" ', 'Average', 300)**는 CPUUtilization 문자열과 정확하게 일치하는 모든 지표 이름, 차원 이름, 차원 값 및 네임스페이스를 반환합니다. AWS 지표 네임스페이스에서는 Amazon EC2, Amazon ECS Amazon SageMaker 및 기타 서비스를 비롯한 여러 서비스의 지표가 포함될 수 있습니다.

이 검색 범위를 AWS 서비스 하나만으로 좁히려면 다음 예제와 같이 지표 스키마에 네임스페이스와 필요한 차원을 지정하는 것이 좋습니다. 이렇게 하면 검색 범위가 AWS/EC2 네임스페이스로 좁혀지지만 CPUUtilization을 해당 지표의 차원 값으로 정의한 경우 다른 지표의 결과가 계속 반환됩니다.

```
SEARCH(' {AWS/EC2, InstanceType} "CPUUtilization" ', 'Average', 300)
```

또는 다음 예제와 같이 SearchTerm에 네임스페이스를 추가할 수 있습니다. 그러나 이 예제에서는 검색 시 사용자 지정 차원 이름 또는 값인 경우에도 AWS/EC2 문자열과 일치합니다.

```
SEARCH(' "AWS/EC2" MetricName="CPUUtilization" ', 'Average', 300)
```

CloudWatch 검색 표현식: 검색에서 속성 이름 지정

"CustomCount1"에 대해 다음의 정확한 일치 검색을 수행하면 해당 이름과 정확하게 일치하는 모든 지표를 반환합니다.

```
SEARCH(' "CustomCount1" ', 'Maximum', 120)
```

그러나 CustomCount1의 차원 이름, 차원 값 또는 네임스페이스가 있는 지표도 반환합니다. 검색을 자세히 구성하려면 검색에서 찾으려는 객체 유형의 속성 이름을 지정하면 됩니다. 다음 예제는 모든 네임스페이스를 검색하고 이름이 CustomCount1인 지표를 반환합니다.

```
SEARCH(' MetricName="CustomCount1" ', 'Maximum', 120)
```

다음 예제와 같이 네임스페이스와 차원 이름/값 페어를 속성 이름으로 사용할 수도 있습니다. 이러한 예제의 첫 번째는 부분 일치 검색에도 속성 이름을 사용할 수 있음을 보여줍니다.

```
SEARCH(' InstanceType=micro ', 'Average', 300)
```

```
SEARCH(' InstanceType="t2.micro" Namespace="AWS/EC2" ', 'Average', 300)
```

CloudWatch 검색 표현식: 영숫자가 아닌 문자

영숫자가 아닌 문자는 구분 기호로 사용되며, 지표, 차원, 네임스페이스 및 검색어의 이름을 토큰으로 구분할 위치를 표시합니다. 용어가 토큰화되면 영숫자가 아닌 문자가 제거되며 토큰에 표시되지 않습니다. 예를 들어, Network-Errors_2는 network, errors 및 2 토큰을 생성합니다.

검색어에 영숫자가 아닌 문자가 포함될 수 있습니다. 검색어에 이러한 문자가 표시되면 해당 문자는 부분 일치에서 복합 토큰을 지정할 수 있습니다. 예를 들어, 다음 검색에서는 이름이 Network-Errors-2 또는 NetworkErrors2인 지표를 모두 찾습니다.

```
network/errors
network+errors
network-errors
Network_Errors
```

정확한 값 검색을 수행할 경우 정확한 검색에 사용된 영숫자가 아닌 문자는 검색할 문자열에 표시되는 정확한 문자여야 합니다. 예를 들어, Network-Errors-2를 찾으려는 경우 "Network-Errors-2"에 대한 검색은 성공하지만 "Network_Errors_2"에 대한 검색은 실패합니다.

정확한 일치 검색을 수행할 경우 다음 문자는 백슬래시로 이스케이프되어야 합니다.

```
" \ ( )
```

예를 들어, 정확한 일치로 이름이 Europe\France Traffic(Network)인 지표를 찾으려면 **"Europe\\France Traffic\(Network\)"** 검색어를 사용해야 합니다.

CloudWatch 검색 표현식: 부울 연산자

검색 시 SearchTerm 내에 부울 연산자 AND, OR 및 NOT을 사용할 수 있습니다. 부울 연산자는 전체 검색어를 묶을 때 사용하는 작은따옴표로 묶습니다. 부울 연산자는 대/소문자를 구별하므로 and, or 및 not은 부울 연산자로 사용할 수 없습니다.

SEARCH('{AWS/EC2,InstanceId} network AND packets', 'Average', 300) 같이 검색에서 AND를 명시적으로 사용할 수 있습니다. 검색어 사이에 부울 연산자를 사용하지 않으면 암시적으로 AND 연산자가 있는 것처럼 검색합니다. 따라서 **SEARCH(' {AWS/EC2,InstanceId} network packets ', 'Average', 300)**의 경우 동일한 검색 결과가 발생합니다.

결과에서 데이터의 하위 집합을 제외하려면 NOT을 사용합니다. 예를 들어, **SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT i-1234567890123456**

', 'Average', 300)는 i-1234567890123456 인스턴스를 제외한 모든 인스턴스에 대해 CPUUtilization을 반환합니다. NOT 절을 유일한 검색어로 사용할 수도 있습니다. 예를 들어, **SEARCH('NOT Namespace=AWS ', 'Maximum', 120)**는 사용자 지정 지표(AWS가 포함되지 않은 네임스페이스가 있는 지표)를 모두 생성합니다.

쿼리에 여러 NOT 구문을 사용할 수 있습니다. 예를 들어, **SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT "ProjectA" NOT "ProjectB" ', 'Average', 300)**는 리전에서 ProjectA 또는 ProjectB의 차원 값이 있는 것을 제외한 모든 인스턴스의 CPUUtilization을 반환합니다.

다음 예제와 같이 더욱 강력하고 상세한 검색을 위해 부울 연산자를 결합할 수 있습니다. 연산자를 그룹화하려면 괄호를 사용합니다.

다음의 두 예제는 EC2 및 EBS 네임스페이스 둘 다의 ReadOps이 포함된 지표 이름을 모두 반환합니다.

```
SEARCH(' (EC2 OR EBS) AND MetricName=ReadOps ', 'Maximum', 120)
```

```
SEARCH(' (EC2 OR EBS) MetricName=ReadOps ', 'Maximum', 120)
```

다음 예제는 이전 검색을 ProjectA(차원 값일 수 있음)만 포함된 결과로 범위를 좁힙니다.

```
SEARCH(' (EC2 OR EBS) AND ReadOps AND ProjectA ', 'Maximum', 120)
```

다음 예제는 중첩 그룹화를 사용합니다. 여기에서는 모든 함수의 Errors 및 이름에 ProjectA 또는 ProjectB 문자열이 포함된 함수의 Invocations에 대한 Lambda 지표를 반환합니다.

```
SEARCH(' {AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average', 600)
```

CloudWatch 검색 표현식: 수학 표현식 사용

그래프의 수학 표현식 내에서 검색 표현식을 사용할 수 있습니다.

예를 들어, **SUM(SEARCH(' {AWS/Lambda, FunctionName}, MetricName="Errors" ', 'Sum', 300))**은 모든 Lambda 함수의 Errors 지표에 대한 합계를 반환합니다.

검색 표현식과 수학 표현식에 별도의 줄을 사용하면 더 유용한 결과를 얻을 수 있습니다. 예를 들어, 그래프에서 다음의 두 표현식을 사용한다고 가정해 보겠습니다. 첫 번째 줄은 각 Lambda 함수에 대한 별도의 Errors 줄을 표시합니다. 이 표현식의 ID는 e1입니다. 두 번째 줄은 모든 함수의 오류 합계를 표시하는 다른 줄을 추가합니다.

```
SEARCH(' {AWS/Lambda, FunctionName}, MetricName="Errors" ', 'Sum', 300)
SUM(e1)
```

CloudWatch 검색 표현식 예제

다음 예제는 더 많은 검색 표현식 사용과 구문을 보여줍니다. 리전의 모든 인스턴스에 대한 CPUUtilization을 검색한 후 변형을 살펴보겠습니다.

이 예제는 리전의 각 인스턴스에 대한 줄 하나를 표시합니다. 이 줄은 AWS/EC2 네임스페이스의 CPUUtilization 지표를 보여줍니다.

```
SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average', 300)
```

InstanceId를 InstanceType으로 변경하면 리전에서 사용되는 각 인스턴스 유형마다 한 줄씩 표시되도록 그래프가 변경됩니다. 각 유형의 모든 인스턴스에 있는 데이터는 해당 인스턴스 유형에 대해 한 줄로 집계됩니다.

```
SEARCH(' {AWS/EC2,InstanceType} MetricName="CPUUtilization" ', 'Average', 300)
```

다음 예제와 같이 스키마에서 차원 이름은 제거하지만 네임스페이스는 유지하는 경우 리전의 모든 인스턴스에 대한 CPUUtilization 지표의 집계를 표시하는 단일 줄이 나타납니다.

```
SEARCH(' {AWS/EC2} MetricName="CPUUtilization" ', 'Average', 300)
```

다음 예제는 인스턴스 유형별로 CPUUtilization을 집계하고 micro 문자열이 포함된 각 인스턴스 유형에 대해 한 줄을 표시합니다.

```
SEARCH(' {AWS/EC2,InstanceType} InstanceType=micro MetricName="CPUUtilization" ',
 'Average', 300)
```

이 예제는 이전 예제의 범위를 좁히며, InstanceType을 t2.micro 인스턴스에 대한 정확한 검색으로 변경합니다.

```
SEARCH(' {AWS/EC2,InstanceType} InstanceType="t2.micro" MetricName="CPUUtilization" ',
 'Average', 300)
```

다음 검색은 쿼리의 {metric schema} 부분을 제거하므로 모든 네임스페이스의 CPUUtilization 지표가 그래프에 표시됩니다. 여러 차원으로 집계된 각 AWS 서비스의 CPUUtilization 지표에 대해 여러 줄이 그래프에 포함되어 있으므로 상당히 많은 결과가 반환될 수 있습니다.

```
SEARCH(' MetricName="CPUUtilization" ', 'Average', 300)
```

이러한 결과를 약간 좁히려면 두 개의 특정 지표 네임스페이스를 지정하면 됩니다.

```
SEARCH(' MetricName="CPUUtilization" AND ("AWS/ECS" OR "AWS/ES") ', 'Average', 300)
```

각 쿼리에서 지표 스키마를 하나만 지정할 수 있으므로, 위의 예제는 검색 쿼리 하나로 여러 개의 특정 네임스페이스를 검색하는 유일한 방법입니다. 그러나 더 많은 구조를 추가하기 위해 다음 예제와 같이 그래프에서 쿼리 두 개를 사용할 수 있습니다. 또한 이 예제에서는 Amazon ECS에 대한 데이터를 집계하는 데 사용할 차원을 지정하여 더 많은 구조를 추가합니다.

```
SEARCH(' {AWS/ECS, ClusterName}, MetricName="CPUUtilization" ', 'Average', 300)
SEARCH(' {AWS/EBS}, MetricName="CPUUtilization" ', 'Average', 300)
```

다음 예제는 이름이 ConsumedLCUs인 Elastic Load Balancing 지표와 flow 토큰이 포함된 Elastic Load Balancing 지표 또는 차원도 모두 반환합니다.

```
SEARCH(' {AWS/NetworkELB, LoadBalancer}, "ConsumedLCUs" OR flow ', 'Maximum', 120)
```

다음 예제는 중첩 그룹화를 사용합니다. 여기에서는 모든 함수의 Errors 및 이름에 ProjectA 또는 ProjectB 문자열이 포함된 함수의 Invocations에 대한 Lambda 지표를 반환합니다.

```
SEARCH(' {AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average', 600)
```

다음 예제는 AWS 서비스에서 생성된 지표를 제외한 모든 사용자 지정 지표를 표시합니다.

```
SEARCH(' NOT Namespace=AWS ', 'Average', 120)
```

다음 예제는 이름의 일부로 `Errors` 문자열을 포함하는 지표 이름, 네임스페이스, 차원 이름 및 차원 값이 있는 지표를 표시합니다.

```
SEARCH(' Errors ', 'Average', 300)
```

다음 예제는 정확한 일치로 검색 범위를 좁힙니다. 예를 들어, 이 검색은 `Errors` 지표 이름을 찾지만 이름이 `ConnectionErrors` 또는 `errors`인 지표를 찾지 않습니다.

```
SEARCH(' "Errors" ', 'Average', 300)
```

다음 예제는 검색어의 지표 스키마 부분에 공백이나 특수 문자가 포함된 이름을 지정하는 방법을 보여줍니다.

```
SEARCH(' {"Custom-Namespace", "Dimension Name With Spaces"}, ErrorCount ', 'Maximum', 120)
```

검색 표현식이 있는 CloudWatch 그래프 생성

CloudWatch 콘솔에서 대시보드에 그래프를 추가할 때 또는 지표 보기를 사용하여 검색 기능에 액세스할 수 있습니다.

기존 대시보드에 검색 표현식이 있는 그래프를 추가하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 대시보드를 선택하고 대시보드를 선택합니다.
3. 위젯 추가를 선택합니다.
4. 행 또는 누적 면적을 선택하고 구성을 선택합니다.
5. 그래프로 표시된 지표 탭에서 Add a math expression(수학 표현식 추가)을 선택합니다.
6. 세부 정보에 원하는 검색 표현식을 입력합니다. 예, **`SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average', 300)`**
7. (선택 사항) 그래프에 다른 검색 표현식 또는 수학 표현식을 추가하려면 수학 표현식 추가를 선택합니다.
8. (선택 사항) 검색 표현식을 추가한 후 그래프 범례에서 각 지표에 대해 표시할 동적 레이블을 지정할 수 있습니다. 동적 레이블은 지표에 대한 통계를 표시하고 대시보드 또는 그래프를 새로 고칠 때 자동으로 업데이트됩니다. 동적 레이블을 추가하려면 그래프로 표시된 지표를 선택한 다음 Dynamic labels(동적 레이블)를 선택합니다.

   기본적으로 레이블에 추가하는 동적 값은 레이블 시작 부분에 나타납니다. 그런 다음 지표에 대한 레이블 값을 클릭하여 레이블을 편집할 수 있습니다. 자세한 내용은 동적 레이블 사용 (p. 43) 단원을 참조하십시오.
9. (선택 사항) 그래프에 단일 지표를 추가하려면 모든 지표 탭을 선택하고 원하는 지표로 드릴다운합니다.
10. (선택 사항) 그래프에 표시된 시간 범위를 변경하려면 그래프 맨 위의 사용자 지정이나 사용자 지정 왼쪽에 있는 기간 중 하나를 선택합니다.
11. (선택 사항) 가로 주석을 사용하면 대시보드 사용자는 지표가 특정 레벨까지 급상승하는지 아니면 지표가 사전 정의된 범위 내에 있는지 여부를 빠르게 확인할 수 있습니다. 가로 주석을 추가하려면 그래프 옵션을 선택한 후 가로 주석 추가를 선택합니다.

    a. 레이블에 주석의 레이블을 입력합니다.
    b. 값에 가로 주석이 표시될 지표 값을 입력합니다.
    c. Fill(채우기)에서 이 주석에 채우기 셰이딩을 사용할지 여부를 지정합니다. 예를 들어 채울 영역에 대해 `Above` 또는 `Below`를 선택합니다. `Between`을 지정할 경우 다른 `Value` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.

d. 그래프에 여러 지표가 포함된 경우 축에서 `Value`의 숫자가 왼쪽 Y축과 연결된 지표를 참조하는지 아니면 오른쪽 Y축과 연결된 지표를 참조하는지를 지정합니다.

주석의 왼쪽 열에서 색상 정사각형을 선택하여 주석의 채우기 색상을 변경할 수 있습니다.

동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

주석을 삭제하려면 작업 열에서 x를 선택합니다.

12. (선택 사항) 세로 주석을 사용하면 그래프에 운영 이벤트나 배포의 시작과 끝과 같은 마일스톤을 표시하는 데 도움이 됩니다. 세로 주석을 추가하려면 그래프 옵션을 선택한 후 Add vertical annotation(세로 주석 추가)을 선택합니다.

a. 레이블에 주석의 레이블을 입력합니다. 주석에 날짜와 시간만 표시하려면 레이블 필드를 비워둡니다.
b. 날짜에서, 세로 주석이 표시되는 날짜와 시간을 지정합니다.
c. Fill(채우기)에서 채우기 셰이딩을 세로 주석 앞에 또는 뒤에 사용할지 아니면 두 개의 세로 주석 사이에 사용할지 여부를 지정합니다. 예를 들어 채울 영역에 대해 `Before` 또는 `After`를 선택합니다. `Between`을 지정할 경우 다른 `Date` 필드가 표시되며, 두 값 사이의 그래프 영역이 채워집니다.

동일한 그래프에 여러 세로 주석을 추가하려면 이들 단계를 반복합니다.

주석을 숨기려면 해당 주석의 왼쪽 열에서 확인란을 선택 취소합니다.

주석을 삭제하려면 작업 열에서 x를 선택합니다.

13. 위젯 생성을 선택합니다.
14. 대시보드 저장을 선택합니다.

지표 보기를 사용하여 검색한 지표를 그래프로 작성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 검색 필드에 검색할 토큰을 입력합니다(예: **`cpuutilization t2.small`**).

검색과 일치하는 결과가 표시됩니다.

4. 검색과 일치하는 지표를 모두 그래프로 작성하려면 Graph search(그래프 검색)를 선택합니다.

또는

검색을 구체화하려면 검색 결과에 표시된 네임스페이스 중 하나를 선택합니다.

5. 네임스페이스를 선택하여 결과를 좁힌 경우 다음을 수행할 수 있습니다.

a. 하나 이상의 지표를 그래프 처리하려면 각 지표 옆에 있는 확인란을 선택합니다. 모든 지표를 선택하려면 테이블의 머리글 행에 있는 확인란을 선택합니다.
b. 검색을 구체화하려면 지표 이름 위로 마우스를 이동하고 검색에 추가 또는 이 항목만 검색을 선택합니다.
c. 지표에 대한 도움말을 보려면 지표 이름을 선택하고 이것은 무엇입니까?를 선택합니다.

선택한 지표가 그래프에 표시됩니다.

6. (선택 사항) 검색어의 해당 부분을 편집하려면 검색 표시줄에 있는 버튼 중 하나를 선택합니다.
7. (선택 사항) 대시보드에 그래프를 추가하려면 작업을 선택한 후 대시보드에 추가를 선택합니다.

Amazon CloudWatch 경보 사용

CloudWatch에서 지표 경보와 복합 경보를 모두 생성할 수 있습니다.

- 지표 경보는 단일 CloudWatch 지표를 감시하거나 CloudWatch 지표를 기반으로 한 수학 표현식의 결과를 감시합니다. 이러한 경보는 여러 기간에 대해 지정된 임계값과 지표 또는 표현식의 값 비교하여 하나 이상의 작업을 수행합니다. Amazon EC2 작업, Amazon EC2 Auto Scaling 작업 또는 Amazon SNS 주제로 알림 전송이 이러한 작업에 해당됩니다.
- 복합 경보에는 사용자가 생성한 다른 경보의 경보 상태를 고려하는 규칙 표현식이 포함됩니다. 복합 경보는 규칙의 모든 조건이 충족되는 경우에만 ALARM 상태로 전환됩니다. 복합 경보의 규칙 표현식에 지정된 경보에는 지표 경보 및 기타 복합 경보가 포함될 수 있습니다.

  복합 경보를 사용하면 경보 노이즈를 줄일 수 있습니다. 여러 지표 경보를 생성할 수 있으며, 복합 경보를 생성하고 복합 경보에 대해서만 경보를 설정할 수도 있습니다. 예를 들어 모든 기본 지표 경보가 ALARM 상태인 경우에만 복합 경보가 ALARM 상태로 전환되도록 할 수 있습니다.

  복합 경보는 상태를 변경할 때 Amazon SNS 알림을 보낼 수 있지만, EC2 작업이나 Auto Scaling 작업을 수행할 수 없습니다.

경보를 CloudWatch 대시보드에 추가하여 시각적으로 모니터링할 수 있습니다. 대시보드에서 경보가 발행되면 빨간색으로 `ALARM` 상태라고 표시가 되기 때문에 사전에 손쉽게 상태를 모니터링할 수 있습니다.

경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 특정 상태가 되었다고 해서 작업을 호출하지는 않습니다. 이러한 상태가 변경되어 지정한 기간 동안 유지되어야 합니다.

상태 변경으로 인해 경보가 작업을 호출한 다음의 후속 동작은 경보와 연결한 작업 유형에 따라 달라집니다. Amazon EC2 Auto Scaling 작업의 경우, 경보가 새로운 상태로 유지되는 모든 기간에 대해 지속적으로 작업을 호출합니다. Amazon SNS 알림의 경우 호출되는 추가 작업이 없습니다.

> **Note**
>
> CloudWatch에서는 지정한 작업을 테스트 또는 검증하지 않으며 없는 작업을 호출하려는 시도로 인한 Amazon EC2 Auto Scaling 또는 Amazon SNS 오류를 감지하지도 않습니다. 경보 작업이 존재하는지 확인하십시오.

지표 경보 상태

지표 경보에는 다음과 같은 상태가 있을 수 있습니다.

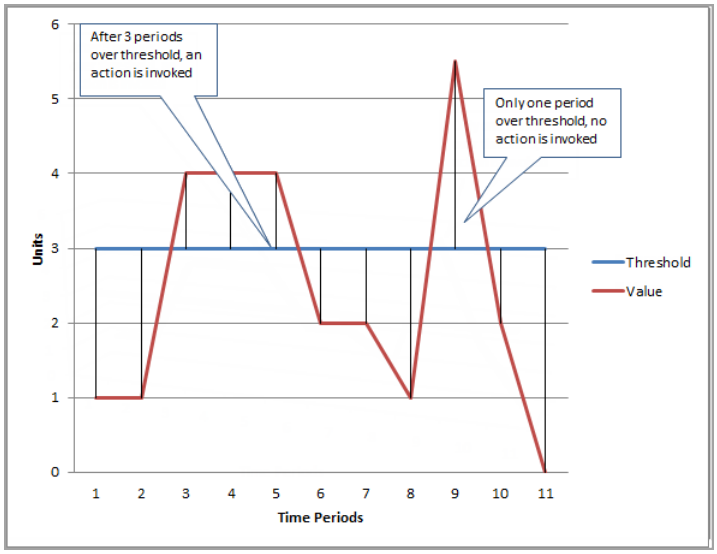
- `OK` – 지표 또는 표현식이 정의된 임계값 내에 있습니다.
- `ALARM` – 지표 또는 표현식이 정의된 임계값을 벗어났습니다.
- `INSUFFICIENT_DATA` – 경보가 방금 시작되었거나, 지표를 사용할 수 없거나, 지표를 통해 경보 상태를 결정하는 데 사용할 충분한 데이터가 없습니다.

경보 평가

경보를 생성할 때, CloudWatch가 경보 상태를 변경할 때를 평가할 수 있도록 3가지 설정을 지정할 수 있습니다.

- 기간은 경보에 대해 개별 데이터 포인트를 생성하기 위해 지표 또는 표현식을 평가하는 기간입니다. 초로 표시됩니다. 1분을 기간으로 선택하면 1분마다 하나의 데이터 포인트가 생성됩니다.
- Evaluation Period(평가 기간)는 경보 상태를 결정할 때 평가할 가장 최근의 기간 또는 데이터 포인트의 수입니다.
- Datapoints to Alarm(경보에 대한 데이터 포인트)은 평가 기간에 경보가 `ALARM` 상태에 도달하게 만드는 위반 데이터 포인트의 수입니다. 위반 데이터 포인트가 연속적일 필요는 없습니다. Evaluation Period(평가 기간)와 동일한 마지막 데이터 포인트의 수 이내이면 됩니다.

다음 그림에서 지표 경보에 대한 경보 임계값은 3개 단위로 설정됩니다. 경보가 `ALARM` 상태가 되도록 구성하며, Evaluation Period(평가 기간) 및 Datapoints to Alarm(경보에 대한 데이터포인트)는 모두 3입니다. 가장 최근의 연속된 3번의 기간에서 모든 기존 데이터 포인트가 모두 임계값 이상일 때 경보가 `ALARM` 상태가 됩니다. 그림에서는 기간 3에서 6 사이에 이러한 일이 발생합니다. 기간 6에서는 이 값이 임계값 아래로 떨어져 평가 대상 기간 중 하나가 위반 상태가 아니기 때문에 경보 상태가 `OK`로 변경됩니다. 9번째 기간에 다시 한 번 임계값이 위반되지만, 오직 하나의 기간 동안에만 그렇습니다. 결과적으로 경보 상태는 `OK`로 남아 있습니다.



Evaluation Period(평가 기간)와 Datapoints to Alarm(경보에 대한 데이터 포인트) 값을 다르게 구성하는 경우 "N 중 M" 경보를 설정합니다. Datapoints to Alarm(경보에 대한 데이터 포인트)은 ("M")이며, Evaluation Period(평가 기간)는 ("N")입니다. 평가 간격은 데이터 포인트의 수에 기간을 곱한 값입니다. 예를 들어, 1분 기간으로 5개의 데이터 포인트 중 4개를 구성하는 경우 평가 간격은 5분입니다. 10분의 기간으로 3개의 데이터 포인트 중 3개를 구성하는 경우 평가 간격은 30분입니다.

CloudWatch 경보가 누락 데이터를 처리하는 방법 구성

때론 특정 지표에 대한 경보 데이터 포인트 가운데 일부가 CloudWatch에 보고되지 않는 경우도 있습니다. 연결이 끊기거나 서버가 정지할 때, 설계에 따라 지표 보고 데이터가 간헐적으로만 전송될 때 이런 일이 일어날 수 있습니다.

CloudWatch의 경우 경보를 평가할 때 누락된 데이터 포인트를 처리하는 방법을 지정할 수 있습니다. 이렇게 하면 모니터링 중인 데이터 유형에 적합한 `ALARM` 상태로 경보를 구성할 수 있습니다. 누락된 데이터에 문제가 없는 경우의 거짓 긍정을 피할 수 있습니다.

각 경보가 항상 세 가지 상태 중 하나인 것과 마찬가지로, CloudWatch에 보고된 각각의 특정 데이터 포인트는 세 가지 범주 중 하나에 속합니다.

- 위반하지 않음(임계값에서)
- 위반(임계값 위반)
- 누락

각 경보에 대해 다음 중 하나로 누락된 데이터 포인트가 처리되도록 CloudWatch를 지정할 수 있습니다.

- `notBreaching` – 누락 데이터 요소를 "양호"하고 임계값 내에 있는 것으로 처리합니다.
- `breaching` – 누락 데이터 요소를 "불량"하고 임계값을 위반한 것으로 처리합니다.
- `ignore` – 현재 경보 상태를 유지합니다.
- `missing` – 경보가 상태 변경 여부를 평가할 때 누락 데이터 포인트를 고려하지 않습니다.

최고의 옵션은 지표 유형에 따라 다릅니다. 인스턴스의 `CPUUtilization`과 같은 데이터를 지속적으로 보고하는 지표의 경우 어떤 문제가 발생했음을 나타내기 위해 누락된 데이터 포인트를 `breaching`으로 처리해야 할 수 있습니다. 그러나 Amazon DynamoDB에서 `ThrottledRequests` 같은 오류가 발생할 때만 데이터 포인트를 생성하는 지표의 경우에는 누락 데이터를 `notBreaching`으로 처리할 수 있습니다. 기본값은 `missing`입니다.

경보에 대한 최상의 옵션을 선택하면 불필요하고 오해의 소지가 있는 경보 조건 변경을 막을 수 있으며, 시스템 상태를 보다 정확하게 나타낼 수 있습니다.

데이터가 누락되었을 때 경보 상태 평가 방법

CloudWatch는 경보 상태 변경 여부를 평가할 때 누락 데이터를 처리하는 방법에 대해 설정한 값에 상관없이 Evaluation Periods(평가 기간)에서 지정한 것보다 더 큰 데이터 포인트 수를 검색하려 시도합니다. 검색을 시도하는 데이터 포인트의 정확한 수는 경보 기간의 길이, 표준 해상도 또는 고해상도 지표에 토대를 두고 있는지 여부에 따라 달라집니다. 검색을 시도하는 데이터 포인트의 기간이 평가 범위입니다.

CloudWatch가 이런 데이터 포인트를 검색한 후에는 다음이 진행됩니다.

- 평가 범위 동안 누락된 데이터 포인트가 없는 경우 CloudWatch는 가장 최근 수집한 데이터 포인트에 따라 경보를 평가합니다.
- 평가 범위 동안 일부 데이터 포인트가 누락되었지만 검색한 기존 데이터 포인트의 수가 경보 Evaluation Periods(평가 기간) 이상인 경우, CloudWatch는 성공적으로 검색을 한 가장 최근의 기존 데이터 포인트에 따라 경보 상태를 평가합니다. 이 경우 누락 데이터 처리 방법에 대한 값이 필요 없으며, 이를 무시합니다.
- 평가 범위 동안 일부 데이터 포인트가 누락되었으며 검색한 기존 데이터 포인트의 수가 경보 평가 기간의 수 이하인 경우, CloudWatch는 사용자가 누락 데이터 처리 방법에 대해 지정한 값으로 누락 데이터 포인트를 채운 다음 경보를 평가합니다. 하지만 보고 시기에 상관없이 평가 범위 동안의 실제 데이터 포인트는 모두 평가에 포함시킵니다. CloudWatch는 가능한 몇 회만 누락 데이터 포인트를 사용합니다.

이 모든 상황에서 평가된 데이터 포인트의 수는 Evaluation Periods(평가 기간)의 값과 동일합니다. Datapoints to Alarm(경보에 대한 데이터 포인트)의 값보다 작은 값만 위반된 경우 경보 상태는 `OK`로 설정됩니다. 나머지 경우는 `ALARM`으로 설정됩니다. 누락된 데이터 포인트를 `missing`으로 처리하도록 경보를 설정한 경우는 예외입니다. 이 경우 CloudWatch에서 평가 범위 동안 충분한 실제 데이터 포인트를 검색할 수 없는 경우 경보 상태가 `INSUFFICIENT_DATA`로 설정됩니다.

> **Note**
>
> 이 동작은 특별한 경우 지표 흐름이 멈춘 후에도 CloudWatch 경보 기능이 일정 시간 동안 최종 데이터 포인트 세트를 계속해서 다시 평가할 수 있습니다. 이 재평가를 통해 지표 스트림 중지 직전에 상태가 변한 경우 경보가 상태를 변경하고 작업을 다시 실행할 수 있습니다. 이 동작을 완화하려면 더 짧은 기간을 사용하십시오.

다음은 경보 평가 동작에 대한 예를 설명한 테이블입니다. 첫 번째 테이블에서 Datapoints to Alarm(경보에 대한 데이터 포인트)과 Evaluation Periods(평가 기간)는 모두 3입니다. CloudWatch는 가장 최근의 데이터

포인트 3개 중 일부가 누락된 경우 경보를 평가할 때 가장 최근의 데이터 포인트 5개를 검색합니다. 5가 경보에 대한 평가 범위입니다.

2열은 필요한 데이터 요소 3개 중 누락된 개수를 표시합니다. 가장 최신 데이터 요소 5개가 평가되었더라도 경보 상태 평가를 위해 3개(Evaluation Periods(평가 기간)에 대한 설정)만 필요합니다. 2열의 데이터 요소 개수는 누락된 데이터 요소 처리 방법에 대한 설정을 사용하여 반드시 "채워야" 하는 데이터 요소의 수입니다.

3-6열은 누락 데이터 처리 방법에 대한 설정 각각에 대해 설정하게 되는 경보 상태입니다(각 열의 맨 위에 표시). 데이터 포인트 열에서 0은 위반되지 않는 데이터 포인트, X는 위반 데이터 포인트, -는 누락 데이터 포인트를 나타냅니다.

데이터 포인트	누락 데이터 포인트 가운데 수(#)	누락	IGNORE	위반	위반하지 않음
0 - X - X	0	OK	OK	OK	OK
0 - - - -	2	OK	OK	OK	OK
- - - - -	3	INSUFFICIENT_DATA	현재 상태 유지	ALARM	OK
0 X X - X	0	ALARM	ALARM	ALARM	ALARM
- - X - -	2	ALARM	ALARM	ALARM	OK

앞 테이블의 2행에서는 누락 데이터를 위반으로 처리하는 경우에도 경보 상태는 OK로 유지됩니다. 기존 데이터 포인트 중 하나가 위반 상태가 아니며, 위반으로 처리되는 2개의 누락 데이터 포인트와 함께 이를 평가하기 때문입니다. 다음 번에 이 경보를 평가할 때 이 데이터가 여전히 누락된 경우에는 ALARM 상태가 됩니다. 위반하지 않은 데이터 포인트가 가장 최근 검색한 5개 데이터 포인트에서 빠지기 때문입니다.

4행에서는 경보가 모두 ALARM 상태입니다. 실제 데이터 포인트가 충분해 누락 데이터 처리 방법에 대한 설정을 고려할 필요가 없기 때문입니다.

마지막 행에서는 경보의 가장 최근 3개 데이터 요소가 위반되거나 존재하지 않으면 경보가 ALARM 상태로 전환되기 때문에 경보는 ALARM 상태로 전환됩니다.

다음 테이블의 경우 기간은 다시 5분이며, Datapoints to Alarm(경보에 대한 데이터 포인트)는 2, Evaluation Periods(평가 기간)는 3입니다. 'N 중 M' 경보는 '3 중 2'입니다.

평가 범위는 5입니다. 이것은 검색되는 최근 데이터 포인트의 최대 수이며 일부 데이터 포인트가 누락된 경우 사용할 수 있습니다.

데이터 포인트	누락 데이터 포인트 가운데 수(#)	누락	IGNORE	위반	위반하지 않음
0 - X - X	0	ALARM	ALARM	ALARM	ALARM
0 0 X 0 X	0	ALARM	ALARM	ALARM	ALARM
0 - X - -	1	OK	OK	ALARM	OK
- - - - 0	2	OK	OK	ALARM	OK
- - X - -	2	ALARM	현재 상태 유지	ALARM	OK

경보를 생성한 직후에 데이터 포인트가 누락되었으며 경보를 생성하기 전에 CloudWatch에 지표가 보고된 경우, CloudWatch는 경보가 생성되기 전 가장 최근의 데이터 포인트를 검색해 경보를 평가합니다.

고분해능 경보

고분해능 지표에 대해 경보를 설정할 경우 고분해능 경보를 10초 또는 30초 기간으로 지정하거나 60초의 배수 기간으로 정기 경보를 설정할 수 있습니다. 고분해능 경보는 요금이 더 비쌉니다. 고분해능 지표에 대한 자세한 내용은 사용자 지정 메트릭 게시 (p. 47) 단원을 참조하십시오.

수학 표현식에 대한 경보

하나 이상의 CloudWatch 지표를 기반으로 하는 수학 표현식의 결과에 대한 알림을 설정할 수 있습니다. 경보에 사용되는 수학 표현식에는 지표를 10개까지 포함할 수 있습니다. 각 지표의 기간은 동일해야 합니다.

수학 표현식을 기반으로 하는 경보의 경우 경보를 평가할 때 CloudWatch에서 기본 지표에 대해 누락된 데이터 포인트를 처리하는 방법을 지정할 수 있습니다.

수학 표현식을 기반으로 하는 경보는 Amazon EC2 작업을 수행할 수 없습니다.

지표 수학 표현식 및 구문에 대한 자세한 내용은 지표 수식 사용 (p. 49) 단원을 참조하십시오.

백분위수 기반 CloudWatch 경보 및 데이터 샘플 부족

경보를 위한 통계로 백분위수를 설정하면 정확한 통계 평가를 위한 데이터가 충분하지 않을 때 어떻게 할 것인지 지정할 수 있습니다. 경보가 통계를 어떻게든 평가하도록 하고 가능하면 경보 상태를 변경하도록 선택할 수 있습니다. 또는 샘플 크기가 작을 때 경보가 지표를 무시하고 통계적으로 의미가 있을 정도로 충분한 데이터가 모일 때까지 기다렸다가 평가할 수 있습니다.

0.5(포함) ~ 1.00(제외) 범위 백분위수의 경우, 평가 기간 동안 10/(1-백분위수) 보다 적은 데이터 포인트가 있을 때 이 설정이 사용됩니다. 예를 들어 p99 백분위수에서 경보 샘플이 1,000개보다 적을 경우 이 설정이 사용됩니다. 0 ~ 0.5(제외) 범위 백분위수의 경우, 10/백분위수 보다 적은 백분위수가 있을 때 이 설정이 사용됩니다.

CloudWatch 경보의 일반적인 기능

아래 기능은 모든 CloudWatch 경보에 적용됩니다.

- AWS 계정마다 리전당 최대 5,000개까지 경보를 만들 수 있습니다. 경보를 만들거나 업데이트하려면 `PutMetricAlarm` API 작업(`mon-put-metric-alarm` 명령)을 사용합니다.
- 경보 이름은 ASCII 문자만 포함해야 합니다.
- 현재 구성된 경보의 일부 또는 전체를 나열하고 `DescribeAlarms`(`mon-describe-alarms`)를 사용하여 특정 상태에 있는 모든 경보를 나열할 수 있습니다. 시간 범위를 기준으로 목록을 추가로 필터링할 수 있습니다.
- `DisableAlarmActions` 및 `EnableAlarmActions`(`mon-disable-alarm-actions` 및 `mon-enable-alarm-actions`)를 사용하여 경보를 활성화 및 비활성화할 수 있습니다.
- `SetAlarmState`(`mon-set-alarm-state`)를 사용하여 어떤 상태로든 경보를 설정하여 테스트를 할 수 있습니다. 이러한 일시적인 상태 변경은 다음 경보 비교 시까지만 지속됩니다.
- 사용자 지정 지표를 생성하기 전에 `PutMetricAlarm`(`mon-put-metric-alarm`)를 사용하여 경보를 생성할 수 있습니다. 경보가 유효하려면 사용자 지정 지표에 대한 모든 차원을 비롯해 지표 네임스페이스 및 지표 이름을 경보 정의에 포함시켜야 합니다.

- `DescribeAlarmHistory(mon-describe-alarm-history)`를 사용하여 경보 기록을 볼 수 있습니다. CloudWatch에서는 경보 기록을 2주 동안 보관합니다. 각 상태 전환은 고유한 타임스탬프로 표시됩니다. 드문 경우지만 기록에 상태 변경에 대한 알림이 두 개 이상 있을 수 있습니다. 이 경우 타임스탬프를 사용하여 고유한 상태 변경을 확인할 수 있습니다.
- 경보에 대한 평가 기간의 수에 각 평가 기간의 길이를 곱한 값이 1일을 초과할 수 없습니다.

Note

일부 AWS 리소스에서는 특정한 상황에서 지표 데이터를 CloudWatch에 전송하지 않습니다. 예를 들어 Amazon EC2 인스턴스에 연결되지 않은 사용 가능한 볼륨에 대해 모니터링할 지표 활동이 없으므로 Amazon EBS에서는 이러한 볼륨에 대한 지표 데이터를 전송할 수 없습니다. 이러한 지표에 대한 경보 세트가 있으면 상태가 `INSUFFICIENT_DATA`로 변경됩니다. 이는 리소스가 비활성 상태임을 나타내지만 그렇다고 반드시 문제가 있음을 의미하지는 않습니다.

Amazon SNS 알림 설정

Amazon CloudWatch는 이메일을 보내기 위해 Amazon SNS를 사용합니다. 먼저, SNS 주제를 생성 및 구독합니다. CloudWatch 경보를 만들면 이 SNS 주제를 추가하여 경보 상태 변경 시 이메일 알림을 보낼 수 있습니다. 자세한 내용은 Amazon Simple Notification Service 시작 안내서를 참조하십시오.

Note

또는 AWS Management 콘솔을 사용하여 CloudWatch 경보를 생성할 계획이라면 이 절차를 건너뛸 수 있습니다. 왜냐하면 경보 생성 마법사를 통해 주제를 생성할 수 있기 때문입니다.

AWS Management 콘솔을 사용하여 Amazon SNS 주제 설정

먼저, 주제를 생성한 다음 구독합니다. 선택적으로 테스트 메시지를 주제에 게시할 수 있습니다.

SNS 주제를 생성하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. Amazon SNS 대시보드의 일반 작업에서 주제 생성을 선택합니다.
3. 새로운 주제 생성 대화 상자의 주제 이름에 주제 이름(예: **`my-topic`**)을 입력합니다.
4. 주제 생성을 선택합니다.
5. 다음 작업에서 주제 ARN을 복사합니다(예: arn:aws:sns:us-east-1:111122223333:my-topic).

SNS 주제를 구독하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 구독과 구독 생성을 선택합니다.
3. 구독 생성 대화 상자의 주제 ARN에서 이전 작업에서 생성한 주제 ARN을 붙여 넣습니다.
4. 프로토콜에서 이메일을 선택합니다.
5. 엔드포인트에 알림을 받는 데 사용할 수 있는 이메일 주소를 입력한 다음 구독 생성을 선택합니다.
6. 이메일 애플리케이션에서 AWS 알림에서 보낸 메시지를 연 다음, 구독을 확인합니다.

Amazon SNS로부터의 확인 반응이 웹 브라우저에 표시됩니다.

SNS 주제에 테스트 메시지를 게시하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 주제를 선택합니다.
3. 주제 페이지에서 주제를 선택하고 주제 게시를 선택합니다.
4. 메시지 게시 페이지의 제목에 메시지에 대한 제목 줄을 입력하고 메시지에 간단한 메시지를 입력합니다.
5. 메시지 게시를 선택합니다.
6. 해당 메시지를 받았는지 이메일을 확인합니다.

AWS CLI를 사용하여 SNS 주제 설정

먼저 SNS 주제를 생성한 다음, 해당 주제에 직접 메시지를 게시해서 제대로 구성이 되었는지 테스트합니다.

SNS 주제를 설정하려면

1. 아래와 같이 create-topic 명령을 사용하여 주제를 생성합니다.

```
aws sns create-topic --name my-topic
```

Amazon SNS는 다음의 형식으로 주제 ARN을 반환합니다.

```
{
    "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic"
}
```

2. subscribe 명령을 사용하여 구독 이메일 주소를 주제에 연결합니다. 구독 요청이 성공하면 구독 확인 이메일 메시지를 받게 됩니다.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:111122223333:my-topic --protocol
 email --notification-endpoint my-email-address
```

Amazon SNS에서 다음을 반환합니다.

```
{
    "SubscriptionArn": "pending confirmation"
}
```

3. 이메일 애플리케이션에서 AWS 알림에서 보낸 메시지를 연 다음, 구독을 확인합니다.

Amazon Simple Notification Service로부터의 확인 반응이 웹 브라우저에 표시됩니다.

4. list-subscriptions-by-topic 명령을 사용하여 구독을 확인합니다.

```
aws sns list-subscriptions-by-topic --topic-arn arn:aws:sns:us-east-1:111122223333:my-
topic
```

Amazon SNS에서 다음을 반환합니다.

```
{
  "Subscriptions": [
    {
        "Owner": "111122223333",
        "Endpoint": "me@mycompany.com",
        "Protocol": "email",
```

```
        "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic",
        "SubscriptionArn": "arn:aws:sns:us-east-1:111122223333:my-topic:64886986-
bf10-48fb-a2f1-dab033aa67a3"
    }
  ]
}
```

5. (선택 사항) publish 명령을 사용하여 해당 주제로 테스트 메시지를 게시합니다.

```
aws sns publish --message "Verification" --topic arn:aws:sns:us-east-1:111122223333:my-
topic
```

Amazon SNS에서 다음을 반환합니다.

```
{
    "MessageId": "42f189a0-3094-5cf6-8fd7-c2dde61a4d7d"
}
```

6. 해당 메시지를 받았는지 이메일을 확인합니다.

정적 임계점을 기반으로 CloudWatch 알람을 생성

감시할 경보에 대한 CloudWatch 지표를 선택하고 해당 지표에 대한 임계값을 선택합니다. 지표가 지정된 수의 평가 기간에 대한 임계값을 위반할 경우 경보가 ALARM 상태가 됩니다.

CloudWatch에서 경보 사용자 인터페이스를 변경했습니다. 기본적으로 새 사용자 인터페이스가 표시되지만 이전 사용자 인터페이스로 돌아갈 수 있습니다. 이 주제에는 각각에 대한 단계가 포함되어 있습니다.

새 경보 사용자 인터페이스를 사용하여 단일 지표를 기반으로 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택을 선택하고 다음 중 하나를 수행합니다.
   - 원하는 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 원하는 지표 옆에 있는 확인란을 선택합니다.
   - 검색 상자에 지표 이름, 차원 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 그런 다음 결과 중 하나를 선택하고 지표 목록이 표시될 때까지 계속 진행합니다. 원하는 지표 옆의 확인란을 선택합니다.
4. 그래프로 표시된 지표 탭을 선택합니다.
   a. 통계에서 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: **p95.45**)를 지정합니다.
   b. 기간에서 경보에 대한 평가 기간을 선택합니다. 경보를 평가할 때 각 기간이 하나의 데이터 포인트로 집계됩니다.

      또한 경보를 생성할 때 Y축 범례를 왼쪽 또는 오른쪽에 표시할지 여부를 선택할 수도 있습니다. 이러한 기본 설정은 경보를 생성하는 동안에만 사용됩니다.
   c. 지표 선택을 선택합니다.

      선택한 지표 및 통계에 대한 그래프와 기타 정보가 표시된 Specify metric and conditions(지표 및 조건 지정) 페이지가 나타납니다.
5. 조건에서 다음을 지정합니다.
   a. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다.

b. Whenever ***metric*** is(지표가 다음인 경우 항상)에서 지표가 임계값보다 크거나, 작거나, 같아야 하는지 여부를 지정합니다. than...(비교 대상)에서 임계값을 지정합니다.

c. 추가 구성을 선택합니다. Datapoints to alarm(경보에 대한 데이터 포인트)에서 경보를 트리거하기 위해 평가 기간(데이터 포인트)이 `ALARM` 상태로 유지해야 하는 기간을 지정합니다. 두 값이 일치하는 경우 다수의 연속 기간이 위반되면 `ALARM` 상태가 되는 경보가 생성됩니다.

   N 중 M 경보를 생성하려면 두 번째 값에 지정한 값보다 낮은 값을 첫 번째 값에 지정합니다. 자세한 내용은 경보 평가 (p. 67) 단원을 참조하십시오.

d. Missing data treatment(누락 데이터 처리)에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 내용은 CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 (p. 68) 단원을 참조하십시오.

e. 경보가 모니터링된 통계 값으로 백분위수를 사용하는 경우에는 샘플이 부족한 백분위수 상자가 표시됩니다. 샘플 비율이 낮은 사례를 평가 또는 무시할지 여부를 선택할 때 이 상자를 사용합니다. ignore (maintain alarm state)(무시(경보 상태 유지))를 선택하면 샘플 크기가 너무 작을 때 현재 경보 상태가 항상 유지됩니다. 자세한 내용은 백분위수 기반 CloudWatch 경보 및 데이터 샘플 부족 (p. 71) 단원을 참조하십시오.

6. 다음을 선택합니다.
7. 알림에서 경보가 `ALARM` 상태, `OK` 상태 또는 `INSUFFICIENT_DATA` 상태일 때 알릴 SNS 주제를 선택합니다.

   경보가 동일한 경보 상태 또는 다른 경보 상태에 대해 여러 개의 알림을 보내도록 설정하려면 알림 추가를 선택합니다.

   경보에서 알림을 보내지 않게 하려면 제거를 선택합니다.
8. 경보가 Auto Scaling 또는 EC2 작업을 수행하도록 하려면 해당 버튼을 선택하고 경보 상태 및 수행할 작업을 선택합니다.
9. 마친 후에는 다음을 선택합니다.
10. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다. 다음을 선택합니다.
11. Preview and create(미리 보기 및 생성)에서 정보 및 조건이 원하는 내용인지 확인한 다음 경보 생성을 선택합니다.

이전 사용자 인터페이스를 사용하여 단일 지표를 기반으로 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택을 선택하고 다음 중 하나를 수행합니다.
   - 원하는 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 원하는 지표 옆에 있는 확인란을 선택합니다.
   - 검색 상자에 지표 이름, 차원 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 그런 다음 결과 중 하나를 선택하고 지표 목록이 표시될 때까지 계속 진행합니다. 원하는 지표 옆의 확인란을 선택합니다.
4. 그래프로 표시된 지표 탭을 선택합니다.

   a. 통계에서 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: **`p95.45`**)를 지정합니다.

   b. 기간에서 경보에 대한 평가 기간을 선택합니다. 경보를 평가할 때 각 기간이 하나의 데이터 포인트로 집계됩니다.

      또한 경보를 생성할 때 Y축 범례를 왼쪽 또는 오른쪽에 표시할지 여부를 선택할 수도 있습니다. 이러한 기본 설정은 경보를 생성하는 동안에만 사용됩니다.

   c. 지표 선택을 선택합니다.
5. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다.

6. 다음 경우 항상에서 경보 조건을 지정합니다.

   a. is:(조건:)에서 지표가 임계값보다 크거나, 작거나, 같아야 하는지 지정하고 임계값을 지정합니다.

   b. for:(기간:)에서 경보를 트리거하기 위해 평가 기간(데이터 포인트)이 `ALARM` 상태로 유지해야 하는 기간을 지정합니다. 처음에는 두 번째 값만 변경할 수 있습니다. 첫 번째 값은 입력한 두 번째 값에 맞춰 바뀝니다. 그러면 다수의 연속 기간이 위반될 경우 `ALARM` 상태가 되는 경보가 생성됩니다.

      N 중 M 경보를 생성하려면 연필 아이콘을 선택합니다. 그런 다음 M 숫자를 N 숫자와 다르게 변경할 수 있습니다. 자세한 내용은 경보 평가 (p. 67) 단원을 참조하십시오.

7. 추가 설정의 누락 데이터 처리에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 내용은 CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 (p. 68) 단원을 참조하십시오.
8. 경보가 모니터링된 통계 값으로 백분위수를 사용하는 경우에는 샘플이 부족한 백분위수 상자가 표시됩니다. 샘플 비율이 낮은 사례를 평가 또는 무시할지 여부를 선택할 때 이 상자를 사용합니다. 무시(경보 상태 유지)를 선택하면 샘플 크기가 너무 작을 때 현재 경보 상태가 항상 유지됩니다. 자세한 내용은 백분위수 기반 CloudWatch 경보 및 데이터 샘플 부족 (p. 71) 단원을 참조하십시오.
9. 작업에서 경보 트리거 시 수행할 작업의 유형을 선택합니다. 경보가 여러 작업을 수행하도록 하려면 +알림, +AutoScaling 작업 및 +EC2 작업 버튼을 사용합니다. 작업을 한 가지 이상 지정합니다.
10. 경보 생성을 선택합니다.

대시보드에 경보를 추가할 수도 있습니다. 자세한 내용은 CloudWatch 대시보드에서 경보를 추가 또는 제거 (p. 26) 단원을 참조하십시오.

이상 탐지에 기반한 CloudWatch 경보 생성

과거 지표 데이터를 마이닝하고 예상 값의 모델을 생성하는 CloudWatch 이상 탐지를 기반으로 경보를 생성할 수 있습니다. 기댓값은 지표의 일반적인 시간별, 일별, 주별 패턴을 고려합니다.

이상 탐지 임계값에 대한 값을 설정합니다. 그러면 CloudWatch에서 모델과 함께 이 임계값을 사용하여 지표 값의 "정상" 범위를 결정합니다. 임계값에 대한 값이 클수록 "정상" 값의 밴드가 더 두꺼워집니다.

지표 값이 기댓값 밴드 이상이거나 이하일 때, 아니면 두 경우 모두 경보가 트리거되도록 설정할 수 있습니다.

자세한 내용은 CloudWatch 이상 탐지 사용 (p. 123) 단원을 참조하십시오.

Note

시각화 목적으로 지표 콘솔에서 이상 탐지를 이미 사용 중인 지표에 이상 탐지 경보를 생성한 경우 경보에 설정한 임계값은 시각화를 위해 이미 사용 중인 임계값을 변경하지 않습니다. 자세한 내용은 그래프 작성 (p. 41) 단원을 참조하십시오.

이상 탐지에 기반하여 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택을 선택하고 다음 중 하나를 수행합니다.
   - 원하는 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 원하는 지표 옆에 있는 확인란을 선택합니다.
   - 검색 상자에 지표 이름, 차원 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 그런 다음 결과 중 하나를 선택하고 지표 목록이 표시될 때까지 계속 진행합니다. 원하는 지표 옆의 확인란을 선택합니다.

4. 그래프로 표시된 지표 탭을 선택합니다.

   a. 통계에서 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: **p95.45**)를 지정합니다.

   b. 기간에서 경보에 대한 평가 기간을 선택합니다. 경보를 평가할 때 각 기간이 하나의 데이터 포인트로 집계됩니다. 이상 탐지 경보의 경우, 값은 1분 이상이어야 합니다.

      또한 경보를 생성할 때 Y축 범례를 왼쪽 또는 오른쪽에 표시할지 여부를 선택할 수도 있습니다. 이러한 기본 설정은 경보를 생성하는 동안에만 사용됩니다.

   c. 지표 선택을 선택합니다.

      선택한 지표 및 통계에 대한 그래프와 기타 정보가 표시된 Specify metric and conditions(지표 및 조건 지정) 페이지가 나타납니다.

5. 조건에서 다음을 지정합니다.

   a. Anomaly Detection(이상 탐지)를 선택합니다.

      이 지표와 통계에 대한 모델이 이미 있는 경우, CloudWatch에서 화면 상단의 샘플 그래프에 이상 탐지 밴드를 표시합니다. 모델이 없는 경우, 경보 생성을 완료하면 모델이 생성됩니다. 모델에서 생성한 실제 이상 탐지 밴드가 그래프에 표시되는 데 최대 15분이 걸립니다. 그 전에 표시되는 밴드는 이상 탐지 밴드의 근사치입니다. 더 긴 기간에서 그래프를 보려면 페이지 오른쪽 위에 있는 편집을 선택합니다.

   b. Whenever ***metric*** is(지표가 다음일 경우 항상)에서 경보를 트리거하기 위해 지표가 밴드보다 크거나, 보다 작거나, 밴드 외부(어느 방향이든)여야 하는지 여부를 지정합니다.

   c. Anomaly detection threshold(이상 탐지 임계값)에서 이상 탐지 임계값에 사용할 숫자를 선택합니다. 숫자가 클수록 지표 변화에 보다 관대한 "정상" 값의 밴드가 두꺼워지며, 숫자가 낮을수록 작은 지표 편차에도 `ALARM` 상태로 전환되는 얇은 밴드가 생성됩니다. 이 숫자는 정수일 필요는 없습니다.

   d. 추가 구성을 선택합니다. Datapoints to alarm(경보에 대한 데이터 포인트)에서 경보를 트리거하기 위해 평가 기간(데이터 포인트)이 `ALARM` 상태로 유지해야 하는 기간을 지정합니다. 두 값이 일치하는 경우 다수의 연속 기간이 위반되면 `ALARM` 상태가 되는 경보가 생성됩니다.

      N 중 M 경보를 생성하려면 두 번째 값에 지정한 값보다 낮은 값을 첫 번째 값에 지정합니다. 자세한 내용은 경보 평가 (p. 67) 단원을 참조하십시오.

   e. Missing data treatment(누락 데이터 처리)에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 내용은 CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 (p. 68) 단원을 참조하십시오.

   f. 경보가 모니터링된 통계 값으로 백분위수를 사용하는 경우에는 샘플이 부족한 백분위수 상자가 표시됩니다. 샘플 비율이 낮은 사례를 평가 또는 무시할지 여부를 선택할 때 이 상자를 사용합니다. ignore (maintain alarm state)(무시(경보 상태 유지))를 선택하면 샘플 크기가 너무 작을 때 현재 경보 상태가 항상 유지됩니다. 자세한 내용은 백분위수 기반 CloudWatch 경보 및 데이터 샘플 부족 (p. 71) 단원을 참조하십시오.

6. 다음을 선택합니다.

7. 알림에서 경보가 `ALARM` 상태, `OK` 상태 또는 `INSUFFICIENT_DATA` 상태일 때 알릴 SNS 주제를 선택합니다.

   경보가 동일한 경보 상태 또는 다른 경보 상태에 대해 여러 개의 알림을 보내도록 설정하려면 알림 추가를 선택합니다.

   경보에서 알림을 보내지 않게 하려면 제거를 선택합니다.

8. 경보가 EC2 작업을 수행하도록 하려면 해당 버튼을 선택하고 경보 상태 및 수행할 작업을 선택합니다.

9. 마친 후에는 다음을 선택합니다.

10. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다. 다음을 선택합니다.

11. Preview and create(미리 보기 및 생성)에서 정보 및 조건이 원하는 내용인지 확인한 다음 경보 생성을 선택합니다.

이상 탐지 모델 수정

경보를 생성하면 이상 탐지 모델을 수정할 수 있습니다. 모델 생성에 특정 기간이 사용되지 않도록 제외할 수 있습니다. 교육 데이터에서 시스템 중단, 배포 및 휴일과 같은 비정상적인 이벤트를 제외하는 것이 중요합니다. 일광 절약 시간제 변경에 대해 모델을 조정할지 여부도 지정할 수 있습니다.

경보에 대한 이상 탐지 모델을 수정하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. 경보 이름을 선택합니다. 필요하면 검색 상자를 사용하여 경보를 찾습니다.
4. 지표에서 보기.를 선택합니다.
5. 화면 하단에서 Edit model(모델 편집)을 선택합니다.
6. 모델을 생성할 때 사용하지 않을 기간을 제외하려면 Add another time range to exclude from training(교육에서 제외할 다른 시간 범위 추가)를 선택합니다. 그런 다음 교육에서 제외할 날짜와 시간을 선택하거나 입력하고 적용을 선택합니다.
7. 지표가 일광절약시간제 변화에 민감한 경우, Metric timezone(지표 시간대) 상자에서 적절한 시간대를 선택합니다.
8. 업데이트를 선택합니다.

이상 탐지 모델 삭제

경보에 대한 이상 탐지를 사용하면 AWS 계정에 요금이 발생합니다. 경보에 대한 이상 탐지 모델이 필요하지 않으면 경보를 삭제하고 모델을 삭제해야 합니다. 경보를 삭제하지 않고 모델을 삭제하면 알람이 자동으로 모델을 다시 생성합니다.

경보를 삭제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. 경보 이름을 선택합니다.
4. [Actions], [Delete]를 선택합니다.

경보에 사용된 이상 탐지 모델을 삭제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 모든 지표 탭의 검색 필드에 검색어(예: 지표 이름 또는 리소스 이름)를 입력하고 Enter 키를 누릅니다.

   예를 들어, `CPUUtilization` 지표를 검색하면 이 지표와 함께 네임스페이스와 차원이 표시됩니다.
4. 결과에서 이상 탐지 모델이 있었던 지표를 선택합니다.
5. 그래프로 표시된 지표 탭을 선택합니다.
6. 화면 하단에서 Edit model(모델 편집)을 선택한 후 Delete model(모델 삭제)을 선택합니다.

지표 수학 표현식을 기반으로 CloudWatch 경보 생성

지표 수학 표현식을 기반으로 경보를 생성하려면 표현식에서 사용할 CloudWatch 지표를 하나 이상 선택합니다. 그런 다음 표현식, 임계값 및 평가 기간을 지정합니다.

CloudWatch에서 경보 사용자 인터페이스를 변경했습니다. 기본적으로 새 사용자 인터페이스가 표시되지만 이전 사용자 인터페이스로 돌아갈 수 있습니다. 이 주제에는 각각에 대한 단계가 포함되어 있습니다.

새 경보 사용자 인터페이스를 사용하여 수학 표현식을 기반으로 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택을 선택하고 다음 중 하나를 수행합니다.
   - 특정 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 오른쪽 지표 옆에 있는 확인란을 선택합니다.
   - 검색 상자에 지표 이름, 차원 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 결과 중 하나를 선택하고 지표 목록이 표시될 때까지 계속 진행합니다. 오른쪽 지표 옆의 확인란을 선택합니다.

   (선택 사항) 수학 표현식에서 사용할 다른 지표를 추가하려면 모든 지표에서 모두를 선택하고 특정 지표를 찾은 다음 해당 지표 옆에 있는 확인란을 선택합니다. 최대 10개의 지표를 추가할 수 있습니다.
4. 그래프로 표시된 지표를 선택합니다. 추가된 각 지표에 대해 다음을 수행합니다.

   a. 통계에서 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: **`p95.45`**)를 지정합니다.
   b. 기간에서 경보에 대한 평가 기간을 선택합니다. 경보를 평가할 때 각 기간이 하나의 데이터 포인트로 집계됩니다.

      또한 경보를 생성할 때 Y축 범례를 왼쪽 또는 오른쪽에 표시할지 여부를 선택할 수도 있습니다. 이러한 기본 설정은 경보를 생성하는 동안에만 사용됩니다.
5. Add a math expression(수학 표현식 추가)을 선택합니다. 표현식을 표시하는 새 행이 나타납니다.
6. 새 행의 세부 정보 열에 수학 표현식을 입력하고 Enter 키를 누릅니다. 사용 가능한 함수 및 구문에 대한 자세한 내용은 지표 수학 구문 및 함수 (p. 50) 단원을 참조하십시오.

   이 표현식을 위한 공식의 일부로 지표 또는 다른 표현식 결과를 사용하려면 Id 열에 표시된 값을 사용합니다(예: m1+m2 또는 e1-MIN(e1)).

   Id 값은 변경할 수 없습니다. 숫자, 문자, 밑줄을 포함할 수 있으며, 소문자로 시작해야 합니다. Id의 값을 좀 더 의미 있는 이름으로 변경하면 경보 그래프를 이해하기가 더욱 쉬워집니다.
7. (선택 사항) 새 수학 표현식의 수식에 다른 수학 표현식의 지표 및 결과를 사용해 수학 표현식을 추가합니다.
8. 경보에 사용할 표현식이 있는 경우 페이지에서 다른 모든 표현식 및 지표 왼쪽에 있는 확인란을 선택 취소합니다. 경보에 사용할 표현식 옆의 확인란만 선택해야 합니다. 경보에 사용하려고 선택한 표현식은 단일 시계열을 생성하고, 그래프에 선을 하나만 표시해야 합니다. 그런 다음 지표 선택을 선택합니다.

   선택한 수학 표현식에 대한 그래프와 기타 정보가 표시된 Specify metric and conditions(지표 및 조건 지정) 페이지가 나타납니다.
9. Whenever ***`expression`*** is(표현식이 다음인 경우 항상)에서 표현식이 임계값보다 크거나, 작거나, 같아야 하는지 여부를 지정합니다. than...(비교 대상)에서 임계값을 지정합니다.
10. 추가 구성을 선택합니다. Datapoints to alarm(경보에 대한 데이터 포인트)에서 경보를 트리거하기 위해 평가 기간(데이터 포인트)이 `ALARM` 상태로 유지해야 하는 기간을 지정합니다. 두 값이 일치하는 경우 다수의 연속 기간이 위반되면 `ALARM` 상태가 되는 경보가 생성됩니다.

    N 중 M 경보를 생성하려면 두 번째 값에 지정한 값보다 낮은 값을 첫 번째 값에 지정합니다. 자세한 내용은 경보 평가 (p. 67) 단원을 참조하십시오.
11. Missing data treatment(누락 데이터 처리)에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 내용은 CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 (p. 68) 단원을 참조하십시오.
12. 다음을 선택합니다.
13. 알림에서 경보가 `ALARM` 상태, `OK` 상태 또는 `INSUFFICIENT_DATA` 상태일 때 알릴 SNS 주제를 선택합니다.

경보가 동일한 경보 상태 또는 다른 경보 상태에 대해 여러 개의 알림을 보내도록 설정하려면 알림 추가를 선택합니다.

경보에서 알림을 보내지 않게 하려면 제거를 선택합니다.

14. 경보가 Auto Scaling 또는 EC2 작업을 수행하도록 하려면 해당 버튼을 선택하고 경보 상태 및 수행할 작업을 선택합니다.
15. 마친 후에는 다음을 선택합니다.
16. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다. 다음을 선택합니다.
17. Preview and create(미리 보기 및 생성)에서 정보 및 조건이 원하는 내용인지 확인한 다음 경보 생성을 선택합니다.

대시보드에 경보를 추가할 수도 있습니다. 자세한 내용은 CloudWatch 대시보드에서 경보를 추가 또는 제거 (p. 26) 단원을 참조하십시오.

이전 사용자 인터페이스를 사용하여 수학 표현식을 기반으로 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택을 선택하고 다음 중 하나를 수행합니다.
   - 특정 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 오른쪽 지표 옆에 있는 확인란을 선택합니다.
   - 검색 상자에 지표 이름, 차원 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 결과 중 하나를 선택하고 지표 목록이 표시될 때까지 계속 진행합니다. 오른쪽 지표 옆의 확인란을 선택합니다.

   (선택 사항) 수학 표현식에서 사용할 다른 지표를 추가하려면 모든 지표에서 모두를 선택하고 특정 지표를 찾은 다음 해당 지표 옆에 있는 확인란을 선택합니다. 최대 10개의 지표를 추가할 수 있습니다.
4. 그래프로 표시된 지표를 선택합니다. 추가된 각 지표에 대해 다음을 수행합니다.

   a. 통계에서 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: **`p95.45`**)를 지정합니다.

   b. 기간에서 경보에 대한 평가 기간을 선택합니다. 경보를 평가할 때 각 기간이 하나의 데이터 포인트로 집계됩니다.

   또한 경보를 생성할 때 Y축 범례를 왼쪽 또는 오른쪽에 표시할지 여부를 선택할 수도 있습니다. 이러한 기본 설정은 경보를 생성하는 동안에만 사용됩니다.
5. Add a math expression(수학 표현식 추가)을 선택합니다. 표현식을 표시하는 새 행이 나타납니다.
6. 새 행의 세부 정보 열에 수학 표현식을 입력하고 Enter 키를 누릅니다. 사용 가능한 함수 및 구문에 대한 자세한 내용은 지표 수학 구문 및 함수 (p. 50) 단원을 참조하십시오.

   지표를 사용하거나 이 표현식을 위한 공식의 일부로 다른 표현식 결과를 사용하려면 Id 열에 표시된 값을 사용합니다. 예: m1+m2 또는 e1-MIN(e1).

   Id 값은 변경할 수 없습니다. 숫자, 문자, 밑줄을 포함할 수 있으며, 소문자로 시작해야 합니다. Id의 값을 좀 더 의미 있는 이름으로 변경하면 경보 그래프를 이해하기가 더욱 쉬워집니다.
7. (선택 사항) 새 수학 표현식의 수식에 다른 수학 표현식의 지표 및 결과를 사용해 수학 표현식을 추가합니다.
8. 경보에 사용할 표현식이 있는 경우 페이지에서 다른 모든 표현식 및 지표 왼쪽에 있는 확인란을 선택 취소합니다. 경보에 사용할 표현식 옆의 확인란만 선택해야 합니다. 경보에 사용하려고 선택한 표현식은 단일 시계열을 생성하고, 그래프에 선을 하나만 표시해야 합니다. 그런 다음 지표 선택을 선택합니다.
9. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다.
10. 다음 경우 항상에서 경보 조건을 지정합니다.

a. is:(조건:)에서 표현식 결과가 임계값보다 크거나 작아야 하는지 아니면 임계값과 같아야 하는지 지정하고 임계값을 지정합니다.

b. for:(기간:)에서 경보를 트리거하기 위해 평가 기간(데이터 포인트)이 `ALARM` 상태로 유지해야 하는 기간을 지정합니다. 처음에는 두 번째 값만 변경할 수 있습니다. 첫 번째 값은 입력한 두 번째 값에 맞춰 바뀝니다. 그러면 다수의 연속 기간이 위반될 경우 `ALARM` 상태가 되는 경보가 생성됩니다.

   N 중 M 경보를 생성하려면 연필 아이콘을 선택합니다. 그런 다음 M 숫자를 N 숫자와 다르게 변경할 수 있습니다. 자세한 내용은 경보 평가 (p. 67) 단원을 참조하십시오.

11. 추가 설정의 누락 데이터 처리에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 내용은 CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 (p. 68) 단원을 참조하십시오.
12. 작업에서 경보 트리거 시 수행할 작업의 유형을 선택합니다. 경보가 여러 작업을 수행하도록 하려면 +알림 또는 +AutoScaling 작업을 선택합니다. 작업을 한 가지 이상 지정합니다.
13. 경보 생성을 선택합니다.

복합 경보 생성

복합 경보는 다른 경보의 경보 상태를 감시하여 경보 상태를 결정하는 경보입니다.

복합 경보를 사용하면 경보 노이즈를 줄이는 데 도움이 될 수 있습니다. 상태 변경을 알리도록 복합 경보를 설정했지만 알림을 직접 보내지 않도록 기본 지표 경보를 설정한 경우, 복합 경보의 경보 상태가 변경될 때만 알림을 받게 됩니다. 예를 들어, CPU 사용률과 디스크 읽기 작업 모두를 기반으로 지표 경보를 생성하고, 조치를 취하지 않도록 이러한 경보를 지정할 수 있습니다. 그런 다음, 이러한 지표 경보가 모두 ALARM 상태에 있는 경우에만 ALARM 상태로 전환되고 사용자에게 알리는 복합 경보를 생성할 수 있습니다.

현재, 복합 경보를 통해 수행할 수 있는 유일한 경보 작업은 SNS 주제를 알리는 것입니다.

규칙 표현식

복합 경보에는 복합 경보의 상태를 결정하기 위해 평가할 다른 경보를 지정하는 규칙 표현식이 포함됩니다. 규칙 표현식에서 참조하는 각 경보에 대해, 해당 경보가 ALARM 상태, OK 상태, INSUFFICIENT_DATA 상태 중 어떤 상태여야 하는지를 지정하는 함수를 지정합니다. 연산자(AND, OR 및 NOT)를 사용하여 여러 함수를 단일 표현식으로 결합할 수 있습니다. 괄호를 사용하여 표현식에서 함수를 논리적으로 그룹화할 수 있습니다.

규칙 표현식은 지표 경보 및 기타 복합 경보를 모두 참조할 수 있습니다.

함수에는 다음이 포함될 수 있습니다.

- 명명된 경보가 ALARM 상태이면 `ALARM("alarm-name or alarm-ARN")`이 TRUE입니다.
- 명명된 경보가 OK 상태이면 `OK("alarm-name or alarm-ARN")`가 TRUE입니다.
- 명명된 경보가 INSUFFICIENT_DATA 상태이면 `INSUFFICIENT_DATA("alarm-name or alarm-ARN")`가 TRUE입니다.
- `TRUE`는 항상 TRUE로 평가됩니다.
- `FALSE`는 항상 FALSE로 평가됩니다.

TRUE 및 FALSE는 복잡한 `AlarmRule` 구조를 테스트하고 경보 동작을 테스트하는 데 유용합니다.

다음은 `AlarmRule`의 몇 가지 예입니다.

- `ALARM(CPUUtilizationTooHigh) AND ALARM(DiskReadOpsTooHigh)`은 CPUUtilizationTooHigh 및 DiskReadOpsTooHigh 경보가 모두 ALARM 상태인 경우에만 복합 경보가 ALARM 상태로 전환되도록 지정합니다.

- `ALARM(CPUUtilizationTooHigh) AND NOT ALARM(DeploymentInProgress)`은 CPUUtilizationTooHigh가 ALARM 상태이고 DeploymentInProgress가 ALARM 상태가 아닌 경우 경보가 ALARM 상태로 전환되도록 지정합니다. 이 예에서는 알려진 배포 기간 동안 경보 노이즈를 줄입니다.
- CPUUtilizationTooHigh OR DiskReadOpsTooHigh가 ALARM 상태이고 NetworkOutTooHigh가 OK 상태인 경우 `(ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)) AND OK(NetworkOutTooHigh)`는 ALARM 상태로 전환됩니다. 여기서 복합 경보를 사용하여 노이즈를 방지하는 또 다른 예를 볼 수 있습니다. 이 규칙은 알려진 네트워크 문제도 발생하는 경우 높은 CPU 또는 디스크 사용량에 대한 경보 작업을 통해 알림을 받지 않도록 합니다.

`AlarmRule`에서 최대 100개의 "하위" 경보를 지정할 수 있습니다. `AlarmRule` 표현식에는 최대 500개의 요소가 있을 수 있습니다. 요소는 하위 경보, TRUE 또는 FALSE 문 및 괄호입니다.

복합 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. 경보 목록에서, 새 복합 경보에서 참조하려는 각 기존 경보 옆에 있는 확인란을 선택합니다. 그런 다음 복합 경보 생성을 선택합니다.
4. 조건 상자에서 복합 경보에 사용할 규칙 표현식을 지정합니다. 처음에는 선택한 경보가 나열되고 OR 논리 연산자를 통해 조인됩니다. 이러한 각 경보에는 ALARM 상태가 지정되어 있습니다.

   생성 중인 복합 경보의 경보 조건을 수정할 수 있습니다.

   a. 나열된 각 기본 경보에 대해, 필요한 상태를 ALARM에서 OK 또는 INSUFICENT_DATA로 변경할 수 있습니다.

   b. 각 OR 연산자를 AND 또는 NOT으로 변경할 수 있습니다. 괄호를 추가하여 논리 연산자를 그룹화할 수도 있습니다.

   c. 복합 경보 조건에 더 많은 경보를 추가할 수 있습니다. 조건 상자에 현재 나열된 경보를 삭제할 수도 있습니다.

   예를 들어, 다음과 같은 조건을 지정하여 `CPUUtilizationTooHigh` OR `DiskReadOpsTooHigh`가 ALARM 상태이고 동시에 `NetworkOutTooHigh`가 OK 상태인 경우 ALARM 상태로 전환되는 복합 경보를 생성할 수 있습니다.

   ```
   (ALARM("CPUUtilizationTooHigh") OR
   ALARM("DiskReadOpsTooHigh")) AND
   OK("NetworkOutTooHigh")
   ```

5. 경보 조건이 만족스러우면 다음을 선택합니다.
6. 알림에서 복합 경보 상태가 변경될 때 알릴 SNS 주제를 선택합니다.

   경보를 통해 알림을 보내지 않거나 작업을 전혀 수행하지 않도록 하려면 제거를 선택합니다.

   경보가 동일한 경보 상태 또는 다른 경보 상태에 대해 여러 개의 알림을 보내도록 설정하려면 알림 추가를 선택합니다.
7. 마친 후에는 다음을 선택합니다.
8. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다. 다음을 선택합니다.
9. Preview and create(미리 보기 및 생성)에서 정보 및 조건이 원하는 내용인지 확인한 다음 경보 생성을 선택합니다.

Note

복합 경보 A가 복합 경보 B에 의존하고 복합 경보 B도 복합 경보 A에 의존하는 경우 복합 경보의 루프 또는 주기를 생성할 수 있습니다. 이 시나리오에서는 삭제하려는 경보에 의존하는 복합 경보가 여전히 있기 때문에 주기의 일부인 복합 경보를 삭제할 수 없습니다.
이러한 상황에서 벗어나려면 주기를 생성하는 종속성을 제거하도록 주기의 복합 경보 중 하나의 규칙을 변경하여 주기를 중단해야 합니다. 주기를 중단하기 위해 수행할 가장 간단한 변경 사항은 경보 중 하나의 `AlarmRule`을 `False`로 변경하는 것입니다.
또한 CloudWatch가 평가 경로에서 주기를 감지하면 복합 경보의 평가가 중지됩니다.

CloudWatch 경보 편집

기존 경보를 편집하고, 해당 경보가 사용하는 Amazon SNS 이메일 알림 목록을 업데이트할 수 있습니다.

CloudWatch에서 경보 사용자 인터페이스를 변경했습니다. 기본적으로 새 사용자 인터페이스가 표시되지만 이전 사용자 인터페이스로 돌아갈 수 있습니다. 이 주제에는 각각에 대한 단계가 포함되어 있습니다.

새 경보 사용자 인터페이스를 사용하여 경보를 편집하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Alarms를 선택합니다.
3. 경보 이름을 선택합니다.
4. 편집을 선택합니다.

   선택한 지표 및 통계에 대한 그래프와 기타 정보가 표시된 Specify metric and conditions(지표 및 조건 지정) 페이지가 나타납니다.
5. 지표를 변경하려면 편집을 선택하고 모든 지표 탭을 선택한 후 다음 중 하나를 수행합니다.
   - 원하는 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 원하는 지표 옆에 있는 확인란을 선택합니다.
   - 검색 상자에 지표 이름, 차원 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 그런 다음 결과 중 하나를 선택하고 지표 목록이 표시될 때까지 계속 진행합니다. 원하는 지표 옆의 확인란을 선택합니다.

   지표 선택을 선택합니다.
6. 경보의 다른 측면을 변경하려면 적절한 옵션을 선택합니다. 경보가 `ALARM` 상태가 되거나 누락된 데이터가 처리되는 방법을 변경하기 위해 위반해야 하는 데이터 포인트 수를 변경하려면 추가 구성을 선택합니다.
7. 다음을 선택합니다.
8. 알림, Auto Scaling action(Auto Scaling 작업) 및 EC2 작업에서 경보가 트리거될 때 수행한 동작을 선택적으로 편집합니다. 다음을 선택합니다.
9. 선택적으로 경보 설명을 변경합니다.

   기존 경보의 이름을 변경할 수 없습니다. 경보를 복사하고 새 경보에 다른 이름을 지정할 수 있습니다. 경보를 복사하려면 경보 목록에서 경보 이름 옆에 있는 확인란을 선택하고 작업, 복사를 선택합니다.
10. 다음을 선택합니다.
11. Preview and create(미리 보기 및 생성)에서 정보 및 조건이 원하는 내용인지 확인한 다음 Update alarm(경보 업데이트)을 선택합니다.

이전 경보 사용자 인터페이스를 사용하여 경보를 편집하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.

2. 탐색 창에서 Alarms를 선택합니다.
3. 경보를 선택하고 작업 및 수정을 선택합니다.
4. 경보 수정 페이지에서 필요에 따라 경보를 업데이트하고 변경 사항 저장을 선택합니다. 표현식, 지표, 기간 또는 통계를 수정하려면 먼저 화면의 상단 근처에서 편집을 선택합니다.

   기존 경보의 이름을 변경할 수 없습니다. 하지만 설명은 변경할 수 있습니다. 또는 경보를 복사하고 새 경보에 다른 이름을 지정할 수 있습니다. 경보를 복사하려면 해당 경보를 선택한 다음 작업, 복사를 선택합니다.

Amazon SNS 콘솔을 사용하여 생성된 이메일 알림 목록을 업데이트하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 주제를 선택한 다음, 알림 목록(주제)에 대한 ARN을 선택합니다.
3. 다음 중 하나를 수행하십시오.
   - 이메일 주소를 추가하려면 구독 생성을 선택합니다. 프로토콜에서 이메일을 선택합니다. 엔드포인트에 새 수신자의 이메일 주소를 입력합니다. 구독 생성을 선택합니다.
   - 이메일 주소를 제거하려면 구독 ID를 선택합니다. 기타 구독 작업과 구독 삭제를 선택합니다.
4. 주제 게시를 선택합니다.

이메일을 전송하는 CPU 사용률 경보 생성

경보 상태가 `OK`에서 `ALARM`으로 변경될 때 Amazon SNS를 사용하여 이메일 메시지를 전송하는 CloudWatch 경보를 생성할 수 있습니다.

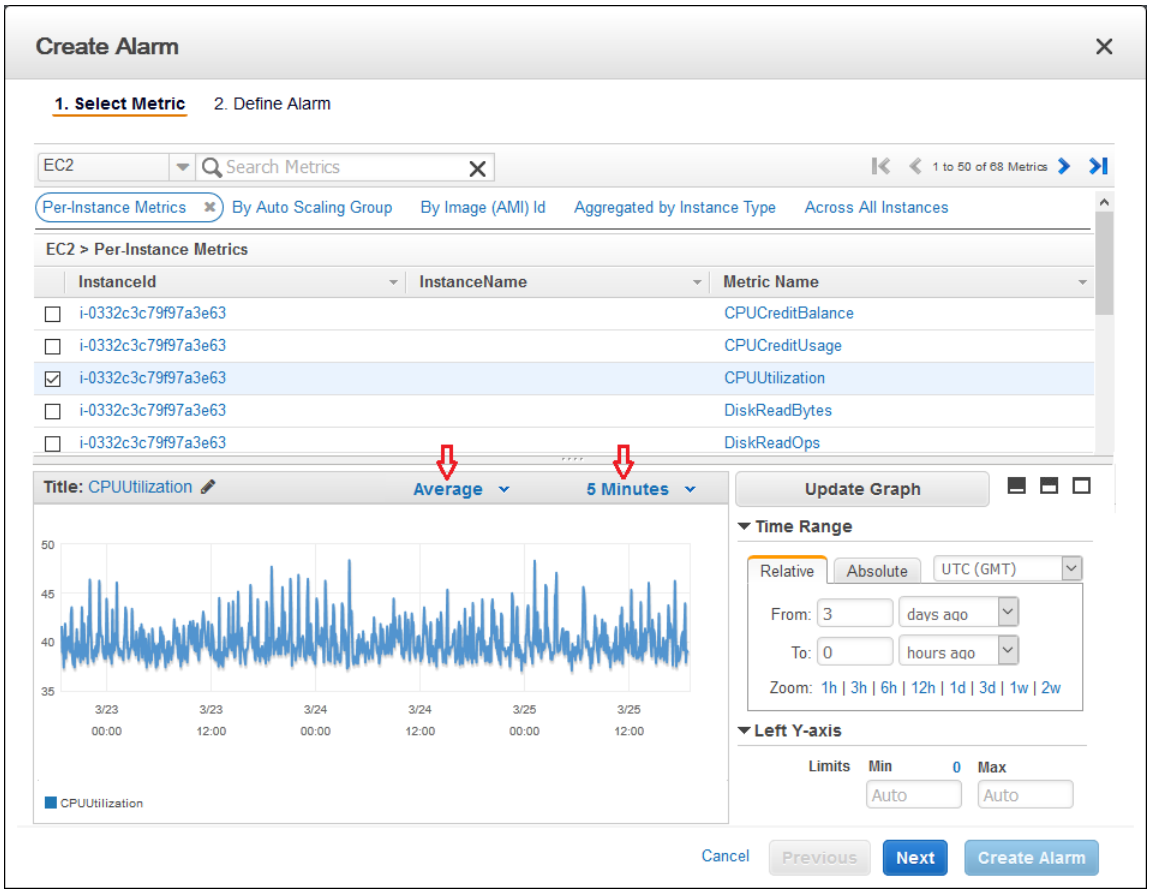
EC2 인스턴스의 평균 CPU 사용률이 지정된 기간 동안 연속해서 지정된 임계값을 초과하면 경보 상태가 `ALARM`으로 바뀝니다.

AWS Management 콘솔을 사용하여 CPU 사용률 경보 설정

다음 단계에 따라 AWS Management 콘솔을 사용해 CPU 사용량 경보를 생성합니다.

CPU 사용률을 기반으로 이메일을 보내는 경보를 만들려면

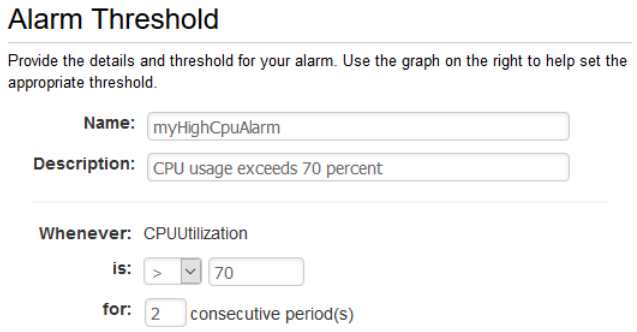
1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. EC2 지표에서 지표 범주(예: 인스턴스별 지표)를 선택합니다.
4. 아래와 같이 지표를 선택합니다.
   a. 해당 인스턴스와 CPUUtilization 지표가 있는 행을 선택합니다.
   b. 통계의 경우 평균을 선택하거나, 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: **`p95.45`**)를 지정합니다.
   c. 기간(예: **`5 minutes`**)을 선택합니다.
   d. 다음을 선택합니다.



5. 다음과 같이 경보를 정의합니다:

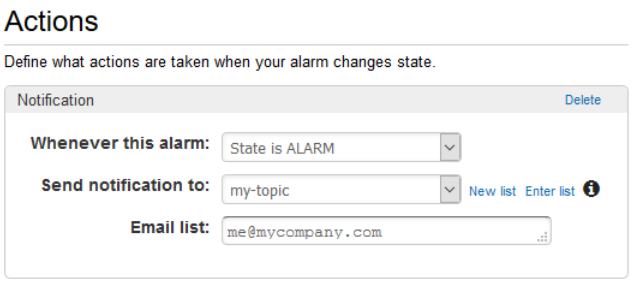
   a. 경보 임계값에 경보의 고유한 이름(예: **`myHighCpuAlarm`**)과 경보에 대한 설명(예: **`CPU usage exceeds 70 percent`**)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

   b. 다음 경우 항상의 조건에서 **>**를 선택하고 **70**을 입력합니다. 기간에 **2**를 입력합니다. 이는 CPU 사용량이 2회의 샘플링 기간 연속으로 70% 이상인 경우 트리거되는 경보를 지정합니다.



   c. 누락 데이터 요소가 인스턴스 다운을 나타내면 추가 설정 아래의 누락 데이터 처리에서 bad(breaching threshold)(불량(임계값 위반))를 선택합니다.

   d. 작업의 이 경보가 발생할 경우 항상에서 상태가 ALARM입니다를 선택합니다. 알림 보내기에 대해 기존 SNS 주제를 선택하거나 새로 만듭니다.



e. SNS 주제를 새로 생성하려면 새 목록을 선택합니다. 알림 보내기에 SNS 주제 이름(예: **myHighCpuAlarm**)을 입력하고 경보가 ALARM 상태로 변경될 때 알림을 보낼 이메일 주소 목록을 쉼표로 구분하여 이메일 목록에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 알림을 받으려면 먼저 구독을 확정해야 합니다.

f. 경보 생성을 선택합니다.

AWS CLI를 사용하여 CPU 사용률 경보 설정

다음 단계에 따라 AWS CLI를 사용해 CPU 사용량 경보를 생성합니다.

CPU 사용률을 기반으로 이메일을 보내는 경보를 만들려면

1. SNS 주제를 설정합니다. 자세한 내용은 Amazon SNS 알림 설정 (p. 72) 단원을 참조하십시오.
2. 아래와 같이 put-metric-alarm 명령을 사용하여 경보를 생성합니다.

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2 --statistic
 Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --
dimensions  Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-actions
 arn:aws:sns:us-east-1:111122223333:my-topic --unit Percent
```

3. set-alarm-state 명령으로 경보 상태를 강제로 변경하여 경보를 테스트합니다.

   a. 경보 상태를 INSUFFICIENT_DATA에서 OK로 변경합니다.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason "initializing"
    --state-value OK
   ```

   b. 경보 상태를 OK에서 ALARM으로 변경합니다.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason "initializing"
    --state-value ALARM
   ```

   c. 경보에 대한 이메일 알림을 받았음을 확인합니다.

이메일을 전송하는 로드 밸런서 지연 경보 생성

Amazon SNS 알림을 설정하고 Classic Load Balancer에서 지연 시간이 100ms를 초과하는지 모니터링하는 경보를 생성할 수 있습니다.

AWS Management 콘솔을 사용하여 지연 시간 경보 설정

다음 단계에 따라 AWS Management 콘솔을 사용해 로드 밸런서 지연 시간 경보를 생성합니다.

이메일을 전송하는 로드 밸런서 지연 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 범주별 CloudWatch 지표에서 ELB 지표 범주를 선택합니다.
4. Classic Load Balancer와 지연 시간 지표가 있는 행을 선택합니다.
5. 통계의 경우 평균을 선택하거나, 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: **p95.45**)를 지정합니다.
6. 기간의 경우 1분을 선택합니다.
7. 다음을 선택합니다.
8. 경보 임계값에 경보의 고유한 이름(예: **myHighCpuAlarm**)과 경보에 대한 설명(예: **Alarm when Latency exceeds 100s**)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.
9. 다음 경우 항상의 조건에서 **>**를 선택하고 **0.1**을 입력합니다. 기간에 **3**을 입력합니다.
10. 누락 데이터 포인트가 경보 상태 변경을 트리거하지 않도록 추가 설정의 누락 데이터 처리에서 무시(경보 상태 유지)를 선택합니다.

    경보가 적정 수의 데이터 샘플이 있는 상황만 평가하도록 샘플이 부족한 백분위수에서 ignore (maintain alarm state)(무시(경보 상태 유지))를 선택합니다.
11. 작업의 이 경보가 발생할 경우 항상에서 상태가 ALARM입니다를 선택합니다. 알림 보내기에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

    SNS 주제를 생성하려면 새 목록을 선택합니다. 알림 보내기에 SNS 주제 이름(예: **myHighCpuAlarm**)을 입력하고 경보가 ALARM 상태로 변경될 때 알림을 보낼 이메일 주소 목록을 쉼표로 구분하여 이메일 목록에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 알림을 받으려면 먼저 구독을 확정해야 합니다.
12. 경보 생성을 선택합니다.

AWS CLI를 사용하여 지연 시간 경보 설정

다음 단계에 따라 AWS CLI를 사용해 로드 밸런서 지연 시간 경보를 생성합니다.

이메일을 전송하는 로드 밸런서 지연 경보를 생성하려면

1. SNS 주제를 설정합니다. 자세한 내용은 Amazon SNS 알림 설정 (p. 72) 단원을 참조하십시오.
2. 아래와 같이 put-metric-alarm 명령을 사용하여 경보를 생성합니다.

```
aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description "Alarm
 when Latency exceeds 100s" --metric-name Latency --namespace AWS/ELB --statistic
 Average --period 60 --threshold 100 --comparison-operator GreaterThanThreshold --
dimensions Name=LoadBalancerName,Value=my-server --evaluation-periods 3 --alarm-actions
 arn:aws:sns:us-east-1:111122223333:my-topic --unit Seconds
```

3. set-alarm-state 명령으로 경보 상태를 강제로 변경하여 경보를 테스트합니다.

   a. 경보 상태를 INSUFFICIENT_DATA에서 OK로 변경합니다.

```
aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "initializing" --
state-value OK
```

b. 경보 상태를 OK에서 ALARM으로 변경합니다.

```
aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "initializing" --
state-value ALARM
```

c. 경보에 대한 이메일 알림을 받았음을 확인합니다.

이메일을 전송하는 스토리지 처리량 경보 생성

SNS 알림을 설정하고 Amazon EBS의 처리량이 100MB를 초과할 때 이메일을 전송하는 경보를 생성할 수 있습니다.

AWS Management 콘솔을 사용하여 스토리지 처리량 경보 설정

다음 단계에 따라 AWS Management 콘솔을 사용하여 Amazon EBS 처리량을 기반으로 하는 경보를 만듭니다.

이메일을 보내는 스토리지 처리량 경보를 만들려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. EBS 지표에서 지표 범주를 선택합니다.
4. 볼륨과 VolumeWriteBytes 지표가 있는 행을 선택합니다.
5. 통계의 경우 평균을 선택합니다. 기간의 경우 5분을 선택합니다. 다음을 선택합니다.
6. 경보 임계값에 경보의 고유한 이름(예: **myHighWriteAlarm**)과 경보에 대한 설명(예: **VolumeWriteBytes exceeds 100,000 KiB/s**)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.
7. 다음 경우 항상의 조건에서 **>**를 선택하고 **100000**을 입력합니다. 기간에 연속 기간으로 **15**를 입력합니다.

   경보 미리 보기 아래에 임계값이 그래픽으로 표시됩니다.
8. 누락 데이터 포인트가 경보 상태 변경을 트리거하지 않도록 추가 설정의 누락 데이터 처리에서 무시(경보 상태 유지)를 선택합니다.
9. 작업의 이 경보가 발생할 경우 항상에서 상태가 ALARM입니다를 선택합니다. 알림 보내기에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

   SNS 주제를 생성하려면 새 목록을 선택합니다. 알림 보내기에 SNS 주제 이름(예: **myHighCpuAlarm**)을 입력하고 경보가 ALARM 상태로 변경될 때 알림을 보낼 이메일 주소 목록을 쉼표로 구분하여 이메일 목록에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.
10. 경보 생성을 선택합니다.

AWS CLI를 사용하여 스토리지 처리량 경보 설정

다음 단계에 따라 AWS CLI를 사용하여 Amazon EBS 처리량을 기반으로 하는 경보를 만듭니다.

이메일을 보내는 스토리지 처리량 경보를 만들려면

1. SNS 주제를 생성합니다. 자세한 내용은 Amazon SNS 알림 설정 (p. 72) 단원을 참조하십시오.
2. 경보를 만듭니다.

```
aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description "Alarm when
 EBS volume exceeds 100MB throughput" --metric-name VolumeReadBytes --namespace AWS/
EBS --statistic Average --period 300 --threshold 100000000 --comparison-operator
 GreaterThanThreshold --dimensions Name=VolumeId,Value=my-volume-id --evaluation-
periods 3 --alarm-actions arn:aws:sns:us-east-1:111122223333:my-alarm-topic --
insufficient-data-actions arn:aws:sns:us-east-1:111122223333:my-insufficient-data-topic
```

3. set-alarm-state 명령으로 경보 상태를 강제로 변경하여 경보를 테스트합니다.

   a. 경보 상태를 INSUFFICIENT_DATA에서 OK로 변경합니다.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason "initializing"
    --state-value OK
   ```

   b. 경보 상태를 OK에서 ALARM으로 변경합니다.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason "initializing"
    --state-value ALARM
   ```

   c. 경보 상태를 ALARM에서 INSUFFICIENT_DATA로 변경합니다.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason "initializing"
    --state-value INSUFFICIENT_DATA
   ```

   d. 경보에 대한 이메일 알림을 받았음을 확인합니다.

인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성

Amazon CloudWatch 경보 작업을 사용하면 EC2 인스턴스를 자동으로 중지, 종료, 재부팅 또는 복구하는 경보를 만들 수 있습니다. 인스턴스를 더 이상 실행할 필요가 없을 때 중지 또는 종료 작업을 사용하여 비용을 절약할 수 있습니다. 재부팅 및 복구 작업을 사용하면 시스템 장애가 발생할 경우 인스턴스를 자동으로 재부팅하거나 새로운 하드웨어로 인스턴스를 복구할 수 있습니다.

인스턴스를 자동으로 중지하거나 종료해야 하는 경우는 매우 다양합니다. 예를 들어 일정 기간 동안 실행한 다음 작업을 완료하는 일괄 급여 처리 작업 또는 과학적 컴퓨팅 작업 전용 인스턴스가 있을 수 있습니다. 이러한 인스턴스를 유휴 상태로 유지하여 비용이 발생하도록 하는 대신 중지하거나 종료하면 비용을 절감할 수 있습니다. 경보 작업 중지와 종료 간의 주요 차이는 나중에 다시 실행해야 하는 경우 중지된 인스턴스는 쉽게 다시 시작할 수 있다는 점입니다. 또한 동일한 인스턴스 ID 및 루트 볼륨을 유지할 수 있습니다. 그러나 종료된 인스턴스를 다시 시작할 수는 없습니다. 대신, 새 인스턴스를 시작해야 합니다.

Amazon CloudWatch에서 제공하는 기본 및 세부 모니터링 지표(AWS/EC2 네임스페이스)을 비롯한 인스턴스 지표당 Amazon EC2 및 InstanceId 값이 실행 중인 유효한 Amazon EC2 인스턴스를 참조하는 경우 "InstanceId=" 차원을 포함하는 모든 사용자 지정 지표에 대해 설정된 경보에 중지, 종료 재부팅 또는 복구 작업을 추가할 수 있습니다.

인스턴스를 재부팅, 중지 또는 종료할 수 있는 CloudWatch 경보 작업을 설정하려면 서비스 연결 IAM 역할, AWSServiceRoleForCloudWatchEvents을 사용해야 합니다. AWSServiceRoleForCloudWatchEvents IAM 역할은 AWS가 사용자를 대신하여 경보 작업을 수행하도록 해줍니다.

CloudWatch 이벤트에 대한 서비스 연결 역할을 만들려면 다음 명령을 사용합니다.

```
aws iam create-service-linked-role --aws-service-name events.amazonaws.com
```

콘솔 지원

CloudWatch 콘솔 또는 Amazon EC2 콘솔을 사용하여 경보를 만들 수 있습니다. 이 문서의 절차는 CloudWatch 콘솔을 사용합니다. Amazon EC2 콘솔을 사용하는 절차는 Linux 인스턴스용 Amazon EC2 사용 설명서의 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성을 참조하십시오.

권한

AWS Identity and Access Management(IAM) 계정을 사용하여 경보를 생성하거나 수정할 경우 다음 권한이 있어야 합니다.

- `iam:CreateServiceLinkedRole`, `iam:GetPolicy`, `iam:GetPolicyVersion`, 및 `iam:GetRole` - Amazon EC2 작업을 수반하는 모든 경보 —
- `ec2:DescribeInstanceStatus` 및 `ec2:DescribeInstances` - Amazon EC2 인스턴스 상태 지표에 대한 모든 경보 —
- `ec2:StopInstances` - 중지 작업을 수반하는 경보 —
- `ec2:TerminateInstances` - 종료 작업을 수반하는 경보 —
- 복구 작업을 수반하는 경보는 권한 제한이 없습니다.

읽기/쓰기 권한이 Amazon CloudWatch에 대해서는 있지만 Amazon EC2에 대해서는 없는 경우 경보를 만들 수는 있지만 인스턴스에 대해 중지 또는 종료 작업이 수행되지 않습니다. 그러나 이후에 연결된 Amazon EC2 API 작업을 사용하도록 권한을 부여 받은 경우 앞서 만든 경보 작업이 수행됩니다. 자세한 내용은 IAM 사용 설명서에서 권한 및 정책을 참조하십시오.

IAM 역할을 사용하여 경보 작업으로 인스턴스를 중지하거나 종료하거나 재부팅하려면 EC2AWSServiceRoleForCloudWatchEvents 역할만 사용할 수 있습니다. 다른 IAM 역할은 지원되지 않습니다. 그러나 경보 상태는 계속 표시되고 Amazon SNS 알림 또는 Amazon EC2 Auto Scaling 정책 등의 다른 작업을 수행할 수 있습니다.

목차

- Amazon CloudWatch 경보에 중지 작업 추가 (p. 90)
- Amazon CloudWatch 경보에 종료 작업 추가 (p. 91)
- Amazon CloudWatch 경보에 재부팅 작업 추가 (p. 92)
- Amazon CloudWatch 경보에 복구 작업 추가 (p. 93)
- 트리거된 경보 및 작업 기록 보기 (p. 94)

Amazon CloudWatch 경보에 중지 작업 추가

특정 임계값에 도달한 경우 Amazon EC2 인스턴스를 중지하는 경보를 만들 수 있습니다. 예를 들어 개발 또는 테스트 인스턴스를 실행한 후 종료하는 것을 잊을 수 있습니다. 24시간 동안 평균 CPU 사용률이 10% 아래로 떨어지는 경우 즉, 유휴 상태로 더 이상 사용되지 않는 경우 트리거되는 경보를 만들 수 있습니다. 필요에 맞춰 임계값 및 기간을 조정할 수 있습니다. 또한 경보가 트리거되면 이메일을 받을 수 있도록 SNS 알림을 추가할 수 있습니다.

Amazon Elastic Block Store 볼륨을 루트 장치로 사용하는 Amazon EC2 인스턴스는 중지하거나 종료할 수 있지만 인스턴스 스토어를 루트 장치로 사용하는 인스턴스는 종료만 할 수 있습니다.

Amazon CloudWatch 콘솔을 사용하여 유휴 인스턴스를 중지하는 경보를 만들려면

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

3. 지표 선택 단계에서 다음을 수행합니다.

   a. EC2 지표에서 인스턴스별 지표를 선택합니다.
   b. 해당 인스턴스와 CPUUtilization 지표가 있는 행을 선택합니다.
   c. 통계의 경우 평균을 선택합니다.
   d. 기간(예: **1 Hour**)을 선택합니다.
   e. 다음을 선택합니다.
4. 경보 정의 단계에서 다음을 수행합니다.

   a. 경보 임계값에 경보의 고유 이름(예: Stop EC2 instance)과 경보에 대한 설명(예: CPU 유휴 시간이 너무 길어서 EC2 인스턴스 중지)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.
   b. Whenever의 is에서 **<**를 선택하고 **10**을 입력합니다. 기간에 연속 기간으로 **24**를 입력합니다.

      경보 미리 보기 아래에 임계값이 그래픽으로 표시됩니다.
   c. 알림의 알림 보내기에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

      SNS 주제를 생성하려면 새 목록을 선택합니다. 알림 보내기에 SNS 주제의 이름을 입력합니다(예: Stop_EC2_Instance). **ALARM** 상태로 경보가 변경되면 알릴 이메일 주소를 쉼표로 구분하여 Email list(이메일 목록)에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.
   d. EC2 작업을 선택합니다.
   e. 이 경보가 발생할 경우 항상에서 상태가 ALARM입니다를 선택합니다. 이 작업을 수행에서 인스턴스 중지를 선택합니다.
   f. 경보 생성을 선택합니다.

Amazon CloudWatch 경보에 종료 작업 추가

인스턴스에 대해 종료 보호가 비활성화되어 있는 경우에 한해서 특정 임계값에 도달한 경우 EC2 인스턴스를 자동으로 종료하는 경보를 만들 수 있습니다. 예를 들어 인스턴스의 작업 완료 후 해당 인스턴스가 다시 필요 없는 경우 인스턴스를 종료하려고 할 수 있습니다. 나중에 인스턴스를 사용하려는 경우에는 종료하지 말고 중지해야 합니다. 인스턴스에 대한 종료 보호 활성화 및 비활성화에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 인스턴스에 대한 종료 보호 활성화를 참조하십시오.

Amazon CloudWatch 콘솔을 사용하여 유휴 인스턴스를 종료하는 경보를 만들려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택 단계에서 다음을 수행합니다.

   a. EC2 지표에서 인스턴스별 지표를 선택합니다.
   b. 해당 인스턴스와 CPUUtilization 지표가 있는 행을 선택합니다.
   c. 통계의 경우 평균을 선택합니다.
   d. 기간(예: **1 Hour**)을 선택합니다.
   e. 다음을 선택합니다.
4. 경보 정의 단계에서 다음을 수행합니다.

   a. 경보 임계값에 경보의 고유 이름(예: Terminate EC2 instance)과 경보에 대한 설명(예: CPU 유휴 시간이 너무 길어서 EC2 인스턴스 종료)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.
   b. Whenever의 is에서 **<**를 선택하고 **10**을 입력합니다. 기간에 연속 기간으로 **24**를 입력합니다.

      경보 미리 보기 아래에 임계값이 그래픽으로 표시됩니다.
   c. 알림의 알림 보내기에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

SNS 주제를 생성하려면 새 목록을 선택합니다. 알림 보내기에 SNS 주제의 이름을 입력합니다(예: Terminate_EC2_Instance). `ALARM` 상태로 경보가 변경되면 알릴 이메일 주소를 쉼표로 구분하여 Email list(이메일 목록)에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.

d. EC2 작업을 선택합니다.

e. 이 경보가 발생할 경우 항상에서 상태가 ALARM입니다를 선택합니다. 이 작업을 수행에서 인스턴스 종료를 선택합니다.

f. 경보 생성을 선택합니다.

Amazon CloudWatch 경보에 재부팅 작업 추가

Amazon EC2 인스턴스를 모니터링하고 인스턴스를 자동으로 재부팅하는 Amazon CloudWatch 경보를 만들 수 있습니다. 재부팅 경보 작업은 인스턴스 상태 확인 오류(복구 경보 작업은 시스템 상태 확인 오류에 적합)에 권장됩니다. 인스턴스 재부팅은 운영 체제 재부팅과 같습니다. 대부분의 경우 인스턴스를 재부팅하는 데는 몇 분 밖에 걸리지 않습니다. 인스턴스를 재부팅하는 경우 동일한 물리적 호스트에 남아 있으므로 퍼블릭 DNS 이름, 프라이빗 IP 주소 및 인스턴스 스토어 볼륨의 모든 데이터가 유지됩니다.

인스턴스를 재부팅해도 인스턴스를 중지했다가 다시 시작할 때와는 달리 새 인스턴스 청구 시간이 시작되지 않습니다. 인스턴스 재부팅에 대한 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서에서 인스턴스 재부팅을 참조하십시오.

> **Important**
>
> 재부팅과 복원 작업 간에 경합 상태가 발생하지 않도록 하려면 재부팅 경보와 복원 경보에 동일한 평가 기간을 설정하지 마십시오. 재부팅 경보를 각각 1분의 평가 기간 3회로 설정하는 것이 좋습니다.

Amazon CloudWatch 콘솔을 사용하여 인스턴스를 재부팅하는 경보를 만들려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택 단계에서 다음을 수행합니다.

   a. EC2 지표에서 인스턴스별 지표를 선택합니다.

   b. 해당 인스턴스와 StatusCheckFailed_Instance 지표가 있는 행을 선택합니다.

   c. 통계의 경우 최소를 선택합니다.

   d. 기간(예: **`1 Minute`**)을 선택하고 다음을 선택합니다.
4. 경보 정의 단계에서 다음을 수행합니다.

   a. 경보 임계값에 경보의 고유 이름(예: Reboot EC2 instance)과 경보에 대한 설명(예: CPU 유휴 시간이 너무 길어서 EC2 인스턴스 재부팅)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

   b. Whenever의 is에서 **>**를 선택하고 **0**을 입력합니다. 기간에 연속 기간으로 **3**를 입력합니다.

   경보 미리 보기 아래에 임계값이 그래픽으로 표시됩니다.

   c. 알림의 알림 보내기에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

   SNS 주제를 생성하려면 새 목록을 선택합니다. 알림 보내기에 SNS 주제의 이름을 입력합니다(예: Reboot_EC2_Instance). `ALARM` 상태로 경보가 변경되면 알릴 이메일 주소를 쉼표로 구분하여 Email list(이메일 목록)에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.

   d. EC2 작업을 선택합니다.

   e. 이 경보가 발생할 경우 항상에서 상태가 ALARM입니다를 선택합니다. 이 작업을 수행에서 인스턴스 재부팅을 선택합니다.

f. 경보 생성을 선택합니다.

Amazon CloudWatch 경보에 복구 작업 추가

사용자는 Amazon EC2 인스턴스를 모니터링하고 기본 하드웨어 장애나 복구에 AWS 개입이 필요한 문제로 인해 인스턴스가 손상된 경우 인스턴스를 자동으로 복구하는 Amazon CloudWatch 경보를 만들 수 있습니다. 종료한 인스턴스는 복구할 수 없습니다. 복구된 인스턴스는 인스턴스 ID, 프라이빗 IP 주소, 탄력적 IP 주소 및 모든 인스턴스 메타데이터를 포함하여 원본 인스턴스와 동일합니다.

`StatusCheckFailed_System` 경보가 트리거되고 복구 작업이 시작되는 경우 경보를 생성하고 복구 작업을 연결할 때 선택한 Amazon SNS 주제로 통지됩니다. 인스턴스 복구 중에 인스턴스를 재부팅할 때 인스턴스가 마이그레이션되고 모든 인 메모리 데이터가 손실됩니다. 프로세스가 완료되면 해당 경보를 위해 구성해 둔 SNS 주제로 정보가 게시됩니다. 이 SNS 주제에 가입되어 있는 사람은 누구나 복구 시도 상태와 세부 지침이 포함된 이메일 알림을 받게 됩니다. 복구된 인스턴스에서 인스턴스를 재부팅하라는 메시지가 나타납니다.

복구 작업은 `StatusCheckFailed_Instance`가 아닌 `StatusCheckFailed_System`을 통해서만 사용할 수 있습니다.

시스템 상태 확인이 실패하게 되는 문제의 예를 들면 다음과 같습니다.

- 네트워크 연결 끊김
- 시스템 전원 중단
- 물리적 호스트의 소프트웨어 문제
- 네트워크 연결성에 영향을 주는 물리적 호스트의 하드웨어 문제

복구 작업은 다음에 대해서만 지원됩니다.

- A1, C3, C4, C5, C5n, M3, M4, M5, M5a, P3, R3, R4, R5, R5a, T2, T3, X1 및 X1e 인스턴스 유형
- VPC의 인스턴스
- `default` 또는 `dedicated` 인스턴스 테넌시가 있는 인스턴스
- Amazon EBS 볼륨만 사용하는(인스턴스 스토어 볼륨을 구성하지 않는) 인스턴스

인스턴스에 퍼블릭 IPv4 주소가 있는 경우 복구 후에도 해당 퍼블릭 IP 주소를 유지합니다.

> **Important**
>
> 재부팅과 복원 작업 간에 경합 상태가 발생하지 않도록 하려면 재부팅 경보와 복원 경보에 동일한 평가 기간을 설정하지 마십시오. 복원 경보는 각각 1분의 평가 기간 2회로 설정하고 재부팅 경보는 각각 1분의 평가 기간 3회로 설정하는 것이 좋습니다.

Amazon CloudWatch 콘솔을 사용하여 인스턴스를 복구하는 경보를 만들려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보, 경보 생성을 선택합니다.
3. 지표 선택 단계에서 다음을 수행합니다.
   a. EC2 지표에서 인스턴스별 지표를 선택합니다.
   b. 해당 인스턴스와 StatusCheckFailed_System 지표가 있는 행을 선택합니다.
   c. 통계의 경우 최소를 선택합니다.
   d. 기간(예: **`1 Minute`**)을 선택합니다.

Important

재부팅과 복원 작업 간에 경합 상태가 발생하지 않도록 하려면 재부팅 경보와 복원 경보에 동일한 평가 기간을 설정하지 마십시오. 복구 경보는 각각 1분의 평가 기간 2회로 설정하는 것이 좋습니다.

e. 다음을 선택합니다.

4. 경보 정의 단계에서 다음을 수행합니다.

   a. 경보 임계값에 경보의 고유 이름(예: Recover EC2 instance)과 경보에 대한 설명(예: 상태 확인 실패 시 EC2 인스턴스 복구)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.

   b. Whenever의 is에서 **>**를 선택하고 **0**을 입력합니다. 기간에 연속 기간으로 **2**를 입력합니다.

   c. 알림의 알림 보내기에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

      SNS 주제를 생성하려면 새 목록을 선택합니다. 알림 보내기에 SNS 주제의 이름을 입력합니다(예: Recover_EC2_Instance). `ALARM` 상태로 경보가 변경되면 알릴 이메일 주소를 쉼표로 구분하여 Email list(이메일 목록)에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.

   d. EC2 작업을 선택합니다.

   e. 이 경보가 발생할 경우 항상에서 상태가 ALARM입니다를 선택합니다. 이 작업을 수행에서 인스턴스 복구를 선택합니다.

   f. 경보 생성을 선택합니다.

트리거된 경보 및 작업 기록 보기

Amazon CloudWatch 콘솔에서 경보 및 작업 기록을 볼 수 있습니다. Amazon CloudWatch에서는 지난 2주간의 경보 및 작업 기록을 보관합니다.

트리거된 경보 및 작업 기록을 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보를 선택한 후 특정 경보를 선택합니다.
3. 가장 최근의 상태 변화와 함께 시간 및 지표 값을 보려면 세부 정보를 선택합니다.
4. 최신 기록 항목을 보려면 내역을 선택합니다.

예상 AWS 요금을 모니터링하기 위한 결제 경보 생성

Amazon CloudWatch를 사용하여 예상 AWS 요금을 모니터링할 수 있습니다. AWS 계정에 대한 예상 요금 모니터링을 활성화하면 예상 요금이 계산되어 지표 데이터로서 하루에 여러 번 CloudWatch에 전송됩니다.

결제 지표 데이터는 미국 동부(버지니아 북부) 리전에 저장되며 전 세계 요금을 반영합니다. 이 데이터에는 사용한 AWS의 모든 서비스에 대한 예상 요금과 전반적인 총 AWS 예상 요금이 들어 있습니다.

계정 결제가 지정한 임계값을 초과하면 경보가 작동합니다. 실제 결제가 임계값을 초과할 때만 경보가 작동합니다. 해당 시점까지의 월 사용량을 기준으로 추정하지 않습니다.

요금이 임계값을 초과한 시점에 결제 경보를 생성할 경우 경보가 즉시 `ALARM` 상태가 됩니다.

작업

- 결제 알림 활성화 (p. 95)

- 결제 경보 생성 (p. 95)
- 결제 경보 삭제 (p. 96)

결제 알림 활성화

예상 요금에 대한 경보를 생성할 수 있으려면 먼저 결제 알림을 활성화해야 합니다. 그래야만 예상되는 AWS 요금을 모니터링하고 결제 지표 데이터를 사용하여 경보를 생성할 수 있습니다. 결제 알림을 활성화하고 나면 데이터 수집을 비활성화할 수 없습니다. 그러나 생성된 모든 결제 경보를 삭제할 수는 있습니다.

결제 알림을 처음 활성화하고 나서 결제 데이터를 확인하고 결제 경보를 설정할 수 있기까지 약 15분 정도의 시간이 걸립니다.

요구 사항

- 계정 루트 사용자 자격 증명을 사용하여 로그인하거나 결제 정보를 볼 수 있는 권한이 있는 IAM 사용자로 로그인해야 합니다.
- 통합 결제 계정의 경우 결제 계정으로 로그인하면 연결된 각 계정에 대한 결제 데이터를 찾을 수 있습니다. 통합 계정에 대해서뿐만 아니라 연결된 각 계정에 대한 서비스별 총 예상 요금 및 예상 요금에 대한 결제 데이터를 볼 수 있습니다.

예상 요금 모니터링을 활성화하려면

1. https://console.aws.amazon.com/billing/에서 Billing and Cost Management 콘솔을 엽니다.
2. 탐색 창에서 결제 기본 설정을 선택합니다.
3. 결제 알림 받기를 선택합니다.
4. 기본 설정 저장을 선택합니다.

결제 경보 생성

결제 알림을 활성화했으면 결제 경보를 생성할 수 있습니다. 이 절차를 통해 AWS에 대한 예상 요금이 지정된 임계값을 초과할 때 이메일 메시지를 전송하는 경보를 생성합니다.

> **Note**
>
> 이 절차는 고급 옵션을 사용합니다. 간단한 옵션 사용에 대한 자세한 내용은 CloudWatch를 사용하여 예상 요금 모니터링의 결제 경보 생성 (p. 352)을 참조하십시오.

CloudWatch 콘솔을 사용하여 결제 경보를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 미국 동부(버지니아 북부)로 변경합니다. 결제 지표 데이터는 이 리전에 저장되어 전 세계 요금을 나타냅니다.
3. 탐색 창에서 경보, 경보 생성을 선택합니다.
4. 지표 선택을 선택합니다. 모든 지표 탭에서 결제, 예상 요금 합계를 선택합니다.
5. EstimatedCharges 옆에 있는 확인란을 선택하고 지표 선택을 선택합니다.
6. 조건에서 정적을 선택합니다.
7. Whenever EstimatedCharges is(EstimatedCharges가 다음인 경우 항상)에서 Greater(보다 큼)를 선택합니다.
8. than(기준)에 경보를 트리거하기 위해 초과해야 하는 금액(예: **200**)을 입력합니다.

> **Note**
>
> 미리 보기 그래프에 당월 요금이 표시됩니다.

9. 다음을 선택합니다.
10. 알림에서 경보가 `ALARM` 상태일 때 알릴 SNS 주제를 선택합니다.

    경보가 동일한 경보 상태 또는 다른 경보 상태에 대해 여러 개의 알림을 보내도록 설정하려면 알림 추가를 선택합니다.
11. 마친 후에는 다음을 선택합니다.
12. 경보 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다. 다음을 선택합니다.
13. Preview and create(미리 보기 및 생성)에서 정보 및 조건이 원하는 내용인지 확인한 다음 경보 생성을 선택합니다.

결제 경보 삭제

결제 경보가 더 이상 필요하지 않다면 삭제할 수 있습니다.

결제 경보를 삭제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 미국 동부(버지니아 북부)로 변경합니다. 결제 지표 데이터는 이 리전에 저장되며 전 세계 요금을 반영합니다.
3. 탐색 창에서 Alarms를 선택합니다.
4. 경보 옆의 확인란을 선택하고 작업, 삭제를 차례로 선택합니다.
5. 확인 메시지가 나타나면 예, 삭제를 선택합니다.

Amazon EC2 Auto Scaling 경보 숨기기

AWS Management 콘솔에서 경보를 확인할 때, Amazon EC2 Auto Scaling와 관련된 경보를 숨길 수 있습니다. 이 기능은 AWS Management 콘솔에서만 사용할 수 있습니다.

Amazon EC2 Auto Scaling 경보를 일시적으로 숨기려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 경보를 선택한 후 Hide all AutoScaling alarms(모든 AutoScaling 경보 숨기기)를 선택합니다.

카나리아 사용

CloudWatch Synthetics is in open preview. The preview is currently available in the following Regions:

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(스톡홀름)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)

The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at cw-synthetics@amazon.com

Amazon CloudWatch Synthetics를 사용하면 카나리아를 생성하여 엔드포인트와 API를 모니터링할 수 있습니다. 카나리아는 고객과 동일한 경로를 따르고 동일한 작업을 수행하는 구성 가능한 스크립트입니다. 이를 통해 고객의 관점에서 고객의 경험과 회사의 서비스 가용성을 외부에서 볼 수 있습니다.

카나리아를 한 번 실행하거나 정기적으로 실행할 수 있습니다. 예약된 카나리아는 분당 한 번씩 하루 24시간 실행할 수 있습니다. 카나리아는 엔드포인트의 가용성과 지연 시간을 확인하고 로드 시간 데이터 및 UI의 스크린샷을 저장할 수 있습니다.

카나리아 생성 및 실행 시 고려해야 할 보안 문제에 대한 자세한 내용은 Synthetics 카나리아에 대한 보안 고려 사항 (p. 384) 단원을 참조하십시오.

CloudWatch Synthetics는 CloudWatch ServiceLens와 완벽하게 통합됩니다. ServiceLens는 CloudWatch와 AWS X-Ray를 통합하여 서비스에 대한 전체적인 보기를 제공함으로써 보다 효율적으로 성능 병목 현상을 파악하고 영향을 받는 사용자를 식별할 수 있게 합니다. CloudWatch Synthetics를 사용하여 생성한 카나리아는 ServiceLens 서비스 맵에 나타납니다. ServiceLens에 대한 자세한 내용은 ServiceLens를 사용하여 애플리케이션의 상태 모니터링 (p. 106) 단원을 참조하십시오.

주제

- CloudWatch 카나리아에 필요한 역할 및 권한 (p. 98)
- 카나리아 생성 (p. 101)
- 카나리아 통계 및 세부 정보 보기 (p. 103)
- 카나리아 삭제 (p. 104)

CloudWatch 카나리아에 필요한 역할 및 권한

CloudWatch Synthetics is in open preview. The preview is currently available in the following Regions:

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(스톡홀름)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)

The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at cw-synthetics@amazon.com

카나리아 세부 정보 및 카나리아 테스트 실행 결과를 보려면 CloudWatchSyntheticsFullAccess 또는 CloudWatchSyntheticsReadOnlyAccess가 연결되어 있는 IAM 사용자로 로그인해야 합니다. 콘솔에서 모든 Synthetics 데이터를 읽으려면 AmazonS3ReadOnlyAccess, CloudWatchReadOnlyAccess 정책도 필요합니다. 또한 카나리아가 사용하는 소스 코드를 보려면 AWSLambdaReadOnlyAccess 정책이 필요합니다.

카나리아를 생성할 수 있으려면 CloudWatchSyntheticsFullAccess 정책 또는 이와 유사한 권한 집합을 가진 IAM 사용자로 로그인해야 합니다. 또한 만든 카나리아에 새 IAM 역할이 생성되면 다음 인라인 정책 설명도 필요합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

Important

사용자에게 `iam:CreateRole`, `iam:CreatePolicy` 및 `iam:AttachRolePolicy` 권한을 부여하면 해당 사용자에게 AWS 계정에 대한 전체 관리 액세스 권한이 부여됩니다. 예를 들어 이러한 권한을 가진 사용자는 모든 리소스에 대한 전체 권한을 가진 정책을 생성하고 해당 정책을 모든 역할에 연결할 수 있습니다. 이러한 권한을 부여한 사람에게 매우 주의해야 합니다.

정책 연결 및 사용자에게 권한 부여에 대한 자세한 내용은 IAM 사용자의 권한 변경 및 사용자 또는 역할의 인라인 정책을 포함하려면을 참조하십시오.

CloudWatchSyntheticsFullAccess

CloudWatchSyntheticsFullAccess 정책의 내용은 다음과 같습니다.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "synthetics:*"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:CreateBucket",
            "s3:PutBucketEncryption",
            "s3:PutEncryptionConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::cw-syn-results-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles",
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource":"arn:aws:s3:::cw-syn-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::aws-synthetics-library-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
```

```
      "Resource":[
         "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
      ],
      "Condition":{
         "StringEquals":{
            "iam:PassedToService":[
               "lambda.amazonaws.com",
               "synthetics.amazonaws.com"
            ]
         }
      }
   },
   {
      "Effect":"Allow",
      "Action":[
         "iam:GetRole"
      ],
      "Resource":[
      "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
      ]
   },
   {
      "Effect":"Allow",
      "Action":[
         "cloudwatch:GetMetricData",
         "cloudwatch:GetMetricStatistics"
      ],
      "Resource":"*"
   },
   {
      "Effect":"Allow",
      "Action":[
         "cloudwatch:PutMetricAlarm",
         "cloudwatch:DeleteAlarms"
      ],
      "Resource":[
         "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
      ]
   },
   {
      "Effect":"Allow",
      "Action":[
         "cloudwatch:DescribeAlarms"
      ],
      "Resource":[
         "arn:aws:cloudwatch:*:*:alarm:*"
      ]
   },
   {
      "Effect":"Allow",
      "Action":[
         "lambda:CreateFunction",
         "lambda:AddPermission",
         "lambda:PublishVersion",
         "lambda:UpdateFunctionConfiguration"
      ],
      "Resource":[
         "arn:aws:lambda:*:*:function:cwsyn-*"
      ]
   },
   {
      "Effect":"Allow",
      "Action":[
         "lambda:GetLayerVersionByArn",
         "lambda:GetLayerVersion",
         "lambda:PublishLayerVersion"
```

```
            ],
            "Resource":[
               "arn:aws:lambda:*:*:layer:cwsyn-*",
               "arn:aws:lambda:*:*:layer:Synthetics:*"
            ]
         }
   ]
}
```

CloudWatchSyntheticsReadOnlyAccess

CloudWatchSyntheticsReadOnlyAccess 정책의 내용은 다음과 같습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

카나리아 생성

CloudWatch Synthetics is in open preview. The preview is currently available in the following Regions:

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(스톡홀름)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)

The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at cw-synthetics@amazon.com

CloudWatch에서 제공하는 “블루프린트”를 사용하여 카나리아를 생성하거나 자신의 스크립트를 작성할 수 있습니다. 다음과 같은 카나리아 유형에 대해 블루프린트가 제공됩니다.

- 하트비트 모니터
- API 카나리아
- 잘못된 링크 검사기
- GUI 워크플로우

> **Important**
>
> 소유권 또는 권한이 있는 엔드포인트 및 API만 모니터링하려면 Synthetics 카나리아를 사용해야 합니다. 카나리아 실행 빈도 설정에 따라 이러한 엔드포인트에서 트래픽이 증가할 수 있습니다.

카나리아를 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Canaries(카나리아)를 선택합니다.

   CloudWatch에는 이미 생성한 카나리아에 대한 세부 정보가 포함된 화면이 표시됩니다.
3. Create Canary(카나리아 생성)를 선택합니다.
4. 다음 페이지에서 다음 중 하나를 선택합니다.

   - 블루프린트 스크립트를 기반으로 카나리아를 생성하려면 Use a blueprint(블루프린트 사용)를 선택합니다. 그런 다음 생성하려는 카나리아의 유형을 선택합니다.
   - 사용자 고유의 node.js 스크립트를 업로드하여 사용자 지정 카나리아를 생성하려면 Upload a script(스크립트 업로드)를 선택합니다.

     그런 다음 스크립트를 스크립트로 드래그하거나 Browse files(파일 찾아보기)를 선택하여 파일 시스템의 스크립트로 이동할 수 있습니다.
   - Amazon S3 버킷에서 스크립트를 가져오려면 Import from S3(S3에서 가져오기)을 선택합니다. 그런 다음 소스 위치에서 카나리아에 대한 전체 경로를 입력하거나 S3 찾아보기를 선택합니다.

     사용하는 S3 버킷에 대한 `s3:GetObject` 및 `s3:GetObjectVersion` 권한이 있어야 합니다. 버킷은 카나리아를 생성할 리전과 동일한 리전에 있어야 합니다.
5. 이름에 카나리아의 이름을 입력합니다. 이 이름은 여러 페이지에서 사용되므로 생성한 다른 카나리아와 구별되는 설명이 포함된 이름을 지정하는 것이 좋습니다.
6. 카나리아에 대한 블루프린트를 사용하는 경우 카나리아를 테스트할 애플리케이션 또는 엔드포인트 URL을 입력해야 할 수 있습니다. 이 URL에는 프로토콜(예: https://)이 포함되어야 합니다.
7. 카나리아에 대해 사용자 고유의 스크립트를 사용하는 경우 Lambda handler(Lambda 핸들러)에 카나리아를 시작할 진입점을 입력합니다. 입력하는 문자열은 `.handler`로 끝나야 합니다.
8. 일정에서 이 카나리아를 한 번 실행할지 아니면 정기적으로 실행할지를 선택합니다.
9. Data retention(데이터 보존)에서 실패한 카나리아 실행과 성공한 카나리아 실행에 대한 정보를 보존할 기간을 지정합니다. 범위는 1-600일입니다.
10. 경보에서 이 카나리아에 대해 CloudWatch 경보를 생성할지 여부를 선택합니다. 경보 설정은 경보를 사용하는 모든 카나리아에 대해 동일합니다.
11. Access permissions(액세스 권한)에서 카나리아를 실행할 새 IAM 역할을 생성할지 아니면 기존 역할을 사용할지 여부를 선택합니다.

    이전 카나리아에 대해 콘솔에서 생성한 IAM 역할은 해당 이전 카나리아 이름에만 해당되므로 다시 사용할 수 없습니다. 여러 카나리아에 사용할 수 있는 역할을 수동으로 생성한 경우에만 기존 역할을 사용할 수 있습니다.

    기존 역할을 사용하려면 해당 역할을 Synthetics 및 Lambda에 전달할 수 있는 `iam:PassRole` 권한이 있어야 하며 `iam:GetRole` 권한도 있어야 합니다.

카나리아에 대해 생성되는 리소스

카나리아를 만들 때 카나리아에 대해 다음 리소스가 생성됩니다.

- `CloudWatchSyntheticsRole-`*`canary-name`*`-`*`uuid` 이름을 가진 IAM 역할(AWS Management 콘솔을 사용하여 카나리아를 생성하고 카나리아에 대해 새 역할을 생성하도록 지정하는 경우)
- `CloudWatchSyntheticsPolicy-`*`canary-name`*`-`*`uuid` 이름을 가진 IAM 정책
- `cw-syn-results-`*`accountID`*`-`*`region` 이름을 가진 Amazon S3 버킷
- `Synthetics-Alarm-`*`MyCanaryName`* 이름을 가진 경보(카나리아에 대해 경보를 생성하도록 지정한 경우)
- Lambda 함수와 계층(블루프린트를 사용하여 카나리아를 생성하는 경우). 해당 이름에 접두사 `cwsyn-`*`MyCanaryName`*이 있습니다.
- `/aws/lambda/cwsyn-`*`MyCanaryName`* 이름을 가진 CloudWatch Logs 로그 그룹

카나리아 라이브러리 및 구성 정보

카나리아를 생성할 때 다음 리소스와 버전이 사용됩니다.

- Synthetics 라이브러리 버전 1.0
- Puppeteer-core 버전 1.14.0
- Puppeteer-core 1.14.0과 일치하는 Chromium 버전

카나리아의 Lambda 코드는 최대 1GB의 메모리를 갖도록 구성됩니다. 카나리아의 각 실행은 10분 또는 카나리아 빈도 값 중 더 짧은 시간 후에 시간이 초과됩니다.

카나리아 통계 및 세부 정보 보기

CloudWatch Synthetics is in open preview. The preview is currently available in the following Regions:

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(스톡홀름)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)

The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at cw-synthetics@amazon.com

카나리아에 대한 세부 정보를 보고 실행에 대한 통계를 볼 수 있습니다.

카나리아 테스트 실행 결과에 대한 모든 세부 정보를 보려면 충분한 권한이 있는 계정에 로그인해야 합니다. 자세한 내용은 CloudWatch 카나리아에 필요한 역할 및 권한 (p. 98) 단원을 참조하십시오.

카나리아 통계 및 세부 정보를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Canaries(카나리아)를 선택합니다.

   CloudWatch에는 사용자가 생성한 카나리아에 대한 세부 정보가 포함된 화면이 표시됩니다.

   - 상태는 가장 최근 테스트 실행을 통과한 카나리아 수를 시각적으로 보여줍니다.
   - Canary test runs(카나리아 테스트 실행) 아래의 그래프에서 각 포인트는 카나리아 테스트 실행의 1분을 나타냅니다. 포인트에서 일시 중지하여 세부 정보를 볼 수 있습니다.
3. 단일 카나리아에 대한 세부 정보를 보려면 상태 그래프에서 포인트를 선택하거나 Canaries(카나리아) 표에서 카나리아의 이름을 선택합니다.

   CloudWatch에는 해당 카나리아에 대한 세부 정보가 포함된 화면이 표시됩니다.

   - Canary test runs(카나리아 테스트 실행)에서 특정 테스트 실행에 대한 세부 정보를 보기 위한 라인 중 하나를 선택할 수 있습니다.
   - 그래프 아래에서 스크린샷, HAR file(HAR 파일) 또는 로그를 선택하여 이러한 유형의 세부 정보를 볼 수 있습니다.

     카나리아 테스트 실행에 대한 로그는 CloudWatch Logs가 아닌 Amazon S3 버킷에 저장됩니다.

카나리아 삭제

CloudWatch Synthetics is in open preview. The preview is currently available in the following Regions:

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(스톡홀름)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)

- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)

The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at cw-synthetics@amazon.com

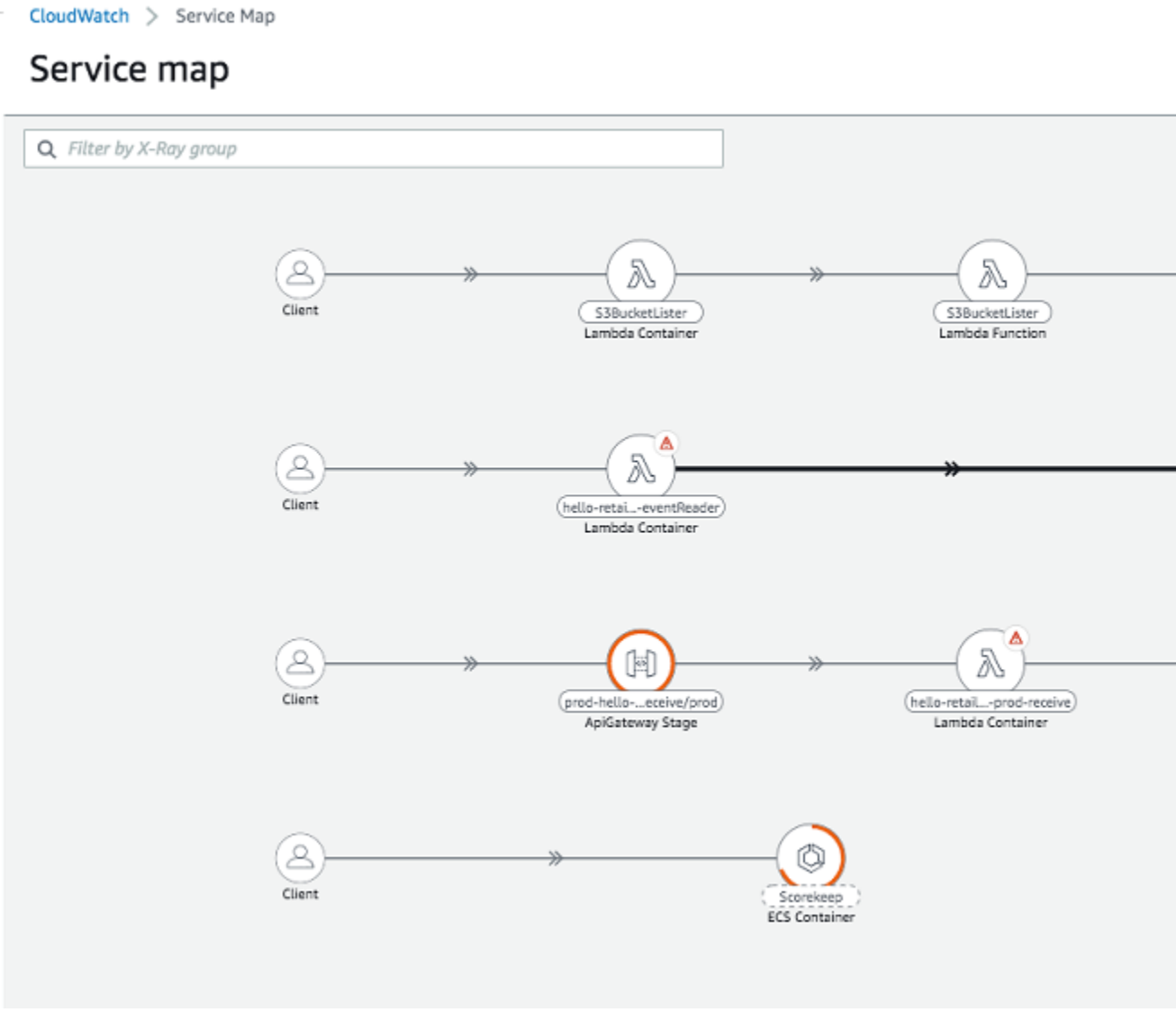
카나리아를 삭제할 때 카나리아에 의해 사용 및 생성된 리소스는 자동으로 삭제되지 않습니다. 다시 사용하지 않을 카나리아를 삭제한 후에는 다음 항목도 삭제해야 합니다.

- 이 카나리아에 사용되는 Lambda 함수 및 계층. 해당 이름에 접두사 `cwsyn-`*`MyCanaryName`*이 있습니다.
- 이 카나리아에 대해 생성된 CloudWatch 경보. 이러한 경보는 `Synthetics-Alarm-`*`MyCanaryName`* 이름을 가집니다.
- Amazon S3 객체 및 버킷(예: 카나리아의 결과 위치 및 아티팩트 위치)
- 카나리아에 대해 생성된 IAM 역할. 이러한 역할은 `role/service-role/CloudWatchSyntheticsRole-`*`MyCanaryName`* 이름을 가집니다.
- 카나리아에 대해 생성된 CloudWatch Logs 로그 그룹. 이러한 로그 그룹은 `/aws/lambda/cwsyn-`*`MyCanaryName`* 이름을 가집니다.

카나리아를 삭제한 후 올바른 리소스를 삭제할 수 있도록 카나리아를 삭제하기 전에 카나리아 세부 정보를 보고 해당 카나리아에 대한 이 정보를 기록해야 할 수 있습니다.

ServiceLens를 사용하여 애플리케이션의 상태 모니터링

CloudWatch ServiceLens는 추적, 지표, 로그 및 경보를 한 곳에 통합할 수 있게 하여 서비스 및 애플리케이션의 관찰 가능성을 향상시킵니다. ServiceLens는 CloudWatch와 AWS X-Ray를 통합하여 애플리케이션에 대한 전체적인 보기를 제공함으로써 보다 효율적으로 성능 병목 현상을 파악하고 영향을 받는 사용자를 식별할 수 있게 합니다. 서비스 맵은 서비스 엔드포인트와 리소스를 "노드"로 표시하고 각 노드 및 해당 연결에 대한 트래픽, 지연 시간 및 오류를 강조 표시합니다. 노드를 선택하여 서비스의 해당 부분과 연관된 상관관계가 있는 지표, 로그 및 추적에 대한 자세한 정보를 볼 수 있습니다. 이를 통해 문제와 해당 문제가 애플리케이션에 미치는 영향을 조사할 수 있습니다.



ServiceLens 및 상관관계가 있는 지표, 로그 및 추적을 완전히 활용하려면 X-Ray SDK를 업데이트해야 하며 애플리케이션의 계측이 필요합니다. ServiceLens는 Lambda 함수, API 게이트웨이, Amazon EC2에서 실행되는 Java 기반 애플리케이션 및 배포된 Container Insights가 있는 Kubernetes 또는 Amazon EKS에서 실행되는 Java 기반 애플리케이션에 대한 로그 상관관계를 지원합니다.

ServiceLens는 카나리아를 생성하여 외부로부터 엔드포인트 및 API를 모니터링할 수 있게 하는 완전 관리형 서비스인 Amazon CloudWatch Synthetics와 통합됩니다. 생성한 카나리아는 ServiceLens 서비스 맵에 나타납니다. 자세한 내용은 카나리아 사용 (p. 97) 단원을 참조하십시오.

AWS GovCloud (US)를 제외하고 X-Ray를 사용할 수 있는 모든 리전에서 ServiceLens를 사용할 수 있습니다. AWS GovCloud (US)에서는 ServiceLens를 사용할 수 없습니다.

주제

- ServiceLens 배포 (p. 107)
- ServiceLens에서 서비스 맵 사용 (p. 115)
- ServiceLens에서 추적 보기 사용 (p. 116)
- ServiceLens 문제 해결 (p. 117)

ServiceLens 배포

ServiceLens 배포하려면 다음 두 단계가 필요합니다.

- 서비스 맵을 볼 수 있도록 AWS X-Ray를 배포합니다.
- CloudWatch 에이전트와 X-Ray 데몬을 배포하여 CloudWatch 지표 및 CloudWatch Logs와의 서비스 맵 통합을 활성화합니다.

주제

- AWS X-Ray 배포 (p. 107)
- CloudWatch 에이전트 및 X-Ray 데몬 배포 (p. 109)

AWS X-Ray 배포

모든 AWS X-Ray SDK를 사용하여 X-Ray를 활성화할 수 있습니다. 그러나 로그 및 지표와 추적의 상관관계는 Java SDK를 사용하는 경우에만 지원됩니다.

X-Ray를 배포하려면 표준 X-Ray 설정을 따릅니다. 자세한 내용은 다음 자료를 참조하십시오.

- Java용 AWS X-Ray SDK(로그 상관관계 지원)
- Node.js용 X-Ray SDK
- .NET용 AWS X-Ray SDK
- Go용 AWS X-Ray SDK
- Python용 AWS X-Ray SDK
- Ruby용 AWS X-Ray SDK

X-Ray 설정을 완료한 후 다음 단원의 단계에 따라 X-Ray를 CloudWatch Logs와 통합하고 세그먼트 지표를 활성화합니다.

주제

- CloudWatch Logs와 통합 (p. 107)
- X-Ray에서 세그먼트 지표 활성화 (p. 108)

CloudWatch Logs와 통합

CloudWatch Logs와의 통합을 활성화하려면 다음 두 단계가 있습니다.

- 로그 상관관계에 대한 추적을 활성화합니다. 이 기능은 Java용 SDK를 사용할 경우에만 지원됩니다.
- 추적 ID 주입을 구성합니다.

로그 상관관계에 대한 추적 활성화

Java용 SDK는 표준 애플리케이션 로깅 프레임워크 집합과 CloudWatch Logs 기본 지원을 모두 지원합니다. 다음 단계를 완료하기 전에 Java용 AWS X-Ray SDK의 표준 설정을 완료해야 합니다.

지원되는 실행 시간은 Amazon EC2, CloudWatch Container Insights가 활성화된 Amazon ECS 및 Lambda입니다.

- Amazon EC2에서 로그 상관관계에 대한 추적을 활성화하려면 X-Ray EC2 플러그인을 활성화합니다. 자세한 내용은 서비스 플러그인을 참조하십시오.
- Amazon EKS에서 로그 상관관계에 대한 추적을 활성화하려면 Container Insights를 아직 활성화하지 않은 경우 먼저 활성화합니다. 자세한 내용은 Container Insights 사용 (p. 136) 단원을 참조하십시오.

  그런 다음 X-Ray SDK EKS 플러그인을 활성화합니다. 자세한 내용은 서비스 플러그인을 참조하십시오.
- Lambda에서 로그 상관관계에 대한 추적을 활성화하려면 Lambda에서 X-Ray를 활성화해야 합니다. 자세한 내용은 AWS Lambda 및 AWS X-Ray 단원을 참조하십시오.

추적 ID 주입 활성화

추적 ID 주입을 활성화하는 방법에 대한 자세한 내용은 로깅을 참조하십시오.

X-Ray에서 세그먼트 지표 활성화

Java용 AWS X-Ray SDK는 세그먼트에 대한 여러 지표를 CloudWatch에 내보내 지연 시간, 스로틀, 오류 및 장애 발생률의 샘플링되지 않은 보기를 제공할 수 있습니다. CloudWatch 에이전트를 사용하여 이러한 지표를 내보내 애플리케이션 성능에 미치는 영향을 최소화합니다. 세그먼트에 대한 자세한 내용은 세그먼트를 참조하십시오.

세그먼트 지표를 활성화할 경우 XRayApplicationMetrics라는 로그 그룹이 생성되고 ErrorRate, FaultRate, ThrottleRate 및 Latency 지표가 관찰 가능성이라는 사용자 지정 CloudWatch 지표 네임스페이스에 게시됩니다.

세그먼트 지표는 현재 Lambda에서 지원되지 않습니다.

Java용 AWS X-Ray SDK에서 세그먼트 지표를 게시할 수 있게 하려면 다음 예제를 사용합니다.

```
AWSXRayRecorderBuilder builder = AWSXRayRecorderBuilder.standard().withSegmentListener(new
 MetricsSegmentListener());
```

Amazon EKS 및 Container Insights와 함께 ServiceLens를 사용하는 경우 다음 예제와 같이 `AWS_XRAY_METRICS_DAEMON_ADDRESS` 환경 변수를 `HOST_IP`에 추가합니다.

```
env:
- name: HOST_IP
  valueFrom:
    fieldRef:
      apiVersion: v1
      fieldPath: status.hostIP
- name: AWS_XRAY_METRICS_DAEMON_ADDRESS
  value: $(HOST_IP):25888
```

자세한 내용은 X-Ray CloudWatch 지표 활성화를 참조하십시오.

CloudWatch 에이전트 및 X-Ray 데몬 배포

이 단원에서는 CloudWatch 에이전트 및 X-Ray 데몬을 배포하는 방법에 대해 설명합니다. 다음 환경에서 에이전트와 데몬을 배포할 수 있습니다.

- Amazon ECS 또는 Fargate
- Amazon EC2에서 호스팅된 Kubernetes 또는 Amazon EKS
- Amazon EC2

이러한 각 환경에 대한 배포 단계는 다음 단원에서 설명합니다.

주제

- Amazon ECS에 CloudWatch 에이전트 및 X-Ray 데몬 배포 (p. 109)
- Amazon EKS 또는 Kubernetes에 CloudWatch 에이전트 및 X-Ray 데몬 배포 (p. 113)
- Amazon EC2에 CloudWatch 에이전트 및 X-Ray 데몬 배포 (p. 115)

Amazon ECS에 CloudWatch 에이전트 및 X-Ray 데몬 배포

Amazon ECS에서는 CloudWatch 에이전트를 사이드카로 애플리케이션 컨테이너에 배포하여 지표를 수집합니다. SSM 파라미터 스토어를 통해 CloudWatch 에이전트를 구성할 수 있습니다.

IAM 역할 생성

두 가지 IAM 역할을 생성해야 합니다. 이러한 역할을 이미 생성한 경우 권한을 추가해야 할 수 있습니다.

- ECS 작업 역할 — 컨테이너는 이 역할을 사용하여 실행됩니다. CloudWatchAgentServerPolicy 및 AWSXRayDaemonWriteAccess 외에도 권한은 애플리케이션에 필요한 것이어야 합니다.
- ECS 작업 정의 역할 — Amazon ECS는 이 역할을 사용하여 컨테이너를 시작하고 실행합니다. 이 역할을 이미 생성한 경우 AmazonSSMReadOnlyAccess, AmazonECSTaskExecutionRolePolicy 및 CloudWatchAgentServerPolicy 정책을 해당 역할에 연결합니다.

  Amazon ECS에서 사용하기 위해 더 중요한 데이터를 저장해야 하는 경우 자세한 내용은 중요한 데이터 지정을 참조하십시오.

IAM 역할 생성에 대한 자세한 내용은 IAM 역할 생성을 참조하십시오.

에이전트 구성을 SSM 파라미터 스토어에 저장

에이전트 구성 파일에 다음 섹션이 있는지 확인한 다음 SSM 파라미터 스토어에 업로드해야 합니다.

```
{
  "logs": {
    "metrics_collected": {
      "emf": {}
    }
  }
}
```

에이전트 구성을 SSM 파라미터 스토어에 업로드하려면

1. 에이전트 구성 내용을 로컬 파일 `/tmp/ecs-cwagent.json`에 넣습니다.
2. 다음 명령을 입력합니다. *region*을 클러스터의 리전으로 바꿉니다.

```
aws ssm put-parameter \
```

```
--name "ecs-cwagent" \
--type "String" \
--value "`cat /tmp/ecs-cwagent.json`" \
--region "region"
```

작업 정의 생성 및 작업 시작

이 작업의 단계는 EC2 시작 유형을 사용할지 아니면 Fargate 시작 유형을 사용할지 여부에 따라 다릅니다.

EC2 시작 유형

먼저 작업 정의를 생성합니다. 이 예제에서 컨테이너 "demo-app"은 X-Ray SDK 지표를 CloudWatch 에이전트로 전송하고 추적 정보를 X-Ray 데몬으로 전송합니다.

`/tmp/ecs-cwagent-ec2.json`과 같은 로컬 JSON 파일에 다음 작업 정의를 복사합니다. 다음과 같이 자리 표시자를 바꿉니다.

- *{{ecs-task-role}}*을 ECS 작업 역할의 ARN으로 바꿉니다.
- *{{ecs-task-execution-role}}*을 ECS 작업 실행 역할의 ARN으로 바꿉니다.
- *{{demo-app-image}}*를 X-Ray SDK 통합이 활성화된 애플리케이션 이미지로 바꿉니다. demo-app의 이름을 사용자 고유의 애플리케이션 이름으로 변경합니다.
- *{{region}}*을 컨테이너에 대한 로그를 전송하려는 AWS 리전의 이름으로 바꿉니다. 예, `us-west-2`.

```
{
    "family": "ecs-cwagent-ec2",
    "taskRoleArn": "{{ecs-task-role}}",
    "executionRoleArn": "{{ecs-task-execution-role}}",
    "networkMode": "bridge",
    "containerDefinitions": [
        {
            "name": "demo-app",
            "image": "{{demo-app-image}}",
            "links": [
                "cloudwatch-agent",
                "xray-daemon"
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                  "awslogs-create-group": "True",
                  "awslogs-group": "/ecs/ecs-cwagent-ec2",
                  "awslogs-region": "{{region}}",
                  "awslogs-stream-prefix": "ecs"
                }
            }
        },
        {
            "name": "xray-daemon",
            "image": "amazon/aws-xray-daemon:latest",
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                  "awslogs-create-group": "True",
                  "awslogs-group": "/ecs/ecs-cwagent-ec2",
                  "awslogs-region": "{{region}}",
                  "awslogs-stream-prefix": "ecs"
                }
            }
        },
        {
```

```
            "name": "cloudwatch-agent",
            "image": "amazon/cloudwatch-agent:latest",
            "secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent"
                }
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                  "awslogs-create-group": "True",
                  "awslogs-group": "/ecs/ecs-cwagent-ec2",
                  "awslogs-region": "{{region}}",
                  "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ],
    "requiresCompatibilities": [
        "EC2"
    ],
    "cpu": "256",
    "memory": "256"
}
```

다음 명령을 입력하여 작업 정의를 생성합니다. *{{region}}*을 클러스터의 리전으로 바꿉니다.

```
aws ecs register-task-definition \
    --cli-input-json file:///tmp/ecs-cwagent-ec2.json \
    --region {{region}}
```

다음 명령을 입력하여 작업을 시작합니다. *{{cluster-name}}* 및 *{{region}}*을 클러스터의 이름 및 리전으로 바꿉니다.

```
aws ecs run-task \
    --cluster {{cluster-name}} \
    --task-definition ecs-cwagent-ec2 \
    --region {{region}} \
    --launch-type EC2
```

Fargate 시작 유형

먼저 작업 정의를 생성합니다. 이 예제에서 컨테이너 "demo-app"은 X-Ray SDK 지표를 CloudWatch 에이전트로 전송하고 추적 정보를 X-Ray 데몬으로 전송합니다.

`/tmp/ecs-cwagent-ec2.json`과 같은 로컬 JSON 파일에 다음 작업 정의를 복사합니다. 다음과 같이 자리 표시자를 바꿉니다.

- *{{ecs-task-role}}*을 ECS 작업 역할의 Amazon 리소스 이름(ARN)으로 바꿉니다.
- *{{ecs-task-execution-role}}*을 ECS 작업 실행 역할의 ARN으로 바꿉니다.
- *{{demo-app-image}}*를 X-Ray SDK 통합이 활성화된 애플리케이션 이미지로 바꿉니다. demo-app의 이름을 사용자 고유의 애플리케이션 이름으로 변경합니다.
- *{{region}}*을 컨테이너에 대한 로그를 전송하려는 AWS 리전의 이름으로 바꿉니다. 예, `us-west-2`.

```
{
    "family": "ecs-cwagent-fargate",
    "taskRoleArn": "{{ecs-task-role}}",
    "executionRoleArn": "{{ecs-task-execution-role}}",
```

```
    "networkMode": "awsvpc",
    "containerDefinitions": [
        {
            "name": "demo-app",
            "image": "{{demo-app-image}}",
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                  "awslogs-create-group": "True",
                  "awslogs-group": "/ecs/ecs-cwagent-fargate",
                  "awslogs-region": "{{region}}",
                  "awslogs-stream-prefix": "ecs"
                }
            }
        },
        {
            "name": "xray-daemon",
            "image": "amazon/aws-xray-daemon:latest",
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                  "awslogs-create-group": "True",
                  "awslogs-group": "/ecs/ecs-cwagent-fargate",
                  "awslogs-region": "{{region}}",
                  "awslogs-stream-prefix": "ecs"
                }
            }
        },
        {
            "name": "cloudwatch-agent",
            "image": "amazon/cloudwatch-agent:latest",
            "secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent"
                }
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                  "awslogs-create-group": "True",
                  "awslogs-group": "/ecs/ecs-cwagent-fargate",
                  "awslogs-region": "{{region}}",
                  "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "512",
    "memory": "1024"
}
```

다음 명령을 입력하여 작업 정의를 생성합니다. *{{region}}*을 클러스터의 리전으로 바꿉니다.

```
aws ecs register-task-definition \
    --cli-input-json file:///tmp/ecs-cwagent-fargate.json \
    --region {{region}}
```

Fargate 클러스터가 이미 설정되어 있는 경우 방금 생성한 작업 정의를 사용하여 작업을 시작할 수 있습니다. 아직 Fargate 클러스터가 없는 경우 Fargate를 설정하기 위한 나머지 단계에 대한 자세한 내용은 서비스 구성을 참조하십시오.

Amazon EKS 또는 Kubernetes에 CloudWatch 에이전트 및 X-Ray 데몬 배포

이 주제에서는 Amazon EKS 또는 Kubernetes에 X-Ray 데몬 및 CloudWatch 에이전트를 설치하는 방법에 대해 설명합니다.

Amazon EKS 또는 Kubernetes에 X-Ray 데몬 배포

Amazon EKS 또는 Kubernetes에 CloudWatch 에이전트 및 X-Ray 데몬을 설치하기 위해 빠른 설정을 사용할 수 있습니다.

CloudWatch 에이전트 및 X-Ray 데몬을 Amazon EKS 또는 Kubernetes에 설치하려면

1. EC2 인스턴스 또는 Kubernetes 작업자 노드에 연결된 IAM 역할에 CloudWatchAgentServerPolicy 및 AWSXRayDaemonWriteAccess 정책이 연결되어 있는지 확인합니다.
2. 다음 명령을 입력합니다.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/master/k8s-deployment-manifest-templates/deployment-mode/daemonset/cwagent-fluentd-xray/cwagent-fluentd-xray-quickstart.yaml | sed "s/{{cluster_name}}/cluster-name/;s/{{region_name}}/region/" | kubectl apply -f -
```

빠른 시작이 수행하는 작업

이 단원에서는 CloudWatch 에이전트 및 X-Ray 데몬의 빠른 설정에 대해 설명합니다.

- 빠른 설정은 다음 파일 내용과 함께 `kubectl apply -f` 명령을 통해 X-Ray 데몬을 설치합니다.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: xray-daemon
  namespace: amazon-cloudwatch
spec:
  selector:
    matchLabels:
      name: xray-daemon
  template:
    metadata:
      labels:
        name: xray-daemon
    spec:
      containers:
        - name: xray-daemon
          image: amazon/aws-xray-daemon:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 2000
              hostPort: 2000
              protocol: UDP
          resources:
            limits:
              cpu:  100m
              memory: 256Mi
            requests:
              cpu: 50m
              memory: 50Mi
      terminationGracePeriodSeconds: 60
```

- 빠른 설정은 도커 이미지 버전/레이블 1.231221.0 이상 또는 최신 버전을 사용하여 CloudWatch 에이전트를 업데이트합니다. https://hub.docker.com/r/amazon/cloudwatch-agent에서 이미지를 확인할 수 있습니다.
- X-Ray SDK가 클러스터 이름 및 리전 정보를 읽을 수 있도록 하기 위해 빠른 설정은 도커 이미지 버전/레이블 1.231221.0 이상 또는 최신 버전을 사용하여 CloudWatch 에이전트를 업데이트합니다. https://hub.docker.com/r/amazon/cloudwatch-agent에서 이미지를 확인할 수 있습니다.
- X-Ray SDK가 클러스터 이름 및 리전 정보를 읽을 수 있도록 하기 위해 빠른 설정에서 다음 콘텐츠가 포함된 파일을 생성한 후 **`kubectl apply -f`** 명령과 함께 적용했습니다.

```
---
# create role binding for XRay SDK to read config map
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
name: container-insights-discovery-role
namespace: amazon-cloudwatch
rules:
- apiGroups:
- ""
resourceNames:
- cluster-info
resources:
- configmaps
verbs:
- get

---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
name: service-users-cloudwatch-discovery-role-binding
namespace: amazon-cloudwatch
roleRef:
apiGroup: rbac.authorization.k8s.io
kind: Role
name: container-insights-discovery-role
subjects:
- apiGroup: rbac.authorization.k8s.io
kind: Group

name: system:serviceaccounts
```

- 빠른 설정은 X-Ray SDK 지표를 수신하는 CloudWatch 에이전트 포트를 표시합니다. 기본 포트는 UDP 25888입니다.

```
ports:
  - containerPort: 25888
    hostPort: 25888
    protocol: UDP
```

- 빠른 설정은 에이전트 구성 JSON을 다음 JSON이 있는 X-Ray SDK 지표 구성과 병합합니다.

```
{
  "logs": {
    "metrics_collected": {
      "emf": {}
    }
  }
}
```

Amazon EC2에 CloudWatch 에이전트 및 X-Ray 데몬 배포

CloudWatch 에이전트 및 X-Ray 데몬의 표준 설치는 다음 CloudWatch 에이전트 구성 섹션 예제를 추가하여 Amazon EC2에서 ServiceLens를 활성화하기에 충분합니다. 에이전트 설치에 대한 자세한 내용은 CloudWatch 에이전트 설치 (p. 190) 단원을 참조하십시오. CloudWatch 에이전트 구성 파일에 대한 자세한 내용은 CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 (p. 227) 단원을 참조하십시오.

CloudWatch 에이전트를 구성할 때 구성 파일에 다음 섹션을 포함합니다.

```
{
  "logs": {
    "metrics_collected": {
      "emf": {}
    }
  }
}
```

X-Ray 데몬 설치에 대한 자세한 내용은 X-Ray 데몬 구성을 참조하십시오.

ServiceLens에서 서비스 맵 사용

이 단원에서는 서비스 맵을 소개하고 서비스 맵을 탐색하는 방법을 배웁니다.

서비스 맵을 보려면 AWS X-Ray를 설치하고 다른 ServiceLens 배포 단계를 완료했어야 합니다. 자세한 내용은 ServiceLens 배포 (p. 107) 단원을 참조하십시오.

또한 CloudWatch 콘솔을 볼 수 있는 권한뿐만 아니라 `AWSXrayReadOnlyAccess` 관리형 정책이 있는 계정에 로그인해야 합니다. 자세한 내용은 IAM에서 AWS X-Ray를 사용하는 방식 및 Amazon CloudWatch 대시보드 사용 (p. 16) 단원을 참조하십시오.

서비스 맵 사용을 시작하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Service Map(서비스 맵)을 선택합니다.

   서비스 맵이 나타납니다. 다음 부분이 포함되어 있습니다.

   - 추적을 활성화한 AWS 서비스 및 사용자 지정 애플리케이션이 원 또는 "노드"로 표시됩니다. 각 노드의 크기는 해당 서비스로 이동하는 추적된 요청의 상대적 수를 나타냅니다.
   - 가장자리 또는 노드 간의 연결은 노드를 연결하는 선으로 표시됩니다. 기본적으로 선의 두께는 해당 노드 간에 추적된 요청의 상대적 수를 나타냅니다.

     오른쪽 상단의 드롭다운 메뉴를 사용하여 노드 및 가장자리 크기 조정에 추적된 요청 수를 사용할지 아니면 평균 지연 시간을 사용할지 선택할 수 있습니다. 모든 노드 및 가장자리에 대해 일정한 크기를 사용하도록 선택할 수도 있습니다.
   - 노드의 진입점은 왼쪽에 "클라이언트"로 표시됩니다. "클라이언트"는 웹 서버 트래픽과 추적된 API 작업 요청을 모두 나타냅니다.
   - 부분적으로 빨간색, 주황색 또는 보라색으로 외곽선이 표시된 노드에 문제가 있습니다. 이러한 노드에 대한 일부 추적된 요청에는 결함, 오류 또는 병목 현상이 있습니다. 색상 외곽선의 백분율은 문제가 있는 추적된 요청의 백분율을 나타냅니다.
   - 노드 옆에 느낌표가 있는 삼각형이 있으면 해당 노드와 관련된 CloudWatch 경보 중 하나 이상이 경보 상태에 있습니다.
3. 기본적으로 맵의 데이터는 가장 최근 6시간 동안의 데이터입니다. 기간의 시간 프레임을 변경하려면 화면 오른쪽 위에 있는 컨트롤을 사용합니다. 표시할 시간 범위는 최대 6시간이며 과거 30일까지 가능합니다.

4. X-Ray 그룹을 활성화한 경우 필터에서 X-ray 그룹을 선택하여 맵을 필터링할 수 있습니다.
5. 노드의 수신 및 발신 연결에 초점을 맞추려면 노드를 선택하고 서비스 맵의 맨 위에 있는 연결 보기를 선택합니다.
6. 노드에 대한 지연 시간, 오류, 요청 및 경보 요약 통계를 표시하는 팝업을 보려면 해당 노드에서 일시 중지합니다.
7. 가장자리 연결에 대한 지연 시간 통계를 보려면 해당 가장자리를 나타내는 줄에서 일시 중지합니다.
8. 서비스에 대한 경보 상태를 지연 시간, 오류 및 추적 수에 대한 선 차트와 함께 표시하려면 맵에서 해당 서비스 노드를 선택합니다.

   이 보기에 대한 자세한 내용은 다음 절차를 참조하십시오.
9. 서비스 맵을 테이블로 보려면 화면 상단의 List view(목록 보기)를 선택합니다. 이 보기에서는 맵에 표시되는 노드와 경보를 필터링하고 정렬할 수 있습니다.
10. 특정 노드에 대한 지표가 포함된 대시보드를 보려면 노드를 선택한 다음 화면 아래쪽에 있는 대시보드 보기를 선택합니다.

서비스 맵에서 서비스 또는 애플리케이션에 대한 추적을 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Service Map(서비스 맵)을 선택합니다.
3. 조사하려는 서비스 또는 애플리케이션을 나타내는 노드를 선택합니다.

   CloudWatch에는 해당 서비스에 대한 지연 시간, 오류 및 추적 수의 선 차트와 경보 상태 요약이 표시됩니다.

   이러한 차트 위에는 서비스에 대한 로그와 추적을 탐색하는 옵션이 있습니다.
4. 서비스와 관련된 추적을 보려면 View traces(추적 보기)를 선택합니다.

   콘솔은 조사 중인 서비스에 초점을 맞춘 Traces(추적) 보기로 전환됩니다. 자세한 내용은 ServiceLens에서 추적 보기 사용 (p. 116) 단원을 참조하십시오.

ServiceLens에서 추적 보기 사용

추적 보기를 사용하면 애플리케이션에서 최근 X-Ray 추적을 볼 수 있습니다.

추적을 보고 추가 정보를 탐색하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Traces(추적)를 선택합니다.
3. Filter type(필터 유형)에서 추적을 정렬할 다른 기준을 선택할 수 있습니다. 이렇게 하면 CloudWatch는 선택 사항에 따라 추적의 성공률과 응답 시간을 표시합니다. 페이지 하단에 표시되는 추적 목록은 필터와 일치하는 추적으로 범위가 좁아집니다. 필터와 일치하는 처음 1000개의 추적이 검색됩니다.

   필터 테이블의 값은 필터가 반환하는 추적으로 채워지고 필터에서 선택 사항을 구체화하면 업데이트됩니다.

   Filter type(필터 유형)에서 Custom query(사용자 지정 쿼리) 옵션을 사용하여 사용자 지정 표현식을 필터로 추가합니다. 사용자 지정 표현식에는 키워드, 단항 또는 이진 연산자 및 키워드의 비교 값이 포함될 수 있습니다. 키워드는 `responsetime`, `duration` 및 상태 코드와 같은 추적의 일부에 해당할 수 있습니다. 자세한 내용은 필터 표현식 구문을 참조하십시오.

   Traces by(추적 기준) 아래의 행 옆에 있는 확인란을 선택하고 Add to filter(필터에 추가)를 선택하여 필터에 더 초점을 맞출 수 있습니다. 선택한 행의 필터 값이 필터에 추가됩니다.

4. 현재 필터에서 반환한 추적의 응답 시간 분포를 나타내는 히스토그램과 각 개별 추적을 표시하는 표를 보려면 화면 아래쪽으로 스크롤합니다.

   추적에 대한 세부 정보를 보려면 표에서 추적 중 하나를 선택합니다.

   추적 세부 정보 페이지에는 추적을 구성하는 각 세그먼트가 포함된 타임라인이 표시됩니다. 해당하는 경우 오류를 포함하여 추가 관련 메타데이터를 볼 세그먼트를 선택합니다. 로그를 연결할 수 있는 추적의 경우 추적과 연결된 로그 줄이 타임라인 아래에 표시됩니다.

   그런 다음 선택적으로 CloudWatch Logs Insights에서 해당 로그 항목을 볼 수 있습니다. 이렇게 하려면 View in CloudWatch Logs Insights(CloudWatch Logs Insights에서 보기)를 선택한 다음 쿼리 실행을 선택합니다.

   일부 추적에는 연결된 로그 항목이 없습니다.

   - Amazon EC2 또는 Amazon EKS에 대한 로그 항목이 표시되지 않을 경우 최신 버전의 X-Ray SDK로 업데이트해야 합니다. 자세한 내용은 AWS X-Ray 배포 (p. 107) 단원을 참조하십시오.
   - ServiceLens는 이름이 `/ecs/`로 시작하는 로그 그룹이 20개 이하인 경우에만 Amazon ECS에 대한 로그 항목을 표시합니다.
   - 현재 Fargate 추적에 대한 로그 항목을 사용할 수 없습니다.

ServiceLens 문제 해결

다음 단원은 CloudWatch ServiceLens에 문제가 있는 경우 도움이 될 수 있습니다.

내 로그가 모두 표시되지 않음

ServiceLens에 표시되도록 로그를 구성하는 방법은 서비스에 따라 다릅니다.

- API 게이트웨이 로그는 API 게이트웨이에서 로깅이 설정된 경우 나타납니다.
- Amazon ECS 및 Amazon EKS 로그는 최신 버전의 X-Ray SDK 및 CloudWatch 에이전트를 사용하는 경우 나타납니다. 자세한 내용은 ServiceLens 배포 (p. 107) 단원을 참조하십시오.
- Lambda 로그는 요청 ID가 로그 항목에 있는 경우 나타납니다. 이 경우는 다음 표에 나열된 상황에서 자동으로 발생합니다. 실행 시간에 추적 ID가 자동으로 포함되지 않는 다른 경우에는 추적 ID를 수동으로 포함할 수 있습니다.

Runtime	방법	로그 항목에 요청 ID가 자동으로 있습니까?
Java	context.getLogger.log aws-lambda-java-log4j2	예
Java	System.out.println	아니요
Python	context.log logging.info/error/log/etc...	예
Python	print	아니요
Node.js	context.log console.log/info/error/etc...	예

Runtime	방법	로그 항목에 요청 ID가 자동으로 있습니까?
dotnet	context.Logger.log Console.WriteLine()	아니요
Go	Fmt.printf log.Print	아니요
Ruby	puts	아니요

서비스 맵에 내 경보가 모두 표시되지 않음

ServiceLens는 해당 노드와 연결된 경보가 ALARM 상태인 경우에만 노드에 대한 경고 아이콘을 표시합니다.

ServiceLens는 다음 논리를 사용하여 경보를 노드와 연결합니다.

- 노드가 AWS 서비스를 나타내는 경우 해당 서비스와 연결된 네임스페이스를 가진 모든 경보는 노드와 연결됩니다. 예를 들어 `AWS::Kinesis` 유형의 노드는 CloudWatch 네임스페이스 `AWS/Kinesis`의 지표를 기반으로 하는 모든 경보와 연결됩니다.
- 노드가 AWS 리소스를 나타내는 경우 해당 특정 리소스에 대한 경보가 연결됩니다. 예를 들어 이름이 “MyTable”인 `AWS::DynamoDB::Table` 유형의 노드는 네임스페이스 `AWS/DynamoDB`가 있는 지표를 기반으로 하며 `TableName` 차원이 `MyTable`로 설정된 모든 경보에 연결됩니다.
- 노드가 이름 주위의 테두리가 파선으로 식별되는 알 수 없는 유형이면 해당 노드와 연결된 경보가 없습니다.

서비스 맵에 일부 AWS 리소스가 표시되지 않음

서비스 맵에서 AWS 리소스를 추적하려면 X-Ray SDK를 사용하여 AWS SDK를 캡처해야 합니다. X-Ray에 대한 자세한 내용은 AWS X-Ray란 무엇입니까?를 참조하십시오.

모든 AWS 리소스가 전용 노드로 표시되는 것은 아닙니다. 일부 AWS 서비스는 서비스에 대한 모든 요청의 단일 노드로 표시됩니다. 다음 리소스 유형은 리소스별 노드와 함께 표시됩니다.

- `AWS::DynamoDB::Table`
- `AWS::Lambda::Function`

  Lambda 함수는 두 개의 노드로 표시되는데 하나는 Lambda 컨테이너에 대한 것이고 하나는 함수에 대한 것입니다. 이것은 Lambda 함수의 콜드 스타트 문제를 식별하는 데 도움이 됩니다. Lambda 컨테이너 노드는 Lambda 함수 노드와 동일한 방식으로 경보 및 대시보드와 연결됩니다.
- `AWS::ApiGateway::Stage`
- `AWS::SQS::Queue`
- `AWS::SNS::Topic`

내 서비스 맵에 너무 많은 노드가 있음

X-Ray 그룹을 사용하여 맵을 여러 맵으로 분할합니다. 자세한 내용은 그룹에 필터 표현식 사용을 참조하십시오.

교차 계정 교차 리전 CloudWatch 콘솔

CloudWatch 콘솔에 교차 계정 교차 리전 기능을 추가할 수 있습니다. 이 기능은 다른 계정에서 로그인하고 로그아웃하지 않고도 대시보드, 경보, 지표 및 자동 대시보드에 대한 교차 계정 및 교차 리전 가시성을 제공합니다.

그런 다음 여러 AWS 계정 및 여러 AWS 리전의 CloudWatch 데이터를 단일 대시보드로 요약하는 대시보드를 생성할 수 있습니다.

많은 조직에서는 결제 및 보안 경계를 제공하기 위해 AWS 리소스를 여러 계정에 배포하고 있습니다. 이 경우 하나 이상의 사용자 계정을 모니터링 계정으로 지정하고 이러한 계정에 교차 계정 교차 리전 대시보드를 구축하는 것이 좋습니다.

교차 계정 교차 리전 기능이 AWS Organizations와 통합되어 교차 계정 교차 리전 대시보드를 효율적으로 구축할 수 있습니다.

CloudWatch에 교차 계정 교차 리전 기능 활성화

CloudWatch 콘솔에서 교차 계정 교차 리전 기능을 설정하려면 AWS Management 콘솔을 사용하여 공유 계정과 모니터링 계정을 설정합니다.

공유 계정 설정

모니터링 계정에서 데이터를 사용할 수 있도록 각 계정에서 공유를 활성화해야 합니다.

사용자 계정에서 다른 계정과 CloudWatch 데이터를 공유하도록 활성화하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 구성을 선택합니다.
3. 데이터 공유를 선택합니다.
4. 공유에서 특정 계정을 선택하고 데이터를 공유할 계정의 ID를 입력합니다.

   여기에서 지정하는 모든 계정은 사용자 계정의 CloudWatch 데이터를 볼 수 있습니다. 알고 신뢰하는 계정의 ID만 지정합니다.
5. 권한에 대해 다음 옵션 중 하나를 사용하여 데이터를 공유하는 방법을 지정합니다.

   - CloudWatch 지표, 대시보드 및 경보에 대한 읽기 전용 액세스 권한을 제공합니다. 이 옵션을 사용하면 모니터링 계정에서 사용자의 CloudWatch 데이터가 포함된 위젯이 있는 교차 계정 교차 리전 대시보드를 생성할 수 있습니다.
   - CloudWatch 자동 대시보드 포함. 이 옵션을 선택하면 모니터링 계정의 사용자가 이 계정의 자동 대시보드에서 정보를 볼 수도 있습니다. 자세한 내용은 Amazon CloudWatch 시작하기 (p. 10) 단원을 참조하십시오.
   - 사용자 계정의 모든 항목에 대한 전체 읽기 전용 액세스 권한. 이 옵션을 사용하면 공유에 사용하는 계정에서 사용자 계정의 CloudWatch 데이터가 포함된 위젯이 있는 교차 계정 간 리전 대시보드를 생성할 수 있습니다. 또한 이러한 계정에서 사용자 계정을 더 자세히 살펴보고 다른 AWS 서비스의 콘솔에서 사용자 계정의 데이터를 볼 수 있습니다.

6. CloudFormation 템플릿 실행을 선택합니다.

   확인 화면에서 **Confirm**을 입력하고 템플릿 실행을 선택합니다.
7. 확인란을 선택하고 스택 생성을 선택합니다.

전체 조직과 공유.

위의 절차를 완료하면 사용자 계정에서 하나의 계정과 데이터를 공유할 수 있는 IAM 역할이 생성됩니다. 조직의 모든 계정과 데이터를 공유하는 IAM 역할을 생성하거나 편집할 수 있습니다. 조직의 모든 계정을 알고 신뢰하는 경우에만 이 작업을 수행하십시오.

조직의 모든 계정과 CloudWatch 계정 데이터를 공유하려면

1. 아직 수행하지 않은 경우 이전 절차를 완료하여 한 AWS 계정과 데이터를 공유하십시오.
2. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
3. 탐색 창에서 Roles를 선택합니다.
4. 역할 목록에서 CloudWatch-CrossAccountSharingRole을 선택합니다.
5. 신뢰 관계, 신뢰 관계 편집을 차례대로 선택합니다.

   다음과 같은 정책이 표시됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. 정책을 다음과 같이 변경하여 *org-id* 를 조직의 ID로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "org-id"
        }
      }

    }
  ]
}
```

7. Update Trust Policy(신뢰 정책 업데이트)를 선택합니다.

모니터링 계정 설정

교차 계정 교차 리전 CloudWatch 데이터를 보려면 각 모니터링 계정을 활성화합니다.

다음 절차를 완료하면 CloudWatch는 모니터링 계정에서 CloudWatch를 사용하여 다른 계정에서 공유된 데이터에 액세스하는 서비스 연결 역할을 생성합니다. 이 서비스 연결 역할을 AWSServiceRoleForCloudWatchCrossAccount라고 합니다. 자세한 내용은 CloudWatch에 서비스 연결 역할 사용 (p. 369)를 참조하십시오.

사용자 계정에서 교차 계정 교차 리전 CloudWatch 데이터를 볼 수 있도록 활성화하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 설정을 선택한 다음 구성을 선택합니다.
3. 교차 계정 교차 리전 보기에서 다음 옵션 중 하나를 선택합니다.

   - 계정 ID 입력. 이 옵션은 교차 계정 데이터를 볼 때 계정을 전환할 때마다 계정 ID를 수동으로 입력하라는 메시지를 표시합니다.
   - AWS 조직 계정 선택기. 이 옵션을 사용하면 조직와의 교차 계정 교차 리전 통합을 완료할 때 지정한 계정이 표시됩니다. 다음에 콘솔을 사용하면 교차 계정 데이터를 볼 때 선택할 수 있는 이러한 계정의 드롭다운 목록이 CloudWatch에 표시됩니다.

     이렇게 하려면 먼저 CloudWatch에서 조직의 계정 목록을 볼 수 있도록 조직 마스터 계정을 사용해야 합니다. 자세한 내용은 (선택 사항) AWS Organizations와 통합 (p. 121) 단원을 참조하십시오.
   - 사용자 지정 계정 선택기. 이 옵션은 계정 ID 목록을 입력하라는 메시지를 표시합니다. 다음에 콘솔을 사용하면 교차 계정 데이터를 볼 때 선택할 수 있는 이러한 계정의 드롭다운 목록이 CloudWatch에 표시됩니다.

     또한 보려는 계정을 선택할 때 계정을 식별하는 데 도움이 되도록 각 계정에 대한 레이블을 입력할 수도 있습니다.
4. [Enable]을 선택합니다.

이 설정을 완료한 후 교차 계정 교차 리전 대시보드를 생성할 수 있습니다. 자세한 내용은 교차 계정 교차 리전 대시보드 (p. 17) 단원을 참조하십시오.

(선택 사항) AWS Organizations와 통합

교차 계정 교차 리전 기능을 AWS Organizations와 통합하려면 조직의 모든 계정 목록을 모니터링 계정에서 사용할 수 있도록 만들어야 합니다.

교차 계정 교차 리전 CloudWatch 기능을 사용하여 조직의 모든 계정 목록에 액세스하려면

1. 조직의 마스터 계정에 로그인합니다.
2. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
3. 탐색 창에서 설정을 선택한 다음 구성을 선택합니다.
4. 조직의 계정 목록을 볼 수 있는 권한 부여에 대해 특정 계정을선택하여 계정 ID 목록을 입력하라는 메시지를 표시합니다. 조직의 계정 목록은 여기에서 지정한 계정과만 공유됩니다.
5. 조직 계정 목록 공유를 선택합니다.
6. CloudFormation 템플릿 실행을 선택합니다.

   확인 화면에서 **`Confirm`**을 입력하고 템플릿 실행을 선택합니다.

CloudWatch 교차 계정 교차 리전 설정 문제 해결

이 섹션에는 CloudWatch의 교차 계정, 교차 리전 콘솔 배포에 대한 문제 해결 팁이 포함되어 있습니다.

교차 계정 데이터를 표시하는 데 액세스 거부 오류가 발생했습니다.

다음을 확인하십시오.

- 모니터링 계정에는 AWSServiceRoleForCloudWatchCrossAccount라는 역할이 있어야 합니다. 그렇지 않은 경우 이 역할을 생성해야 합니다. 자세한 내용은 모니터링 계정 설정 (p. 120) 단원을 참조하십시오.
- 각 공유 계정에는 CloudWatch-CrossAccountSharingRole이라는 역할이 있어야 합니다. 그렇지 않은 경우 이 역할을 생성해야 합니다. 자세한 내용은 공유 계정 설정 (p. 119) 단원을 참조하십시오.
- 공유 역할은 모니터링 계정을 신뢰해야 합니다.

CloudWatch 교차 계정 콘솔에 대해 역할이 올바르게 설정되어 있는지 확인하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Roles를 선택합니다.
3. 역할 목록에서 필요한 역할이 있는지 확인합니다. 공유 계정에서 CloudWatch-CrossAccountSharingRole을 찾습니다. 모니터링 계정에서 AWSServiceRoleForCloudWatchCrossAccount를 찾습니다.
4. 공유 계정에 CloudWatch-CrossAccountSharingRole이 이미 있는 경우 CloudWatch-CrossAccountSharingRole을 선택합니다.
5. 신뢰 관계, 신뢰 관계 편집을 차례대로 선택합니다.
6. 정책에 모니터링 계정의 계정 ID 또는 모니터링 계정이 포함된 조직의 조직 ID가 나열되어 있는지 확인합니다.

콘솔에 계정 드롭다운이 표시되지 않습니다.

먼저 앞의 문제 해결 섹션에서 설명한 대로 올바른 IAM 역할을 생성했는지 확인합니다. 이러한 항목이 올바르게 설정된 경우 사용자 계정에서 교차 계정 교차 리전 데이터 활성화 (p. 121)에 설명된 대로 이 계정에서 교차 계정 데이터를 볼 수 있도록 활성화해야 합니다.

CloudWatch 이상 탐지 사용

지표에 대해 이상 탐지를 활성화하면 CloudWatch가 통계 및 기계 학습 알고리즘을 적용합니다. 이러한 알고리즘은 시스템 및 애플리케이션의 지표를 지속적으로 분석하고, 정상 기준을 결정하며, 최소한의 사용자 개입으로 이상을 나타냅니다.

알고리즘은 이상 탐지 모델을 생성합니다. 모델은 정상 지표 동작을 나타내는 예상 값의 범위를 생성합니다.

기댓값 모델을 두 가지 방법으로 사용할 수 있습니다.

- 지표의 예상 값을 기반으로 이상 탐지 경보를 생성합니다. 이러한 유형의 경보에는 경보 상태를 확인하기 위한 정적 임계값이 없습니다. 대신 이상 탐지 모델을 기반으로 지표의 값을 예상 값과 비교합니다.

  지표 값이 예상 값의 밴드보다 높거나, 밴드보다 낮거나, 두 경우에 모두 해당할 때 경보를 트리거하도록 선택할 수 있습니다.

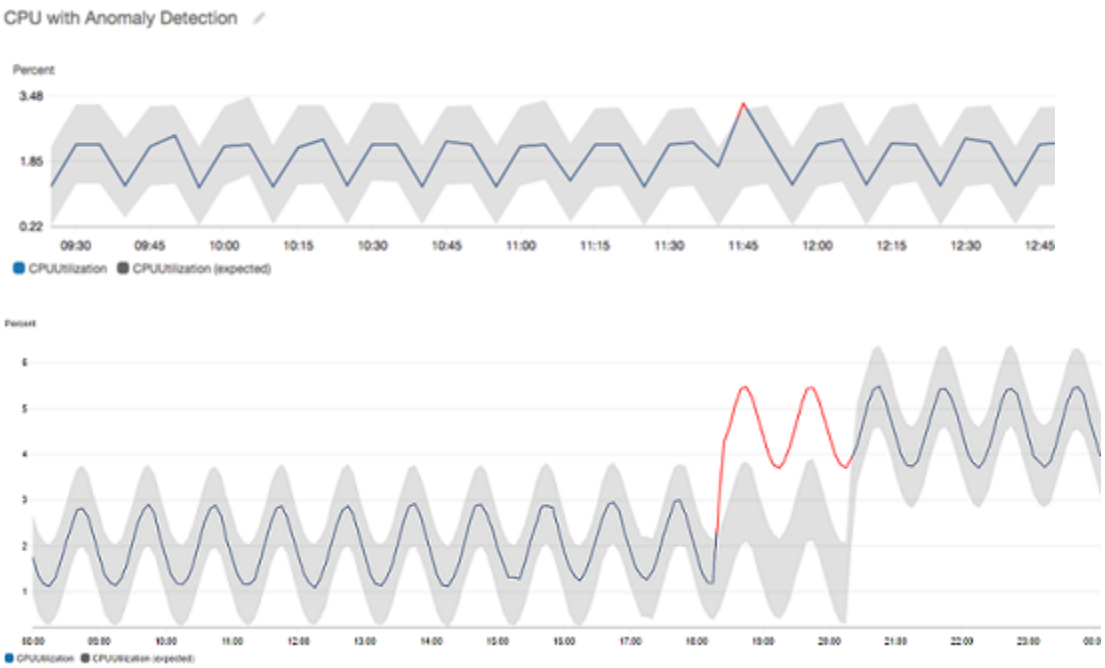
  자세한 내용은 이상 탐지에 기반한 CloudWatch 경보 생성 (p. 76) 단원을 참조하십시오.
- 지표 데이터의 그래프를 볼 때 예상 값을 그래프에 밴드로 오버레이합니다. 이렇게 하면 그래프의 어떤 값이 정상 범위를 벗어났는지 시각적으로 알 수 있습니다. 자세한 내용은 그래프 작성 (p. 41) 단원을 참조하십시오.

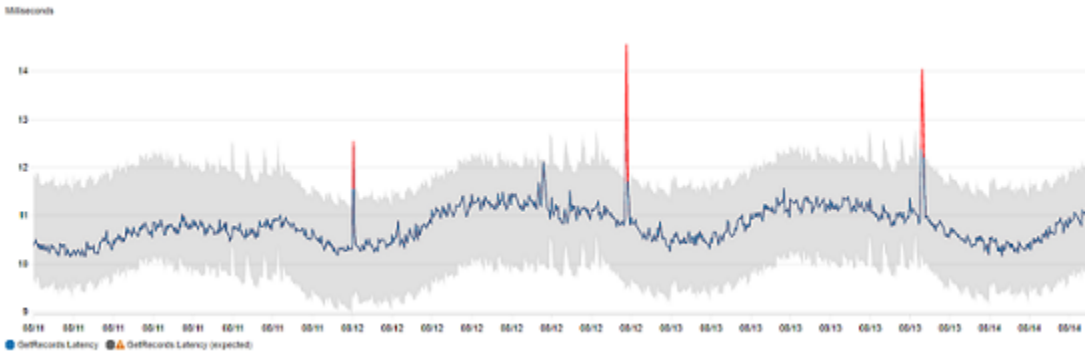
  AWS Management 콘솔, AWS CLI, AWS CloudFormation 또는 AWS SDK를 사용하여 이상 탐지를 활성화할 수 있습니다. AWS에서 판매한 지표에 대해 및 사용자 정의 지표에 대해 이상 탐지를 활성화할 수 있습니다.

  또한, `ANOMALY_DETECTION_BAND` 지표 수학 함수가 있는 `GetMetricData` API 요청을 사용하여 모델 밴드의 상위 값과 하위 값을 검색할 수 있습니다. 자세한 내용은 GetMetricData 단원을 참조하십시오.

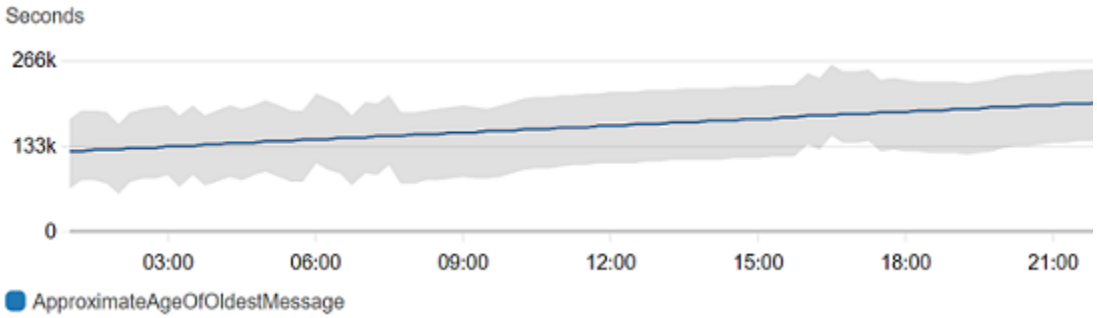
이상 탐지가 있는 그래프에서 예상 값의 범위는 회색 밴드로 표시됩니다. 지표의 실제 값이 이 밴드를 초과하면 해당 시간 동안 빨간색으로 표시됩니다.

이상 탐지 알고리즘은 지표의 계절성 및 추세 변화를 설명합니다. 계절성 변화는 다음 예와 같이 시간별, 일별 또는 주별일 수 있습니다.

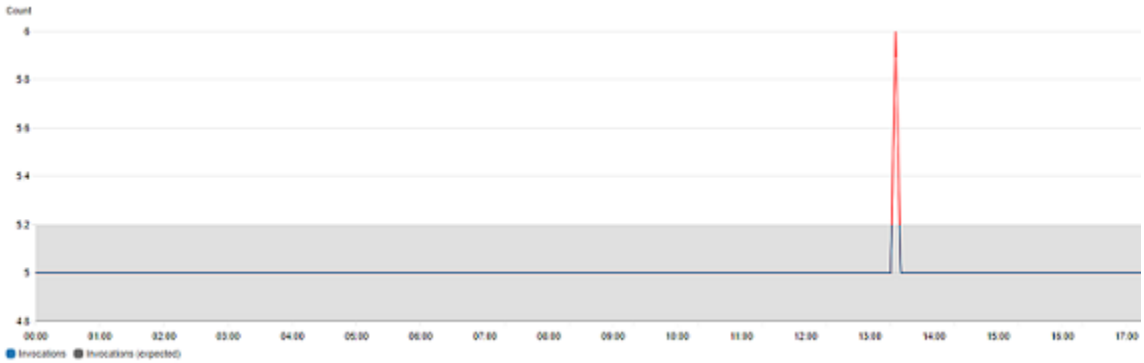




장거리 추세는 하향 또는 상향 추세가 될 수 있습니다.



이상 탐지는 플랫 패턴이 있는 지표에서도 잘 작동합니다.



CloudWatch 이상 탐지 작동 방식

지표에 대해 이상 탐지를 활성화하면, CloudWatch에서는 지표의 이전 데이터에 기계 학습 알고리즘을 적용하여 지표 예상 값의 모델을 생성합니다. 모델은 지표의 추세와 시간별, 일별 및 주별 패턴을 모두 평가합니다. 이 알고리즘은 최대 2주의 지표 데이터를 교육할 수 있지만, 지표에 2주분의 데이터가 없더라도 지표에 이상 탐지를 활성화할 수 있습니다.

CloudWatch에서 지표 값의 "정상" 범위를 결정하기 위해 모델과 함께 사용하는 이상 탐지 임계값에 대한 값을 지정합니다. 이상 탐지 임계값의 값이 높을수록 "정상" 값의 밴드가 두꺼워집니다.

기계 학습 모델은 지표 및 통계에 고유합니다. 예를 들어, `AVG` 통계를 사용한 지표에 대한 이상 탐지를 활성화하면, 해당 모델은 `AVG` 통계에 고유합니다.

모델을 생성하면 지표의 최신 데이터를 이용하여 계속 자동으로 업데이트됩니다.

지표에 이상 탐지를 활성화한 다음 모델을 교육할 때, 지표에 특정 기간을 사용하지 않도록 제외할 수 있습니다. 이렇게 하면, 모델 교육을 할 때 배포 또는 다른 이상 이벤트 등을 제외할 수 있어 가장 정확한 모델을 생성할 수 있습니다.

경보용 이상 탐지 모델을 사용하면 AWS 계정에 비용이 청구됩니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

Contributor Insights를 사용하여 높은 카디널리티 데이터 분석

CloudWatch Contributor Insights is in open preview. The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at contributorinsightsfeedback@amazon.com

Contributor Insights를 사용하면 로그 데이터를 분석하고 기고자 데이터를 표시하는 시계열을 생성할 수 있습니다. 상위 N개의 기고자, 총 고유 기고자 수 및 사용량에 대한 지표를 볼 수 있습니다. 이를 통해 시스템 성능에 영향을 미치는 사람 또는 대상을 이해하고 최고의 대화자를 찾을 수 있습니다. 예를 들어 잘못된 호스트를 찾거나 사용량이 가장 많은 네트워크 사용자를 식별하거나 가장 많은 오류를 생성하는 URL을 찾을 수 있습니다.

CloudWatch Logs 로그 이벤트를 평가하는 규칙을 생성합니다. 규칙은 `IpAddress`와 같이 기고자 정의에 사용할 로그 필드를 정의합니다. 또한 로그 데이터를 필터링하여 개별 기고자의 동작을 찾고 분석할 수 있습니다.

CloudWatch는 다른 AWS 서비스의 지표를 분석하는 데 사용할 수 있는 기본 제공 규칙도 제공합니다. 현재 Amazon DynamoDB에 기본 제공 규칙을 사용할 수 있습니다.

모든 규칙은 수신 데이터를 실시간으로 분석합니다.

> **Note**
>
> Contributor Insights를 사용하면 로그 이벤트와 일치하는 규칙이 발생할 때마다 비용이 발생합니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

주제

- Contributor Insights 규칙 생성 (p. 126)
- Contributor Insights 규칙 구문 (p. 128)
- Contributor Insights 규칙 예제 (p. 131)
- Contributor Insights 보고서 보기 (p. 133)
- 규칙에 의해 생성된 지표 그래프 작성 (p. 133)
- Contributor Insights 기본 제공 규칙 사용 (p. 135)

Contributor Insights 규칙 생성

CloudWatch Contributor Insights is in open preview. The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at contributorinsightsfeedback@amazon.com

규칙을 생성하여 로그 데이터를 분석할 수 있습니다. JSON 또는 CLF(일반적 로그 형식)의 모든 로그를 평가할 수 있습니다. 여기에는 Amazon VPC 흐름 로그, Amazon Route 53 DNS 쿼리 로그, Amazon ECS 컨테이너 로그 및 AWS CloudTrail, Amazon SageMaker, Amazon RDS, AWS AppSync 및 API 게이트웨이의 로그와 같은 AWS 서비스의 로그뿐만 아니라 이러한 형식 중 하나를 따르는 사용자 지정 로그도 포함됩니다.

규칙에서 필드 이름이나 값을 지정할 때 일치하는 모든 항목은 대/소문자를 구분합니다.

Contributor Insights에서 규칙을 생성하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Contributor Insights를 선택합니다.
3. [Create rule]을 선택합니다.

   규칙 생성 마법사가 나타납니다.
4. 규칙 이름에 이름을 지정합니다. 유효한 문자는 A-Z, a-z, 0-9, "-", "_" 및 "."입니다.
5. 마법사를 사용하거나 Syntax(구문) 탭을 선택하고 규칙 구문을 수동으로 지정하여 규칙 생성을 완료할 수 있습니다.

   마법사를 계속 사용하려면 다음을 수행합니다.

   a. Log group(s)(로그 그룹)에서 규칙을 모니터링할 로그 그룹을 선택합니다. 선택적으로 Select by prefix match(접두사 일치 항목으로 선택)를 선택하고 해당 접두사를 입력하여 이름이 특정 문자열로 시작하는 모든 로그 그룹을 선택할 수 있습니다.

      Select by prefix match(접두사 일치 항목으로 선택)를 사용할 경우 접두사와 일치하고 규칙에 의해 분석되는 로그 그룹의 수에 주의하십시오. 규칙과 일치하는 각 로그 이벤트에 대해 요금이 발생합니다. 의도한 것보다 많은 로그 그룹을 실수로 검색하면 예기치 않은 요금이 발생할 수 있습니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

   b. 로그 형식에서 JSON 또는 CLF를 선택합니다.

   c. Contribution(기고), 키에 보고할 기고자 유형을 입력합니다. 보고서에는 이 기고자 유형에 대한 상위 N개의 값이 표시됩니다.

      유효한 항목은 값이 있는 모든 로그 필드입니다. 예를 들면 **`requestId`**, **`sourceIPaddress`** 및 **`containerID`**입니다.

      특정 로그 그룹에서 로그의 로그 필드 이름을 찾는 방법에 대한 자세한 내용은 로그 필드 찾기 (p. 128) 단원을 참조하십시오.

      1KB보다 큰 키는 1KB 단위로 잘립니다.

   d. 선택적으로 Contribution(기고), 키에 대한 값을 더 추가합니다. 규칙에 최대 4개의 키를 포함할 수 있습니다. 두 개 이상의 키를 입력하면 보고서의 기고자는 키의 고유한 값 조합으로 정의됩니다. 예를 들어 세 개의 키를 지정하면 세 개의 키에 대한 각각의 고유한 값 조합이 고유한 기고자로 계산됩니다.

   e. 선택적으로 로그 이벤트의 발생 횟수 대신 숫자 로그 필드 값으로 기고자의 순위를 매기려면 Contribution(기고), 값 필드를 사용합니다. 기고자 순위를 결정하기 위해 합계할 숫자 로그 필드의 이름을 입력합니다. 예를 들어 네트워크를 통해 가장 많은 바이트를 전송하는 소스 IP 주소를 찾으려면 **`bytes`**가 로그 이벤트의 해당 필드에 대한 올바른 키워드라고 가정하여 **`bytes`**를 값으로 추가합니다.

   f. 결과를 더 좁은 범위로 필터링하려면 필터 추가를 선택합니다. Match(일치)에 필터링 기준으로 사용할 로그 필드를 입력합니다. 그런 다음 조건에서 비교 연산자를 선택하고 이 필드에서 필터링할 값을 입력합니다. 값에서 **`*`**를 와일드카드로 사용할 수 있습니다.

      예를 들어, 오류가 포함된 Apache 로그의 로그 이벤트만 분석하려는 경우 **`RESPONSE_CODE`**를 Match(일치)로 지정하고 EqualTo를 조건으로 지정한 다음 **`5**`**를 필터링할 값으로 입력합니다.

      규칙에 필터를 4개까지 추가할 수 있습니다. 여러 필터는 AND 논리로 결합되므로 모든 필터와 일치하는 로그 이벤트만 평가됩니다.

   g. Aggregate on(집계 기준)에서 COUNT 또는 SUM을 선택합니다. COUNT를 선택하면 기고자 순위가 발생 횟수를 기준으로 합니다. SUM을 선택하면 Contribution(기고), 값에 지정한 필드 값의 집계된 합계를 기준으로 순위가 결정됩니다.
6. 마법사를 사용하는 대신 규칙을 JSON 객체로 입력하려면 다음을 수행합니다.

   a. Syntax(구문) 탭을 선택합니다.
   b. Rule body(규칙 본문)에 규칙의 JSON 객체를 입력합니다. 규칙 구문에 대한 자세한 내용은 Contributor Insights 규칙 구문 (p. 128) 단원을 참조하십시오.
7. 규칙을 활성화된 또는 비활성화된 상태로 생성할지 여부를 선택합니다. 규칙을 활성화하도록 선택하면 즉시 데이터 분석이 시작됩니다. 활성화된 규칙을 실행하면 비용이 발생합니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

   Contributor Insights는 규칙이 생성된 후 새 로그 이벤트만 분석합니다. 규칙은 이전에 CloudWatch Logs에서 처리한 로그 이벤트를 처리할 수 없습니다.
8. 생성을 선택합니다.

생성한 규칙을 비활성화, 활성화 또는 삭제할 수 있습니다.

Contributor Insights에서 규칙을 활성화, 비활성화 또는 삭제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 Contributor Insights를 선택합니다.
3. 왼쪽의 규칙 목록에서 단일 규칙 옆에 있는 확인란을 선택합니다.

   기본 제공 규칙은 AWS 서비스에서 생성되며 편집, 비활성화 또는 삭제할 수 없습니다.
4. 작업을 선택한 다음 원하는 옵션을 선택합니다.

로그 필드 찾기

규칙을 생성할 때 로그 그룹의 로그 항목에 있는 필드의 이름을 알아야 합니다.

로그 그룹에서 로그 필드를 찾으려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창의 로그에서 Insights를 선택합니다.
3. 쿼리 편집기 위에서 쿼리할 로그 그룹을 하나 이상 선택합니다.

   로그 그룹을 선택하면 CloudWatch Logs가 로그 그룹 내 데이터에서 필드를 자동으로 검색하고 오른쪽 창에 있는 Discovered fields(검색된 필드)에 표시합니다.

Contributor Insights 규칙 구문

> CloudWatch Contributor Insights is in open preview. The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at contributorinsightsfeedback@amazon.com

이 단원에서는 Contributor Insights 규칙의 구문에 대해 설명합니다. JSON 블록을 입력하여 규칙을 생성하는 경우에만 이 구문을 사용합니다. 마법사를 사용하여 규칙을 생성하는 경우 구문을 알 필요가 없습니다. 마법사를 사용하여 규칙을 생성하는 방법에 대한 자세한 내용은 Contributor Insights 규칙 생성 (p. 126) 단원을 참조하십시오.

이벤트 필드 이름 및 값을 기록하기 위한 규칙과 일치하는 모든 항목은 대/소문자를 구분합니다.

다음 예제에서는 JSON 로그의 구문을 보여줍니다.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*",
        "Log-group-name2"
    ],
    "LogFormat": "JSON",
    "Contribution": {
        "Keys": [
            "$.ip"
        ],
        "ValueOf": "$.requestBytes",
        "Filters": [
            {
                "Match": "$.httpMethod",
                "In": [
                    "PUT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Contributor Insights 규칙의 필드

Schema

CloudWatch Logs 데이터를 분석하는 규칙에 대한 `Schema`의 값은 항상 `{"Name": "CloudWatchLogRule", "Version": 1}`이어야 합니다.

LogGroupNames

문자열 배열입니다. 배열의 각 요소에 대해 선택적으로 문자열 끝에 `*`를 사용하여 해당 접두사로 시작하는 이름을 가진 모든 로그 그룹을 포함할 수 있습니다.

로그 그룹 이름과 함께 와일드카드를 사용할 때는 주의해야 합니다. 규칙과 일치하는 각 로그 이벤트에 대해 요금이 발생합니다. 의도한 것보다 많은 로그 그룹을 실수로 검색하면 예기치 않은 요금이 발생할 수 있습니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

LogFormat

유효 값은 `JSON` 및 `CLF`입니다.

Contribution

이 객체에는 최대 4개의 멤버가 있는 `Keys` 배열, 선택적으로 단일 `ValueOf` 및 선택적으로 최대 4개의 `Filters`가 있는 배열이 포함됩니다.

키

기고자를 분류하는 차원으로 사용될 최대 4개의 로그 필드의 배열입니다. 두 개 이상의 키를 입력하면 키에 대한 각각의 고유한 값 조합이 고유한 기고자로 계산됩니다. 필드는 JSON 속성 형식 표기법을 사용하여 지정해야 합니다.

ValueOf

(선택 사항) `SUM`을 `AggregateOn`의 값으로 지정하는 경우에만 이 옵션을 지정합니다. `ValueOf`는 숫자 값을 갖는 로그 필드를 지정합니다. 이 유형의 규칙에서 기고자 순위는 로그 항목에서의 발생 횟수 대신 이 필드 값의 합계로 결정됩니다. 예를 들어 기고자를 일정 기간 동안 총 `BytesSent`로 정렬하려는 경우 `ValueOf`를 `BytesSent`로 설정하고 `AggregateOn`에 `Sum`을 지정합니다.

필터

(선택 사항) 보고서에 포함된 로그 이벤트의 범위를 좁히기 위해 최대 4개의 필터가 있는 배열을 지정합니다. 여러 필터를 지정하면 Contributor Insights는 논리 AND 연산자를 사용하여 필터를 평가합니다. 이 옵션을 사용하여 검색에서 관련 없는 로그 이벤트를 필터링하거나 단일 기고자를 선택하여 해당 동작을 분석할 수 있습니다.

배열의 각 멤버는 `Match` 필드와 사용할 일치하는 연산자 유형을 나타내는 필드를 포함해야 합니다.

`Match` 필드는 필터에서 평가할 로그 필드를 지정합니다. 로그 필드는 JSON 속성 형식 표기법을 사용하여 지정됩니다.

일치하는 연산자 필드는 `In`, `NotIn`, `StartsWith`, `GreaterThan`, `LessThan`, `EqualTo`, `NotEqualTo` 또는 `IsPresent` 중 하나여야 합니다. 연산자 필드가 `In`, `NotIn` 또는 `StartsWith`인 경우 확인할 문자열 값의 배열이 뒤에 옵니다. Contributor Insights는 OR 연산자를 사용하여 문자열 값의 배열을 평가합니다. 배열은 최대 10개의 문자열 값을 포함할 수 있습니다.

연산자 필드가 `GreaterThan`, `LessThan`, `EqualTo` 또는 `NotEqualTo`인 경우 비교할 단일 숫자 값이 뒤에 옵니다.

연산자 필드가 `IsPresent`인 경우 뒤에 `true` 또는 `false`가 옵니다. 이 연산자는 로그 이벤트에 지정된 로그 필드가 있는지 여부에 따라 로그 이벤트를 일치시킵니다. `isPresent`는 JSON 속성의 리프 노드에 있는 값에서만 작동합니다. 예를 들어 `c-count`와 일치하는 항목을 찾는 필터는 `details.c-count.c1` 값이 있는 로그 이벤트를 평가하지 않습니다.

필터 예제는 다음을 참조하십시오.

```
{"Match": "$.httpMethod", "In": [ "PUT", ] }
{"Match": "$.StatusCode", "EqualTo": 200 }
{"Match": "$.BytesReceived", "GreaterThan": 10000}
{"Match": "$.eventSource", "StartsWith": [ "ec2", "ecs" ] }
```

AggregateOn

유효 값은 `COUNT` 및 `SUM`입니다. 발생 횟수 또는 `ValueOf` 필드에 지정된 필드 값의 합계를 기준으로 보고서를 집계할지 여부를 지정합니다.

JSON 속성 형식 표기법

`Keys`, `ValueOf` 및 `Match` 필드는 점 표기법이 있는 JSON 속성 형식을 따르며 여기서 `$`는 JSON 객체의 루트를 나타냅니다. 그 뒤에는 마침표와 하위 속성의 이름을 가진 영숫자 문자열이 옵니다. 여러 속성 레벨이 지원됩니다.

다음 목록에서는 JSON 속성 형식의 유효한 예를 보여 줍니다.

```
$.userAgent
$.endpoints[0]
$.users[1].name
$.requestParameters.instanceId
```

CLF 로그에 대한 규칙의 추가 필드

CLF(일반적 로그 형식) 로그 이벤트에는 JSON과 같은 필드에 대한 이름이 없습니다. Contributor Insights 규칙에 사용할 필드를 제공하기 위해 CLF 로그 이벤트를 인덱스가 `1`부터 시작하는 배열로 취급할 수 있습니다. 예를 들어 첫 번째 필드를 **`"1"`**로, 두 번째 필드를 **`"2"`**로 지정할 수 있습니다.

CLF 로그에 대한 규칙을 읽기 쉽게 만들려면 `Fields`를 사용합니다. 이렇게 하면 CLF 필드 위치에 대한 이름 지정 별칭을 제공할 수 있습니다. 예를 들어 위치 "4"가 IPAddress가 되도록 지정할 수 있습니다. 지정한 후에는 `IpAddress`를 규칙에서 `Keys`, `ValueOf` 및 `Filters`의 속성으로 사용할 수 있습니다.

다음은 `Fields` 필드를 사용하는 CLF 로그에 대한 규칙의 예입니다.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "IpAddress",
        "7": "StatusCode"
    },
    "Contribution": {
        "Keys": [
            "IpAddress"
        ],
        "Filters": [
            {
                "Match": "StatusCode",
                "EqualTo": 200
            }
        ]
    },
    "AggregateOn": "Count"
}
```

Contributor Insights 규칙 예제

이 단원에는 Contributor Insights 규칙의 사용 사례를 보여주는 예제가 포함되어 있습니다.

VPC 흐름 로그: 소스 및 대상 IP 주소별 바이트 전송

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "srcaddr",
        "5": "dstaddr",
        "10": "bytes"
    },
    "Contribution": {
        "Keys": [
            "srcaddr",
            "dstaddr"
        ],
        "ValueOf": "bytes",
        "Filters": []
    },
    "AggregateOn": "Sum"
}
```

VPC 흐름 로그: HTTPS 요청 수가 가장 많음

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "5": "destination address",
        "7": "destination port",
        "9": "packet count"
    },
    "Contribution": {
        "Keys": [
            "destination address"
        ],
        "ValueOf": "packet count",
        "Filters": [
            {
                "Match": "destination port",
                "EqualTo": 443
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

VPC 흐름 로그: 거부된 TCP 연결

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
```

```
    "AggregateOn": "Sum"
}
```

Contributor Insights 보고서 보기

보고서 데이터의 그래프와 규칙에 의해 발견된 기고자의 순위가 매겨진 목록을 보려면 다음 단계를 따르십시오.

규칙 보고서를 보려면

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

   왼쪽에는 계정의 규칙 목록이 있습니다.
3. 규칙 목록에서 규칙 이름을 선택합니다.

   그래프는 지난 3시간 동안의 규칙 결과를 표시합니다. 그래프 아래의 표에는 상위 10개의 기고자가 나와 있습니다.
4. 표에 표시된 기고자 수를 변경하려면 그래프 상단에 있는 Top 10 contributors(상위 10개의 기고자)를 선택합니다.
5. 단일 기고자의 결과만 표시하도록 그래프를 필터링하려면 표 범례에서 해당 기고자를 선택합니다. 모든 기고자를 다시 표시하려면 범례에서 동일한 기고자를 다시 선택합니다.
6. 보고서에 표시된 시간 범위를 변경하려면 그래프 상단에서 15m, 30m, 1h, 2h, 3h 또는 custom(사용자 지정)을 선택합니다.

   보고서의 최대 시간 범위는 24시간이지만 최대 15일 전에 발생한 24시간 기간을 선택할 수 있습니다. 과거의 기간을 선택하려면 custom(사용자 지정), absolute(절대값)를 선택한 다음 기간을 지정합니다.
7. 기고자의 집계 및 순위에 사용되는 기간의 길이를 변경하려면 그래프 상단에서 period(기간)를 선택합니다. 더 긴 기간을 보면 일반적으로 스파이크가 거의 없는 더 부드러운 보고서가 표시됩니다. 더 짧은 기간을 선택하면 스파이크가 나타날 가능성이 더 큽니다.
8. 이 그래프를 CloudWatch 대시보드에 추가하려면 화면 상단에 있는 대시보드에 추가를 선택합니다.
9. 이 보고서의 로그 그룹이 쿼리 상자에 미리 로드된 상태로 CloudWatch Logs Insights 쿼리 창을 열려면 로그 보기를 선택합니다.
10. 보고서 데이터를 클립보드나 CSV 파일로 내보내려면 내보내기를 선택합니다.

규칙에 의해 생성된 지표 그래프 작성

Contributor Insights는 지표 수학 함수 `INSIGHT_RULE_METRIC`을 제공합니다. 이 함수를 사용하면 Contributor Insights 보고서의 데이터를 CloudWatch 콘솔의 지표 탭에 있는 그래프에 추가할 수 있습니다. 이 수학 함수를 기반으로 경보를 설정할 수도 있습니다. 지표 수학 함수에 대한 자세한 내용은 지표 수식 사용 (p. 49) 단원을 참조하십시오.

이 지표 수학 함수를 사용하려면 `cloudwatch:GetMetricData` 및 `cloudwatch:GetInsightRuleReport` 권한이 모두 있는 계정에 로그인해야 합니다.

구문은 `INSIGHT_RULE_METRIC(`*`ruleName`*`, `*`metricName`*`)`입니다. *`ruleName`*은 Contributor Insights 규칙의 이름이고 *`metricName`*은 다음 목록의 값 중 하나입니다. *`metricName`*의 값은 수학 함수가 반환하는 데이터의 유형을 결정합니다.

- `UniqueContributors` — 각 데이터 포인트에 대한 고유 기고자 수입니다.
- `MaxContributorValue` — 각 데이터 포인트에 대한 최상위 기고자의 값입니다. 그래프의 각 데이터 포인트에 대해 기고자의 ID가 변경될 수 있습니다.

이 규칙이 COUNT로 집계되는 경우 각 데이터 포인트의 최상위 기고자는 해당 기간에 가장 많이 발생한 기고자입니다. 규칙이 SUM으로 집계되는 경우 최상위 기고자는 해당 기간 동안 규칙의 `Value`에 의해 지정된 로그 필드에서 가장 큰 합계를 가진 기고자입니다.

- `SampleCount` — 규칙과 일치하는 데이터 포인트의 수입니다.
- `Sum` — 해당 데이터 포인트가 나타내는 기간 동안 모든 기고자 값의 합계입니다.
- `Minimum` — 해당 데이터 포인트가 나타내는 기간 동안 단일 관측치의 최소값입니다.
- `Maximum` — 해당 데이터 포인트가 나타내는 기간 동안 단일 관측치의 최대값입니다.
- `Average` — 해당 데이터 포인트가 나타내는 기간 동안 모든 기고자의 평균값입니다.

Contributor Insights 지표 데이터에 대한 경보 설정

`INSIGHT_RULE_METRIC`을 사용하여 Contributor Insights에 의해 생성된 지표에 경보를 설정할 수 있습니다. 예를 들어 거부된 TCP 연결의 비율을 기준으로 경보를 생성할 수 있습니다. 먼저 다음과 같은 두 가지 규칙을 생성합니다.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
```

```
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            }
    "AggregateOn": "Sum"
}
```

그런 다음 콘솔의 지표 탭에서 다음 지표 수학 표현식을 사용하여 그래프를 생성합니다.

```
e1 INSIGHT_RULE_METRIC("RejectedConnectionsRule", "Sum")
e2 INSIGHT_RULE_METRIC("TotalConnectionsRule", "Sum")
e3 (e1/e2)*100
```

지표 그래프 작성 및 지표 수학 함수 사용에 대한 자세한 내용은 CloudWatch 그래프에 수학 표현식 추가 (p. 50) 단원을 참조하십시오.

이 예제에서 `e3` 표현식은 거부된 연결의 백분율을 반환합니다. 20%의 연결이 거부될 때 알림을 받으려면 **20**을 임계값으로 설정하여 해당 표현식에 경보를 설정할 수 있습니다. 지표 탭에서 보고 있는 지표에 경보를 설정하려면 경보를 표시할 지표 행에서 경보 아이콘을 선택합니다. 경보 아이콘은 종처럼 보입니다.

지표 그래프 작성 및 지표 수학 함수 사용에 대한 자세한 내용은 CloudWatch 그래프에 수학 표현식 추가 (p. 50) 단원을 참조하십시오.

Contributor Insights 기본 제공 규칙 사용

CloudWatch Contributor Insights is in open preview. The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at contributorinsightsfeedback@amazon.com

다른 AWS 서비스는 해당 AWS 서비스의 지표를 평가하는 기본 제공 Contributor Insights 규칙을 생성합니다. 현재 Amazon DynamoDB는 기본 제공 규칙을 지원합니다.

자세한 내용은 Contributor Insights for Amazon DynamoDB를 참조하십시오.

Container Insights 사용

CloudWatch Container Insights를 사용해 컨테이너식 애플리케이션 및 마이크로서비스의 지표 및 로그를 수집하고 집계하며 요약할 수 있습니다. Container Insights는 Amazon Elastic Container Service와 Amazon Elastic Kubernetes Service, 그리고 Amazon EC2의 Kubernetes 플랫폼에 사용할 수 있습니다. 이 지표에는 CPU, 메모리, 디스크, 네트워크 같은 리소스 사용률이 포함되어 있습니다. 또한 Container Insights는 컨테이너 재시작 오류 같은 진단 정보를 제공하여 문제를 격리하고 신속하게 해결할 수 있도록 도와줍니다. Container Insights에서 수집한 지표에 대해 CloudWatch 경보를 설정할 수도 있습니다.

Container Insights에서 수집한 지표는 CloudWatch 자동 대시보드에서 사용할 수 있습니다. CloudWatch Logs Insights를 사용해 컨테이너 성능 및 로그를 분석하고 이와 관련해 발생하는 문제를 해결할 수 있습니다.

운영 데이터는 성능 로그 이벤트로서 수집됩니다. 이들은 정형 JSON 스키마를 사용하는 항목이기 때문에 카디널리티가 높은 데이터를 적정 규모로 수집 및 저장할 수 있습니다. 이러한 데이터를 토대로 CloudWatch는 집계 지표를 클러스터, 노드, Pod, 작업 및 서비스 수준에서 CloudWatch 지표로 생성합니다.

Container Insights에서 수집된 지표는 사용자 지정 지표로 청구됩니다. CloudWatch 요금에 대한 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

Amazon EKS 및 Kubernetes에서 Container Insights는 컨테이너화된 버전의 CloudWatch 에이전트를 사용하여 클러스터에서 실행 중인 모든 컨테이너를 검색합니다. 이어서 성능 스택의 모든 계층에서 성능 데이터를 수집합니다.

Container Insights는 자체적으로 수집하는 로그 및 지표에 대해 고객 마스터 키(CMK)를 이용한 암호화를 지원합니다. 이 암호화를 활성화하려면 Container Insights 데이터를 수신하는 로그 그룹에 대한 KMS 암호화를 수동으로 활성화해야 합니다. 그러면 Container Insights에서는 제공된 CMK를 사용해 이 데이터를 암호화하게 됩니다. 대칭 CMK만 지원됩니다. 비대칭 CMK를 사용하여 로그 그룹을 암호화하지 마십시오.

자세한 내용은 AWS KMS를 사용하여 CloudWatch Logs에서 로그 데이터 암호화 단원을 참조하십시오.

지원되는 플랫폼

Container Insights는 Amazon EC2 인스턴스의 Amazon Elastic Container Service, Amazon Elastic Kubernetes Service 및 Kubernetes 플랫폼에 사용할 수 있습니다.

- Amazon ECS의 경우 Container Insights는 Linux와 Windows Server 인스턴스 모두에서 클러스터, 작업 및 서비스 레벨의 지표를 수집합니다. Linux 인스턴스에서만 인스턴스 레벨에서 지표를 수집할 수 있습니다.

  Amazon ECS의 경우 네트워크 지표는 `bridge` 네트워크 모드에서 컨테이너에 대해서만 제공되며 `awsvpc` 네트워크 모드 또는 `host` 네트워크 모드에서는 컨테이너에 대해 제공되지 않습니다.
- Amazon Elastic Kubernetes Service 및 Amazon EC2 인스턴스의 Kubernetes 플랫폼의 경우 Container Insights는 Linux 인스턴스에서만 지원됩니다.
- 현재 AWS Batch.에서는 Container Insights가 지원되지 않습니다.

Amazon ECS 지원 리전

Amazon ECS용 Container Insights는 다음 리전에서 지원됩니다.

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)

- 유럽(런던)
- 유럽(파리)
- 유럽(스톡홀름)
- 아시아 태평양(홍콩)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)

유럽(파리) 또는 남아메리카(상파울루)에서는 AWS Fargate가 지원되지 않습니다.

Amazon EKS 및 Kubernetes 지원 리전

Amazon EKS 및 Kubernetes용 Container Insights는 다음 리전에서 지원됩니다.

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 아시아 태평양(뭄바이)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 남아메리카(상파울루)

주제

- Container Insights 설정 (p. 137)
- Container Insights 지표 보기 (p. 159)
- Container Insights에서 수집된 지표 (p. 161)
- Container Insights 성능 로그 참조 (p. 167)
- Container Insights 문제 해결 (p. 187)
- 자체 CloudWatch 에이전트 도커 이미지 빌드 (p. 188)

Container Insights 설정

Amazon ECS, Amazon EKS 및 Kubernetes에 대한 Container Insights 설정 프로세스는 다릅니다.

주제

- Amazon ECS에서 Container Insights 설정 (p. 138)

- Amazon EKS 및 Kubernetes에서 Container Insights 설정 (p. 145)

Amazon ECS에서 Container Insights 설정

다음 옵션 중 하나 또는 모두를 사용하여 Amazon ECS 클러스터에서 Container Insights를 활성화할 수 있습니다.

- AWS Management 콘솔 또는 AWS CLI를 사용하여 클러스터 레벨, 작업 레벨 및 서비스 레벨 지표 수집을 시작합니다.
- CloudWatch 에이전트를 DaemonSet로 배포하여 Amazon EC2 인스턴스에서 호스팅되는 클러스터에서 인스턴스 레벨 지표 수집을 시작합니다.

주제

- 클러스터 및 서비스 레벨 지표를 위해 Amazon ECS에서 Container Insights 설정 (p. 138)
- Amazon ECS에서 EC2 인스턴스 레벨 지표를 수집하기 위해 CloudWatch 에이전트 배포 (p. 139)

클러스터 및 서비스 레벨 지표를 위해 Amazon ECS에서 Container Insights 설정

신규 및 기존 Amazon ECS 클러스터에서 Container Insights를 활성화할 수 있습니다. Container Insights는 클러스터, 작업 및 서비스 레벨에서 지표를 수집합니다. 기존 클러스터의 경우 AWS CLI를 사용합니다. 새 클러스터의 경우 Amazon ECS 콘솔 또는 AWS CLI를 사용합니다.

Amazon EC2 인스턴스에서 Amazon ECS를 사용 중이고 Container Insights에서 네트워크 및 스토리지 지표를 수집하려는 경우 Amazon ECS 에이전트 버전 1.29를 포함한 AMI를 사용하여 해당 인스턴스를 시작합니다. 에이전트 버전 업데이트에 대한 자세한 내용은 Amazon ECS Container Agent 업데이트를 참조하십시오.

AWS CLI를 사용하여 계정에 생성된 새로운 Amazon ECS 클러스터에 대해 Container Insights를 활성화할 수 있는 계정 수준 권한을 설정할 수 있습니다. 이를 위해 다음 명령을 입력합니다.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

기존 Amazon ECS 클러스터에서 Container Insights 설정

기존 Amazon ECS 클러스터에서 Container Insights를 활성화하려면 다음 명령을 입력합니다. 다음 명령이 작동하려면 AWS CLI 버전 1.16.200 이상을 실행 중이어야 합니다.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

새로운 Amazon ECS 클러스터에서 Container Insights 설정

새로운 Amazon ECS 클러스터에서 Container Insights를 활성화할 수 있는 방법이 두 가지 있습니다. 기본적으로 모든 새 클러스터가 Container Insights에 대해 활성화되도록 Amazon ECS를 구성할 수 있습니다. 그렇지 않으면 새 클러스터를 생성할 때 활성화할 수 있습니다.

AWS Management 콘솔 사용

Container Insights는 모든 새 클러스터에서 기본적으로 활성화하거나 새로 생성하는 개별 클러스터에서 활성화할 수 있습니다.

기본적으로 모든 새 클러스터에서 Container Insights를 활성화하려면

1. https://console.aws.amazon.com/ecs/에서 Amazon ECS 콘솔을 엽니다.

2. 탐색 창에서 Account Settings(계정 설정)를 선택합니다.
3. Container Insights 기본을 활성화하려면 페이지 하단의 확인란을 선택합니다.

모든 새 클러스터에 기본적으로 Container Insights를 활성화하기 위해 앞에 나온 절차를 사용하지 않은 경우 Container Insights가 활성화된 상태에서 다음 단계에 따라 클러스터를 생성하십시오.

Container Insights 활성화되었을 때 클러스터를 생성하려면

1. https://console.aws.amazon.com/ecs/에서 Amazon ECS 콘솔을 엽니다.
2. 탐색 창에서 Clusters를 선택합니다.
3. [Create cluster]를 선택합니다.
4. 다음 페이지에서 다음을 수행합니다.
   a. 클러스터 이름을 지정합니다.
   b. VPC가 없는 경우, 확인란을 선택하여 새로 생성합니다. VPC에 기본값을 사용할 수 있습니다.
   c. 인스턴스 유형을 포함한 다른 필요한 모든 정보를 입력합니다.
   d. Enabled Container Insights(활성화된 Container Insights)를 선택합니다.
   e. Create를 선택합니다.

이제 클러스터에서 작업 정의 생성, 작업 실행, 서비스 시작을 할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- 작업 정의 생성하기
- 작업 실행하기
- 서비스 생성하기

AWS CLI를 사용해 새로운 Amazon ECS 클러스터에서 Container Insights 설정

모든 새 클러스터에서 Container Insights를 기본적으로 활성화하려면 다음 명령을 입력합니다.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

모든 새로운 클러스터에 기본적으로 Container Insights를 활성화하기 위해 앞에 나온 명령을 사용하지 않은 경우, Container Insights 활성화되어 있을 때 새로운 클러스터를 생성하기 위해 다음 명령을 입력합니다. 다음 명령이 작동하려면 AWS CLI 버전 1.16.200 이상을 실행 중이어야 합니다.

```
aws ecs create-cluster --cluster-name myCICluster --settings
 "name=containerInsights,value=enabled"
```

Amazon ECS 클러스터에서 Container Insights 비활성화

기존 Amazon ECS 클러스터에서 Container Insights를 활성화하려면 다음 명령을 입력합니다.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=disabled
```

Amazon ECS에서 EC2 인스턴스 레벨 지표를 수집하기 위해 CloudWatch 에이전트 배포

EC2 인스턴스에서 호스팅되는 Amazon ECS 클러스터에서 인스턴스 레벨 지표를 수집하도록 CloudWatch 에이전트를 배포하려면 기본 구성으로 빠른 시작 설정을 사용하거나 에이전트를 사용자 지정할 수 있도록 수동으로 설치합니다.

두 방법 모두 EC2 시작 유형으로 배포된 Amazon ECS 클러스터가 이미 하나 이상 있어야 합니다. 이러한 방법은 AWS CLI도 설치되어 있다고 가정합니다. 또한 다음 절차에서 명령을 실행하려면 IAMFullAccess 및 AmazonECS_FullAccess 정책이 있는 계정 또는 역할에 로그인해야 합니다.

주제

- AWS CloudFormation을 사용한 빠른 설정 (p. 140)
- 수동 및 사용자 지정 설치 (p. 141)

AWS CloudFormation을 사용한 빠른 설정

빠른 설정을 사용하려면 다음 명령을 입력하여 AWS CloudFormation을 사용해 에이전트를 설치합니다. *cluster-name* 및 *cluster-region*을 Amazon ECS 클러스터의 이름 및 리전으로 바꿉니다.

이 명령은 IAM 역할 CWAgentECSTaskRole 및 CWAgentECSExecutionRole을 생성합니다. 이러한 역할이 계정에 이미 있는 경우 명령을 입력할 때 `ParameterKey=CreateIAMRoles,ParameterValue=True` 대신 `ParameterKey=CreateIAMRoles,ParameterValue=False`를 사용합니다. 그렇지 않으면 명령이 실패합니다.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/
cwagent-ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
 \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

(대안) 사용자 고유의 IAM 역할 사용

CWAgentECSTaskRole 및 CWAgentECSExecutionRole 역할 대신 고유한 사용자 지정 ECS 작업 역할 및 ECS 작업 실행 역할을 사용하려면 먼저 ECS 작업 역할로 사용할 역할에 CloudWatchAgentServerPolicy가 연결되어 있는지 확인합니다. 또한 ECS 작업 실행 역할로 사용할 역할에 CloudWatchAgentServerPolicy 및 AmazonECSTaskExecutionRolePolicy 정책이 모두 연결되어 있는지 확인합니다. 이어서 다음 명령을 입력합니다. 명령에서 *task-role-arn*을 사용자 지정 ECS 작업 역할의 ARN으로 바꾸고 *execution-role-arn*을 사용자 지정 ECS 작업 실행 역할의 ARN으로 바꿉니다.

```
ClusterName=cluster-name
Region=cluster-region
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/
cwagent-ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
 \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=TaskRoleArn,ParameterValue=${TaskRoleArn} \
                 ParameterKey=ExecutionRoleArn,ParameterValue=${ExecutionRoleArn} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

빠른 설정 문제 해결

AWS CloudFormation 스택의 상태를 확인하려면 다음 명령을 입력합니다.

```
ClusterName=cluster-name
```

```
Region=cluster-region
aws cloudformation describe-stacks --stack-name CWAgentECS-$ClusterName-$Region --region
 $Region
```

CREATE_COMPLETE 또는 CREATE_IN_PROGRESS 이외의 StackStatus가 표시되면 스택 이벤트를 확인하여 오류를 찾습니다. 다음 명령을 입력합니다.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name CWAgentECS-$ClusterName-$Region --
region $Region
```

cwagent 데몬 서비스의 상태를 확인하려면 다음 명령을 입력합니다. 출력에서 runningCount가 deployment 섹션의 desiredCount와 같은 것을 볼 수 있습니다. 같지 않은 경우 출력에서 failures 섹션을 확인합니다.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services cwagent-daemon-service --cluster $ClusterName --region
 $Region
```

CloudWatch Logs 콘솔을 사용하여 에이전트 로그를 확인할 수도 있습니다. /ecs/ecs-cwagent-daemon-service 로그 그룹을 찾습니다.

CloudWatch 에이전트에 대한 AWS CloudFormation 스택 삭제

AWS CloudFormation 스택을 삭제해야 하는 경우 다음 명령을 입력합니다.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation delete-stack --stack-name CWAgentECS-${ClusterName}-${Region} --region
 ${Region}
```

수동 및 사용자 지정 설치

이 단원의 단계에 따라 CloudWatch 에이전트를 수동으로 배포하여 EC2 인스턴스에서 호스팅되는 Amazon ECS 클러스터에서 인스턴스 레벨 지표를 수집합니다.

필요한 IAM 역할 및 정책

두 가지 IAM 역할이 필요합니다. 아직 존재하지 않는 경우 생성해야 합니다. 이러한 역할에 대한 자세한 내용은 Amazon ECS 작업 역할 및 Amazon ECS 작업 실행 역할을 참조하십시오.

- CloudWatch 에이전트가 지표를 게시하는 데 사용하는 ECS 작업 역할. 이 역할이 이미 있는 경우 CloudWatchAgentServerPolicy 정책이 연결되어 있는지 확인해야 합니다.
- Amazon ECS 에이전트가 CloudWatch 에이전트를 시작하는 데 사용하는 ECS 작업 실행 역할. 이 역할이 이미 있는 경우 AmazonECSTaskExecutionRolePolicy 및 CloudWatchAgentServerPolicy 정책이 연결되어 있는지 확인해야 합니다.

이러한 역할이 아직 없는 경우 다음 명령을 사용하여 역할을 생성하고 필요한 정책을 연결할 수 있습니다. 이 첫 번째 명령은 ECS 작업 역할을 생성합니다.

```
aws iam create-role --role-name CWAgentECSTaskRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid\":
 \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com\"},
\"Action\": \"sts:AssumeRole\"}]}"
```

위 명령을 입력한 후 명령 출력에서 `Arn` 값을 "TaskRoleArn"으로 기록해 둡니다. 나중에 작업 정의를 생성할 때 사용해야 합니다. 이어서 다음 명령을 입력하여 필요한 정책을 연결합니다.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
 \
    --role-name CWAgentECSTaskRole
```

다음 명령은 ECS 작업 실행 역할을 생성합니다.

```
aws iam create-role --role-name CWAgentECSExecutionRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid\":
 \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com\"},
\"Action\": \"sts:AssumeRole\"}]}"
```

위 명령을 입력한 후 명령 출력에서 `Arn` 값을 "ExecutionRoleArn"으로 기록해 둡니다. 나중에 작업 정의를 생성할 때 사용해야 합니다. 이어서 다음 명령을 입력하여 필요한 정책을 연결합니다.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
 \
    --role-name CWAgentECSExecutionRole

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy \
    --role-name CWAgentECSExecutionRole
```

작업 정의 생성 및 데몬 서비스 시작

작업 정의를 생성하고 이를 사용하여 CloudWatch 에이전트를 데몬 서비스로 시작합니다. 작업 정의를 생성하려면 다음 명령을 입력합니다. 첫 번째 줄에서 자리 표시자를 배포의 실제 값으로 바꿉니다. *logs-region*은 CloudWatch Logs가 위치한 리전이고 *cluster-region*은 클러스터가 위치한 리전입니다. *task-role-arn*은 사용 중인 ECS 작업 역할의 Arn이고 *execution-role-arn*은 ECS 작업 실행 역할의 Arn입니다.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/
latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-ecs-instance-
metric/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-json
```

이어서 다음 명령을 실행하여 데몬 서비스를 시작합니다. *cluster-name* 및 *cluster-region*을 Amazon ECS 클러스터의 이름 및 리전으로 바꿉니다.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

`An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent` 오류 메시지가 표시되면 `cwagent-`

daemon-service라는 데몬 서비스를 이미 생성한 것입니다. 다음 명령을 예로 사용하여 먼저 해당 서비스를 삭제해야 합니다.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

(선택 사항) 고급 구성

선택적으로 SSM을 사용하여 EC2 인스턴스에서 호스팅되는 Amazon ECS 클러스터의 CloudWatch 에이전트에 대한 다른 구성 옵션을 지정할 수 있습니다. 이러한 옵션은 다음과 같습니다.

- `metrics_collection_interval` – CloudWatch 에이전트가 지표를 수집하는 빈도(초)입니다. 기본값은 60입니다. 범위는 1–172,000입니다.
- `endpoint_override` – (선택 사항) 로그를 전송할 다른 엔드포인트를 지정합니다. VPC의 클러스터에서 게시하는 중이고 로그 데이터를 VPC 엔드포인트로 이동하려는 경우 이 작업을 원할 수 있습니다.

  `endpoint_override`의 값은 URL인 문자열이어야 합니다.
- `force_flush_interval` – 로그가 서버로 전송되기 전에 메모리 버퍼에 남아 있을 최대 시간(초)을 지정합니다. 이 필드에 대한 설정과 상관없이 버퍼 내 로그의 크기가 1MB에 도달하면 로그가 즉시 서버로 전송됩니다. 기본 값은 5초입니다.
- `region` – 기본적으로 에이전트는 Amazon ECS 컨테이너 인스턴스가 있는 리전과 동일한 리전에 지표를 게시합니다. 이를 무시하려면 여기에서 다른 리전을 지정합니다. 예, `"region" : "us-east-1"`

다음은 사용자 지정된 구성의 예입니다.

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "ecs": {
                "metrics_collection_interval": 30
            }
        },
        "force_flush_interval": 5
    }
}
```

Amazon ECS 컨테이너에서 CloudWatch 에이전트 구성을 사용자 지정하려면

1. AmazonSSMReadOnlyAccess 정책이 Amazon ECS 작업 실행 역할에 연결되어 있는지 확인합니다. 이렇게 하려면 다음 명령을 입력합니다. 이 예제에서는 Amazon ECS 작업 실행 역할이 CWAgentECSExecutionRole이라고 가정합니다. 다른 역할을 사용하는 경우 다음 명령에서 해당 역할 이름으로 바꿉니다.

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess
    \
           --role-name CWAgentECSExecutionRole
   ```
2. 앞의 예제와 비슷한 사용자 지정된 구성 파일을 생성합니다. 이 파일의 이름을 `/tmp/ecs-cwagent-daemon-config.json`으로 지정합니다.

3. 이 구성을 Parameter Store에 넣으려면 다음 명령을 실행합니다. *cluster-region*을 Amazon ECS 클러스터의 리전으로 바꿉니다. 이 명령을 실행하려면 AmazonSSMFullAccess 정책이 있는 사용자 또는 역할에 로그인해야 합니다.

```
Region=cluster-region
aws ssm put-parameter \
    --name "ecs-cwagent-daemon-service" \
    --type "String" \
    --value "`cat /tmp/ecs-cwagent-daemon-config.json`" \
    --region $Region
```

4. 작업 정의 파일을 로컬 파일(예 `/tmp/cwagent-ecs-instance-metric.json`)로 다운로드합니다.

```
chttps://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/
latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-ecs-
instance-metric/cwagent-ecs-instance-metric.json -o /tmp/cwagent-ecs-instance-
metric.json
```

5. 작업 정의 파일을 수정합니다. 다음 섹션을 삭제합니다.

```
"environment": [
                {
                    "name": "USE_DEFAULT_CONFIG",
                    "value": "True"
                }
            ],
```

해당 섹션을 다음으로 바꿉니다.

```
"secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent-daemon-service"
                }
            ],
```

6. 다음 단계에 따라 에이전트를 데몬 서비스로 다시 시작합니다.

   a. 다음 명령을 실행합니다.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
cat /tmp/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-json
```

   b. 데몬 서비스를 시작하려면 다음 명령을 실행합니다. *cluster-name* 및 *cluster-region*을 Amazon ECS 클러스터의 이름 및 리전으로 바꿉니다.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

`An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent` 오류 메시지가 표시되면 `cwagent-daemon-service`라는 데몬 서비스를 이미 생성한 것입니다. 다음 명령을 예로 사용하여 먼저 해당 서비스를 삭제해야 합니다.

```
ClusterName=cluster-name
Region=Region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

Amazon EKS 및 Kubernetes에서 Container Insights 설정

Amazon EKS 또는 Kubernete에서 Container Insights를 설정하는 전반적인 프로세스는 다음과 같습니다.

1. 필요한 사전 조건을 갖추었는지 확인합니다.
2. CloudWatch에 지표를 전송하려면 Amazon EKS 클러스터 또는 Kubernetes 클러스터에서 CloudWatch 에이전트를 DaemonSet로 설정합니다. 그리고 CloudWatch Logs에 로그를 전송하려면 FluentID를 DaemonSet로 설정합니다.

   이 단계를 빠른 시작 설정의 일부로 수행하거나 별도로 수행할 수 있습니다.
3. (선택 사항) Amazon EKS 제어 플레인 로깅을 설정합니다.
4. (선택 사항) CloudWatch에 StatsD 지표를 전송하려면 클러스터에서 CloudWatch 에이전트를 StatsD 엔드포인트로 설정합니다.

주제

- 사전 조건 확인 (p. 145)
- Amazon EKS에서 Container Insights 빠른 시작 설정 (p. 146)
- 클러스터 지표를 수집하도록 CloudWatch 에이전트 설정 (p. 147)
- DaemonSet으로 FluentD를 설정하여 CloudWatch Logs에 로그 전송 (p. 150)
- (선택 사항) Amazon EKS 제어 플레인 로깅 설정 (p. 154)
- (선택 사항) CloudWatch에 StatsD 지표를 전송하기 위해 CloudWatch 에이전트를 StatsD 엔드포인트로 설정 (p. 154)
- Amazon EKS 및 Kubernetes에서 Container Insights 업데이트 또는 삭제 (p. 157)

사전 조건 확인

Amazon EKS 또는 Kubernetes에 Container Insights를 설치하기 전에 다음을 확인합니다.

- Amazon EKS 및 Kubernetes용 Container Insights를 지원하는 리전 중 하나에 노드가 연결되어 있는 Amazon EKS 클러스터 또는 Kubernetes 클러스터가 작동 중입니다. 지원되는 리전 목록은 Container Insights 사용 (p. 136) 단원을 참조하십시오.
- `kubectl`을 설치하여 실행하고 있습니다. 자세한 내용은 Amazon EKS 사용 설명서의 `kubectl` 설치를 참조하십시오.
- Amazon EKS를 사용하지 않고 AWS 에서 Kubernetes를 실행 중이라면 다음과 같은 사전 조건을 충족해야 합니다.

- Kubernetes 클러스터가 역할 기반 액세스 제어(RBAC)를 지원하는지 확인합니다. 자세한 내용은 Kubernetes 참조 문서의 RBAC 승인 사용을 참조하십시오.
- 여러분의 Kubelet에서는 Webhook 인증 모드를 지원해왔습니다. 자세한 내용은 Kubernetes 참조 문서의 Kubelet 인증/권한 부여를 참조하십시오.
- 컨테이너 실행 시간은 Docker입니다.

또한 Amazon EKS 작업자 노드의 IAM 역할에 정책을 연결하여 CloudWatch으로 지표 및 로그를 전송할 수 있도록 해야 합니다.

작업자 노드의 IAM 역할에 필요한 정책을 추가하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 작업자 노드 인스턴스 중 하나를 선택하고 설명에서 IAM 역할을 선택합니다.
3. IAM 역할 페이지에서 Attach policies(정책 연결)를 선택합니다.
4. 정책 목록에서 CloudWatchAgentServerPolicy 옆의 확인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 이 정책을 찾습니다.
5. 정책 연결을 선택합니다.

Amazon EKS에서 벗어나 Kubernetes 클러스터를 실행하는 경우, 작업자 노드에 연결된 IAM 역할이 없을 수도 있습니다. 이러한 경우에는 먼저 인스턴스에 IAM 역할을 연결하고 이전 단계에서 설명한 대로 정책을 추가해야 합니다. 인스턴스에 정책을 연결하는 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 IAM 역할을 인스턴스에 연결을 참조하십시오.

Amazon EKS 외부에서 Kubernetes 클러스터를 실행하는 중 지표에서 EBS 볼륨 ID를 수집하려면 인스턴스에 연결된 IAM 역할에 다른 정책을 추가해야 합니다. 다음 내용을 인라인 정책으로 추가합니다. 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거를 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Amazon EKS에서 Container Insights 빠른 시작 설정

Container Insights 설정을 완료하려면 이 단원의 빠른 시작 지침을 따르십시오.

또는 클러스터 지표를 수집하도록 CloudWatch 에이전트 설정 (p. 147) 및 DaemonSet으로 FluentD를 설정하여 CloudWatch Logs에 로그 전송 (p. 150) 단원의 지침을 따를 수 있습니다. 이 단원에서는 CloudWatch 에이전트가 Amazon EKS 및 구성과 어떤 방식으로 연동되는지에 관한 추가 세부 정보를 제공합니다. 하지만 추가 설치 단계를 수행해야 합니다.

빠른 시작을 사용해 Container Insights를 배포하려면 다음 명령을 입력하십시오.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/
latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-
monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/{{cluster_name}}/cluster-
name/;s/{{region_name}}/cluster-region/" | kubectl apply -f -
```

이 명령에서 *cluster-name*은 Amazon EKS 또는 Kubernetes 클러스터의 이름이며 *cluster-region*은 로그가 게시되는 리전의 이름입니다. AWS 아웃바운드 데이터 전송 비용을 줄이기 위해 클러스터가 배포되는 리전과 동일한 리전을 사용하는 것이 좋습니다.

예를 들어 `MyCluster`라는 클러스터에서 Container Insights를 배포하고 로그 및 지표를 미국 서부(오레곤)에 게시하려면 다음 명령을 입력합니다.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/
latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-
monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/{{cluster_name}}/
MyCluster/;s/{{region_name}}/us-west-2/" | kubectl apply -f -
```

Container Insights 삭제

빠른 시작 설정을 사용한 후 Container Insights를 제거하려면 다음 명령을 입력하십시오.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/
latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-
monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/{{cluster_name}}/cluster-
name/;s/{{region_name}}/cluster-region/" | kubectl delete -f -
```

클러스터 지표를 수집하도록 CloudWatch 에이전트 설정

Container Insights를 설정하여 지표를 수집하려면 Amazon EKS에서 Container Insights 빠른 시작 설정 (p. 146)의 절차를 따르거나 이 단원의 절차를 따르면 됩니다. 다음 단계에 따라 클러스터에서 지표를 수집할 수 있도록 CloudWatch 에이전트를 설정합니다.

1단계: CloudWatch에 대한 네임스페이스 생성

다음 단계를 통해 CloudWatch에 대해 `amazon-cloudwatch`라는 Kubernetes 네임스페이스를 생성합니다. 이 네임스페이스를 이미 생성했다면 이 단계를 건너뛸 수 있습니다.

CloudWatch에 대한 네임스페이스를 생성하려면

- 다음 명령을 입력합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cloudwatch-namespace.yaml
```

2단계: 클러스터에서 서비스 계정 생성

아직 서비스 계정이 없다면 다음 단계를 통해 CloudWatch 에이전트에 대한 서비스 계정을 생성합니다.

CloudWatch 에이전트에 대한 서비스 계정을 생성하려면

- 다음 명령을 입력합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-serviceaccount.yaml
```

이전 단계를 따르지 않았지만 사용하고자 하는 CloudWatch 에이전트에 대한 서비스 계정이 이미 있는 경우에는 다음 규칙을 가지고 있는지 확인해야 합니다. 뿐만 아니라 Container Insights 설치를 위한 나머지 단계에서 `cloudwatch-agent` 대신 이 서비스 계정의 이름을 사용해야 합니다.

```
rules:
  - apiGroups: [""]
    resources: ["pods", "nodes", "endpoints"]
    verbs: ["watch", "list"]
  - apiGroups: [""]
    resources: ["nodes/proxy"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes/stats", "configmaps", “events”]
    verbs: ["create"]
  - apiGroups: [""]
    resources: ["configmaps"]
    resourceNames: ["cwagent-clusterleader"]
    verbs: ["get", "update"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["get"]
```

3단계 CloudWatch 에이전트에 대한 ConfigMap 생성

다음 단계를 통해 CloudWatch 에이전트에 대한 ConfigMap을 생성합니다.

CloudWatch 에이전트에 대한 ConfigMap 생성하려면

1. 다음 명령을 실행하여 `kubectl` 클라이언트 호스트로 ConfigMap YAML을 다운로드합니다.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-configmap.yaml
   ```

2. 다운로드한 YAML 파일을 다음과 같이 편집합니다.

   - cluster_name – `kubernetes` 섹션에서 `{{cluster-name}}`를 클러스터 이름으로 바꿉니다. `{{}}` 문자를 제거합니다. 대신에 Amazon EKS 클러스터를 사용하고 있는 경우에는 `"cluster_name"` 필드 및 값을 삭제할 수 있습니다. 이렇게 하면 CloudWatch 에이전트가 Amazon EC2 태그에서 클러스터 이름을 삭제합니다.

3. (선택 사항) 다음과 같이 모니터링 요구사항에 따라 ConfigMap을 추가로 변경합니다.

   - metrics_collection_interval – `kubernetes` 섹션에서 에이전트가 지표를 수집하는 간격을 지정할 수 있습니다. 기본값은 60초입니다. Kubelet의 기본 cadvisor 수집 간격은 15초이기 때문에 이 값을 15초 미만으로 설정해서는 안 됩니다.
   - endpoint_override–기본 엔드포인트를 재정의하려면 `logs` 섹션에서 CloudWatch Logs 엔드포인트를 지정할 수 있습니다. VPC의 클러스터에서 게시 중인 데이터를 VPC 종단점으로 이동시키고 싶은 경우에 재정의를 원할 수 있습니다.
   - force_flush_interval– `logs` 섹션에서 CloudWatch Logs에 게시되기 전에 로그 이벤트를 배치화하는 간격을 지정할 수 있습니다. 기본값은 5초입니다.
   - region– 기본적으로 에이전트가 작업자 노드가 위치한 리전에 지표를 게시했습니다. 이를 재정의하기 위해 `"region":"us-west-2"`처럼 `agent` 섹션에서 `region` 필드를 추가할 수 있습니다.
   - StatsD 섹션 – 클러스터의 각 작업자 노드에서 StatsD 리스너로서 CloudWatch Logs 에이전트를 실행하려면 다음 예제와 같이 `metrics` 섹션에 `statsd` 섹션을 추가할 수 있습니다. 이 섹션의 다른 StatsD 옵션에 대한 자세한 내용은 StatsD로 시작하는 사용자 지정 지표 검색 (p. 253) 단원을 참조하십시오.

     ```
     "metrics": {
       "metrics_collected": {
         "statsd": {
           "service_address":":8125"
         }
       }
     ```

```
}
```

JSON 섹션에 대한 전체 예는 다음과 같습니다.

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "kubernetes": {
                "cluster_name": "MyCluster",
                "metrics_collection_interval": 60
            }
        },
        "force_flush_interval": 5,
        "endpoint_override": "logs.us-east-1.amazonaws.com"
    },
    "metrics": {
        "metrics_collected": {
            "statsd": {
                "service_address": ":8125"
            }
        }
    }
}
```

4. 다음 명령을 실행하여 클러스터에서 ConfigMap을 생성합니다.

```
kubectl apply -f cwagent-configmap.yaml
```

4단계: CloudWatch DaemonSet으로서 에이전트를 배치

CloudWatch 에이전트의 설치를 완료하고 컨테이너 지표 수집을 시작하려면 다음 단계를 따르십시오.

DaemonSet으로서 CloudWatch 에이전트를 시작하려면

1. • 클러스터에서 StatsD를 사용하려면 다음 명령을 입력하십시오.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

   • StatsD를 사용하지 않으려면 다음 절차를 따르십시오.

   a. 다음 명령을 실행하여 `kubectl` 클라이언트 호스트에 DaemonSet YAML을 다운로드합니다.

```
curl -O  https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

   b. 다음과 같이 `cwagent-daemonset.yaml` 파일에서 `port` 섹션의 주석 처리를 해제합니다.

```
ports:
  - containerPort: 8125
    hostPort: 8125
    protocol: UDP
```

   c. 다음 명령을 실행하여 클러스터에서 CloudWatch 에이전트를 배포합니다.

```
kubectl apply -f cwagent-daemonset.yaml
```

2. 다음 명령을 실행하여 에이전트가 배포되었는지 확인합니다.

```
kubectl get pods -n amazon-cloudwatch
```

확인이 완료되면 CloudWatch 에이전트가 `/aws/containerinsights/Cluster_Name/performance`라는 로그 그룹을 생성하고 이 로그 그룹에 성능 로그 이벤트를 전송합니다. 또한 StatsD 리스너로서 에이전트를 설정하는 경우, 에이전트는 애플리케이션 Pod가 예약된 노드의 IP 주소를 통해 포트 8125에서 StatsD 지표를 수신합니다.

문제 해결

에이전트에서 배포가 올바르게 되지 않으면 다음을 수행해 보십시오.

- 다음 명령을 실행하여 Pod 목록을 가져옵니다.

```
kubectl get pods -n amazon-cloudwatch
```

- 다음 명령을 실행하고 출력 하단에서 이벤트를 확인합니다.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- 다음 명령을 실행하여 로그를 확인합니다.

```
kubectl logs pod-name  -n amazon-cloudwatch
```

DaemonSet으로 FluentD를 설정하여 CloudWatch Logs에 로그 전송

FluentD를 설정하여 컨테이너에서 로그를 수집하려면 Amazon EKS에서 Container Insights 빠른 시작 설정 (p. 146)의 절차를 따르거나 이 단원의 절차를 따르면 됩니다. 다음 단계에서는 DaemonSet으로 FluentD를 설정하여 CloudWatch Logs에 로그를 전송합니다. 이 단계가 완료되고 로그 그룹이 아직 존재하지 않으면 FluentD가 다음과 같이 로그 그룹을 생성합니다.

로그 그룹 이름	로그 소스
`/aws/containerinsights/Cluster_Name/application`	`/var/log/containers`의 모든 로그 파일
`/aws/containerinsights/Cluster_Name/host`	`/var/log/dmesg`, `/var/log/secure` 및 `/var/log/messages`에서의 로그
`/aws/containerinsights/Cluster_Name/dataplane`	`kubelet.service`, `kubeproxy.service` 및 `docker.service`에 대한 `/var/log/journal`에서의 로그.

1단계: CloudWatch에 대한 네임스페이스 생성

다음 단계를 통해 CloudWatch에 대해 `amazon-cloudwatch`라는 Kubernetes 네임스페이스를 생성합니다. 이 네임스페이스를 이미 생성했다면 이 단계를 건너뛸 수 있습니다.

CloudWatch에 대한 네임스페이스를 생성하려면

- 다음 명령을 입력합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cloudwatch-namespace.yaml
```

2단계: FluentD 설치

FluentD를 다운로드하여 이 프로세스를 시작합니다. 이러한 단계를 완료하면 배포된 에이전트에서 클러스터에 대해 다음과 같은 리소스가 생성됩니다.

- `amazon-cloudwatch` 네임스페이스의 `fluentd`이라는 서비스 계정. 이 서비스 계정은 FluentD DaemonSet을 실행하는 데 사용됩니다. 자세한 내용은 Kubernetes 참조 문서의 서비스 계정 관리를 참조하십시오.
- `amazon-cloudwatch` 네임스페이스의 `fluentd`이라는 클러스터 역할. 이 클러스터 역할은 `fluentd` 서비스 계정에 대해 Pod 로그에서 `get`, `list` 및 `watch` 권한을 부여합니다. 자세한 내용은 Kubernetes 참조 문서의 API 개요를 참조하십시오.
- `amazon-cloudwatch` 네임스페이스의 `fluentd-config`이라는 ConfigMap. 이 ConfigMap에는 FluentD에서 사용되는 구성이 포함되어 있습니다. 자세한 내용은 Kubernetes 작업 설명서의 ConfigMap을 사용하도록 Pod 구성을 참조하십시오.

FluentD를 설치하려면

1. 클러스터 이름과 로그가 전송될 AWS 리전을 포함하여 `cluster-info`이라는 ConfigMap을 생성합니다. 다음 명령을 실행하여 클러스터와 리전 이름으로 자리표시자를 업데이트합니다.

```
kubectl create configmap cluster-info \
--from-literal=cluster.name=cluster_name \
--from-literal=logs.region=region_name -n amazon-cloudwatch
```

2. 다음 명령을 실행하여 FluentD DaemonSet을 다운로드한 후 클러스터에 배포합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/fluentd/fluentd.yaml
```

3. 다음 명령을 실행하여 배포를 확인합니다. 각 노드에는 `fluentd-cloudwatch-*`라는 이름의 Pod가 하나 포함되어야 합니다.

```
kubectl get pods -n amazon-cloudwatch
```

3단계: FluentD 설정 확인

다음 단계를 이용하여 FluentD 설정을 확인합니다.

Container Insights를 위한 FluentD 설정을 확인하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 로그를 선택합니다. 컨테이너에 FluentD를 배포한 리전에 있는지 확인합니다.

   해당 리전의 로그 그룹 목록에서 다음을 확인해야 합니다.

- /aws/containerinsights/*Cluster_Name*/application
- /aws/containerinsights/*Cluster_Name*/host
- /aws/containerinsights/*Cluster_Name*/dataplane

이러한 로그 그룹을 보면 FluentD 설정이 확인됩니다.

여러 줄 로그 지원

2019년 8월 19일 FluentD에서 수집한 로그에 대한 여러 줄 로그 지원을 추가했습니다.

기본적으로 여러 줄 로그 항목 시작자는 공백이 포함되지 않은 임의의 문자입니다. 이는 공백이 포함되지 않은 문자로 시작하는 모든 로그 행이 새로운 여러 줄 로그 항목으로 간주됨을 뜻합니다.

자체 애플리케이션 로그에서 다른 여러 줄 시작자를 사용하는 경우 `fluentd.yaml` 파일에서 두 가지 사항을 변경하여 이를 지원할 수 있습니다.

먼저 `fluentd.yaml`의 `containers` 섹션에 있는 `exclude_path` 필드에 로그 파일의 경로 이름을 추가하여 기본 여러 줄 지원에서 해당 로그를 제외합니다. 다음은 예제입니다.

```
<source>
  @type tail
  @id in_tail_container_logs
  @label @containers
  path /var/log/containers/*.log
  exclude_path ["full_pathname_of_log_file*", "full_pathname_of_log_file2*"]
```

그 다음 로그 파일에 대한 블록을 `fluentd.yaml` 파일에 추가합니다. 아래 예제는 타임스탬프 정규식을 여러 줄 시작자로 사용하는 CloudWatch 에이전트의 로그 파일에 사용됩니다. 이 블록을 복사하여 `fluentd.yaml`에 추가할 수 있습니다. 표시된 줄을 변경하여 사용하려는 애플리케이션 로그 파일 이름과 여러 줄 시작자를 반영하게 합니다.

```
<source>
  @type tail
  @id in_tail_cwagent_logs
  @label @cwagentlogs
  path /var/log/containers/cloudwatch-agent*
  pos_file /var/log/cloudwatch-agent.log.pos
  tag *
  read_from_head true
 <parse>
    @type json
    time_format %Y-%m-%dT%H:%M:%S.%NZ
 </parse>
</source>
```

```
<label @cwagentlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_cwagent
  </filter>

  <filter **>
    @type record_transformer
```

```
    @id filter_cwagent_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
 </filter>

  <filter **>
    @type concat
    key log
    multiline_start_regexp /^\d{4}[-/]\d{1,2}[-/]\d{1,2}/
    separator ""
    flush_interval 5
    timeout_label @NORMAL
 </filter>

 <match **>
    @type relabel
    @label @NORMAL
 </match>
</label>
```

FluentD의 로그 볼륨 축소(선택 사항)

기본적으로 AWS는 FluentD 애플리케이션 로그 및 Kubernetes 메타데이터를 CloudWatch로 전송합니다. CloudWatch로 전송되는 데이터의 볼륨을 줄이려면 이 데이터 원본 전체 또는 그 중 하나가 CloudWatch로 전송되지 못하게 할 수 있습니다.

FluentD 애플리케이션 로그를 중단하려면 `fluentd.yaml` 파일에서 다음 섹션을 제거하십시오.

```
<source>
  @type tail
  @id in_tail_fluentd_logs
  @label @fluentdlogs
  path /var/log/containers/fluentd*
  pos_file /var/log/fluentd.log.pos
  tag *
  read_from_head true
   <parse>
   @type json
   time_format %Y-%m-%dT%H:%M:%S.%NZ
  </parse>
</source>

<label @fluentdlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_fluentd
  </filter>

  <filter **>
    @type record_transformer
    @id filter_fluentd_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
   </filter>

   <match **>
     @type relabel
     @label @NORMAL
   </match>
</label>
```

Kubernetes 메타데이터를 제거하여 CloudWatch로 전송되는 로그 이벤트에 추가되지 못하게 하려면 `fluentd.yaml` 파일의 `record_transformer` 섹션에 한 줄을 추가하십시오. 이 메타데이터를 제거하려는 로그 소스에서 다음 줄을 추가합니다.

```
remove_keys $.kubernetes.pod_id, $.kubernetes.master_url, $.kubernetes.container_image_id,
 $.kubernetes.namespace_id
```

예:

```
<filter **>
  @type record_transformer
  @id filter_containers_stream_transformer
  <record>
    stream_name ${tag_parts[3]}
  </record>
  remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
</filter>
```

문제 해결

이러한 로그 그룹을 확인하지 않고 올바른 리전을 살펴보고 있는 경우, 오류를 찾으려면 FluentD DaemonSet Pod에 대한 로그를 확인합니다.

다음 명령을 실행하여 상태가 `Running`인지 확인합니다.

```
kubectl get pods -n amazon-cloudwatch
```

이전 명령의 결과에서 `fluentd-cloudwatch`로 시작하는 Pod 이름을 적어둡니다. 다음 명령에서 이 Pod 이름을 사용합니다.

```
kubectl logs pod_name -n amazon-cloudwatch
```

로그에 IAM 권한과 관련된 오류가 있으면 클러스터 노드에 연결된 IAM 역할을 확인합니다. Amazon EKS 클러스터를 실행하기 위해 필요한 권한에 대한 자세한 내용은 Amazon EKS 사용 설명서의 Amazon EKS IAM 정책, 역할 및 권한을 참조하십시오.

Pod 상태가 `CreateContainerConfigError`이면 다음 명령을 실행하여 정확한 오류를 가져옵니다.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

(선택 사항) Amazon EKS 제어 플레인 로깅 설정

Amazon EKS을 사용 중이라면 선택에 따라 Amazon EKS 제어 플레인 로깅을 활성화하여 Amazon EKS 제어 플레인에서 CloudWatch Logs로 직접 감사 및 진단 로그를 제공할 수 있습니다. 자세한 내용은 Amazon EKS 제어 플레인 로깅을 참조하십시오.

(선택 사항) CloudWatch에 StatsD 지표를 전송하기 위해 CloudWatch 에이전트를 StatsD 엔드포인트로 설정

이 섹션에서는 Amazon EKS 클러스터 또는 Kubernetes 클러스터 상에서 StatsD 엔드포인트로서 CloudWatch 에이전트를 설정 및 구성함으로써 컨테이너로부터 StatsD 지표를 수집하여 이를 CloudWatch에 게시하는 방법을 설명합니다. StatsD를 이미 사용 중이라면 이러한 방법을 알고 싶을 수 있습니다.

클러스터에서 StatsD 엔드포인트로서 CloudWatch 에이전트를 배포하는 방법은 다음과 같이 세 가지가 있습니다.

클러스터 내부에 단일 CloudWatch 에이전트를 서비스로 배포

CloudWatch에 게시하기 전에 클러스터에서 모든 StatsD 지표를 집계하고 싶은 경우에 이 옵션을 추천합니다. 이 옵션 역시 확장이 잘 되며 에이전트를 사이드카로 배포하는 것보다 더 적은 리소스를 소비합니다.

CloudWatch 에이전트를 각 작업자 노드에 DaemonSet으로 배포

Kubernetes 작업자 노드에서 StatsD 지표를 집계하고 싶거나 전체 클러스터에서 모든 StatsD 지표를 지원하는 단일 CloudWatch 에이전트의 확장성이 우려되는 경우에는 이 옵션을 추천합니다. 이 옵션 역시 에이전트를 사이드카로 배포하는 것보다 더 적은 리소스를 소비합니다.

단점은 각 노드에 하나의 컨테이너만 실행되기 때문에 단일 노드에서는 확장이 불가능하다는 것입니다.

CloudWatch 에이전트를 사이드카로 배포

여러 Pod에 걸쳐 StatsD 지표를 집계하지 않으려면 이 옵션을 추천합니다. CPU, 메모리, 네트워크 등의 리소스는 고객이 에이전트를 Pod에 연결할 수 있도록 유연해야 하므로 포드 수준에서는 격리되어 있습니다.

이 방법의 단점은 리소스 소비가 많다는 것입니다. – 예를 들어 1000개의 Pod에 1000개의 CloudWatch 에이전트 컨테이너가 있습니다. 또한 다양한 작업에서 발생하는 지표를 집계할 수 없습니다.

1단계: CloudWatch에 대한 네임스페이스 생성

다음 단계를 통해 CloudWatch에 대해 `amazon-cloudwatch`이라는 Kubernetes 네임스페이스를 생성합니다. 이러한 네임스페이스가 이미 생성되어 있는 경우 이러한 단계들을 건너뜁니다.

CloudWatch에 대한 네임스페이스를 생성하려면

- 다음 명령을 실행하여 네임스페이스 YAML을 다운로드한 후 `kubectl` 클라이언트 호스트에 적용합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/master/k8s-yaml-templates/cloudwatch-namespace.yaml
```

2단계: StatsD에 대한 ConfigMap 생성

어떤 옵션을 선택하든 StatsD에 대한 ConfigMap을 생성해야 합니다.

클러스터에서 지표를 전송하기 위해 StatsD에 대한 ConfigMap을 생성하려면

1. 다음 명령을 실행하여 `kubectl` 클라이언트 호스트로 ConfigMap YAML을 다운로드합니다.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/master/k8s-yaml-templates/cwagent-statsd/cwagent-statsd-configmap.yaml
```

   이 YAML 파일에는 JSON 구성 BLOB가 포함되어 있습니다.
2. JSON 구성 BLOB를 편집합니다. 편집이 끝났을 때 JSON BLOB이 유효한 JSON 형식을 유지하고 있는지 확인합니다.

   a. `statsd` 섹션에서 StatsD 항목을 편집할 수 있습니다. 자세한 내용은 StatsD로 시작하는 사용자 지정 지표 검색 (p. 253) 단원을 참조하십시오.

   b. `agent` 섹션에서 일반 항목을 편집할 수 있습니다. 자세한 내용은 CloudWatch 에이전트 구성 파일: 에이전트 섹션 (p. 227) 단원을 참조하십시오.

`agent` 섹션에서 `omit_hostname`를 `true`으로 설정하는 것이 좋습니다. 그렇지 않으면 에이전트가 StatsD 지표의 차원으로서 에이전트 Pod가 실행 중인 작업자 노드의 호스트 이름을 추가합니다. 이 호스트 이름을 차원으로 사용하면 에이전트 Pod가 다른 작업자 노드로 다시 예약된 경우에 지표를 확인하는 것이 더 까다로울 수 있습니다.

기본적으로 CloudWatch은 작업자 노드가 위치한 AWS 리전에 지표를 게시합니다. 이를 재정의하기 위해 `agent` 섹션에 `region` 필드를 추가할 수 있습니다. 예를 들어 `"region":"us-west-2"`와 같은 경로를 사용할 수 있습니다.

3. 다음 명령을 실행하여 클러스터에서 ConfigMap을 생성합니다.

```
kubectl apply -f cwagent-statsd-configmap.yaml
```

3단계: CloudWatch 에이전트 배포

CloudWatch 에이전트를 배포하여 StatsD 지표를 전송하려면 다음 섹션 중 하나에서 이러한 단계를 이용합니다.

옵션 1: 클러스터에 단일 CloudWatch 에이전트를 배포

이 섹션에서는 클러스터에 단일 CloudWatch 에이전트를 배포하여 CloudWatch에 StatsD 지표를 전송하는 방법을 설명합니다. 배포가 완료되고 에이전트가 실행 중이면 URL `cloudwatch-agent-statsd.amazon-cloudwatch.svc:8125`에서 수신이 시작됩니다. 클러스터 내의 애플리케이션은 이 URL로 StatsD 지표를 방출할 수 있습니다.

클러스터에서 단일 CloudWatch 에이전트를 배포하려면

1. 다음 명령을 실행하여 배포 YAML을 다운로드 및 배포합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/master/k8s-yaml-templates/cwagent-statsd/cwagent-statsd-deployment.yaml
```

2. 다음 명령을 입력하여 배포를 검증합니다.

```
kubectl get pods -n amazon-cloudwatch
```

옵션 2: 각 작업자 노드에 DaemonSet으로서 CloudWatch 에이전트를 배포

이 옵션은 DaemonSet로서 CloudWatch 에이전트를 각 작업자 노드에 배포하는 방법입니다. 배포가 완료되고 에이전트가 실행 중이면 `NodeIP:8125`에서 수신이 시작됩니다. 클러스터 내의 애플리케이션은 애플리케이션 Pod가 예약된 노드의 IP를 사용하여 StatsD 지표를 포트 8125로 방출할 수 있습니다.

각 작업자 노드에서 CloudWatch 에이전트를 배포하려면

1. 다음 명령을 실행하여 DaemonSet YAML을 다운로드 및 배포합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/master/k8s-yaml-templates/cwagent-statsd/cwagent-statsd-daemonset.yaml
```

2. 다음 명령을 실행하여 배포를 확인합니다.

```
kubectl get pods -n amazon-cloudwatch
```

옵션 3: 사이드카로서 CloudWatch 에이전트를 배포

이 섹션에서는 사이드카로서 CloudWatch 에이전트를 배포하는 방법을 설명합니다. 이 에이전트는 같은 Pod에서 실행 중인 애플리케이션으로부터 StatsD 지표를 수집합니다. 다음은 Amazon Linux 컨테이너에 대한 사이드카로서 CloudWatch 에이전트를 시작하는 YAML 파일의 예제입니다.

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: amazonlinux
spec:
  containers:
    - name: amazonlinux
      image: amazonlinux
      command: ["/bin/sh"]
      args: ["-c", "sleep 300"]
    - name: cloudwatch-agent
      image: amazon/cloudwatch-agent
      imagePullPolicy: Always
      resources:
        limits:
          cpu:  200m
          memory: 100Mi
        requests:
          cpu: 200m
          memory: 100Mi
      volumeMounts:
        - name: cwagentconfig
          mountPath: /etc/cwagentconfig
  volumes:
    - name: cwagentconfig
      configMap:
        name: cwagentstatsdconfig
  terminationGracePeriodSeconds: 60
```

Amazon EKS 및 Kubernetes에서 Container Insights 업데이트 또는 삭제

이 단원의 절차를 통해 CloudWatch 에이전트 컨테이너 이미지를 업데이트하거나 Amazon EKS 또는 Kubernetes 클러스터에서 Container Insights를 제거합니다.

주제

- CloudWatch 에이전트 컨테이너 이미지 업데이트 (p. 157)
- Container Insights용 CloudWatch 에이전트 및 FluentD 삭제 (p. 158)

CloudWatch 에이전트 컨테이너 이미지 업데이트

컨테이너 이미지를 최신 버전으로 업데이트해야 하는 경우 이 단원의 절차를 따르십시오.

컨테이너 이미지를 업데이트하려면

1. 다음 명령을 입력하여 최신 `cwagent-serviceaccount.yaml` 파일을 적용합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-serviceaccount.yaml
```

2. 이 단계는 컨테이너식 CloudWatch 에이전트를 2019년 8월 20일에 릴리스된 1.226589.0 버전 이전 버전에서 업그레이드한 고객에게만 필수입니다.

   Configmap 파일인 `cwagentconfig`에서 `structuredlogs`라는 키워드를 `logs`로 변경합니다.

   a. 먼저 다음 명령을 입력하여 편집 모드에서 기존 파일인 `cwagentconfig`를 엽니다.

   ```
   kubectl edit cm cwagentconfig -n amazon-cloudwatch
   ```

   이 파일에 `structuredlogs`라는 키워드가 있으면 `logs`로 변경합니다.

   b. **`wq`**를 입력하여 파일을 저장하고 편집 모드를 종료합니다.

3. 다음 명령을 입력하여 최신 `cwagent-daemonset.yaml` 파일을 적용합니다.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

`cwagent-configmap.yaml`에서 구성을 변경하는 CloudWatch 에이전트 DaemonSet의 단계적 업데이트는 언제든지 완료할 수 있습니다. 이를 위해서는 `cwagent-daemonset.yaml` 파일의 `.spec.template` 섹션에 변경 사항이 있는지 확인해야 합니다. 그렇지 않으면 Kubernetes는 DaemonSet을 변경되지 않은 상태로 처리합니다. 일반적인 방법은 다음 예제에서와 같이 `.spec.template.metadata.annotations.configHash`에 ConfigMap의 해시 값을 추가하는 것입니다.

```
yq w -i cwagent-daemonset.yaml spec.template.metadata.annotations.configHash $(kubectl get
 cm/cwagentconfig -n amazon-cloudwatch -o yaml | sha256sum)
```

이렇게 하면 다음 예제에서와 같이 `cwagent-daemonset.yaml` 파일에 해시 값이 추가됩니다.

```
spec:
  selector:
    matchLabels:
      name: cloudwatch-agent
  template:
    metadata:
      labels:
        name: cloudwatch-agent
      annotations:
        configHash: 88915de4cf9c3551a8dc74c0137a3e83569d28c71044b0359c2578d2e0461825
```

그리고 다음 명령을 실행하면 새 구성이 선택됩니다.

```
kubectl apply -f cwagent-daemonset.yaml
```

yq에 대한 자세한 내용은 yq를 참조하십시오.

Container Insights용 CloudWatch 에이전트 및 FluentD 삭제

CloudWatch 에이전트 및 FluentD와 관련된 모든 리소스를 삭제하려면 다음 명령을 입력하십시오. 이 명령에서 *`Cluster`*는 Amazon EKS 또는 Kubernetes 클러스터의 이름이며, *`Region`*은 로그가 게시되는 리전의 이름입니다.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
```

```
container-insights-monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/
{{cluster_name}}/Cluster_Name/;s/{{region_name}}/Region/" | kubectl delete -f -
```

Container Insights 지표 보기

Container Insights를 설정하여 지표를 수집하고 난 후에는 CloudWatch 자동 대시보드에서 이러한 지표를 볼 수 있습니다.

대시보드에 Container Insights 지표를 표시하려면 Container Insights 설정을 완료해야 합니다. 자세한 내용은 Container Insights 설정 (p. 137) 단원을 참조하십시오.

Container Insights 지표를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 화면 왼쪽 상단에서 개요 옆의 아래쪽 화살표를 선택하고 Container Insights를 선택합니다.

   클러스터에 대한 지표를 보여주는 그래프가 몇 개 나타납니다. 그래프 아래의 클러스터 목록에는 각 클러스터에 대해 상태 및 기본 지표가 표시됩니다.

   특정 클러스터에 대해 보기를 필터링하려면 가운데 상단의 상자를 사용합니다.
3. Clusters(클러스터)가 표시된 왼쪽 상단에서 모드를 바꿔가며 노드, Pod, 서비스, 작업 및 네임스페이스 수준의 지표를 확인합니다. 이때 개별 Pod 및 노드만 보도록 이러한 보기를 필터링할 수도 있습니다.

   또한 Amazon EKS 지표와 Amazon ECS 지표 사이를 전환할 수 있습니다.
4. 어떤 그래프에서든 범례 줄에서 멈춰 있으면 해당 리소스에 대한 자세한 정보를 확인할 수 있습니다.
5. 대시보드에서 클러스터 중 하나에 관한 클러스터 수준의 로그를 보려면 화면 하단의 목록에서 해당 이름 옆에 있는 버튼을 선택합니다. 그런 다음 작업을 선택하고 목록에서 옵션 중 하나를 선택합니다.

   새 브라우저가 나타나고 해당 리소스에 대한 CloudWatch Logs Insights 쿼리가 표시됩니다.

   쿼리 실행을 선택합니다. 쿼리 결과가 표시됩니다.

   CloudWatch Logs Insights에 대한 자세한 내용은 CloudWatch Logs Insights로 로그 데이터 분석을 참조하십시오.
6. 노드, Pod 또는 작업 수준에서 페이지 하단의 목록에 현재 표시된 한 개 이상의 리소스에 관한 로그를 보려면 해당 리소스의 이름 옆에 있는 확인란을 선택합니다. 그런 다음 작업을 선택하고 목록에서 옵션 중 하나를 선택합니다. 그러면 해당 리소스의 로그 또는 AWS X-Ray 추적을 볼 수 있습니다.

   새 브라우저가 나타나고 요청한 정보가 표시됩니다.

Container Insights에서 수집되는 모든 지표에 대해 CloudWatch 경보를 설정할 수 있습니다. 자세한 내용은 Amazon CloudWatch 경보 사용 (p. 67) 단원을 참조하십시오.

CloudWatch Logs Insights을 사용하여 Container Insights 데이터 보기

Container Insights는 CloudWatch Logs에 저장된 성능 로그 이벤트를 사용하여 지표를 수집합니다. 컨테이너 데이터에 대한 추가 보기에서 CloudWatch Logs Insights 쿼리를 사용할 수 있습니다.

CloudWatch Logs Insights에 대한 자세한 내용은 CloudWatch Logs Insights로 로그 데이터 분석을 참조하십시오. 쿼리에서 사용할 수 있는 로그 필드에 대한 자세한 내용은 Amazon EKS 및 Kubernetes에 대한 Container Insights 성능 로그 이벤트 (p. 170) 단원을 참조하십시오.

CloudWatch Logs Insights을 사용하여 컨테이너 지표 데이터를 쿼리하려면

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

   화면 상단 근처에 쿼리 편집기가 있습니다. CloudWatch Logs Insights를 처음으로 열면 이 상자에는 최신 로그 이벤트 20개를 반환하는 기본 쿼리가 포함되어 있습니다.
3. 쿼리 편집기 위의 상자에서 쿼리할 Container Insights 로그 그룹을 선택합니다. 작업할 다음 예제 쿼리에서는 로그 그룹 이름이 performance로 끝나야 합니다.

   로그 그룹을 선택하면 CloudWatch Logs Insights가 로그 그룹 내 데이터에서 필드를 자동으로 검색하고 오른쪽 창에 있는 검색된 필드에 표시합니다. 또한 이 로그 그룹의 로그 이벤트를 시간의 흐름에 따라 보여주는 막대 그래프도 표시합니다. 이 막대 그래프에서는 테이블에 표시된 이벤트뿐만 아니라 쿼리 및 시간 범위와 일치하는 로그 그룹 내 이벤트의 분포를 보여줍니다.
4. 쿼리 편집기에서 기본 쿼리를 다음 쿼리로 바꾸고 쿼리 실행을 선택합니다.

```
STATS avg(node_cpu_utilization) as avg_node_cpu_utilization by NodeName
| SORT avg_node_cpu_utilization DESC
```

   이 쿼리는 노드 목록을 평균적인 노드 CPU 이용률에 따라 정렬하여 보여줍니다.
5. 또 다른 예를 시도하려면 쿼리를 다음 쿼리로 바꾸고 쿼리 실행을 선택합니다. 추가 샘플 쿼리는 이 페이지 후반부에 나열되어 있습니다.

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by PodName
| SORT avg_number_of_container_restarts DESC
```

   이 쿼리는 Pod 목록을 평균적인 컨테이너 재시작 횟수에 따라 정렬하여 보여줍니다.
6. 또 다른 쿼리를 시도하고 싶은 경우에는 화면 오른쪽의 목록에 필드를 포함시킬 수 있습니다. 쿼리 구문에 대한 자세한 내용은 CloudWatch Logs Insights 쿼리 구문을 참조하십시오.

Container Insights에 대한 기타 샘플 쿼리

평균적인 컨테이너 재시작 횟수에 따라 정렬된 Pod 목록

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by PodName
| SORT avg_number_of_container_restarts DESC
```

요청된 Pod와 실행 중인 Pod 간 비교

```
fields @timestamp, @message
| sort @timestamp desc
| filter Type="Pod"
| stats min(pod_number_of_containers) as requested, min(pod_number_of_running_containers)
 as running, ceil(avg(pod_number_of_containers-pod_number_of_running_containers)) as
 pods_missing by kubernetes.pod_name
| sort pods_missing desc
```

클러스터 노드 실패 횟수

```
stats avg(cluster_failed_node_count) as CountOfNodeFailures
| filter Type="Cluster"
| sort @timestamp desc
```

애플리케이션 로그 오류

```
Count by container name: stats count() as countoferrors by kubernetes.container_name
| filter stream="stderr"
| sort countoferrors desc
```

컨테이너 이름의 디스크 사용량

```
stats floor(avg(container_filesystem_usage/1024)) as container_filesystem_usage_avg_kb by
 InstanceId, kubernetes.container_name, device
| filter Type="ContainerFS"
| sort container_filesystem_usage_avg_kb desc
```

컨테이너 이름의 CPU 사용량

```
stats pct(container_cpu_usage_total, 50) as CPUPercMedian by kubernetes.container_name
| filter Type="Container"
```

Container Insights에서 수집된 지표

Container Insights에서는 Amazon ECS 및 AWS Fargate에 대한 지표 한 세트와 Amazon EKS 및 Kubernetes에 대한 다른 세트를 수집합니다.

주제

- Amazon ECS Container Insights 지표 (p. 161)
- Amazon EKS 및 and Kubernetes Container Insights 지표 (p. 165)

Amazon ECS Container Insights 지표

아래 표에는 Amazon ECS에 대해 Container Insights에서 수집한 지표 및 차원의 목록이 나와 있습니다. 이러한 지표는 `ECS/ContainerInsights` 네임스페이스에 있습니다. 자세한 내용은 지표 (p. 3) 단원을 참조하십시오.

콘솔에 Container Insights 지표가 보이지 않는 경우, Container Insights 설정을 완료했는지 확인합니다. Container Insights 설정이 완료되기 전에는 지표가 나타나지 않습니다. 자세한 내용은 Container Insights 설정 (p. 137) 단원을 참조하십시오.

> **Note**
>
> 수집된 네트워크 및 디스크 메트릭은 Amazon ECS 작업당 누적됩니다. 이러한 누적 지표가 CloudWatch에서 집계된 다음 `RATE` 함수가 적용되기 때문에 이러한 지표에서 스파이크가 발생할 것으로 예상됩니다. 예를 들어 클러스터에 있는 두 Amazon ECS 작업 중 하나가 실행이 중지되면 누적 네트워크 메트릭의 `RATE`는 방금 중지된 Amazon ECS 작업에서 수집된 네트워크 메트릭 데이터와 동일한 데이터 포인트에 대한 음의 스파이크를 표시합니다. CloudWatch 팀은 이러한 동작을 인지하고 있으며 누적 지표를 수집하고 모니터링하고 경보를 생성할 수 있는 최상의 방법을 연구하고 있습니다.
> `RATE` 함수에 대한 자세한 내용은 지표 수학 구문 및 함수 (p. 50) 단원을 참조하십시오.

다음 지표는 클러스터 및 서비스 레벨 지표를 위해 Amazon ECS에서 Container Insights 설정 (p. 138)의 단계를 완료하면 사용할 수 있습니다.

지표 이름	차원	설명
`ContainerInstanceCount`	ClusterName	클러스터에 등록된 Amazon Amazon ECS

지표 이름	차원	설명
		에이전트를 실행하는 EC2 인스턴스의 숫자입니다.
`CpuUtilized`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용 중인 차원 세트로 지정된 리소스의 작업에서 사용하는 CPU 유닛입니다. 이 지표는 작업 정의에 정의된 CPU 예약이 있는 작업에 대해서만 수집됩니다.
`CpuReserved`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용 중인 차원 세트에서 지정한 리소스의 작업에서 예약된 CPU 유닛입니다. 이 지표는 작업 정의에 정의된 CPU 예약이 있는 작업에 대해서만 수집됩니다.
`DeploymentCount`	ServiceName, ClusterName	Amazon ECS 서비스의 배포 숫자입니다.
`DesiredTaskCount`	ServiceName, ClusterName	Amazon ECS에서 원하는 작업 숫자입니다.
`MemoryUtilized`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용 중인 차원 세트로 지정된 리소스의 작업에서 사용 중인 메모리입니다. 이 지표는 작업 정의에 정의된 메모리 예약이 있는 작업에 대해서만 수집됩니다.
`MemoryReserved`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용 중인 차원 세트에서 지정한 리소스의 작업에서 예약된 메모리입니다. 이 지표는 작업 정의에 정의된 메모리 예약이 있는 작업에 대해서만 수집됩니다.

지표 이름	차원	설명
`NetworkRxBytes`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용중인 차원에서 지정한 리소스에서 수신된 바이트의 숫자입니다. 이 지표는 `bridge` 네트워크 모드에서 컨테이너에 대해서만 제공됩니다. `awsvpc` 네트워크 모드 또는 `host` 네트워크 모드에서는 컨테이너에 대해 제공되지 않습니다.
`NetworkTxBytes`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용중인 차원에서 지정한 리소스에서 전송된 바이트의 숫자입니다. 이 지표는 `bridge` 네트워크 모드에서 컨테이너에 대해서만 제공됩니다. `awsvpc` 네트워크 모드 또는 `host` 네트워크 모드에서는 컨테이너에 대해 제공되지 않습니다.
`PendingTaskCount`	ServiceName, ClusterName	현재 `PENDING` 상태인 작업의 숫자입니다.
`RunningTaskCount`	ServiceName, ClusterName	현재 `RUNNING` 상태인 작업의 숫자입니다.
`ServiceCount`	ServiceName	클러스터의 서비스 숫자입니다.
`StorageReadBytes`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용 중인 차원에서 지정한 리소스의 스토리지에서 읽혀진 바이트의 숫자입니다.
`StorageWriteBytes`	TaskDefinitionFamily, ClusterName ServiceName, ClusterName ClusterName	사용 중인 차원에서 지정한 리소스의 스토리지에서 쓰여진 바이트의 숫자입니다.
`TaskCount`	ServiceName	서비스에서 실행 중인 작업의 숫자입니다.
`TaskSetCount`	ServiceName, ClusterName	서비스의 작업 세트 숫자입니다.

다음 지표는 Amazon ECS에서 EC2 인스턴스 레벨 지표를 수집하기 위해 CloudWatch 에이전트 배포 (p. 139)의 단계를 완료하면 사용할 수 있습니다.

지표 이름	차원	설명
instance_cpu_limit	ClusterName	클러스터의 단일 EC2 인스턴스에 할당할 수 있는 최대 CPU 유닛 수입니다.
instance_cpu_reserved_capacity	ClusterName EC2InstanceId, ContainerInstanceId, ClusterName	클러스터의 단일 EC2 인스턴스에서 현재 예약 중인 CPU의 비율입니다.
instance_cpu_usage_total	ClusterName	클러스터의 단일 EC2 인스턴스에서 사용 중인 CPU 유닛 수입니다.
instance_cpu_utilization	ClusterName EC2InstanceId, ContainerInstanceId, ClusterName	클러스터의 단일 EC2 인스턴스에서 사용 중인 CPU 유닛의 총 비율입니다.
instance_filesystem_utilization	ClusterName EC2InstanceId, ContainerInstanceId, ClusterName	클러스터의 단일 EC2 인스턴스에서 사용 중인 파일 시스템 용량의 총 비율입니다.
instance_memory_limit	ClusterName	이 클러스터의 단일 EC2 인스턴스에 할당할 수 있는 최대 메모리 양(바이트)입니다.
instance_memory_reserved_capacity	ClusterName EC2InstanceId, ContainerInstanceId, ClusterName	클러스터의 단일 EC2 인스턴스에서 현재 예약 중인 메모리의 비율입니다.
instance_memory_utilization	ClusterName EC2InstanceId, ContainerInstanceId, ClusterName	클러스터의 단일 EC2 인스턴스에서 사용 중인 메모리의 총 비율입니다.
instance_memory_working_set	ClusterName	클러스터의 단일 EC2 인스턴스에서 사용 중인 메모리의 양(바이트)입니다.
instance_network_total_bytes	ClusterName	클러스터의 단일 EC2 인스턴스에서 네트워크를 통해 전송 및 수신된 초당 총 바이트 수입니다.
instance_number_of_running_tasks	ClusterName	클러스터의 단일 EC2 인스턴스에서 실행 중인 작업 수입니다.

Amazon EKS 및 and Kubernetes Container Insights 지표

아래 표에는 Container Insights에서 Amazon EKS 및 Kubernetes용으로 수집한 지표 및 차원의 목록이 나와 있습니다. 이러한 지표는 `ContainerInsights` 네임스페이스에 있습니다. 자세한 내용은 지표 (p. 3) 단원을 참조하십시오.

콘솔에 Container Insights 지표가 보이지 않는 경우, Container Insights 설정을 완료했는지 확인합니다. Container Insights 설정이 완료되기 전에는 지표가 나타나지 않습니다. 자세한 내용은 Container Insights 설정 (p. 137) 단원을 참조하십시오.

지표 이름	차원	설명
`cluster_failed_node_count`	ClusterName	클러스터의 실패한 작업자 노드의 숫자입니다.
`cluster_node_count`	ClusterName	클러스터의 작업자 노드의 총 숫자입니다.
`namespace_number_of_running_pods`	Namespace, ClusterName ClusterName	사용 중인 차원에서 지정한 리소스의 네임스페이스당 실행 중인 Pod 숫자입니다.
`node_cpu_limit`	ClusterName	클러스터에서 단일 노드에 할당할 수 있는 최대 CPU 유닛 숫자입니다.
`node_cpu_reserved_capacity`	NodeName, ClusterName, InstanceId ClusterName	kubelet, kube-proxy, Docker 등 노드 구성 요소에 예약된 CPU 유닛의 비율입니다.
`node_cpu_usage_total`	ClusterName	클러스터의 노드에서 사용 중인 CPU 유닛의 숫자입니다.
`node_cpu_utilization`	NodeName, ClusterName, InstanceId ClusterName	클러스터의 노드에서 사용 중인 CPU 유닛의 총 백분율입니다.
`node_filesystem_utilization`	NodeName, ClusterName, InstanceId ClusterName	클러스터에서 노드에 사용하는 파일 시스템 용량의 총 백분율입니다.
`node_memory_limit`	ClusterName	클러스터에서 단일 노드로 할당될 수 있는 최대 메모리의 양(바이트)입니다.
`node_memory_reserved_capacity`	NodeName, ClusterName, InstanceId ClusterName	클러스터의 노드에서 현재 사용 중인 메모리의 비율입니다.
`node_memory_utilization`	NodeName, ClusterName, InstanceId ClusterName	한 개 또는 여러 개의 노드에서 현재 사용 중인 메모리의 비율입니다.

지표 이름	차원	설명
node_memory_working_set	ClusterName	클러스터의 노드 작업 세트에서 사용하는 메모리의 양(바이트)입니다.
node_network_total_bytes	NodeName, ClusterName, InstanceId ClusterName	클러스터에서 노드당 네트워크를 통해 전송 및 수신된 초당 바이트의 합계 수치입니다.
node_number_of_running_containers	NodeName, ClusterName, InstanceId ClusterName	클러스터에서 노드당 실행 중인 컨테이너의 숫자입니다.
node_number_of_running_pods	NodeName, ClusterName, InstanceId ClusterName	클러스터에서 노드당 실행 중인 Pod 숫자입니다.
pod_cpu_reserved_capacity	PodName, Namespace, ClusterName ClusterName	클러스터에서 Pod별로 예약된 CPU 용량입니다.
pod_cpu_utilization	PodName, Namespace, ClusterName Namespace, ClusterName Service, Namespace, ClusterName ClusterName	Pod에서 사용 중인 CPU 유닛의 비율입니다.
pod_cpu_utilization_over_pod_limit	PodName, Namespace, ClusterName Namespace, ClusterName Service, Namespace, ClusterName ClusterName	Pod 한도를 초과하여 Pod에서 사용 중인 CPU 유닛의 비율입니다.
pod_memory_reserved_capacity	PodName, Namespace, ClusterName ClusterName	포드에 예약된 메모리의 비율입니다.
pod_memory_utilization	PodName, Namespace, ClusterName Namespace, ClusterName Service, Namespace, ClusterName ClusterName	한 개 또는 여러 개의 Pod에서 현재 사용 중인 메모리의 비율입니다.
pod_memory_utilization_over_pod_limit	PodName, Namespace, ClusterName Namespace, ClusterName Service, Namespace, ClusterName ClusterName	Pod 한도를 초과하여 Pod에서 사용 중인 메모리의 비율입니다.
pod_number_of_container_restarts	PodName, Namespace, ClusterName	Pod의 컨테이너 재시작 총 횟수입니다.

지표 이름	차원	설명
pod_network_rx_bytes	PodName, Namespace, ClusterName Namespace, ClusterName Service, Namespace, ClusterName ClusterName	Pod에서 네트워크를 통해 수신 중인 초당 바이트 수입니다.
pod_network_tx_bytes	PodName, Namespace, ClusterName Namespace, ClusterName Service, Namespace, ClusterName ClusterName	Pod에서 네트워크를 통해 전송 중인 초당 바이트 수입니다.
service_number_of_running_pods	Service, Namespace, ClusterName ClusterName	클러스터에서 단일 또는 복수의 서비스를 실행하는 Pod의 숫자입니다.

Container Insights 성능 로그 참조

이 단원에는 Container Insights가 성능 로그 이벤트를 사용하여 지표를 수집하는 방법에 관한 참조 정보가 포함되어 있습니다.

주제

- Amazon ECS에 대한 Container Insights 성능 로그 이벤트 (p. 167)
- Amazon EKS 및 Kubernetes에 대한 Container Insights 성능 로그 이벤트 (p. 170)
- Amazon EKS 및 Kubernetes에 대한 성능 로그 이벤트의 관련 필드 (p. 181)

Amazon ECS에 대한 Container Insights 성능 로그 이벤트

다음은 Container Insights가 Amazon ECS에서 수집하는 성능 로그 이벤트의 예입니다.

유형: 컨테이너

```
{
    "Version": "0",
    "Type": "Container",
    "ContainerName": "mysql",
    "TaskId": "7c7bce41-8f2f-4673-b1a3-0e57ef51168f",
    "TaskDefinitionFamily": "hello_world",
    "TaskDefinitionRevision": "2",
    "ContainerInstanceId": "12345678-db61-4b1b-b0ec-93ad21467cc9",
    "EC2InstanceId": "i-1234567890123456",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586749104,
    "CpuUtilized": 1.8381139895790504,
    "CpuReserved": 30.0,
    "MemoryUtilized": 295,
    "MemoryReserved": 400,
    "StorageReadBytes": 10817536,
```

```
    "StorageWriteBytes": 2085376000,
    "NetworkRxBytes": 6522,
    "NetworkRxDropped": 0,
    "NetworkRxErrors": 0,
    "NetworkRxPackets": 83,
    "NetworkTxBytes": 3904,
    "NetworkTxDropped": 0,
    "NetworkTxErrors": 0,
    "NetworkTxPackets": 44
}
```

유형: 작업

```
{
    "Version": "0",
    "Type": "Task",
    "TaskId": "8b03d710-a0de-4458-b6ee-e9829e9d1440",
    "TaskDefinitionFamily": "hello_world",
    "TaskDefinitionRevision": "2",
    "ContainerInstanceId": "12345678-e797-496b-9d70-4edfda2700a4",
    "EC2InstanceId": "i-1234567890123456",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586269429,
    "CpuUtilized": 2.4353688568190526,
    "CpuReserved": 60.0,
    "MemoryUtilized": 314,
    "MemoryReserved": 800,
    "StorageReadBytes": 11124736,
    "StorageWriteBytes": 2100379648,
    "NetworkRxBytes": 12062,
    "NetworkRxDropped": 0,
    "NetworkRxErrors": 0,
    "NetworkRxPackets": 149,
    "NetworkTxBytes": 8064,
    "NetworkTxDropped": 0,
    "NetworkTxErrors": 0,
    "NetworkTxPackets": 96,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "CpuUtilized",
                    "Unit": "None"
                },
                {
                    "Name": "CpuReserved",
                    "Unit": "None"
                },
                {
                    "Name": "MemoryUtilized",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "MemoryReserved",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "StorageReadBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "StorageWriteBytes",
                    "Unit": "Bytes/Second"
```

```
                },
                {
                    "Name": "NetworkRxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkTxBytes",
                    "Unit": "Bytes/Second"
                }
            ],
            "Dimensions": [
                [
                    "ClusterName"
                ],
                [
                    "ServiceName",
                    "ClusterName"
                ],
                [
                    "ClusterName",
                    "TaskDefinitionFamily"
                ]
            ]
        }
    ]
}
```

유형: 서비스

```
{
    "Version": "0",
    "Type": "Service",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586460000,
    "DesiredTaskCount": 2,
    "RunningTaskCount": 2,
    "PendingTaskCount": 0,
    "DeploymentCount": 1,
    "TaskSetCount": 0,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "DesiredTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "RunningTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "PendingTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "DeploymentCount",
                    "Unit": "Count"
                },
                {
                    "Name": "TaskSetCount",
                    "Unit": "Count"
                }
            ],
```

```
            "Dimensions": [
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

유형: 클러스터

```
{
    "Version": "0",
    "Type": "Cluster",
    "ClusterName": "myCICluster",
    "Timestamp": 1561587300000,
    "TaskCount": 5,
    "ContainerInstanceCount": 5,
    "ServiceCount": 2,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "TaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ContainerInstanceCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ServiceCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Amazon EKS 및 Kubernetes에 대한 Container Insights 성능 로그 이벤트

다음은 Container Insights가 Amazon EKS 및 Kubernetes 클러스터에서 수집하는 성능 로그 이벤트의 예입니다.

유형: 노드

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
```

```
        "Unit": "Percent",
        "Name": "node_cpu_utilization"
      },
      {
        "Unit": "Percent",
        "Name": "node_memory_utilization"
      },
      {
        "Unit": "Bytes/Second",
        "Name": "node_network_total_bytes"
      },
      {
        "Unit": "Percent",
        "Name": "node_cpu_reserved_capacity"
      },
      {
        "Unit": "Percent",
        "Name": "node_memory_reserved_capacity"
      },
      {
        "Unit": "Count",
        "Name": "node_number_of_running_pods"
      },
      {
        "Unit": "Count",
        "Name": "node_number_of_running_containers"
      }
    ],
    "Dimensions": [
      [
        "NodeName",
        "InstanceId",
        "ClusterName"
      ]
    ],
    "Namespace": "ContainerInsights"
  },
  {
    "Metrics": [
      {
        "Unit": "Percent",
        "Name": "node_cpu_utilization"
      },
      {
        "Unit": "Percent",
        "Name": "node_memory_utilization"
      },
      {
        "Unit": "Bytes/Second",
        "Name": "node_network_total_bytes"
      },
      {
        "Unit": "Percent",
        "Name": "node_cpu_reserved_capacity"
      },
      {
        "Unit": "Percent",
        "Name": "node_memory_reserved_capacity"
      },
      {
        "Unit": "Count",
        "Name": "node_number_of_running_pods"
      },
      {
        "Unit": "Count",
        "Name": "node_number_of_running_containers"
```

```
        },
        {
          "Name": "node_cpu_usage_total"
        },
        {
          "Name": "node_cpu_limit"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_working_set"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_limit"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "/proc",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567096682364",
  "Type": "Node",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_cpu_limit": 4000,
  "node_cpu_request": 1130,
  "node_cpu_reserved_capacity": 28.249999999999996,
  "node_cpu_usage_system": 33.794636630852764,
  "node_cpu_usage_total": 136.47852169244098,
  "node_cpu_usage_user": 71.67075111567326,
  "node_cpu_utilization": 3.4119630423110245,
  "node_memory_cache": 3103297536,
  "node_memory_failcnt": 0,
  "node_memory_hierarchical_pgfault": 0,
  "node_memory_hierarchical_pgmajfault": 0,
  "node_memory_limit": 16624865280,
  "node_memory_mapped_file": 406646784,
  "node_memory_max_usage": 4230746112,
  "node_memory_pgfault": 0,
  "node_memory_pgmajfault": 0,
  "node_memory_request": 1115684864,
  "node_memory_reserved_capacity": 6.7109407818311055,
  "node_memory_rss": 798146560,
  "node_memory_swap": 0,
  "node_memory_usage": 3901444096,
  "node_memory_utilization": 6.601302600149552,
  "node_memory_working_set": 1097457664,
  "node_network_rx_bytes": 35918.392817386324,
  "node_network_rx_dropped": 0,
  "node_network_rx_errors": 0,
```

```
  "node_network_rx_packets": 157.67565245448117,
  "node_network_total_bytes": 68264.20276554905,
  "node_network_tx_bytes": 32345.80994816272,
  "node_network_tx_dropped": 0,
  "node_network_tx_errors": 0,
  "node_network_tx_packets": 154.21455923431654,
  "node_number_of_running_containers": 16,
  "node_number_of_running_pods": 13
}
```

유형: NodeFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_filesystem_utilization"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097939726",
  "Type": "NodeFS",
  "Version": "0",
  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_filesystem_available": 17298395136,
  "node_filesystem_capacity": 21462233088,
  "node_filesystem_inodes": 10484720,
  "node_filesystem_inodes_free": 10367158,
  "node_filesystem_usage": 4163837952,
  "node_filesystem_utilization": 19.400767547940255
}
```

유형: NodeDiskIO

```
{
```

```
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeDiskIO",
  "Version": "0",
  "device": "/dev/nvme0n1",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_diskio_io_service_bytes_async": 9750.505814277016,
  "node_diskio_io_service_bytes_read": 0,
  "node_diskio_io_service_bytes_sync": 230.6174506688036,
  "node_diskio_io_service_bytes_total": 9981.123264945818,
  "node_diskio_io_service_bytes_write": 9981.123264945818,
  "node_diskio_io_serviced_async": 1.153087253344018,
  "node_diskio_io_serviced_read": 0,
  "node_diskio_io_serviced_sync": 0.03603397666700056,
  "node_diskio_io_serviced_total": 1.1891212300110185,
  "node_diskio_io_serviced_write": 1.1891212300110185
}
```


유형: NodeNet


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeNet",
  "Version": "0",
  "interface": "eni972f6bfa9a0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_interface_network_rx_bytes": 3163.008420864309,
  "node_interface_network_rx_dropped": 0,
  "node_interface_network_rx_errors": 0,
  "node_interface_network_rx_packets": 16.575629266820258,
  "node_interface_network_total_bytes": 3518.3935157426017,
  "node_interface_network_tx_bytes": 355.385094878293,
  "node_interface_network_tx_dropped": 0,
  "node_interface_network_tx_errors": 0,
  "node_interface_network_tx_packets": 3.9997714100370625
}
```


유형: Pod


```
{
```

```
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_rx_bytes"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_tx_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization_over_pod_limit"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization_over_pod_limit"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_reserved_capacity"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
```

```
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "pod_number_of_container_restarts"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "Pod",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_cpu_limit": 200,
  "pod_cpu_request": 200,
  "pod_cpu_reserved_capacity": 5,
  "pod_cpu_usage_system": 1.4504841104992765,
  "pod_cpu_usage_total": 5.817016867430125,
  "pod_cpu_usage_user": 1.1281543081661038,
  "pod_cpu_utilization": 0.14542542168575312,
  "pod_cpu_utilization_over_pod_limit": 2.9085084337150624,
  "pod_memory_cache": 8192,
```

```
  "pod_memory_failcnt": 0,
  "pod_memory_hierarchical_pgfault": 0,
  "pod_memory_hierarchical_pgmajfault": 0,
  "pod_memory_limit": 104857600,
  "pod_memory_mapped_file": 0,
  "pod_memory_max_usage": 25268224,
  "pod_memory_pgfault": 0,
  "pod_memory_pgmajfault": 0,
  "pod_memory_request": 104857600,
  "pod_memory_reserved_capacity": 0.6307275170893897,
  "pod_memory_rss": 22777856,
  "pod_memory_swap": 0,
  "pod_memory_usage": 25141248,
  "pod_memory_utilization": 0.10988455961791709,
  "pod_memory_utilization_over_pod_limit": 17.421875,
  "pod_memory_working_set": 18268160,
  "pod_network_rx_bytes": 9880.697124714186,
  "pod_network_rx_dropped": 0,
  "pod_network_rx_errors": 0,
  "pod_network_rx_packets": 107.80005532263283,
  "pod_network_total_bytes": 10158.829201483635,
  "pod_network_tx_bytes": 278.13207676944796,
  "pod_network_tx_dropped": 0,
  "pod_network_tx_errors": 0,
  "pod_network_tx_packets": 1.146027574644318,
  "pod_number_of_container_restarts": 0,
  "pod_number_of_containers": 1,
  "pod_number_of_running_containers": 1,
  "pod_status": "Running"
}
```


유형: PodNet


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "PodNet",
  "Version": "0",
  "interface": "eth0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
```

```
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_interface_network_rx_bytes": 9880.697124714186,
  "pod_interface_network_rx_dropped": 0,
  "pod_interface_network_rx_errors": 0,
  "pod_interface_network_rx_packets": 107.80005532263283,
  "pod_interface_network_total_bytes": 10158.829201483635,
  "pod_interface_network_tx_bytes": 278.13207676944796,
  "pod_interface_network_tx_dropped": 0,
  "pod_interface_network_tx_errors": 0,
  "pod_interface_network_tx_packets": 1.146027574644318
}
```

유형: 컨테이너

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-NodeGroup-sample",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "Container",
  "Version": "0",
  "container_cpu_limit": 200,
  "container_cpu_request": 200,
  "container_cpu_usage_system": 1.87958283771964,
  "container_cpu_usage_total": 6.159993652997942,
  "container_cpu_usage_user": 1.6707403001952357,
  "container_cpu_utilization": 0.15399984132494854,
  "container_memory_cache": 8192,
  "container_memory_failcnt": 0,
  "container_memory_hierarchical_pgfault": 0,
  "container_memory_hierarchical_pgmajfault": 0,
  "container_memory_limit": 104857600,
  "container_memory_mapped_file": 0,
  "container_memory_max_usage": 24580096,
  "container_memory_pgfault": 0,
  "container_memory_pgmajfault": 0,
  "container_memory_request": 104857600,
  "container_memory_rss": 22736896,
  "container_memory_swap": 0,
  "container_memory_usage": 24453120,
  "container_memory_utilization": 0.10574541028701798,
  "container_memory_working_set": 17580032,
  "container_status": "Running",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id": "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
```

```
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "number_of_container_restarts": 0
}
```

유형: ContainerFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "ContainerFS",
  "Version": "0",
  "container_filesystem_available": 0,
  "container_filesystem_capacity": 21462233088,
  "container_filesystem_usage": 24576,
  "container_filesystem_utilization": 0.0001145081217748071,
  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id": "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  }
}
```

유형: 클러스터

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "cluster_node_count"
        },
        {
          "Unit": "Count",
          "Name": "cluster_failed_node_count"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "Cluster",
  "Version": "0",
  "cluster_failed_node_count": 0,
  "cluster_node_count": 3
}
```

유형: ClusterService

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "service_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "ClusterService",
```

```
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch",
    "service_name": "cloudwatch-agent-statsd"
  },
  "service_number_of_running_pods": 1
}
```

유형: ClusterNamespace

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "namespace_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097594160",
  "Type": "ClusterNamespace",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch"
  },
  "namespace_number_of_running_pods": 7
}
```

Amazon EKS 및 Kubernetes에 대한 성능 로그 이벤트의 관련 필드

Amazon EKS 및 Kubernetes의 경우, 컨테이너화된 CloudWatch 에이전트는 성능 로그 이벤트로 데이터를 방출합니다. 따라서 CloudWatch가 카디널리티가 높은 데이터를 수집 및 저장할 수 있습니다. CloudWatch은 성능 로그 이벤트의 데이터를 사용하여 세부적인 정보를 잃지 않고도 클러스터, 노드 및 Pod 수준에서 집계된 CloudWatch 지표를 생성합니다.

다음 표에는 이러한 성능 로그 이벤트에서 Container Insights 지표 데이터의 수집과 관련된 필드가 나와 있습니다. CloudWatch Logs Insights을 사용해 이러한 필드 중 어떤 것에 대해서든 쿼리를 수행하여 데이터를 수집하거나 문제를 조사할 수 있습니다. 자세한 내용은 CloudWatch Logs Insights로 로그 데이터 분석을 참조하십시오.

유형	로그 필드	Source	공식 또는 노트
Pod	pod_cpu_utilization	계산 완료	공식: pod_cpu_usage_total / node_cpu_limit
Pod	pod_cpu_usage_total	cadvisor	
Pod	pod_cpu_limit	계산 완료	공식: sum(container_cpu_limit) Pod의 어떤 컨테이너에도 CPU 한도가 정의되어 있지 않은 경우에는 로그 이벤트에 이 필드가 나타나지 않습니다.
Pod	pod_cpu_request	계산 완료	공식: sum(container_cpu_reques container_cpu_request이 설정되어 있지 않을 수도 있습니다. 설정된 것만 합계에 포함됩니다.
Pod	pod_cpu_utilization_over_pod_limit	계산 완료	공식: pod_cpu_usage_total / pod_cpu_limit
Pod	pod_cpu_reserved_capacity	계산 완료	공식: pod_cpu_request / node_cpu_limit
Pod	pod_memory_utilization	계산 완료	공식: pod_memory_working_set / node_memory_limit
Pod	pod_memory_working_set	cadvisor	
Pod	pod_memory_limit	계산 완료	공식: sum(container_memory_lim Pod의 어떤 컨테이너에도 메모리 한도가 정의되어 있지 않은 경우에는 로그 이벤트에 이 필드가 나타나지 않습니다.
Pod	pod_memory_request	계산 완료	공식: sum(container_memory_req container_memory_request 설정되어 있지 않

유형	로그 필드	Source	공식 또는 노트
			을 수도 있습니다. 설정된 것만 합계에 포함됩니다.
Pod	pod_memory_utilization_over_pod_limi	계산 완료	공식: pod_memory_working_set / pod_memory_limit Pod의 어떤 컨테이너에도 메모리 한도가 정의되어 있지 않은 경우에는 로그 이벤트에 이 필드가 나타나지 않습니다.
Pod	pod_memory_reserved_capacity	계산 완료	공식: pod_memory_request / node_memory_limit
Pod	pod_network_tx_bytes	계산 완료	공식: sum(pod_interface_networ 이 데이터는 Pod별로 모든 네트워크 인스턴스에서 사용할 수 있습니다. CloudWatch 에이전트는 총 합계를 계산하고 지표 추출 규칙을 추가합니다.
Pod	pod_network_rx_bytes	계산 완료	공식: sum(pod_interface_networ
Pod	pod_network_total_bytes	계산 완료	공식: pod_network_rx_bytes + pod_network_tx_bytes
PodNet	pod_interface_network_rx_bytes	cadvisor	이 데이터는 Pod 네트워크 인터페이스에 대한 초당 네트워크 rx 바이트입니다.
PodNet	pod_interface_network_tx_bytes	cadvisor	이 데이터는 Pod 네트워크 인터페이스에 대한 초당 네트워크 tx 바이트입니다.
컨테이너	container_cpu_usage_total	cadvisor	

유형	로그 필드	Source	공식 또는 노트
컨테이너	container_cpu_limit	cadvisor	설정되어 있지 않을 수도 있습니다. 설정이 되지 않은 경우에는 표시되지 않습니다.
컨테이너	container_cpu_request	cadvisor	설정되어 있지 않을 수도 있습니다. 설정이 되지 않은 경우에는 표시되지 않습니다.
컨테이너	container_memory_working_set	cadvisor	
컨테이너	container_memory_limit	pod	설정되어 있지 않을 수도 있습니다. 설정이 되지 않은 경우에는 표시되지 않습니다.
컨테이너	container_memory_request	pod	설정되어 있지 않을 수도 있습니다. 설정이 되지 않은 경우에는 표시되지 않습니다.
ContainerFS	container_filesystem_capacity	pod	이 데이터는 디스크 디바이스별로 사용할 수 있습니다.
ContainerFS	container_filesystem_usage	pod	이 데이터는 디스크 디바이스별로 사용할 수 있습니다.
ContainerFS	container_filesystem_utilization	계산 완료	공식: container_filesystem_usa container_filesystem_cap 이 데이터는 디스크 이름별로 사용할 수 있습니다.
노드	node_cpu_utilization	계산 완료	공식: node_cpu_usage_total / node_cpu_limit
노드	node_cpu_usage_total	cadvisor	
노드	node_cpu_limit	/proc	
노드	node_cpu_request	계산 완료	공식: sum(pod_cpu_request)

유형	로그 필드	Source	공식 또는 노트
노드	node_cpu_reserved_capacity	계산 완료	공식: node_cpu_request / node_cpu_limit
노드	node_memory_utilization	계산 완료	공식: node_memory_working_set / node_memory_limit
노드	node_memory_working_set	cadvisor	
노드	node_memory_limit	/proc	
노드	node_memory_request	계산 완료	공식: sum(pod_memory_request)
노드	node_memory_reserved_capacity	계산 완료	공식: node_memory_request / node_memory_limit
노드	node_network_rx_bytes	계산 완료	공식: sum(node_interface_netwo
노드	node_network_tx_bytes	계산 완료	공식: sum(node_interface_netwo
노드	node_network_total_bytes	계산 완료	공식: node_network_rx_bytes + node_network_tx_bytes
노드	node_number_of_running_pods	Pod 목록	
노드	node_number_of_running_containers	Pod 목록	
NodeNet	node_interface_network_rx_bytes	cadvisor	이 데이터는 작업자 노드 네트워크 인터페이스에 대한 초당 네트워크 rx 바이트입니다.
NodeNet	node_interface_network_tx_bytes	cadvisor	이 데이터는 작업자 노드 네트워크 인터페이스에 대한 초당 네트워크 tx 바이트입니다.
NodeFS	node_filesystem_capacity	cadvisor	
NodeFS	node_filesystem_usage	cadvisor	
NodeFS	node_filesystem_utilization	계산 완료	공식: node_filesystem_usage / node_filesystem_capacity 이 데이터는 디스크 이름별로 사용할 수 있습니다.

유형	로그 필드	Source	공식 또는 노트
클러스터	cluster_failed_node_count	API 서버	
클러스터	cluster_node_count	API 서버	
서비스	service_number_of_running_pods	API 서버	
네임스페이스	namespace_number_of_running_pods	API 서버	

지표 계산 예제

이 섹션에는 앞의 표에서 몇몇 값들이 어떻게 계산되었는지를 보여주는 예제가 포함되어 있습니다.

클러스터가 다음 상태에 있다고 가정해 보겠습니다.

```
Node1
   node_cpu_limit = 4
   node_cpu_usage_total = 3

   Pod1
     pod_cpu_usage_total = 2

     Container1
        container_cpu_limit = 1
        container_cpu_request = 1
        container_cpu_usage_total = 0.8

     Container2
        container_cpu_limit = null
        container_cpu_request = null
        container_cpu_usage_total = 1.2

   Pod2
     pod_cpu_usage_total = 0.4

     Container3
        container_cpu_limit = 1
        container_cpu_request = 0.5
        container_cpu_usage_total = 0.4

Node2
   node_cpu_limit = 8
   node_cpu_usage_total = 1.5

   Pod3
     pod_cpu_usage_total = 1

     Container4
        container_cpu_limit = 2
        container_cpu_request = 2
        container_cpu_usage_total = 1
```

다음 표에는 이 데이터를 사용하여 Pod CPU 지표를 계산하는 방법이 나와 있습니다.

지표	공식	Pod1	Pod2	Pod3
pod_cpu_utilization	pod_cpu_usage_total / node_cpu_limit	2 / 4 = 50%	0.4 / 4 = 10%	1 / 8 = 12.5%

지표	공식	Pod1	Pod2	Pod3
pod_cpu_utilization_over_pod_limit	pod_cpu_usage_total / sum(container_cpu_limit)	Container2에 대한 CPU 한도가 정의되어 있지 않기 때문에 해당 사항 없음	0.4 / 1 = 40%	1 / 2 = 50%
pod_cpu_reserved_capacity	sum(container_cpu_request) / node_cpu_limit	(1 + 0) / 4 = 25%	0.5 / 4 = 12.5%	2 / 8 = 25%

다음 표에는 이 데이터를 사용하여 노드 CPU 지표를 계산하는 방법이 나와 있습니다.

지표	공식	노드 1	노드 2
node_cpu_utilization	node_cpu_usage_total / node_cpu_limit	3 / 4 = 75%	1.5 / 8 = 18.75%
node_cpu_reserved_capacity	sum(pod_cpu_request) / node_cpu_limit	1.5 / 4 = 37.5%	2 / 8 = 25%

Container Insights 문제 해결

Container Insights에 문제가 있는 경우 다음 단원을 참조하면 도움이 됩니다.

Amazon EKS 또는 Kubernetes에서 실패한 배포

에이전트가 Kubernetes 클러스터에 올바르게 배포하지 않는 경우 다음을 수행해 보십시오.

- 다음 명령을 실행하여 Pod 목록을 가져옵니다.

```
kubectl get pods -n amazon-cloudwatch
```

- 다음 명령을 실행하고 출력 하단에서 이벤트를 확인합니다.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- 다음 명령을 실행하여 로그를 확인합니다.

```
kubectl logs pod-name -n amazon-cloudwatch
```

권한이 없는 패닉: Kubelet에서 cadvisor 데이터를 검색할 수 없음

`Unauthorized panic: Cannot retrieve cadvisor data from kubelet` 오류가 표시되면서 배포에 실패한 경우 kubelet 에 Webhook 인증 모드가 활성화되어 있지 않을 수 있습니다. 이는 Container Insights에 필요한 모드입니다. 자세한 내용은 사전 조건 확인 (p. 145) 단원을 참조하십시오.

삭제했다가 다시 생성한 클러스터에 Container Insights 배포

Container Insights가 활성화되지 않은 기존 클러스터를 삭제했다가 같은 이름으로 다시 생성하는 경우 다시 생성하는 시점에는 이 새 클러스터에서 Container Insights를 활성화할 수 없습니다. 다시 생성한 후 다음 명령을 입력해야만 활성화할 수 있습니다.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

콘솔에 지표가 나타나지 않음

AWS Management 콘솔에 Container Insights 지표가 보이지 않는 경우 Container Insights 설정을 완료했는지 확인합니다. Container Insights가 완전히 설정되지 않으면 지표가 나타나지 않습니다. 자세한 내용은 Container Insights 설정 (p. 137) 단원을 참조하십시오.

CloudWatch 에이전트의 CrashLoopBackoff 오류

CloudWatch 에이전트에 대해 `CrashLoopBackOff` 오류가 표시되면 IAM 권한이 정확하게 설정되어 있는지 확인합니다. 자세한 내용은 사전 조건 확인 (p. 145) 단원을 참조하십시오.

보류 상태에서 멈춘 CloudWatch 에이전트 또는 FluentD Pod

`Pending` 상태에서 멈춰 있거나 `FailedScheduling` 오류가 있는 CloudWatch 에이전트 또는 FluentD Pod가 있는 경우 에이전트에 필요한 코어의 수와 RAM의 양에 근거하여 노드에 충분한 컴퓨팅 리소스가 있는지 확인하십시오. 다음 명령을 입력하여 Pod를 설명합니다.

```
kubectl describe pod cloudwatch-agent-85ppg -n amazon-cloudwatch
```

자체 CloudWatch 에이전트 도커 이미지 빌드

https://github.com/aws-samples/amazon-cloudwatch-container-insights/blob/master/cloudwatch-agent-dockerfile/Dockerfile에 있는 Dockerfile을 참조하여 자체 CloudWatch 에이전트 도커 이미지를 빌드할 수 있습니다.

CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집

통합 CloudWatch 에이전트를 사용하면 다음을 수행할 수 있습니다.

- 운영 체제 전반에 걸쳐 Amazon EC2 인스턴스로부터 더 많은 시스템 수준 지표를 수집할 수 있습니다. 지표에는 EC2 인스턴스 지표뿐만 아니라 인게스트 지표도 포함될 수 있습니다. 수집할 수 있는 추가 지표는 CloudWatch 에이전트가 수집하는 지표 (p. 255)에 나열되어 있습니다.
- 온프레미스 서버로부터 시스템 수준 지표를 수집합니다. 여기에는 AWS가 관리하지 않는 서버뿐만 아니라 하이브리드 환경의 서버도 포함될 수 있습니다.
- `StatsD` 및 `collectd` 프로토콜을 사용하여 애플리케이션 또는 서비스에서 사용자 지정 지표를 검색할 수 있습니다. `StatsD`는 Windows Server가 실행되는 서버와 Linux 서버에서 모두 지원되며, `collectd`는 Linux 서버에서만 지원됩니다.
- Linux 또는 Windows Server를 실행하는 Amazon EC2 인스턴스 및 온프레미스 서버로부터 로그를 수집합니다.

다른 CloudWatch 지표와 마찬가지로 CloudWatch에서 CloudWatch 에이전트를 사용하여 수집한 지표를 저장하고 볼 수 있습니다. CloudWatch 에이전트가 수집하는 지표의 기본 네임스페이스는 `CWAgent`이지만, 에이전트를 구성할 때 다른 네임스페이스를 지정할 수도 있습니다.

통합 CloudWatch 에이전트에 의해 수집된 로그는 기존 CloudWatch Logs 에이전트에 의해 수집된 로그와 마찬가지로 Amazon CloudWatch Logs에서 처리되고 저장됩니다. CloudWatch Logs 요금에 대한 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

CloudWatch 에이전트에 의해 수집되는 지표는 사용자 지정 지표로 청구됩니다. CloudWatch 지표 요금에 대한 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

이 단원의 단계에서는 Amazon EC2 인스턴스 및 온프레미스 서버에 통합 CloudWatch 에이전트를 설치하는 방법을 설명합니다. CloudWatch 에이전트가 수집할 수 있는 지표에 대한 자세한 내용은 CloudWatch 에이전트가 수집하는 지표 (p. 255) 단원을 참조하십시오.

지원되는 운영 체제

CloudWatch 에이전트는 다음 운영 체제를 실행하는 AMD64 아키텍처에서 지원됩니다.

- Amazon Linux버전 2014.03.02 이상
- Amazon Linux 2
- Ubuntu Server 버전 18.04, 16.04, 14.04
- CentOS 버전 7.6, 7.2, 7.0, 6.8, 6.5
- Red Hat Enterprise Linux(RHEL) 버전 8, 7.6, 7.5, 7.4, 7.2, 7.0, 6.5
- Debian 8.0
- SUSE Linux Enterprise Server(SLES) 버전 12 및 버전 15
- Windows Server 2019, Windows Server 2016, Windows Server 2012 및 Windows Server 2008 R2의 64비트 버전

이 에이전트는 다음 운영 체제를 실행하는 ARM64 아키텍처에서 지원됩니다.

- Amazon Linux 2

- Ubuntu Server 버전 18.04
- Red Hat Enterprise Linux(RHEL) 버전 7.6
- SUSE Linux Enterprise Server 15

설치 프로세스 개요

명령줄을 사용하여 CloudWatch 에이전트를 수동으로 다운로드하고 설치하거나, SSM과 통합할 수 있습니다. 이중 어떤 방법을 사용하든 CloudWatch 에이전트를 설치하는 일반적인 흐름은 다음과 같습니다.

1. 에이전트가 서버로부터 지표를 수집할 수 있도록 하거나 필요한 경우 AWS 시스템 관리자와 통합할 수 있도록 하는 IAM 역할 또는 사용자를 만듭니다.
2. 에이전트 패키지를 다운로드합니다.
3. CloudWatch 에이전트 구성 파일을 수정하고 수집하려는 지표를 지정합니다.
4. 서버에 에이전트를 설치하고 시작합니다. EC2 인스턴스에 에이전트를 설치할 때 1단계에서 만든 IAM 역할을 연결합니다. 온프레미스 서버에 에이전트를 설치할 때 1단계에서 만든 IAM 사용자의 자격 증명이 포함되어 있는 명명된 프로필을 지정합니다.

목차

- CloudWatch 에이전트 설치 (p. 190)
- CloudWatch 에이전트 구성 파일 생성 (p. 222)
- CloudWatch 에이전트가 수집하는 지표 (p. 255)
- CloudWatch 에이전트를 사용하는 일반적인 시나리오 (p. 263)
- CloudWatch 에이전트의 문제 해결 (p. 270)

CloudWatch 에이전트 설치

Amazon S3 다운로드 링크가 있는 명령줄을 사용하거나 SSM 또는 AWS CloudFormation 템플릿을 사용하여 CloudWatch 에이전트를 다운로드하고 설치할 수 있습니다.

목차

- 명령줄을 사용하여 CloudWatch 에이전트 설치 (p. 190)
- AWS 시스템 관리자를 사용하여 CloudWatch 에이전트 설치 (p. 201)
- AWS CloudFormation을 사용하여 CloudWatch 에이전트 설치 (p. 213)
- CloudWatch 에이전트 패키지의 서명 확인 (p. 217)

명령줄을 사용하여 CloudWatch 에이전트 설치

CloudWatch 에이전트 패키지 다운로드, 구성 및 설치에 대해서는 다음 주제를 참조하십시오.

주제

- 명령줄을 사용하여 CloudWatch 에이전트 다운로드 및 구성 (p. 190)
- CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 (p. 194)
- 서버에 CloudWatch 에이전트 설치 및 실행 (p. 195)

명령줄을 사용하여 CloudWatch 에이전트 다운로드 및 구성

다음 단계를 통해 CloudWatch 에이전트 패키지를 다운로드하고, IAM 역할 또는 사용자를 생성하고, 필요한 경우 일반 구성 파일을 수정합니다.

S3 다운로드 링크를 사용하여 CloudWatch 에이전트 패키지 다운로드

Amazon S3 다운로드 링크를 사용하여 CloudWatch 에이전트 패키지를 다운로드할 수 있습니다. 아키텍처 및 플랫폼에 따라 이 테이블에서 다운로드 링크를 선택하십시오.

각 다운로드 링크에는 전체 링크뿐 아니라 리전별 링크도 있습니다. 예를 들어 Amazon Linux, Amazon Linux 2 및 AMD64 아키텍처의 경우 세 개의 유효한 다운로드 링크가 있습니다.

- `https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://s3.us-east-1.amazonaws.com/amazoncloudwatch-agent-us-east-1/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://s3.eu-central-1.amazonaws.com/amazoncloudwatch-agent-eu-central-1/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

또한 에이전트의 최신 변경 사항에 대한 README 파일과 다운로드할 수 있는 버전 번호를 나타내는 파일을 다운로드할 수 있습니다. 이러한 파일은 다음 위치에 있습니다.

- `https://s3.amazonaws.com/amazoncloudwatch-agent/info/latest/RELEASE_NOTES` 또는 `https://s3.region.amazonaws.com/amazoncloudwatch-agent-Region/info/latest/RELEASE_NOTES`
- `https://s3.amazonaws.com/amazoncloudwatch-agent/info/latest/CWAGENT_VERSION` 또는 `https://s3.region.amazonaws.com/amazoncloudwatch-agent-Region/info/latest/CWAGENT_VERSION`

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
AMD64	Amazon Linux 및 Amazon Linux 2	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Centos	https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/centos/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Redhat	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
		https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	SUSE	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/suse/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Debian	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig
AMD64	Ubuntu	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
AMD64	Windows	https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
ARM64	Amazon Linux 2	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Redhat	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Ubuntu	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

명령줄을 사용하여 CloudWatch 에이전트 패키지를 다운로드하고 설치하려면

1. CloudWatch 에이전트를 다운로드합니다.

   Linux 서버의 경우 다음을 입력합니다. *download-link*에 이전 테이블의 해당 다운로드 링크를 사용합니다.

```
wget download-link
```

Windows Server가 실행되는 서버의 경우 다음 파일을 다운로드합니다.

```
https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi
```

2. 패키지를 다운로드했으면 선택 사항으로 패키지 서명을 확인할 수 있습니다. 자세한 내용은 CloudWatch 에이전트 패키지의 서명 확인 (p. 217) 단원을 참조하십시오.
3. 패키지를 설치합니다. Linux 서버에서 RPM 패키지를 다운로드한 경우 패키지가 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Linux 서버에서 DEB 패키지를 다운로드한 경우 패키지가 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Windows Server가 실행되는 서버에서 MSI 패키지를 다운로드한 경우 패키지가 있는 디렉터리로 변경하고 다음을 입력합니다.

```
msiexec /i amazon-cloudwatch-agent.msi
```

또한 이 명령은 PowerShell 내에서도 작동합니다. MSI 명령 옵션에 대한 자세한 내용은 Microsoft Windows 문서의 명령줄 옵션을 참조하십시오.

에이전트 구성 파일 생성 및 수정

CloudWatch 에이전트를 다운로드했으면 서버에서 에이전트를 시작하기 전에 구성 파일을 생성해야 합니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.

CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성

AWS 리소스에 액세스하려면 권한이 필요합니다. CloudWatch 에이전트가 CloudWatch에 지표를 쓰는 데 필요한 권한을 부여하는 IAM 역할과 IAM 사용자 중 하나 또는 모두를 만듭니다. Amazon EC2 인스턴스에서 에이전트를 사용하려면 IAM 역할을 생성해야 합니다. 온프레미스 서버에서 에이전트를 사용하려면 IAM 사용자를 생성해야 합니다.

Note

최근에 당사는 고객들에게 정책을 직접 생성하도록 요구하는 대신 Amazon에서 만든 새로운 `CloudWatchAgentServerPolicy` 및 `CloudWatchAgentAdminPolicy` 정책을 사용하여 다음과 같은 절차를 수정했습니다. Parameter Store에서의 파일 쓰기 및 다운로드는 Amazon에서 만든 정책에 따라 이름이 `AmazonCloudWatch-`로 시작하는 파일만 지원합니다. 파일 이름이 `AmazonCloudWatch-`로 시작하지 않는 CloudWatch 에이전트 구성 파일이 있는 경우 Parameter Store에 파일을 쓰거나 Parameter Store에서 파일 다운로드할 때는 이 정책을 사용할 수 없습니다.

Amazon EC2 인스턴스에서 CloudWatch 에이전트를 실행하려면 다음 단계를 통해 필요한 IAM 역할을 만드십시오. 이 역할은 인스턴스로부터 정보를 읽고 이를 CloudWatch에 쓰는 권한을 제공합니다.

EC2 인스턴스에서 CloudWatch 에이전트를 실행하는 데 필요한 IAM 역할을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Roles(역할)를 선택한 후 Create role(역할 생성)을 선택합니다.
3. Choose the service that will use this role(이 역할을 사용할 서비스 선택)에서 EC2 Allows EC2 instances to call AWS services on your behalf(EC2 인스턴스가 대신 AWS 서비스를 호출할 수 있도록 허용)를 선택합니다. Next: Permissions(다음: 권한)를 선택합니다.
4. 정책 목록에서 CloudWatchAgentServerPolicy 옆의 확인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다.
5. Next: Review(다음: 검토)를 선택합니다.
6. CloudWatchAgentServerPolicy가 Policies(정책) 옆에 표시되는지 확인합니다. Role name(역할 이름)에 *CloudWatchAgentServerRole* 등의 역할 이름을 입력합니다. 필요한 경우 설명을 입력합니다. 그런 다음 역할 생성을 선택합니다.

   이제 역할이 생성되었습니다.

온프레미스 서버에서 CloudWatch 에이전트를 실행하려면 다음 단계를 통해 필요한 IAM 사용자를 만드십시오.

온프레미스 서버에서 CloudWatch 에이전트가 실행되도록 하는 데 필요한 IAM 사용자를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 Users(사용자)를 선택한 후 Add user(사용자 추가)를 선택합니다.
3. 새 사용자의 사용자 이름을 입력합니다.
4. Programmatic access(프로그래밍 방식 액세스)를 선택하고 Next: Permissions(다음: 권한)를 선택합니다.
5. Attach existing policies directly(기존 정책 직접 연결)를 선택합니다.
6. 정책 목록에서 CloudWatchAgentServerPolicy 옆의 확인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다.
7. Next: Review(다음: 검토)를 선택합니다.
8. 올바른 정책이 나열되는지 확인하고 사용자 생성을 선택합니다.
9. 새 사용자 이름 옆에서 표시를 선택합니다. 에이전트를 설치할 때 사용할 수 있도록 액세스 키 및 보안 키를 파일에 복사한 다음, 닫기를 선택합니다.

서버에 CloudWatch 에이전트 설치 및 실행

원하는 에이전트 구성 파일을 생성하고 IAM 역할이나 IAM 사용자를 생성한 후 다음 단계를 따라 해당 구성을 사용하여 서버에 에이전트를 설치하고 실행합니다. 먼저 IAM 역할 또는 IAM 사용자를 에이전트가 실행될 서버에 연결합니다. 그런 다음 해당 서버에 에이전트 패키지를 다운로드한 후 만든 에이전트 구성을 사용하여 시작합니다.

S3 다운로드 링크를 사용하여 CloudWatch 에이전트 패키지 다운로드

에이전트가 실행될 각 서버에 에이전트 패키지를 다운로드합니다. 아키텍처 및 플랫폼에 따라 이 테이블에서 다운로드 링크를 선택하십시오.

각 다운로드 링크에는 전체 링크뿐 아니라 리전별 링크도 있습니다. 예를 들어 Amazon Linux, Amazon Linux 2 및 AMD64 아키텍처의 경우 세 개의 유효한 다운로드 링크가 있습니다.

- `https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://s3.us-east-1.amazonaws.com/amazoncloudwatch-agent-us-east-1/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://s3.eu-central-1.amazonaws.com/amazoncloudwatch-agent-eu-central-1/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
AMD64	Amazon Linux 및 Amazon Linux 2	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Centos	https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/centos/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Redhat	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	SUSE	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/suse/amd64/	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/suse/amd64/

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
		latest/amazon-cloudwatch-agent.rpm	latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Debian	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig
AMD64	Ubuntu	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
AMD64	Windows	https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig
ARM64	Amazon Linux 2	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
ARM64	Redhat	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Ubuntu	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

명령줄을 사용하여 Amazon EC2 인스턴스에 CloudWatch 에이전트를 설치하려면

1. CloudWatch 에이전트를 다운로드합니다. Linux 서버의 경우 다음을 입력합니다. *download-link*에 이전 테이블의 해당 다운로드 링크를 사용합니다.

   ```
   wget download-link
   ```

   Windows 서버를 실행하는 서버의 경우, 다음 파일을 다운로드합니다.

   ```
   https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. 패키지를 다운로드했으면 선택 사항으로 패키지 서명을 확인할 수 있습니다. 자세한 내용은 CloudWatch 에이전트 패키지의 서명 확인 (p. 217) 단원을 참조하십시오.
3. 패키지를 설치합니다. Linux 서버에서 RPM 패키지를 다운로드한 경우 패키지가 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Linux 서버에서 DEB 패키지를 다운로드한 경우 패키지가 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Windows Server가 실행되는 서버에서 MSI 패키지를 다운로드한 경우 패키지가 있는 디렉터리로 변경하고 다음을 입력합니다.

```
msiexec /i amazon-cloudwatch-agent.msi
```

또한 이 명령은 PowerShell 내에서도 작동합니다. MSI 명령 옵션에 대한 자세한 내용은 Microsoft Windows 문서의 명령줄 옵션을 참조하십시오.

(EC2 인스턴스에 설치) IAM 역할 연결

CloudWatch 에이전트가 인스턴스의 데이터를 보낼 수 있도록 하려면 IAM 역할을 해당 인스턴스에 연결해야 합니다. 연결할 역할은 CloudWatchAgentServerRole입니다.

인스턴스에 IAM 역할을 연결하는 방법에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 IAM 역할을 인스턴스에 연결 단원을 참조하십시오.

(온프레미스 서버에 설치) IAM 자격 증명 및 AWS 리전 지정

CloudWatch 에이전트가 온프레미스 서버의 데이터를 보낼 수 있도록 하려면 이전에 만든 IAM 사용자의 액세스 키와 보안 키를 지정해야 합니다. 이 사용자를 생성하는 방법에 대한 자세한 내용은 CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 (p. 194) 단원을 참조하십시오.

또한 다음 예에 표시된 대로 AWS 구성 파일의 `[AmazonCloudWatchAgent]` 섹션에 있는 `region` 필드를 사용하여 지표를 보낼 AWS 리전을 지정해야 합니다.

```
[profile AmazonCloudWatchAgent]
region = us-west-1
```

다음은 `aws configure` 명령을 사용하여 CloudWatch 에이전트에 대한 명명된 프로필을 만드는 예입니다. 이 예에서는 `AmazonCloudWatchAgent`의 기본 프로필 이름을 사용하는 것으로 가정합니다.

CloudWatch 에이전트용 AmazonCloudWatchAgent 프로필을 생성하려면

- Linux 서버에서 다음 명령을 입력하고 표시되는 메시지를 따릅니다.

  ```
  sudo aws configure --profile AmazonCloudWatchAgent
  ```

  Windows Server에서는 관리자 권한으로 PowerShell을 열고 다음 명령을 입력한 후 표시되는 메시지를 따릅니다.

  ```
  aws configure --profile AmazonCloudWatchAgent
  ```

(선택 사항) 프록시 또는 리전 정보에 대한 일반 구성 수정

CloudWatch 에이전트에는 `common-config.toml` 구성 파일이 있습니다. 필요한 경우 이 파일을 사용하여 프록시 및 리전 정보를 지정할 수 있습니다.

Linux를 실행하는 서버에서는 이 파일이 `/opt/aws/amazon-cloudwatch-agent/etc` 디렉터리에 있습니다. Windows Server를 실행하는 서버에서는 이 파일이 `C:\ProgramData\Amazon\AmazonCloudWatchAgent` 디렉터리에 있습니다.

기본 `common-config.toml`은 다음과 같습니다.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

처음에는 모든 줄이 코멘트 아웃 처리되어 있습니다. 자격 증명 프로필이나 프록시 설정을 설정하려면 해당 줄에서 #을 제거하고 값을 지정하십시오. 이 파일은 수동으로 편집하거나 시스템 관리자의 `RunShellScript` Run Command를 사용하여 편집할 수 있습니다.

- `shared_credential_profile` – 온프레미스 서버의 경우 이 줄은 CloudWatch에 데이터를 보내는 데 사용할 IAM 사용자 자격 증명 프로필을 지정합니다. 이 줄을 코멘트 아웃 처리된 상태로 유지할 경우 `AmazonCloudWatchAgent`가 사용됩니다. 이 프로필 생성에 대한 자세한 내용은 (온프레미스 서버에 설치) IAM 자격 증명 및 AWS 리전 지정 (p. 199) 단원을 참조하십시오.

  EC2 인스턴스에서 이 줄을 사용하여 CloudWatch 에이전트가 이 인스턴스의 데이터를 다른 AWS 리전의 CloudWatch에 보내도록 할 수 있습니다. 이렇게 하려면 보낼 리전의 이름을 지정하는 `region` 필드가 있는 명명된 프로필을 지정합니다.

  `shared_credential_profile`을 지정하는 경우 `[credentials]` 행의 시작 부분에서 #도 제거해야 합니다.
- `shared_credential_file` – 에이전트가 기본 경로 이외의 다른 경로에 있는 파일에서 자격 증명을 찾도록 하려면 여기에 전체 경로 및 파일 이름을 지정합니다. 기본 경로는 Linux의 경우 `/root/.aws`이며 Windows Server의 경우 `C:\\Users\\Administrator\\.aws`입니다.

  아래의 첫 번째 예는 Linux 서버의 유효한 `shared_credential_file` 행 구문이고 두 번째 예는 Windows Server에 유효한 행 구문입니다. Windows Server에서는 \ 문자를 이스케이프 처리해야 합니다.

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  `shared_credential_file`을 지정하는 경우 `[credentials]` 행의 시작 부분에서 #도 제거해야 합니다.
- 프록시 설정 – 서버가 HTTP 또는 HTTPS 프록시를 사용하여 AWS 서비스에 접근하도록 하려면 해당 프록시를 `http_proxy` 및 `https_proxy` 필드에 지정합니다. 프록시 설정에서 제외해야 하는 URL이 있다면 이를 쉼표로 구분하여 `no_proxy` 필드에 지정하십시오.

명령줄을 사용하여 CloudWatch 에이전트 시작

다음 단계에 따라 명령줄을 사용하여 서버에서 CloudWatch 에이전트를 시작합니다.

명령줄을 사용하여 서버에서 CloudWatch 에이전트를 시작하려면

1. 사용하려는 에이전트 구성 파일을 에이전트를 실행할 서버에 복사합니다. 해당 파일을 복사할 경로 이름을 기록해 둡니다.
2. 이 명령에서 `-a fetch-config`는 에이전트가 최신 버전의 CloudWatch 에이전트 구성 파일을 로드하게 하며, `-s`는 에이전트를 시작합니다.

   Linux가 실행되는 EC2 인스턴스의 경우 다음을 입력합니다.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
   m ec2 -c file:configuration-file-path -s
   ```

   Linux가 실행되는 온프레미스 서버의 경우 다음을 입력합니다.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
   m onPremise -c file:configuration-file-path -s
   ```

   Windows Server가 실행되는 EC2 인스턴스의 경우 PowerShell 콘솔에서 다음을 입력합니다.

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -a
    fetch-config -m ec2 -c file:configuration-file-path -s
   ```

   Windows Server가 실행되는 온프레미스 서버의 경우 PowerShell 콘솔에서 다음을 입력합니다.

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -a
    fetch-config -m onPremise -c file:configuration-file-path -s
   ```

AWS 시스템 관리자를 사용하여 CloudWatch 에이전트 설치

AWS 시스템 관리자를 사용하여 CloudWatch 에이전트를 설치하고 실행하는 방법에 대해서는 다음 주제를 참조하십시오.

주제

- CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 (p. 201)
- CloudWatch 에이전트 다운로드 및 구성 (p. 205)
- 에이전트 구성을 사용하여 EC2 인스턴스에 CloudWatch 에이전트 설치 (p. 206)
- 온프레미스 서버에 CloudWatch 에이전트 설치 (p. 210)

CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성

AWS 리소스에 액세스하려면 권한이 필요합니다. CloudWatch 에이전트가 CloudWatch에 지표를 쓰고 CloudWatch 에이전트가 Amazon EC2 및 AWS 시스템 관리자와 통신하도록 하는 데 필요한 권한이 있는 IAM 역할 및 사용자를 생성할 수 있습니다. Amazon EC2 인스턴스에서는 IAM 역할, 온프레미스 서버에서는 IAM 사용자를 사용합니다.

하나의 역할 또는 사용자를 사용하면 CloudWatch 에이전트가 서버에 설치되어 CloudWatch에 지표를 보낼 수 있습니다. 시스템 관리자 Parameter Store에 CloudWatch 에이전트 구성을 저장하려면 다른 역할 또는 사용자가 필요합니다. 파라미터 스토어를 사용하면 여러 서버가 하나의 CloudWatch 에이전트 구성을 사용할 수 있습니다.

Parameter Store에 쓰는 기능은 광범위하고 강력한 권한이며, 필요 시에만 사용해야 하고, 배포 시 여러 인스턴스에 연결해선 안 됩니다. CloudWatch 에이전트 구성을 Parameter Store에 저장하는 경우에는 다음 설정을 권장합니다.

- 이 구성을 수행할 단일 인스턴스를 설정합니다.
- 이 인스턴스에서만 Parameter Store에 쓰기를 할 수 있는 권한과 함께 IAM 역할을 사용합니다.
- CloudWatch 에이전트 구성 파일로 작업하고 저장하는 동안에만 Parameter Store에 쓰기를 할 수 있는 권한과 함께 IAM 역할을 사용합니다.

  Note

  최근에 당사는 고객들에게 정책을 직접 생성하도록 요구하는 대신 Amazon에서 만든 새로운 `CloudWatchAgentServerPolicy` 및 `CloudWatchAgentAdminPolicy` 정책을 사용하여 다음과 같은 절차를 수정했습니다. 이러한 정책을 사용하여 Parameter Store에 에이전트 구성 파일을 작성한 다음 Parameter Store에서 다운로드하려면, 에이전트 구성 파일에 `AmazonCloudWatch-`로 시작하는 이름이 있어야 합니다. 파일 이름이 `AmazonCloudWatch-`로 시작하지 않는 CloudWatch 에이전트 구성 파일이 있는 경우, Parameter Store에 파일을 쓰거나 Parameter Store에서 파일 다운로드할 때는 이 정책을 사용할 수 없습니다.

Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성

첫 번째 절차에서는 CloudWatch 에이전트를 실행할 각 Amazon EC2 인스턴스에 연결해야 하는 IAM 역할을 생성합니다. 이 역할은 인스턴스로부터 정보를 읽고 이를 CloudWatch에 쓰는 권한을 제공합니다.

두 번째 절차에서는 IAM 역할을 생성하는데 이 역할을 CloudWatch 에이전트 구성 파일 생성에 사용되는 Amazon EC2 인스턴스에 연결해야 합니다. 다른 서버에서 사용할 수 있도록 이 파일을 시스템 관리자 Parameter Store에 저장하려면 이 단계가 필요합니다. 이 역할은 인스턴스의 정보를 읽고 이를 CloudWatch에 쓰는 권한뿐만 아니라 Parameter Store에 쓰는 권한도 제공합니다. 이 역할에는 Parameter Store에 쓰는 데 필요한 권한뿐만 아니라 CloudWatch 에이전트를 실행하기에도 충분한 권한이 들어 있습니다.

각 서버가 CloudWatch 에이전트를 실행하는 데 필요한 IAM 역할을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. Select type of trusted entity(신뢰할 수 있는 유형의 엔터티 선택) 아래에서 AWS 서비스를 선택합니다.
4. Choose the service that will use this role(이 역할을 사용할 서비스 선택)에서 EC2와 Next: Permissions(다음: 권한)를 차례대로 선택합니다.
5. 정책 목록에서 CloudWatchAgentServerPolicy 옆의 확인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다.
6. 시스템 관리자를 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonSSMManagedInstanceCore 옆의 확인란을 선택하십시오. 이 AWS 관리형 정책을 사용하면 인스턴스가 시스템 관리자 서비스 핵심 기능을 사용할 수 있습니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
7. Next: Tags(다음: 태그)를 선택합니다.
8. (선택 사항) 이 역할에 대한 액세스를 구성, 추적 또는 제어할 태그-키 값 페어를 하나 이상 추가한 후 Next: Review(다음: 검토)를 선택합니다.

9. 역할 이름에 새 역할의 이름(예: **CloudWatchAgentServerRole** 또는 자신이 선호하는 다른 이름)을 입력합니다.
10. (선택 사항) 역할 설명에 설명을 입력합니다.
11. CloudWatchAgentServerPolicy 및 AmazonSSMManagedInstanceCore이 정책 옆에 표시되는지 확인하십시오.
12. Create role을 선택합니다.

    이제 역할이 생성되었습니다.

다음 절차에서는 Parameter Store에 대한 쓰기 권한도 있는 IAM 역할을 생성합니다. 이 역할을 사용하여 에이전트 구성 파일을 Parameter Store에 저장하면 다른 서버가 이를 검색할 수 있습니다.

Parameter Store에 대한 쓰기 권한은 광범위한 기능을 제공하므로, 이 역할은 일부 서버에만 연결해야 하며, 관리자만 사용해야 합니다. 에이전트 구성 파일을 생성하고 Parameter Store에 복사하는 작업을 마친 후 이 역할을 인스턴스로부터 분리하고 대신 `CloudWatchAgentServerRole`을 사용해야 합니다.

관리자가 Parameter Store에 쓰기를 할 수 있도록 IAM 역할을 생성하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. Select type of trusted entity(신뢰할 수 있는 유형의 엔터티 선택) 아래에서 AWS 서비스를 선택합니다.
4. Choose the service that will use this role(이 역할을 사용할 서비스 선택)에서 EC2와 Next: Permissions(다음: 권한)를 차례대로 선택합니다.
5. 정책 목록에서 CloudWatchAgentAdminPolicy 옆의 확인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다.
6. 시스템 관리자를 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonSSMManagedInstanceCore 옆의 확인란을 선택하십시오. 이 AWS 관리형 정책을 사용하면 인스턴스가 시스템 관리자 서비스 핵심 기능을 사용할 수 있습니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
7. Next: Tags(다음: 태그)를 선택합니다.
8. (선택 사항) 이 역할에 대한 액세스를 구성, 추적 또는 제어할 태그-키 값 페어를 하나 이상 추가한 후 Next: Review(다음: 검토)를 선택합니다.
9. 역할 이름에 새 역할의 이름(예: **CloudWatchAgentAdminRole** 또는 자신이 선호하는 다른 이름)을 입력합니다.
10. (선택 사항) 역할 설명에 설명을 입력합니다.
11. CloudWatchAgentAdminPolicy 및 AmazonSSMManagedInstanceCore(선택 사항)이 정책 옆에 표시되는지 확인하십시오.
12. Create role을 선택합니다.

    이제 역할이 생성되었습니다.

온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성

첫 번째 절차에서는 CloudWatch 에이전트를 실행하는 데 필요한 IAM 사용자를 생성합니다. 이 사용자는 CloudWatch에 데이터를 전송할 수 있는 권한을 제공합니다.

두 번째 절차에서는 CloudWatch 에이전트 구성 파일을 생성할 때 사용할 수 있는 IAM 사용자를 생성합니다. 다른 서버에서 이를 사용할 수 있도록 이 절차를 사용하여 시스템 관리자 Parameter Store에 이 파일을 저장합니다. 이 사용자는 Parameter Store에 대한 쓰기 권한과 함께 CloudWatch에 대한 데이터 쓰기 권한을 제공합니다.

CloudWatch 에이전트가 CloudWatch에 데이터를 쓰는 데 필요한 IAM 사용자를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자와 Add user(사용자 추가)를 차례대로 선택합니다.
3. 새 사용자의 사용자 이름을 입력합니다.
4. Access type(액세스 유형)에서 Programmatic access(프로그래밍 방식 액세스)를 선택한 다음 Next: Permissions(다음: 권한)를 선택합니다.
5. Set permissions(권한 설정)에서 Attach existing policies directly(기존 정책을 직접 연결)를 선택합니다.
6. 정책 목록에서 CloudWatchAgentServerPolicy 옆의 확인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다.
7. 시스템 관리자를 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonSSMManagedInstanceCore 옆의 확인란을 선택하십시오. 이 AWS 관리형 정책을 사용하면 인스턴스가 시스템 관리자 서비스 핵심 기능을 사용할 수 있습니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
8. Next: Tags(다음: 태그)를 선택합니다.
9. (선택 사항) 이 역할에 대한 액세스를 구성, 추적 또는 제어할 태그-키 값 페어를 하나 이상 추가한 후 Next: Review(다음: 검토)를 선택합니다.
10. 올바른 정책이 나열되는지 확인하고 사용자 생성을 선택합니다.
11. 새 사용자를 위한 행에서 표시를 선택합니다. 에이전트를 설치할 때 사용할 수 있도록 액세스 키 및 보안 키를 파일에 복사한 다음, 닫기를 선택합니다.

다음 절차에서는 Parameter Store에 대한 쓰기 권한도 있는 IAM 사용자를 생성합니다. 에이전트 구성 파일을 Parameter Store에 저장하여 다른 서버가 사용할 수 있도록 하려는 경우 이 IAM 사용자를 사용해야 합니다. 이 IAM 사용자는 Parameter Store에 대한 쓰기 권한을 제공합니다. 이 사용자는 인스턴스로부터 정보를 읽고 이를 CloudWatch에 쓰는 권한을 제공합니다. 시스템 관리자 Parameter Store에 대한 쓰기 권한은 광범위한 기능을 제공하므로, 이 IAM 사용자는 일부 서버에만 연결해야 하며, 관리자만 사용해야 합니다. 에이전트 구성 파일을 Parameter Store에 저장하는 경우에만 이 IAM 사용자를 사용해야 합니다.

Parameter Store에 구성 파일을 저장하고 CloudWatch에 정보를 전송하는 데 필요한 IAM 사용자를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 사용자와 Add user(사용자 추가)를 차례대로 선택합니다.
3. 새 사용자의 사용자 이름을 입력합니다.
4. Access type(액세스 유형)에서 Programmatic access(프로그래밍 방식 액세스)를 선택한 다음 Next: Permissions(다음: 권한)를 선택합니다.
5. Set permissions(권한 설정)에서 Attach existing policies directly(기존 정책을 직접 연결)를 선택합니다.
6. 정책 목록에서 CloudWatchAgentAdminPolicy 옆의 확인란을 선택합니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다.
7. 시스템 관리자를 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonSSMManagedInstanceCore 옆의 확인란을 선택합니다. 이 AWS 관리형 정책을 사용하면 인스턴스가 시스템 관리자 서비스 핵심 기능을 사용할 수 있습니다. 필요한 경우 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
8. Next: Tags(다음: 태그)를 선택합니다.
9. (선택 사항) 이 역할에 대한 액세스를 구성, 추적 또는 제어할 태그-키 값 페어를 하나 이상 추가한 후 Next: Review(다음: 검토)를 선택합니다.
10. 올바른 정책이 나열되는지 확인하고 사용자 생성을 선택합니다.
11. 새 사용자를 위한 행에서 표시를 선택합니다. 에이전트를 설치할 때 사용할 수 있도록 액세스 키 및 보안 키를 파일에 복사한 다음, 닫기를 선택합니다.

CloudWatch 에이전트 다운로드 및 구성

이 단원에서는 시스템 관리자를 사용하여 에이전트를 다운로드하는 방법과 에이전트 구성 파일을 생성하는 방법을 설명합니다. 시스템 관리자를 사용하여 에이전트를 다운로드하기 전에 먼저 인스턴스가 시스템 관리자에 대해 올바르게 구성되어 있는지 확인해야 합니다.

SSM Agent 설치 또는 업데이트

Amazon EC2 인스턴스에서는 CloudWatch 에이전트를 사용하려면 해당 인스턴스가 2.2.93.0 이상 버전을 실행하고 있어야 합니다. CloudWatch 에이전트를 설치하기 전에, 인스턴스에 SSM Agent를 업데이트하거나 설치합니다(아직 하지 않은 경우).

Linux를 실행하는 인스턴스에서 SSM Agent를 설치하거나 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 Linux 인스턴스에서 SSM Agent 설치 및 구성을 참조하십시오.

SSM Agent를 설치하거나 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 SSM Agent로 작업을 참조하십시오.

(선택 사항) 시스템 관리자 사전 조건 확인

시스템 관리자 Run Command를 사용하여 CloudWatch 에이전트를 설치 및 구성하기 전에 인스턴스가 최소 시스템 관리자 요구 사항을 충족하는지 확인하십시오. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 시스템 관리자 사전 조건을 참조하십시오.

인터넷 액세스 확인

데이터를 CloudWatch 또는CloudWatch Logs로 보내려면 Amazon EC2 인스턴스에 아웃바운드 인터넷 액세스 권한이 있어야 합니다. 인터넷 액세스 구성 방법에 대한 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 참조하십시오.

프록시에서 구성할 엔드포인트와 포트는 다음과 같습니다.

- 에이전트를 사용하여 지표를 수집하는 경우 해당 리전의 CloudWatch 엔드포인트를 화이트리스트로 지정해야 합니다. 이러한 엔드포인트 목록은 Amazon Web Services 일반 참조의 Amazon CloudWatch를 참조하십시오.
- 에이전트를 사용하여 로그를 수집하는 경우 해당 리전의 CloudWatch Logs 엔드포인트를 화이트리스트로 지정해야 합니다. 이러한 엔드포인트 목록은 Amazon Web Services 일반 참조의 Amazon CloudWatch Logs를 참조하십시오.
- 시스템 관리자을 사용하여 에이전트를 설치하는 경우 또는 Parameter Store를 사용하여 구성 파일을 저장하는 경우 해당 리전의 시스템 관리자 엔드포인트를 화이트리스트로 지정해야 합니다. 이러한 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 시스템 관리자를 참조하십시오.

다음 단계를 통해 시스템 관리자를 사용하여 CloudWatch 에이전트 패키지를 다운로드합니다.

시스템 관리자를 사용하여 CloudWatch 에이전트를 다운로드하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose Run Command.

   -or-

   If the AWS 시스템 관리자 home page opens, scroll down and choose Explore Run Command.
3. Run 명령을 선택합니다.
4. 명령 문서 목록에서 AWS-ConfigureAWSPackage를 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치할 인스턴스를 선택합니다. 특정 인스턴스가 표시되지 않으면 시스템 관리자에서 사용할 관리형 인스턴스로 구성되지 않은 것일 수 있습니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 하이브리드 환경에 대한 AWS 시스템 관리자 설정을 참조하십시오.

6. 작업 목록에서 설치를 선택합니다.
7. 이름 필드에 *AmazonCloudWatchAgent*를 입력합니다.
8. 최신 버전의 에이전트를 설치하도록 버전을 최신 상태로 설정한 채로 유지합니다.
9. 실행을 선택합니다.
10. 선택적으로, Targets and outputs(대상 및 결과) 영역에서 인스턴스 이름 옆에 있는 버튼을 선택하고 View output(결과 보기)을 선택합니다. 시스템 관리자에 에이전트가 성공적으로 설치되었음이 표시되어야 합니다.

에이전트 구성 파일 생성 및 수정

CloudWatch 에이전트를 다운로드했으면 서버에서 에이전트를 시작하기 전에 구성 파일을 생성해야 합니다.

시스템 관리자 Parameter Store에 에이전트 구성 파일을 저장하려는 경우 EC2 인스턴스를 사용하여 Parameter Store에 저장해야 합니다. 또한 이 인스턴스에 `CloudWatchAgentAdminRole` IAM 역할을 연결해야 합니다. 역할 연결에 대한 자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 IAM 역할을 인스턴스에 연결 단원을 참조하십시오.

CloudWatch 에이전트 구성 파일 생성에 대한 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.

에이전트 구성을 사용하여 EC2 인스턴스에 CloudWatch 에이전트 설치

CloudWatch 에이전트 구성을 Parameter Store에 저장했으면 다른 서버에 에이전트를 설치할 때 이를 사용할 수 있습니다.

주제

- IAM 역할을 인스턴스에 연결 (p. 206)
- Amazon EC2 인스턴스에 CloudWatch 에이전트 패키지 다운로드 (p. 206)
- (선택 사항) CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정 (p. 207)
- CloudWatch 에이전트 시작 (p. 208)

IAM 역할을 인스턴스에 연결

EC2 인스턴스에 CloudWatchAgentServerRole IAM 역할을 연결해야 인스턴스에서 CloudWatch 에이전트를 실행할 수 있습니다. 이 역할은 CloudWatch 에이전트가 인스턴스에 대한 작업을 수행하도록 해줍니다.

자세한 내용은 Windows 인스턴스용 Amazon EC2 사용 설명서의 IAM 역할을 인스턴스에 연결을 참조하십시오.

Amazon EC2 인스턴스에 CloudWatch 에이전트 패키지 다운로드

시스템 관리자 Run Command 또는 Amazon S3 다운로드 링크 중 하나를 사용하여 CloudWatch 에이전트 패키지를 다운로드할 수 있습니다. Amazon S3 다운로드 링크 사용에 대한 자세한 내용은 S3 다운로드 링크를 사용하여 CloudWatch 에이전트 패키지 다운로드 (p. 191) 단원을 참조하십시오.

시스템 관리자를 사용하여 Amazon EC2 인스턴스에 CloudWatch 에이전트 다운로드

시스템 관리자를 사용하여 CloudWatch 에이전트를 설치하기 전에 먼저 인스턴스가 시스템 관리자에 대해 올바르게 구성되어 있는지 확인해야 합니다.

SSM Agent 설치 또는 업데이트

Amazon EC2 인스턴스에서는 CloudWatch 에이전트를 사용하려면 해당 인스턴스가 2.2.93.0 이상 버전을 실행하고 있어야 합니다. CloudWatch 에이전트를 설치하기 전에, 인스턴스에 SSM Agent를 업데이트하거나 설치합니다(아직 하지 않은 경우).

Linux를 실행하는 인스턴스에서 SSM Agent를 설치하거나 업데이트하는 방법에 대한 자세한 내용은 AWS 시스템 관리자 사용 설명서의 Linux 인스턴스에서 SSM Agent 설치 및 구성을 참조하십시오.

Windows Server를 실행하는 인스턴스에서 SSM Agent를 설치하거나 업데이트하는 방법에 대한 자세한 내용은 AWS 시스템 관리자 사용 설명서의 Windows 인스턴스에서 SSM 에이전트 설치 및 구성을 참조하십시오.

(선택 사항) 시스템 관리자 사전 조건 확인

시스템 관리자 Run Command를 사용하여 CloudWatch 에이전트를 설치 및 구성하기 전에 인스턴스가 최소 시스템 관리자 요구 사항을 충족하는지 확인하십시오. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 AWS 시스템 관리자 설정을 참조하십시오.

인터넷 액세스 확인

데이터를 CloudWatch 또는CloudWatch Logs 로 전송하려면 Amazon EC2 인스턴스에 아웃바운드 인터넷 액세스 권한이 있어야 합니다. 인터넷 액세스 구성 방법에 대한 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 참조하십시오.

CloudWatch 에이전트 패키지를 다운로드합니다.

시스템 관리자 Run Command를 사용하면 인스턴스 구성을 관리할 수 있습니다. 시스템 관리자 문서를 지정하고 파라미터를 지정한 다음, 하나 이상의 인스턴스에서 명령을 실행합니다. 인스턴스 상의 SSM Agent가 이 명령을 처리하고 지정된 대로 인스턴스를 구성합니다.

Run Command를 사용하여 CloudWatch 에이전트를 다운로드하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose Run Command.

   -or-

   If the AWS 시스템 관리자 home page opens, scroll down and choose Explore Run Command.
3. Run 명령을 선택합니다.
4. 명령 문서 목록에서 AWS-ConfigureAWSPackage를 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치할 인스턴스를 선택합니다. 특정 인스턴스가 보이지 않으면 Run Command에 대해 구성되지 않은 것일 수 있습니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 하이브리드 환경에 대한 AWS 시스템 관리자 설정을 참조하십시오.
6. 작업 목록에서 설치를 선택합니다.
7. 이름 상자에 `AmazonCloudWatchAgent`를 입력합니다.
8. 최신 버전의 에이전트를 설치하도록 버전을 최신 상태로 설정한 채로 유지합니다.
9. 실행을 선택합니다.
10. 선택적으로, Targets and outputs(대상 및 결과) 영역에서 인스턴스 이름 옆에 있는 버튼을 선택하고 View output(결과 보기)을 선택합니다. 시스템 관리자에 에이전트가 성공적으로 설치되었음이 표시되어야 합니다.

(선택 사항) CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정

CloudWatch 에이전트에는 `common-config.toml` 구성 파일이 있습니다. 필요한 경우 이 파일을 사용하여 프록시 및 리전 정보를 지정할 수 있습니다.

Linux를 실행하는 서버에서는 이 파일이 `/opt/aws/amazon-cloudwatch-agent/etc` 디렉터리에 있습니다. Windows Server를 실행하는 서버에서는 이 파일이 `C:\ProgramData\Amazon\AmazonCloudWatchAgent` 디렉터리에 있습니다.

기본 common-config.toml은 다음과 같습니다.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

처음에는 모든 줄이 코멘트 아웃 처리되어 있습니다. 자격 증명 프로필이나 프록시 설정을 설정하려면 해당 줄에서 #을 제거하고 값을 지정하십시오. 이 파일은 수동으로 편집하거나 시스템 관리자의 RunShellScript Run Command를 사용하여 편집할 수 있습니다.

- shared_credential_profile – 온프레미스 서버의 경우 이 줄은 CloudWatch에 데이터를 보내는 데 사용할 IAM 사용자 자격 증명 프로필을 지정합니다. 이 줄을 코멘트 아웃 처리된 상태로 유지할 경우 AmazonCloudWatchAgent가 사용됩니다.

  EC2 인스턴스에서 이 줄을 사용하여 CloudWatch 에이전트가 이 인스턴스의 데이터를 다른 AWS 리전의 CloudWatch에 보내도록 할 수 있습니다. 이렇게 하려면 보낼 리전의 이름을 지정하는 region 필드가 있는 명명된 프로필을 지정합니다.

  shared_credential_profile을 지정하는 경우 [credentials] 행의 시작 부분에서 #도 제거해야 합니다.
- shared_credential_file – 에이전트가 기본 경로 이외의 다른 경로에 있는 파일에서 자격 증명을 찾도록 하려면 여기에 전체 경로 및 파일 이름을 지정합니다. 기본 경로는 Linux의 경우 /root/.aws이며 Windows Server의 경우 C:\\Users\\Administrator\\.aws입니다.

  아래의 첫 번째 예는 Linux 서버의 유효한 shared_credential_file 행 구문이고 두 번째 예는 Windows Server에 유효한 행 구문입니다. Windows Server에서는 \ 문자를 이스케이프 처리해야 합니다.

```
shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
```

```
shared_credential_file= "/usr/username/credentials"
```

  shared_credential_file을 지정하는 경우 [credentials] 행의 시작 부분에서 #도 제거해야 합니다.
- 프록시 설정 – 서버가 HTTP 또는 HTTPS 프록시를 사용하여 AWS 서비스에 접근하도록 하려면 해당 프록시를 http_proxy 및 https_proxy 필드에 지정합니다. 프록시 설정에서 제외해야 하는 URL이 있다면 이를 쉼표로 구분하여 no_proxy 필드에 지정하십시오.

CloudWatch 에이전트 시작

시스템 관리자 Run Command 또는 명령줄을 사용하여 에이전트를 시작할 수 있습니다.

시스템 관리자 Run Command 를 사용하여 CloudWatch 에이전트 시작

시스템 관리자 Run Command를 사용하여 에이전트를 시작하려면 다음 단계를 따르십시오.

Run Command를 사용하여 CloudWatch 에이전트를 시작하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose Run Command.

   -or-

   If the AWS 시스템 관리자 home page opens, scroll down and choose Explore Run Command.
3. Run 명령을 선택합니다.
4. 명령 문서 목록에서 AmazonCloudWatch-ManageAgent를 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치한 인스턴스를 선택합니다.
6. 작업 목록에서 구성을 선택합니다.
7. Optional Configuration Source(구성 소스(선택 사항)) 목록에서 ssm을 선택합니다.
8. Optional Configuration Location(구성 위치(선택 사항)) 상자에 CloudWatch 에이전트 구성 파일 생성 (p. 222)에 설명된 대로 생성하고 시스템 관리자 Parameter Store에 저장한 에이전트 구성 파일의 이름을 입력합니다.
9. Optional Restart(재시작(선택 사항)) 목록에서 예를 선택하여 해당 단계를 마친 후 에이전트가 시작되도록 합니다.
10. 실행을 선택합니다.
11. 선택적으로, Targets and outputs(대상 및 결과) 영역에서 인스턴스 이름 옆에 있는 버튼을 선택하고 View output(결과 보기)을 선택합니다. 시스템 관리자에 에이전트가 성공적으로 시작되었음이 표시되어야 합니다.

명령줄을 사용하여 Amazon EC2 인스턴스에서 CloudWatch 에이전트 시작

명령줄을 사용하여 Amazon EC2 인스턴스에 CloudWatch 에이전트를 설치하려면 다음 단계를 따르십시오.

명령줄을 사용하여 Amazon EC2 인스턴스에서 CloudWatch 에이전트를 시작하려면

- 이 명령에서 `-a fetch-config`는 에이전트가 최신 버전의 CloudWatch 에이전트 구성 파일을 로드하게 하며, `-s`는 에이전트를 시작합니다.

  Linux: 구성 파일을 시스템 관리자 Parameter Store에 저장한 경우 다음을 입력합니다.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
  m ec2 -c ssm:configuration-parameter-store-name -s
  ```

  Linux: 구성 파일을 로컬 컴퓨터에 저장한 경우 다음을 입력합니다.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
  m ec2 -c file:configuration-file-path -s
  ```

  Windows Server: 에이전트 구성 파일을 시스템 관리자 Parameter Store에 저장한 경우 PowerShell 콘솔에서 다음을 입력합니다.

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -a
   fetch-config -m ec2 -c ssm:configuration-parameter-store-name -s
  ```

  Windows Server: 에이전트 구성 파일을 로컬 컴퓨터에 저장한 경우 PowerShell 콘솔에서 다음을 입력합니다.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\config.json" -s
```

온프레미스 서버에 CloudWatch 에이전트 설치

컴퓨터 한 대에 CloudWatch 에이전트를 다운로드하고 원하는 에이전트 구성 파일을 생성한 경우 해당 구성 파일을 사용하여 다른 온프레미스 서버에 에이전트를 설치할 수 있습니다.

온프레미스 서버에 CloudWatch 에이전트 다운로드

시스템 관리자 Run Command 또는 Amazon S3 다운로드 링크 중 하나를 사용하여 CloudWatch 에이전트 패키지를 다운로드할 수 있습니다. Amazon S3 다운로드 링크 사용에 대한 자세한 내용은 S3 다운로드 링크를 사용하여 CloudWatch 에이전트 패키지 다운로드 (p. 191) 단원을 참조하십시오.

시스템 관리자를 사용하여 다운로드

시스템 관리자 Run Command를 사용하려면 Amazon EC2 Systems Manager을 사용하여 온프레미스 서버를 등록해야 합니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 하이브리드 환경에서 시스템 관리자 설정 단원을 참조하십시오.

서버를 이미 등록한 경우 SSM Agent를 최신 버전으로 업데이트하십시오.

Linux를 실행하는 서버에서 SSM Agent를 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 Linux 하이브리드 환경에 SSM Agent 설치를 참조하십시오.

Windows Server를 실행하는 서버에서 SSM Agent를 업데이트하는 방법에 대한 자세한 내용은 AWS 시스템 관리자 사용 설명서의 Windows 하이브리드 환경에 SSM Agent 설치를 참조하십시오.

SSM Agent를 사용하여 CloudWatch 에이전트 패키지를 온프레미스 서버에 다운로드하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose Run Command.

   -or-

   If the AWS 시스템 관리자 home page opens, scroll down and choose Explore Run Command.
3. Run 명령을 선택합니다.
4. 명령 문서 목록에서 AWS-ConfigureAWSPackage 옆의 버튼을 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치할 서버를 선택합니다. 특정 서버가 표시되지 않으면 Run Command에 대해 구성되지 않은 것일 수 있습니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서의 하이브리드 환경에 대한 AWS 시스템 관리자 설정을 참조하십시오.
6. 작업 목록에서 설치를 선택합니다.
7. 이름 상자에 *AmazonCloudWatchAgent*를 입력합니다.
8. 최신 버전의 에이전트를 설치하도록 버전을 비워 둡니다.
9. 실행을 선택합니다.

   에이전트 패키지가 다운로드되며 다음 단계는 이를 구성하고 시작하는 것입니다.

(온프레미스 서버에 설치) IAM 자격 증명 및 AWS 리전 지정

CloudWatch 에이전트가 온프레미스 서버의 데이터를 보낼 수 있도록 하려면 이전에 만든 IAM 사용자의 액세스 키와 보안 키를 지정해야 합니다. 이 사용자 생성에 대한 자세한 내용은 CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 (p. 201) 단원을 참조하십시오.

또한 region 필드를 사용하여 지표를 보낼 AWS 리전을 지정해야 합니다.

다음은 이 파일의 예입니다.

```
[AmazonCloudWatchAgent]
aws_access_key_id=my_access_key
aws_secret_access_key=my_secret_key
region = us-west-1
```

my_access_key 및 *my_secret_key*의 경우 시스템 관리자 Parameter Store에 대한 쓰기 권한이 없는 IAM 사용자의 키를 사용합니다. CloudWatch 에이전트에 필요한 IAM 사용자에 대한 자세한 내용은 온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성 (p. 203) 단원을 참조하십시오.

이 프로필의 이름을 AmazonCloudWatchAgent로 지정하면 아무 작업도 수행할 필요가 없습니다. 다음 단원에서 설명된 대로 필요한 경우 다른 이름을 지정하고 해당 이름을 common-config.toml 파일에서 shared_credential_profile에 대한 값으로 지정할 수 있습니다.

다음은 aws configure 명령을 사용하여 CloudWatch 에이전트에 대한 명명된 프로필을 생성하는 예입니다. 이 예에서는 AmazonCloudWatchAgent의 기본 프로필 이름을 사용하는 것으로 가정합니다.

CloudWatch 에이전트용 AmazonCloudWatchAgent 프로필을 생성하려면

- Linux 서버에서 다음 명령을 입력하고 표시되는 메시지를 따릅니다.

  ```
  sudo aws configure --profile AmazonCloudWatchAgent
  ```

  Windows Server에서는 관리자 권한으로 PowerShell을 열고 다음 명령을 입력한 후 표시되는 메시지를 따릅니다.

  ```
  aws configure --profile AmazonCloudWatchAgent
  ```

(선택 사항) CloudWatch 에이전트에 대한 일반 구성 및 명명된 프로필 수정

CloudWatch 에이전트에는 common-config.toml 구성 파일이 있습니다. 필요한 경우 이 파일을 사용하여 프록시 및 리전 정보를 지정할 수 있습니다.

Linux를 실행하는 서버에서는 이 파일이 /opt/aws/amazon-cloudwatch-agent/etc 디렉터리에 있습니다. Windows Server를 실행하는 서버에서는 이 파일이 C:\ProgramData\Amazon\AmazonCloudWatchAgent 디렉터리에 있습니다.

기본 common-config.toml은 다음과 같습니다.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
```

```
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

처음에는 모든 줄이 코멘트 아웃 처리되어 있습니다. 자격 증명 프로필이나 프록시 설정을 설정하려면 해당 줄에서 #을 제거하고 값을 지정하십시오. 이 파일은 수동으로 편집하거나 시스템 관리자의 `RunShellScript` Run Command를 사용하여 편집할 수 있습니다.

- `shared_credential_profile` – 온프레미스 서버의 경우 이 줄은 CloudWatch에 데이터를 보내는 데 사용할 IAM 사용자 자격 증명 프로필을 지정합니다. 이 줄을 코멘트 아웃 처리된 상태로 유지할 경우 `AmazonCloudWatchAgent`가 사용됩니다. 이 프로필 생성에 대한 자세한 내용은 (온프레미스 서버에 설치) IAM 자격 증명 및 AWS 리전 지정 (p. 210) 단원을 참조하십시오.

  EC2 인스턴스에서 이 줄을 사용하여 CloudWatch 에이전트가 이 인스턴스의 데이터를 다른 AWS 리전의 CloudWatch에 보내도록 할 수 있습니다. 이렇게 하려면 보낼 리전의 이름을 지정하는 `region` 필드가 있는 명명된 프로필을 지정합니다.

  `shared_credential_profile`을 지정하는 경우 `[credentials]` 행의 시작 부분에서 #도 제거해야 합니다.
- `shared_credential_file` – 에이전트가 기본 경로 이외의 다른 경로에 있는 파일에서 자격 증명을 찾도록 하려면 여기에 전체 경로 및 파일 이름을 지정합니다. 기본 경로는 Linux의 경우 `/root/.aws`이며 Windows Server의 경우 `C:\\Users\\Administrator\\.aws`입니다.

  아래의 첫 번째 예는 Linux 서버의 유효한 `shared_credential_file` 행 구문이고 두 번째 예는 Windows Server에 유효한 행 구문입니다. Windows Server에서는 \ 문자를 이스케이프 처리해야 합니다.

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  `shared_credential_file`을 지정하는 경우 `[credentials]` 행의 시작 부분에서 #도 제거해야 합니다.
- 프록시 설정 – 서버가 HTTP 또는 HTTPS 프록시를 사용하여 AWS 서비스에 접근하도록 하려면 해당 프록시를 `http_proxy` 및 `https_proxy` 필드에 지정합니다. 프록시 설정에서 제외해야 하는 URL이 있다면 이를 쉼표로 구분하여 `no_proxy` 필드에 지정하십시오.

CloudWatch 에이전트 시작

시스템 관리자 Run Command 또는 명령줄을 사용하여 CloudWatch 에이전트를 시작할 수 있습니다.

SSM Agent를 사용하여 온프레미스 서버에서 CloudWatch 에이전트를 시작하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose Run Command.

   -or-

   If the AWS 시스템 관리자 home page opens, scroll down and choose Explore Run Command.
3. Run 명령을 선택합니다.
4. 명령 문서 목록에서 AmazonCloudWatch-ManageAgent 옆의 버튼을 선택합니다.
5. 대상 영역에서 에이전트를 설치한 인스턴스를 선택합니다.
6. 작업 목록에서 구성을 선택합니다.
7. 모드 목록에서 onPremise를 선택합니다.

8. Optional Configuration Location(구성 위치(선택 사항)) 상자에 마법사를 사용하여 생성하고 Parameter Store에 저장한 에이전트 구성 파일의 이름을 입력합니다.
9. 실행을 선택합니다.

   구성 파일에 지정한 구성을 사용하여 에이전트가 시작됩니다.

명령줄을 사용하여 CloudWatch 에이전트를 온프레미스 서버에서 시작하려면

- 이 명령에서 `-a fetch-config`는 에이전트가 최신 버전의 CloudWatch 에이전트 구성 파일을 로드하게 하며, `-s`는 에이전트를 시작합니다.

  Linux: 구성 파일을 시스템 관리자 Parameter Store에 저장한 경우 다음을 입력합니다.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
  m onPremise -c ssm:configuration-parameter-store-name -s
  ```

  Linux: 구성 파일을 로컬 컴퓨터에 저장한 경우 다음을 입력합니다.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
  m onPremise -c file:configuration-file-path -s
  ```

  Windows Server: 에이전트 구성 파일을 시스템 관리자 Parameter Store에 저장한 경우 PowerShell 콘솔에서 다음을 입력합니다.

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -a
   fetch-config -m onPremise -c ssm:configuration-parameter-store-name -s
  ```

  Windows Server: 에이전트 구성 파일을 로컬 컴퓨터에 저장한 경우 PowerShell 콘솔에서 다음을 입력합니다.

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -a
   fetch-config -m onPremise -c file:configuration-file-path -s
  ```

AWS CloudFormation을 사용하여 CloudWatch 에이전트 설치

CloudWatch 에이전트 설치 및 업데이트를 지원하기 위해 Amazon에서는 GitHub에 여러 AWS CloudFormation 템플릿을 업로드했습니다. AWS CloudFormation 사용에 대한 자세한 내용은 AWS CloudFormation란 무엇입니까?를 참조하십시오.

템플릿 위치는 AWS CloudFormation을 사용하여 Amazon CloudWatch 에이전트를 EC2 인스턴스에 배포입니다. 이 위치에는 `inline` 디렉터리와 `ssm` 디렉터리가 모두 포함되어 있습니다. 각 디렉터리에는 Linux 및 Windows 인스턴스 둘 다에 대한 템플릿이 포함되어 있습니다.

- `inline` 디렉터리의 템플릿에는 AWS CloudFormation 템플릿에 포함된 CloudWatch 에이전트 구성이 있습니다. 기본적으로 Linux 템플릿은 지표 `mem_used_percent` 및 `swap_used_percent`를 수집하고, Windows 템플릿은 `Memory % Committed Bytes In Use` 및 `Paging File % Usage`를 수집합니다.

  다른 지표를 수집하도록 이러한 템플릿을 수정하려면 템플릿의 다음 섹션을 수정하십시오. 다음 예제는 Linux 서버용 템플릿에서 가져옵니다. 에이전트 구성 파일의 형식 및 구문에 따라 다음과 같이 변경합니다. 자세한 내용은 CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 (p. 227) 단원을 참조하십시오.

```
{
   "metrics":{
      "append_dimensions":{
         "AutoScalingGroupName":"${!aws:AutoScalingGroupName}",
         "ImageId":"${!aws:ImageId}",
         "InstanceId":"${!aws:InstanceId}",
         "InstanceType":"${!aws:InstanceType}"
      },
      "metrics_collected":{
         "mem":{
            "measurement":[
               "mem_used_percent"
            ]
         },
         "swap":{
            "measurement":[
               "swap_used_percent"
            ]
         }
      }
   }
}
```

Note

인라인 템플릿에서 모든 자리 표시자 변수에는 이스케이프 문자로 앞에 느낌표(!)가 있어야 합니다. 이러한 내용은 예제 템플릿에서 확인할 수 있습니다. 다른 자리 표시자 변수를 추가한 경우 해당 변수의 이름 앞에 느낌표를 추가해야 합니다.

- `ssm` 디렉터리의 템플릿은 Parameter Store에서 에이전트 구성 파일을 로드합니다. 이러한 템플릿을 사용하려면 먼저 구성 파일을 생성한 다음 Parameter Store에 업로드해야 합니다. 그런 다음 해당 템플릿에 파일의 Parameter Store 이름을 입력합니다. 구성 파일은 수동으로 생성하거나 마법사를 사용해 생성할 수 있습니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.

CloudWatch 에이전트를 설치하고 에이전트 구성을 업데이트하는 데 두 가지 템플릿 유형을 모두 사용할 수 있습니다.

자습서: AWS CloudFormation 인라인 템플릿을 사용하여 CloudWatch 에이전트 설치 및 구성

이 자습서에서는 AWS CloudFormation을 사용하여 새 Amazon EC2 인스턴스에 CloudWatch 에이전트를 설치하는 과정을 안내합니다. 이 자습서에서는 인라인 템플릿을 사용하여 Amazon Linux 2를 실행하는 새 인스턴스에 설치하는데, JSON 구성 파일 또는 Parameter Store를 사용할 필요가 없습니다. 인라인 템플릿에는 에이전트 구성이 포함되어 있습니다. 이 자습서에서는 템플릿에 포함된 기본 에이전트 구성을 사용합니다.

에이전트를 설치하는 절차를 수행한 다음 자습서에서는 계속해서 에이전트를 업데이트하는 방법을 설명합니다.

새 인스턴스에서 AWS CloudFormation을 사용해 CloudWatch 에이전트를 설치하려면

1. GitHub에서 템플릿을 다운로드합니다. 이 자습서에서는 다음과 같이 Amazon Linux 2용 인라인 템플릿을 다운로드합니다.

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/master/aws/solutions/AmazonCloudWatchAgent/inline/amazon_linux.template
   ```

2. https://console.aws.amazon.com/cloudformation에서 AWS CloudFormation 콘솔을 엽니다.
3. 스택 생성을 선택합니다.

4. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 다운로드한 템플릿을 선택한 후 다음을 선택합니다.
5. 세부 정보 지정 페이지에 다음 파라미터를 입력하고 다음을 선택합니다.

   - 스택 이름: AWS CloudFormation 스택에 대한 스택 이름을 선택합니다.
   - IAMRole: CloudWatch 지표 및 로그를 쓸 권한이 있는 IAM 역할을 선택합니다. 자세한 내용은 Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 (p. 194) 단원을 참조하십시오.
   - InstanceAMI: 스택을 시작하려는 리전에서 유효한 AMI를 선택합니다.
   - InstanceType: 유효한 인스턴스 유형을 선택합니다.
   - KeyName: 새 인스턴스에 대한 SSH 액세스를 활성화하려면 기존 Amazon EC2 키 페어를 선택합니다. Amazon EC2 키 페어가 아직 없는 경우 AWS Management 콘솔에서 새로 생성할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 Amazon EC2 키 페어를 참조하십시오.
   - SSHLocation: SSH를 사용하여 인스턴스에 연결하는 데 사용할 수 있는 IP 주소 범위를 지정합니다. 기본값은 모든 IP 주소에서의 액세스를 허용합니다.
6. 옵션 페이지에서 스택 리소스에 태그를 지정할 수 있습니다. 다음을 선택합니다.
7. 검토 페이지에서 정보를 검토하고, 스택이 IAM 리소스를 생성할 수 있음을 인지한 다음 생성을 선택합니다.

   콘솔을 새로 고치면 새 스택에 `CREATE_IN_PROGRESS` 상태가 있음을 알 수 있습니다.
8. 인스턴스가 생성되면 Amazon EC2 콘솔에서 해당 인스턴스를 볼 수 있습니다. 필요한 경우 호스트에 연결해 진행 상황을 확인할 수 있습니다.

   다음 명령을 사용하여 에이전트가 설치되었는지 확인합니다.

```
rpm -qa amazon-cloudwatch-agent
```

   다음 명령을 사용하여 에이전트가 실행 중인지 확인합니다.

```
ps aux | grep amazon-cloudwatch-agent
```

다음 절차에서는 AWS CloudFormation에서 인라인 템플릿을 사용해 CloudWatch 에이전트를 업데이트하는 방법을 보여줍니다. 기본 인라인 템플릿은 `mem_used_percent` 지표를 수집합니다. 이 자습서에서는 이 지표 수집을 중지하도록 에이전트 구성을 변경합니다.

AWS CloudFormation을 사용해 CloudWatch 에이전트를 업데이트하려면

1. 이전 절차에서 다운로드한 템플릿에서 다음 줄을 제거하고 템플릿을 저장합니다.

```
"mem": {

    "measurement": [
        "mem_used_percent"
      ]
 },
```

2. https://console.aws.amazon.com/cloudformation에서 AWS CloudFormation 콘솔을 엽니다.
3. AWS CloudFormation 대시보드에서 생성된 스택을 선택하고 스택 업데이트를 선택합니다.
4. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 수정한 템플릿을 선택한 후 다음을 선택합니다.
5. 옵션 페이지에서 다음을 선택한 후 다음을 선택합니다.
6. 검토 페이지에서 정보를 검토하고 업데이트를 선택합니다.

잠시 후 UPDATE_COMPLETE가 표시됩니다.

자습서: AWS CloudFormation 및 Parameter Store를 사용하여 CloudWatch 에이전트 설치

이 자습서에서는 AWS CloudFormation을 사용하여 새 Amazon EC2 인스턴스에 CloudWatch 에이전트를 설치하는 과정을 안내합니다. 이 자습서에서는 생성해 Parameter Store에 저장한 에이전트 구성 파일을 사용하여 Amazon Linux 2를 실행 중인 새 인스턴스에 설치합니다.

에이전트를 설치하는 절차를 수행한 다음 자습서에서는 계속해서 에이전트를 업데이트하는 방법을 설명합니다.

AWS CloudFormation에서 Parameter Store의 구성을 사용해 새 인스턴스에서 CloudWatch 에이전트를 설치하려면

1. 아직 설치하지 않은 경우 에이전트 구성 파일을 생성할 수 있도록 컴퓨터 중 한 대에 CloudWatch 에이전트 패키지를 다운로드합니다. Parameter Store를 사용하여 에이전트를 다운로드하는 방법에 대한 자세한 내용은 CloudWatch 에이전트 다운로드 및 구성 (p. 205) 단원을 참조하십시오. 명령줄을 사용하여 패키지를 다운로드하는 방법에 대한 자세한 내용은 명령줄을 사용하여 CloudWatch 에이전트 다운로드 및 구성 (p. 190) 단원을 참조하십시오.
2. 에이전트 구성 파일을 생성하고 Parameter Store에 저장합니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.
3. 다음과 같이 GitHub에서 템플릿을 다운로드합니다.

```
curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/master/
aws/solutions/AmazonCloudWatchAgent/ssm/amazon_linux.template
```

4. https://console.aws.amazon.com/cloudformation에서 AWS CloudFormation 콘솔을 엽니다.
5. 스택 생성을 선택합니다.
6. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 다운로드한 템플릿을 선택한 후 다음을 선택합니다.
7. 세부 정보 지정 페이지에 다음 파라미터를 입력하고 다음을 선택합니다.

   - 스택 이름: AWS CloudFormation 스택에 대한 스택 이름을 선택합니다.
   - IAMRole: CloudWatch 지표 및 로그를 쓸 권한이 있는 IAM 역할을 선택합니다. 자세한 내용은 Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 (p. 202) 단원을 참조하십시오.
   - InstanceAMI: 스택을 시작하려는 리전에서 유효한 AMI를 선택합니다.
   - InstanceType: 유효한 인스턴스 유형을 선택합니다.
   - KeyName: 새 인스턴스에 대한 SSH 액세스를 활성화하려면 기존 Amazon EC2 키 페어를 선택합니다. Amazon EC2 키 페어가 아직 없는 경우 AWS Management 콘솔에서 새로 생성할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 Amazon EC2 키 페어를 참조하십시오.
   - SSHLocation: SSH를 사용하여 인스턴스에 연결하는 데 사용할 수 있는 IP 주소 범위를 지정합니다. 기본값은 모든 IP 주소에서의 액세스를 허용합니다.
   - SSMKey: 생성해 Parameter Store에 저장한 에이전트 구성 파일을 지정합니다.
8. 옵션 페이지에서 스택 리소스에 태그를 지정할 수 있습니다. 다음을 선택합니다.
9. 검토 페이지에서 정보를 검토하고, 스택이 IAM 리소스를 생성할 수 있음을 인지한 다음 생성을 선택합니다.

   콘솔을 새로 고치면 새 스택에 CREATE_IN_PROGRESS 상태가 있음을 알 수 있습니다.
10. 인스턴스가 생성되면 Amazon EC2 콘솔에서 해당 인스턴스를 볼 수 있습니다. 필요한 경우 호스트에 연결해 진행 상황을 확인할 수 있습니다.

다음 명령을 사용하여 에이전트가 설치되었는지 확인합니다.

```
rpm -qa amazon-cloudwatch-agent
```

다음 명령을 사용하여 에이전트가 실행 중인지 확인합니다.

```
ps aux | grep amazon-cloudwatch-agent
```

다음 절차에서는 AWS CloudFormation에서 Parameter Store에 저장한 에이전트 구성을 사용해 CloudWatch 에이전트를 업데이트하는 방법을 보여줍니다.

AWS CloudFormation에서 Parameter Store의 구성을 사용해 CloudWatch 에이전트를 업데이트하려면

1. Parameter Store에 저장한 에이전트 구성 파일을 원하는 새 구성으로 변경합니다.
2. the section called “자습서: AWS CloudFormation 및 Parameter Store를 사용하여 CloudWatch 에이전트 설치” (p. 216) 주제에서 다운로드한 AWS CloudFormation 템플릿에서 버전 번호를 변경합니다. 예를 들어, `VERSION=1.0`을 `VERSION=2.0`으로 변경할 수 있습니다.
3. https://console.aws.amazon.com/cloudformation에서 AWS CloudFormation 콘솔을 엽니다.
4. AWS CloudFormation 대시보드에서 생성된 스택을 선택하고 스택 업데이트를 선택합니다.
5. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 방금 수정한 템플릿을 선택한 후 다음을 선택합니다.
6. 옵션 페이지에서 다음을 선택한 후 다음을 선택합니다.
7. 검토 페이지에서 정보를 검토하고 업데이트를 선택합니다.

   잠시 후 `UPDATE_COMPLETE`가 표시됩니다.

AWS CloudFormation에서 CloudWatch 에이전트 설치 문제 해결

이 단원에서는 AWS CloudFormation을 사용하여 CloudWatch 에이전트 설치 및 업데이트와 관련된 문제를 해결할 수 있도록 지원합니다.

업데이트에 실패 감지

AWS CloudFormation을 사용하여 CloudWatch 에이전트 구성을 업데이트하는데, 잘못된 구성을 사용한 경우 에이전트는 CloudWatch로 지표 전송을 중지합니다. `cfn-init-cmd.log` 파일을 살펴보면 에이전트 구성 업데이트에 성공했는지 여부를 빠르게 확인할 수 있습니다. Linux 서버에서 이 파일은 `/var/log/cfn-init-cmd.log`에 있습니다. Windows 인스턴스에서 이 파일은 `C:\cfn\log\cfn-init-cmd.log`에 있습니다.

지표가 누락됨

에이전트를 설치 또는 업데이트한 후 원하는 지표가 표시되지 않는 경우 에이전트가 해당 지표를 수집하도록 구성되어 있는지 확인하십시오. 이렇게 하려면 `amazon-cloudwatch-agent.json` 파일을 점검하여 해당 지표가 나열되어 있는지 확인하고 지금 살펴보고 있는 지표 네임스페이스가 올바른 네임스페이스인지를 확인하십시오. 자세한 내용은 CloudWatch 에이전트 파일 및 위치 (p. 272) 단원을 참조하십시오.

CloudWatch 에이전트 패키지의 서명 확인

Linux 서버의 CloudWatch 에이전트 패키지에 대해 GPG 서명 파일이 포함됩니다. 퍼블릭 키를 사용하여 에이전트 다운로드 파일이 원본이며 수정되지 않았는지 확인할 수 있습니다.

Windows Server의 경우 MSI를 사용하여 서명을 확인할 수 있습니다.

올바른 서명 파일을 찾으려면 다음 표를 참조하십시오. 각 아키텍처 및 운영 체제에는 일반 링크와 각 리전별 링크가 있습니다. 예를 들어 Amazon Linux, Amazon Linux 2 및 AMD64 아키텍처의 경우 세 개의 유효한 링크가 있습니다.

- `https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`
- `https://s3.us-east-1.amazonaws.com/amazoncloudwatch-agent-us-east-1/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`
- `https://s3.eu-central-1.amazonaws.com/amazoncloudwatch-agent-eu-central-1/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
AMD64	Amazon Linux 및 Amazon Linux 2	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Centos	https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/centos/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Redhat	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	SUSE	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
		amazoncloudwatch-agent-*region*/suse/amd64/latest/amazon-cloudwatch-agent.rpm	amazoncloudwatch-agent-*region*/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig
AMD64	Debian	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig
AMD64	Ubuntu	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
AMD64	Windows	https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig
ARM64	Amazon Linux 2	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
ARM64	Redhat	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Ubuntu	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://s3.amazonaws.com/amazoncloudwatch-agent/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://s3.*region*.amazonaws.com/amazoncloudwatch-agent-*region*suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Linux 서버에서 CloudWatch 에이전트 패키지를 확인하려면

1. 퍼블릭 키를 다운로드합니다.

```
shell$ wget https://s3.amazonaws.com/amazoncloudwatch-agent/assets/amazon-cloudwatch-agent.gpg
```

2. 퍼블릭 키를 인증 키로 가져옵니다.

```
shell$  gpg --import amazon-cloudwatch-agent.gpg
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

   다음 단계에서 필요하므로 키 값을 적어 둡니다. 이전 예제에서 키 값은 `3B789C72`입니다.

3. *#-#*을 이전 단계의 값으로 대체하고 다음 명령을 실행하여 지문을 확인합니다.

```
shell$  gpg --fingerprint key-value
```

```
pub   2048R/3B789C72 2017-11-14
      Key fingerprint = 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid                  Amazon CloudWatch Agent
```

지문 문자열이 다음과 동일해야 합니다.

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

지문 문자열이 일치하지 않으면 에이전트를 설치하지 말고 Amazon Web Services에 문의하십시오.

지문을 확인한 이후 이를 사용하여 CloudWatch 에이전트 패키지의 서명을 확인할 수 있습니다.

4. wget을 사용하여 패키지 서명 파일을 다운로드합니다. 올바른 서명 파일을 확인하려면 이전 표를 참조하십시오.

```
wget Signature File Link
```

5. 서명을 확인하려면 gpg --verify를 실행합니다.

```
shell$ gpg --verify signature-filename agent-download-filename
gpg: Signature made Wed 29 Nov 2017 03:00:59 PM PST using RSA key ID 3B789C72
gpg: Good signature from "Amazon CloudWatch Agent"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

출력에 `BAD signature` 문구가 포함된 경우 절차를 올바르게 수행했는지 확인합니다. 이 응답이 계속되는 경우, Amazon Web Services에 문의하고, 다운로드한 파일을 사용하지 마십시오.

신뢰에 대한 경고를 참조하십시오. 사용자 또는 사용자가 신뢰하는 사람이 서명한 키만 신뢰됩니다. 이는 서명이 잘못되었음을 의미하지 않으며, 단지 해당 사용자가 퍼블릭 키를 확인하지 않은 것입니다.

Windows Server를 실행하는 서버에서 CloudWatch 에이전트 패키지를 확인하려면

1. https://gnupg.org/download/에서 Windows용 GnuPG를 다운로드 및 설치합니다. 설치할 때 Shell Extension (GpgEx)(셸 확장(GpgEx)) 옵션을 포함합니다.

   나머지 단계는 Windows PowerShell에서 수행할 수 있습니다.

2. 퍼블릭 키를 다운로드합니다.

```
PS> wget https://s3.amazonaws.com/amazoncloudwatch-agent/assets/amazon-cloudwatch-
agent.gpg -OutFile amazon-cloudwatch-agent.gpg
```

3. 퍼블릭 키를 인증 키로 가져옵니다.

```
PS>  gpg --import amazon-cloudwatch-agent.gpg
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

다음 단계에서 필요하므로 키 값을 기록해 둡니다. 이전 예제에서 키 값은 `3B789C72`입니다.

4. *#-#*을 이전 단계의 값으로 대체하고 다음 명령을 실행하여 지문을 확인합니다.

```
PS>  gpg --fingerprint key-value
pub   rsa2048 2017-11-14 [SC]
      9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid           [ unknown] Amazon CloudWatch Agent
```

지문 문자열이 다음과 동일해야 합니다.

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

지문 문자열이 일치하지 않으면 에이전트를 설치하지 말고 Amazon Web Services에 문의하십시오.

지문을 확인한 이후 이를 사용하여 CloudWatch 에이전트 패키지의 서명을 확인할 수 있습니다.

5. wget을 사용하여 패키지 서명 파일을 다운로드합니다. 정확한 서명 파일을 확인하려면 CloudWatch 에이전트 다운로드 링크 (p. 191) 단원을 참조하십시오.
6. 서명을 확인하려면 gpg --verify를 실행합니다.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

출력에 `BAD signature` 문구가 포함된 경우 절차를 올바르게 수행했는지 확인합니다. 이 응답이 계속되는 경우, Amazon Web Services에 문의하고, 다운로드한 파일을 사용하지 마십시오.

신뢰에 대한 경고를 참조하십시오. 사용자 또는 사용자가 신뢰하는 사람이 서명한 키만 신뢰됩니다. 이는 서명이 잘못되었음을 의미하지 않으며, 단지 해당 사용자가 퍼블릭 키를 확인하지 않은 것입니다.

CloudWatch 에이전트 구성 파일 생성

서버에서 CloudWatch 에이전트를 실행하기 전에 CloudWatch 에이전트 구성 파일을 생성해야 합니다.

에이전트 구성 파일은 사용자 지정 지표를 포함하여 에이전트가 수집해야 하는 지표 및 로그가 지정되어 있는 JSON 파일입니다. 마법사를 사용하거나 처음부터 직접 생성하여 이 구성 파일을 생성할 수 있습니다. 또한 마법사를 사용하여 구성 파일을 처음으로 만든 다음, 수동으로 수정할 수 있습니다. 수동으로 파일을 생성하거나 수정하는 경우 프로세스는 더 복잡하지만, 수집된 지표를 더 잘 제어할 수 있으며 마법사를 통해 사용할 수 없는 지표를 지정할 수 있습니다.

에이전트 구성 파일을 변경할 때마다 에이전트를 다시 시작하여 변경 사항이 적용되도록 해야 합니다.

구성 파일을 만든 후 수동으로 JSON 파일로 저장하여 서버에 에이전트를 설치할 때 이 파일을 사용할 수 있습니다. 또는 서버에 에이전트를 설치할 때 시스템 관리자를 사용하려는 경우 해당 파일을 시스템 관리자 Parameter Store에 저장할 수 있습니다.

목차

- 마법사로 CloudWatch 에이전트 구성 파일 만들기 (p. 222)
- CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 (p. 227)

마법사로 CloudWatch 에이전트 구성 파일 만들기

에이전트 구성 파일 마법사 `amazon-cloudwatch-agent-config-wizard`는 다음을 포함하여 일련의 질문을 던집니다.

- 에이전트를 Amazon EC2 인스턴스에 설치할 것입니까 아니면 온프레미스 서버에 설치할 것입니까?
- 서버가 Linux를 실행합니까 아니면 Windows Server를 실행합니까?

- 에이전트가 CloudWatch Logs에도 로그 파일을 보내도록 하시겠습니까? 보내는 경우 기존 CloudWatch Logs 에이전트 구성 파일을 가지고 있습니까? 예인 경우 CloudWatch 에이전트는 이 파일을 사용하여 서버에서 수집할 로그를 결정할 수 있습니다.
- 서버에서 지표를 수집하려는 경우 기본 지표 집합 중 하나를 모니터링하시겠습니까? 아니면 수집할 지표 목록을 사용자 지정하시겠습니까?
- `StatsD` 또는 `collectd`를 사용하여 애플리케이션 또는 서비스에서 사용자 지정 지표를 수집하시겠습니까?
- 기존 SSM Agent로부터 마이그레이션하는 중입니까?

마법사를 시작하기 전에 AWS 자격 증명 및 구성 파일이 제자리에 있는 경우, 마법사가 사용할 자격 증명 및 AWS 리전을 자동 검색할 수 있습니다. 이러한 파일에 대한 자세한 내용은 AWS 시스템 관리자 사용 설명서의 구성 및 자격 증명 파일을 참조하십시오.

AWS 자격 증명 파일에서 마법사는 기본 자격 증명을 확인하고 다음과 같은 `AmazonCloudWatchAgent` 섹션도 찾습니다.

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

마법사는 기본 자격 증명, `AmazonCloudWatchAgent`의 자격 증명 및 `Others` 옵션을 표시합니다. 사용할 자격 증명을 선택할 수 있습니다. `Others`를 선택할 경우 자격 증명을 입력할 수 있습니다.

my_access_key 및 *my_secret_key*의 경우 시스템 관리자 Parameter Store에 대한 쓰기 권한이 있는 IAM 사용자의 키를 사용합니다. CloudWatch 에이전트에 필요한 IAM 사용자에 대한 자세한 내용은 온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성 (p. 203) 단원을 참조하십시오.

AWS 구성 파일에서 에이전트가 지표를 보낼 리전을 지정할 수 있습니다(해당 리전이 `[default]` 섹션의 리전과 다른 경우). 기본값은 Amazon EC2 인스턴스가 위치하는 리전에 지표를 게시하는 것입니다. 지표를 다른 리전에 게시해야 하는 여기에서 리전을 지정합니다. 다음 예에서는 지표가 `us-west-1` 리전에 게시되어 있습니다.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

CloudWatch 에이전트 사전 정의된 지표 집합

마법사는 상세 레벨이 다양한 사전 정의된 지표 집합을 사용하여 구성되어 있습니다. 이러한 지표 집합은 다음 표에 표시되어 있습니다. 지표에 대한 자세한 내용은 CloudWatch 에이전트가 수집하는 지표 (p. 255) 섹션을 참조하십시오.

Linux를 실행하는 Amazon EC2 인스턴스

상세 수준	포함된 지표
기본	Mem: mem_used_percent Disk: disk_used_percent `disk_used_percent`와 같은 `disk` 지표에는 `Partition`의 차원이 있는데, 이는 생성된 사용자 지정 지표의 수가 인스턴스와 연결된 파티션의 수에 따라 달라짐을 뜻합니다. 디스크 파티션의 수는 사용 중인 AMI와 서버에 연결하는 Amazon EBS 볼륨의 수에 따라 달라집니다.

상세 수준	포함된 지표
표준	CPU: `cpu_usage_idle`, `cpu_usage_iowait`, `cpu_usage_user`, `cpu_usage_system` Disk: `disk_used_percent`, `disk_inodes_free` Diskio: `diskio_io_time` Mem: `mem_used_percent` Swap: `swap_used_percent`
Advanced	CPU: `cpu_usage_idle`, `cpu_usage_iowait`, `cpu_usage_user`, `cpu_usage_system` Disk: `disk_used_percent`, `disk_inodes_free` Diskio: `diskio_io_time`, `diskio_write_bytes`, `diskio_read_bytes`, `diskio_writes`, `diskio_reads` Mem: `mem_used_percent` Netstat: `netstat_tcp_established`, `netstat_tcp_time_wait` Swap: `swap_used_percent`

Linux를 실행하는 온프레미스 서버

상세 수준	포함된 지표
기본	Disk: `disk_used_percent` Diskio: `diskio_write_bytes`, `diskio_read_bytes`, `diskio_writes`, `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent`, `net_bytes_recv`, `net_packets_sent`, `net_packets_recv` Swap: `swap_used_percent`
표준	CPU: `cpu_usage_idle`, `cpu_usage_iowait` Disk: `disk_used_percent`, `disk_inodes_free` Diskio: `diskio_io_time`, `diskio_write_bytes`, `diskio_read_bytes`, `diskio_writes`, `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent`, `net_bytes_recv`, `net_packets_sent`, `net_packets_recv` Swap: `swap_used_percent`
Advanced	CPU: `cpu_usage_guest`, `cpu_usage_idle`, `cpu_usage_iowait`, `cpu_usage_steal`, `cpu_usage_user`, `cpu_usage_system`

상세 수준	포함된 지표
	Disk: `disk_used_percent`, `disk_inodes_free` Diskio: `diskio_io_time`, `diskio_write_bytes`, `diskio_read_bytes`, `diskio_writes`, `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent`, `net_bytes_recv`, `net_packets_sent`, `net_packets_recv` Netstat: `netstat_tcp_established`, `netstat_tcp_time_wait` Swap: `swap_used_percent`

Windows Server를 실행하는 Amazon EC2 인스턴스

상세 수준	포함된 지표
기본	Memory: `Memory % Committed Bytes In Use` LogicalDisk: `LogicalDisk % Free Space`
표준	Memory: `Memory % Committed Bytes In Use` Paging: `Paging File % Usage` 프로세서: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` PhysicalDisk: `PhysicalDisk % Disk Time` LogicalDisk: `LogicalDisk % Free Space`
Advanced	Memory: `Memory % Committed Bytes In Use` Paging: `Paging File % Usage` 프로세서: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time`, `PhysicalDisk Disk Write Bytes/sec`, `PhysicalDisk Disk Read Bytes/sec`, `PhysicalDisk Disk Writes/sec`, `PhysicalDisk Disk Reads/sec` TCP: `TCPv4 Connections Established`, `TCPv6 Connections Established`

Windows Server를 실행하는 온프레미스 서버

상세 수준	포함된 지표
기본	페이징: `Paging File % Usage` 프로세서: `Processor % Processor Time`

상세 수준	포함된 지표
	LogicalDisk:`LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk Disk Write Bytes/sec`, `PhysicalDisk Disk Read Bytes/sec`, `PhysicalDisk Disk Writes/sec`, `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` 네트워크 인터페이스: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec`, `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
표준	Paging: `Paging File % Usage` 프로세서: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time`, `PhysicalDisk Disk Write Bytes/sec`, `PhysicalDisk Disk Read Bytes/sec`, `PhysicalDisk Disk Writes/sec`, `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` 네트워크 인터페이스: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec`, `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Advanced	페이징:`Paging File % Usage` 프로세서: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time`, `PhysicalDisk Disk Write Bytes/sec`, `PhysicalDisk Disk Read Bytes/sec`, `PhysicalDisk Disk Writes/sec`, `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` 네트워크 인터페이스: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec`, `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec` TCP: `TCPv4 Connections Established`, `TCPv6 Connections Established`

CloudWatch 에이전트 구성 마법사 실행

CloudWatch 에이전트 구성 파일을 생성하려면

1. 다음을 입력하여 CloudWatch 에이전트 구성 마법사를 시작합니다.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

Windows Server가 실행되는 서버에서 다음을 입력합니다.

```
cd "C:\Program Files\Amazon\AmazonCloudWatchAgent"
amazon-cloudwatch-agent-config-wizard.exe
```

2. 질문에 답하여 서버의 구성 파일을 사용자 지정합니다.
3. 구성 파일을 로컬에 저장하는 경우 구성 파일 `config.json`은 Linux 서버에서는 `/opt/aws/amazon-cloudwatch-agent/bin/`에, Windows 서버에서는 `C:\Program Files\Amazon\AmazonCloudWatchAgent`에 저장됩니다. 그러면 이 파일을 에이전트를 설치하려고 하는 다른 서버에 복사할 수 있습니다.

   시스템 관리자를 사용하여 에이전트를 설치 및 구성하려는 경우 파일을 시스템 관리자 Parameter Store에 저장할지 여부를 묻는 메시지가 표시되면 예로 답해야 합니다. CloudWatch 에이전트를 설치하는 데 SSM Agent를 사용하고 있지 않더라도 파일을 Parameter Store에 저장하도록 선택할 수도 있습니다. 파일을 Parameter Store에 저장할 수 있도록 하려면 충분한 권한이 있는 IAM 역할을 사용해야 합니다. 자세한 내용은 CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 (p. 201) 단원을 참조하십시오.

CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집

CloudWatch 에이전트 구성 파일은 `agent`, `metrics` 및 `logs`의 3개 섹션으로 구성된 JSON 파일입니다.

- `agent` 섹션에는 에이전트의 전체 구성에 대한 필드가 포함되어 있습니다. 마법사를 사용하는 경우 `agent` 섹션이 생성되지 않습니다.
- `metrics` 섹션은 수집하여 CloudWatch에 게시할 사용자 지정 지표를 지정합니다. 에이전트를 사용하여 로그만 수집하는 경우 파일에서 `metrics` 섹션을 생략할 수 있습니다.
- `logs` 섹션은 CloudWatch Logs에 게시되는 로그 파일을 지정합니다. 서버에서 Windows Server가 실행되는 경우 Windows 이벤트 로그에 따른 이벤트가 포함될 수 있습니다.

다음 섹션에서는 이 JSON 파일의 구조 및 필드에 대해 설명합니다. 이 구성 파일에 대한 스키마 정의도 볼 수 있습니다. 스키마 정의는 Linux 서버의 *`installation-directory`*`/doc/amazon-cloudwatch-agent-schema.json`, Windows Server를 실행하는 서버의 *`installation-directory`*`/amazon-cloudwatch-agent-schema.json`에 있습니다.

에이전트 구성 파일을 수동으로 생성하거나 편집할 경우 이름을 지정할 수 있습니다. 문제를 간단하게 해결할 수 있도록 이름을 Linux 서버에서는 `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json`로 지정하고, Windows 서버를 실행하는 서버에서는 `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`로 지정하는 것이 좋습니다. 파일을 생성한 후에는 해당 파일을 에이전트를 설치하려고 하는 다른 서버에 복사할 수 있습니다.

CloudWatch 에이전트 구성 파일: 에이전트 섹션

`agent` 섹션은 다음 필드를 포함할 수 있습니다. 마법사는 `agent` 섹션을 생성하지 않습니다. 대신, 이를 생략하고 이 섹션의 모든 필드에 대해 기본값을 사용합니다.

- `metrics_collection_interval` – 선택 사항. 이 구성 파일에 지정되어 있는 모든 지표가 수집될 빈도를 지정합니다. 특정 유형의 지표에 대해 이 값을 재정의할 수 있습니다.

  이 값은 초 단위로 지정됩니다. 예를 들어, 10을 지정하면 10초마다 지표가 수집되도록 설정되며, 300으로 설정하면 5분마다 지표가 수집되도록 지정됩니다.

  이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.

기본값은 60입니다.

- `region` – Amazon EC2 인스턴스를 모니터링할 때 CloudWatch 엔드포인트에 대해 사용할 리전을 지정합니다. 수집되는 지표는 이 리전(예: `us-west-1`)에 전송됩니다. 이 필드를 생략하면 에이전트가 지표를 Amazon EC2 인스턴스가 있는 리전으로 전송합니다.

  온프레미스 서버를 모니터링하는 경우 이 필드가 사용되지 않으며, 에이전트는 AWS 구성 파일의 `AmazonCloudWatchAgent` 프로필에서 리전을 읽습니다.
- `credentials` – 지표 및 로그를 다른 AWS 계정에 보낼 때 사용할 IAM 역할을 지정합니다. 지정된 경우 이 필드는 1개의 파라미터 `role_arn`를 포함합니다.
  - `role_arn` – 지표 및 로그를 다른 AWS 계정에 보낼 때 인증에 사용할 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다. 자세한 내용은 다른 계정에 지표 및 로그 전송 (p. 268) 단원을 참조하십시오.
- `debug` – 선택 사항. 디버그 로그 메시지와 함께 CloudWatch 에이전트를 실행하도록 지정합니다. 기본값은 `false`입니다.
- `logfile` – CloudWatch 에이전트가 로그 메시지를 쓸 위치를 지정합니다. 비어 있는 문자열을 지정하면 로그가 stderr에 저장됩니다. 이 옵션을 지정하지 않으면 다음과 같이 기본 위치가 사용됩니다.
  - Linux: `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`
  - Windows Server: `c:\\ProgramData\\Amazon\\CloudWatchAgent\\Logs\\amazon-cloudwatch-agent.log`

  CloudWatch 에이전트는 생성한 로그 파일을 자동으로 교체합니다. 로그 파일의 크기가 100MB에 도달하면 교체됩니다. 에이전트는 교체된 로그 파일을 최대 7일 동안 보존하며 교체된 백업 로그 파일을 최대 5개까지 보존합니다. 백업 로그 파일의 파일 이름에는 타임스탬프가 추가됩니다. 이 타임스탬프는 파일이 교체된 날짜와 시간을 표시합니다(예: `amazon-cloudwatch-agent-2018-06-08T21-01-50.247.log.gz`).
- `omit_hostname` – 선택 사항. 기본적으로 호스트 이름은 에이전트에서 수집한 지표의 차원으로 게시됩니다. 호스트 이름이 차원으로 게시되지 않도록 하려면 이 값을 `true`로 설정합니다. 기본값은 `false`입니다.
- `run_as_user` – 선택 사항. CloudWatch 에이전트를 실행하는 데 사용할 사용자를 지정합니다. 이 파라미터를 지정하지 않으면 루트 사용자가 사용됩니다. 이 옵션은 Linux 서버에서만 유효합니다.

  이 옵션을 지정하면 CloudWatch 에이전트를 시작하기 전에 사용자가 존재해야 합니다. 자세한 내용은 다른 사용자로 CloudWatch 에이전트 실행 (p. 263) 단원을 참조하십시오.

다음은 `agent` 섹션의 예입니다.

```
"agent": {
   "metrics_collection_interval": 60,
   "region": "us-west-1",
   "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log",
   "debug": false,
   "run_as_user": "cwagent"
  }
```

CloudWatch 에이전트 구성 파일: 지표 섹션

Linux 또는 Windows Server가 실행되는 서버의 경우 `metrics` 섹션에 다음 필드가 있습니다.

- `namespace` – 선택 사항. 에이전트에 의해 수집되는 지표에 대해 사용할 네임스페이스입니다. 기본값은 `CWAgent`입니다. 최대 길이는 255자입니다.
- `append_dimensions` – 선택 사항. 에이전트에 의해 수집되는 모든 지표에 Amazon EC2 지표 차원을 추가합니다. 다음 목록은 지원되는 키-값 페어만 보여줍니다. 다른 키-값 페어는 무시됩니다.
  - `"ImageID":"${aws:ImageId}"`는 인스턴스의 AMI ID를 `ImageID` 차원의 값으로 설정합니다.

  - `"InstanceId":"${aws:InstanceId}"`는 인스턴스의 인스턴스 ID를 `InstanceID` 차원의 값으로 설정합니다.
  - `"InstanceType":"${aws:InstanceType}"`는 인스턴스의 인스턴스 유형을 `InstanceType` 차원의 값으로 설정합니다.
  - `"AutoScalingGroupName":"${aws:AutoScalingGroupName}"`는 인스턴스의 Auto Scaling 그룹 이름을 `AutoScalingGroupName` 차원의 값으로 설정합니다.

  임의 키-값 페어를 사용하여 지표에 차원을 추가하려면 특정 지표 유형에 대해 필드의 `append_dimensions` 파라미터를 사용해야 합니다.

  Amazon EC2 메타데이터에 따라 좌우되는 값을 지정하고 프록시를 사용할 경우 서버에서 Amazon EC2의 엔드포인트에 액세스할 수 있어야 합니다. 이러한 엔드포인트에 대한 자세한 내용은 Amazon Web Services 일반 참조의 Amazon Elastic Compute Cloud(Amazon EC2)를 참조하십시오.
- `aggregation_dimensions` – 선택 사항. 수집된 지표가 집계될 차원을 지정합니다. 예를 들어, `AutoScalingGroupName` 차원에서 지표를 롤업하는 경우, 각 Auto Scaling 그룹의 모든 인스턴스의 지표가 집계되고 전체로 표시될 수 있습니다.

  하나 또는 여러 차원에 따라 지표를 롤업할 수 있습니다. 예를 들어, `[["InstanceId"], ["InstanceType"], ["InstanceId","InstanceType"]]`을 지정하면 인스턴스 ID 단일, 인스턴스 유형 단일 및 두 차원의 조합에 대해 지표를 집계할 수 있습니다.

  `[]`를 지정하면 모든 차원을 무시하고 모든 지표를 하나의 모음에 롤업할 수도 있습니다.
- `endpoint_override` – 에이전트가 지표를 보내는 엔드포인트로 사용할 FIPS 엔드포인트 또는 프라이빗 링크를 지정합니다. 이 옵션을 지정하고 프라이빗 연결을 설정하면 Amazon VPC 엔드포인트로 지표를 전송할 수 있습니다. 자세한 내용은 Amazon VPC이란 무엇입니까?단원을 참조하십시오.

  `endpoint_override`의 값은 URL인 문자열이어야 합니다.
- `metrics_collected` – 필수 사항. `StatsD` 또는 `collectd`를 통해 수집된 사용자 지정 지표를 포함하여 수집할 지표를 지정합니다. 이 단원에는 여러 하위 섹션이 포함되어 있습니다.

  `metrics_collected` 섹션의 내용은 이 구성 파일이 Linux를 실행하는 서버용인지 아니면 Windows Server를 실행하는 서버용인지에 따라 달라집니다.
- `force_flush_interval` – 서버로 전송되기 전에 지표가 메모리 버퍼에 남아 있을 최대 시간(초)을 지정합니다. 이 설정과 상관없이 버퍼 내 지표의 크기가 40KB에 도달하거나 지표 개수가 20개가 되면 지표가 즉시 서버로 전송됩니다.

  기본값은 60입니다.
- `credentials` – 지표를 다른 계정에 보낼 때 사용할 IAM 역할을 지정합니다. 지정된 경우 이 필드는 1개의 파라미터 `role_arn`를 포함합니다.
  - `role_arn` – 지표를 다른 계정에 보낼 때 인증에 사용할 IAM 역할의 ARN을 지정합니다. 자세한 내용은 다른 계정에 지표 및 로그 전송 (p. 268) 단원을 참조하십시오. 여기에 지정한 경우 이 값은 구성 파일의 `agent` 섹션에 지정되어 있는 `role_arn`을 재정의합니다(있는 경우).

Linux

Linux가 실행되는 서버에서는 구성 파일의 metrics_collected 섹션에 다음 필드도 포함될 수 있습니다.

이러한 필드 중 다수에는 해당 리소스에 대해 수집하고 싶은 지표나 나열된 `measurement` 섹션이 포함되어 있습니다. 이러한 `measurement` 섹션은 `swap_used` 같이 완전한 척도 이름을 지정하거나, 리소스 유형에 추가되는 척도 이름의 일부만 지정할 수 있습니다. 예를 들어 `diskio` 섹션의 `measurement` 섹션에 `reads`를 지정하면 `diskio_reads` 지표가 수집될 수 있습니다.

- `collectd` – 선택 사항. `collectd` 프로토콜을 사용하여 사용자 지정 지표를 검색하려 한다는 것을 지정합니다. `collectd` 소프트웨어를 사용하여 CloudWatch 에이전트에 지표를 보냅니다. 자세한 내용은 collectd로 사용자 지정 지표 검색 (p. 254) 단원을 참조하십시오.

- `cpu` – 선택 사항. CPU 지표가 수집되도록 지정합니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 수집할 CPU 지표에 대해 `resources` 및 `totalcpu` 필드 중 하나 이상을 포함해야 합니다. 이 섹션은 다음 필드를 포함할 수 있습니다.
  - `resources` – 선택 사항. CPU당 지표가 수집되도록 하려면 이 필드의 값을 `*`로 지정해야 합니다. 허용되는 유일한 값은 `*`입니다.
  - `totalcpu` – 선택 사항. 모든 CPU 코어에서 집계되는 CPU 지표를 보고할지 여부를 지정합니다. 기본값은 true입니다.
  - `measurement` – 수집될 CPU 지표의 어레이를 지정합니다. 가능한 값은 `time_active`, `time_guest`, `time_guest_nice`, `time_idle`, `time_iowait`, `time_irq`, `time_nice`, `time_softirq`, `time_steal`, `time_system`, `time_user`, `usage_active`, `usage_guest`, `usage_guest_nice`, `usage_idle`, `usage_iowait`, `usage_irq`, `usage_nice`, `usage_softirq`, `usage_steal`, `usage_system`, `usage_user`입니다. `cpu`를 포함하는 경우 이 필드는 필수입니다.

    기본적으로 `cpu_usage_*` 지표의 단위는 `Percent`이며, `cpu_time_*` 지표에는 단위가 없습니다.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.
    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `metrics_collection_interval` – 선택 사항. CPU 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

    이 값은 초 단위로 지정됩니다. 예를 들어, 10을 지정하면 10초마다 지표가 수집되도록 설정되며, 300으로 설정하면 5분마다 지표가 수집되도록 지정됩니다.

    이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.
  - `append_dimensions` – 선택 사항. CPU 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트가 수집하는 모든 유형의 지표에 대해 사용되는 글로벌 `append_dimensions` 필드에 지정되어 있는 차원 외에 이 차원도 사용됩니다.
- `disk` – 선택 사항. 디스크 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 2개 필드만큼만 포함할 수 있습니다.
  - `resources` – 선택 사항. 디스크 탑재 지점에 대한 배열을 지정합니다. 이 필드는 CloudWatch가 나열된 탑재 지점에서만 지표를 수집하도록 제한합니다. 값으로 `*`를 지정하면 모든 탑재 지점에서 지표를 수집할 수 있습니다. 기본값은 모든 탑재 지점에서 지표를 수집하는 것입니다.
  - `measurement` – 수집될 디스크 지표의 어레이를 지정합니다. 가능한 값은 `free`, `total`, `used`, `used_percent`, `inodes_free`, `inodes_used`, `inodes_total`입니다. `disk`를 포함하는 경우 이 필드는 필수입니다.

    Note

    `disk` 지표에는 `Partition`의 차원이 있는데, 이는 생성된 사용자 지정 지표의 수가 인스턴스와 연결된 파티션의 수에 따라 달라짐을 뜻합니다. 디스크 파티션의 수는 사용 중인 AMI와 서버에 연결하는 Amazon EBS 볼륨의 수에 따라 달라집니다.

    각 `disk` 지표에 대한 기본 단위를 보려면 Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 (p. 255) 섹션을 참조하십시오.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.
    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `ignore_file_system_types` – 디스크 지표를 수집할 때 제외할 파일 시스템 유형을 지정합니다. 유효한 값으로는 `sysfs`, `devtmpfs` 등이 있습니다.
  - `drop_device` – 이 값을 `true`로 설정하면 `Device`가 디스크 지표의 차원으로 포함되지 않습니다.

`Device`가 차원으로 사용되지 않도록 하면 Nitro 시스템을 사용하는 인스턴스에서 인스턴스가 재부팅될 때 각 디스크 마운트마다 디바이스 이름이 변경되기 때문에 유용합니다. 이 경우 지표의 데이터가 일관되지 않으므로 이러한 지표를 기반으로 하는 경보가 `INSUFFICIENT DATA` 상태가 될 수 있습니다.

기본값은 `false`입니다.

- `metrics_collection_interval` – 선택 사항. 디스크 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

  이 값은 초 단위로 지정됩니다.

  이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.

- `append_dimensions` – 선택 사항. 디스크 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트에 의해 수집되는 모든 유형의 지표에 대해 사용되는 `append_dimensions` 필드에 지정되어 있는 차원 외에 지정한 차원도 사용됩니다.

- `diskio` – 선택 사항. 디스크 I/O 지표가 수집되도록 지정합니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 2개 필드만큼만 포함할 수 있습니다.

  - `resources` – 선택 사항. 디바이스 어레이를 지정하는 경우, CloudWatch가 해당 디바이스에서만 지표를 수집합니다. 그렇지 않으면, 모든 디바이스의 지표가 수집됩니다. 값으로 `*`를 지정하여 모든 디바이스에서 지표를 수집할 수도 있습니다.
  - `measurement` – 수집될 diskio 지표의 어레이를 지정합니다. 가능한 값은 `reads`, `writes`, `read_bytes`, `write_bytes`, `read_time`, `write_time`, `io_time`, `iops_in_progress`입니다. `diskio`를 포함하는 경우 이 필드는 필수입니다.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.

    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `metrics_collection_interval` – 선택 사항. diskio 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

    이 값은 초 단위로 지정됩니다.

    이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.

  - `append_dimensions` – 선택 사항. diskio 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트에 의해 수집되는 모든 유형의 지표에 대해 사용되는 `append_dimensions` 필드에 지정되어 있는 차원 외에 지정한 차원도 사용됩니다.

- `swap` – 선택 사항. 스왑 메모리 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 3개 필드만큼만 포함할 수 있습니다.

  - `measurement` – 수집될 스왑 지표의 어레이를 지정합니다. 가능한 값은 `free`, `used`, `used_percent`입니다. `swap`를 포함하는 경우 이 필드는 필수입니다.

    각 `swap` 지표에 대한 기본 단위를 보려면 Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 (p. 255) 섹션을 참조하십시오.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.

    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `metrics_collection_interval` – 선택 사항. 스왑 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

    이 값은 초 단위로 지정됩니다.

이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.

- `append_dimensions` – 선택 사항. 스왑 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트에 의해 수집되는 모든 유형의 지표에 대해 사용되는 글로벌 `append_dimensions` 필드에 지정되어 있는 차원 외에 지정한 차원도 사용됩니다. 고해상도 지표로 수집됩니다.

- `mem` – 선택 사항. 메모리 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 3개 필드만큼만 포함할 수 있습니다.
  - `measurement` – 수집될 메모리 지표의 어레이를 지정합니다. 가능한 값은 `active`, `available`, `available_percent`, `buffered`, `cached`, `free`, `inactive`, `total`, `used`, `used_percent`입니다. `mem`를 포함하는 경우 이 필드는 필수입니다.

    각 `mem` 지표에 대한 기본 단위를 보려면 Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 (p. 255) 섹션을 참조하십시오.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.
    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `metrics_collection_interval` – 선택 사항. mem 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

    이 값은 초 단위로 지정됩니다.

    이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.
  - `append_dimensions` – 선택 사항. mem 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트가 수집하는 모든 유형의 지표에 대해 사용되는 `append_dimensions` 필드에 지정되어 있는 차원 외에 이 차원도 사용됩니다.

- `net` – 선택 사항. 네트워킹 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음과 같이 최대 4개의 필드가 포함될 수 있습니다.
  - `resources` – 선택 사항. 네트워크 인터페이스 어레이를 지정하는 경우, CloudWatch가 해당 인터페이스에서만 지표를 수집합니다. 그렇지 않으면, 모든 디바이스의 지표가 수집됩니다. 또한 값으로 `*`를 지정하면 모든 인터페이스에서 지표를 수집할 수 있습니다.
  - `measurement` – 수집될 네트워킹 지표의 어레이를 지정합니다. 가능한 값은 `bytes_sent`, `bytes_recv`, `drop_in`, `drop_out`, `err_in`, `err_out`, `packets_sent`, `packets_recv`입니다. `net`를 포함하는 경우 이 필드는 필수입니다.

    각 `net` 지표에 대한 기본 단위를 보려면 Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 (p. 255) 섹션을 참조하십시오.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.
    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `metrics_collection_interval` – 선택 사항. net 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

    이 값은 초 단위로 지정됩니다. 예를 들어, 10을 지정하면 10초마다 지표가 수집되도록 설정되며, 300으로 설정하면 5분마다 지표가 수집되도록 지정됩니다.

    이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.

  - `append_dimensions` – 선택 사항. net 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트에 의해 수집되는 모든 유형의 지표에 대해 사용되는 `append_dimensions` 필드에 지정되어 있는 차원 외에 이 차원도 사용됩니다.
- `netstat` – 선택 사항. TCP 연결 상태 및 UDP 연결 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 3개 필드만큼만 포함할 수 있습니다.
  - `measurement` – 수집될 netstat 지표의 어레이를 지정합니다. 가능한 값은 `tcp_close`, `tcp_close_wait`, `tcp_closing`, `tcp_established`, `tcp_fin_wait1`, `tcp_fin_wait2`, `tcp_last_ack`, `tcp_listen`, `tcp_none`, `tcp_syn_sent`, `tcp_syn_recv`, `tcp_time_wait` 및 `udp_socket`입니다. `netstat`를 포함하는 경우 이 필드는 필수입니다.

    각 `netstat` 지표에 대한 기본 단위를 보려면 Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 (p. 255) 섹션을 참조하십시오.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.
    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `metrics_collection_interval` – 선택 사항. netstat 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

    이 값은 초 단위로 지정됩니다.

    이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 고분해능 지표에 대한 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.
  - `append_dimensions` – 선택 사항. netstat 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트에 의해 수집되는 모든 유형의 지표에 대해 사용되는 `append_dimensions` 필드에 지정되어 있는 차원 외에 이 차원도 사용됩니다.
- `processes` – 선택 사항. 프로세스 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 3개 필드만큼만 포함할 수 있습니다.
  - `measurement` – 수집될 프로세스 지표의 어레이를 지정합니다. 가능한 값은 `blocked`, `dead`, `idle`, `paging`, `running`, `sleeping`, `stopped`, `total`, `total_threads`, `wait`, `zombies`입니다. `processes`를 포함하는 경우 이 필드는 필수입니다.

    모든 `processes` 지표에 대해 기본 단위는 `Count`입니다.

    각 개별 지표의 항목 내에서 다음 중 하나 또는 둘 다를 선택적으로 지정할 수 있습니다.
    - `rename` – 이 지표에 대해 다른 이름을 지정합니다.
    - `unit` – 이 지표에 대해 사용할 단위를 지정하여 해당 지표의 기본 단위를 재정의합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.
  - `metrics_collection_interval` – 선택 사항. 프로세스 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

    이 값은 초 단위로 지정됩니다. 예를 들어, 10을 지정하면 10초마다 지표가 수집되도록 설정되며, 300으로 설정하면 5분마다 지표가 수집되도록 지정됩니다.

    이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.
  - `append_dimensions` – 선택 사항. 프로세스 지표에만 사용할 수 있는 추가 차원입니다. 이 필드를 지정하면 에이전트에 의해 수집되는 모든 유형의 지표에 대해 사용되는 `append_dimensions` 필드에 지정되어 있는 차원 외에 이 차원도 사용됩니다.
- `procstat` – 선택 사항. 개별 프로세스에서 지표를 검색하려 한다고 지정합니다. 자세한 내용은 procstat 플러그인을 사용하여 프로세스 지표 수집 (p. 246) 단원을 참조하십시오.
- `statsd` – 선택 사항. `StatsD` 프로토콜을 사용하여 사용자 지정 지표를 검색하려 한다는 것을 지정합니다. CloudWatch 에이전트는 프로토콜용 데몬으로 작용합니다. 표준 `StatsD` 클라이언트를 사용하

여 CloudWatch 에이전트에 지표를 보냅니다. 자세한 내용은 StatsD로 시작하는 사용자 지정 지표 검색 (p. 253) 단원을 참조하십시오.

다음은 Linux 서버용 `metrics` 섹션의 예입니다. 이 예에서는 3개의 CPU 지표, 3개의 netstat 지표, 3개의 프로세스 지표 및 1개의 디스크 지표가 수집되며, 에이전트는 `collectd` 클라이언트로부터 추가 지표를 받도록 설정됩니다.

```
  "metrics": {
    "metrics_collected": {
      "collectd": {},
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit": "Percent"},
          {"name": "cpu_usage_nice", "unit": "Percent"},
          "cpu_usage_guest"
        ],
        "totalcpu": false,
        "metrics_collection_interval": 10,
        "append_dimensions": {
          "test": "test1",
          "date": "2017-10-01"
        }
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_syn_sent",
          "tcp_close"
        ],
        "metrics_collection_interval": 60
      },
       "disk": {
        "measurement": [
          "used_percent"
        ],
        "resources": [
          "*"
        ],
        "drop_device": true
      },
      "processes": {
        "measurement": [
          "running",
          "sleeping",
          "dead"
        ]
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    },
    "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId", "InstanceType"],
[]]
  }
```

Windows Server

Windows Server의 `metrics_collected` 섹션에서는 `Memory`, `Processor` 및 `LogicalDisk`와 같은 각 Windows 성능 객체에 대한 하위 섹션을 생성할 수 있습니다. 사용 가능한 객체 및 카운터에 대한 자세한 내용은 Microsoft Windows 설명서를 참조하십시오.

각 객체의 하위 섹션 내에서 수집할 카운터의 `measurement` 어레이를 지정합니다. `measurement` 어레이는 구성 파일에서 지정하는 객체마다 있어야 합니다. 또한 `resources` 필드를 지정하여 지표를 수집할 인스턴스의 이름을 지정할 수 있습니다. 또한 `resources`에 대해 `*`를 지정하면 모든 인스턴스에 대해 별도의 지표를 수집할 수 있습니다. `resources`를 생략하면 모든 인스턴스의 데이터가 하나의 집합으로 집계됩니다. 인스턴스가 없는 객체의 경우 `resources`를 생략합니다.

각 객체 섹션 내에서 다음 선택 필드를 지정할 수도 있습니다.

- `metrics_collection_interval` – 선택 사항. 이 객체에 대해 지표를 수집하여 구성 파일의 `agent` 섹션에 지정되어 있는 글로벌 `metrics_collection_interval` 값을 재정의할 빈도를 지정합니다.

  이 값은 초 단위로 지정됩니다. 예를 들어, 10을 지정하면 10초마다 지표가 수집되도록 설정되며, 300으로 설정하면 5분마다 지표가 수집되도록 지정됩니다.

  이 값을 60초 미만으로 설정하면 각 지표가 고분해능 지표로 수집됩니다. 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.
- `append_dimensions` – 선택 사항. 이 객체의 지표에만 사용할 추가 차원을 지정합니다. 이 필드를 지정하면 에이전트에 의해 수집되는 모든 유형의 지표에 대해 사용되는 글로벌 `append_dimensions` 필드에 지정되어 있는 차원 외에 이 차원도 사용됩니다.

각 카운터 섹션 내에서 다음 선택 필드를 지정할 수도 있습니다.

- `rename` – 이 지표에 대해 CloudWatch에서 사용될 다른 이름을 지정합니다.
- `unit` – 이 지표에 대해 사용할 단위를 지정합니다. 지정하는 단위는 MetricDatum의 `Unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.

`metrics_collected`에는 다음과 같이 두 가지 다른 선택적 섹션을 포함할 수 있습니다.

- `statsd` – `StatsD` 프로토콜을 사용하여 사용자 지정 지표를 검색할 수 있도록 합니다. CloudWatch 에이전트는 프로토콜용 데몬으로 작용합니다. 표준 `StatsD` 클라이언트를 사용하여 CloudWatch 에이전트에 지표를 보냅니다. 자세한 내용은 StatsD로 시작하는 사용자 지정 지표 검색 (p. 253) 단원을 참조하십시오.
- `procstat` – 개별 프로세스에서 지표를 검색할 수 있도록 합니다. 자세한 내용은 procstat 플러그인을 사용하여 프로세스 지표 수집 (p. 246) 단원을 참조하십시오.

다음은 Windows Server에서 사용하기 위한 `metrics` 섹션의 예입니다. 이 예에서는 다양한 Windows 지표가 수집되며, 컴퓨터가 `StatsD` 클라이언트로부터 추가 지표를 받도록 설정되기도 합니다.

```
"metrics": {
    "metrics_collected": {
      "statsd": {},
      "Processor": {
        "measurement": [
          {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
          "% Interrupt Time",
          "% User Time",
          "% Processor Time"
        ],
        "resources": [
          "*"
        ],
```

```
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      },
      "LogicalDisk": {
        "measurement": [
          {"name": "% Idle Time", "unit": "Percent"},
          {"name": "% Disk Read Time", "rename": "DISK_READ"},
          "% Disk Write Time"
        ],
        "resources": [
          "*"
        ]
      },
      "Memory": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Available Bytes",
          "Cache Faults/sec",
          "Page Faults/sec",
          "Pages/sec"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "Network Interface": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Bytes Received/sec",
          "Bytes Sent/sec",
          "Packets Received/sec",
          "Packets Sent/sec"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "System": {
        "measurement": [
          "Context Switches/sec",
          "System Calls/sec",
          "Processor Queue Length"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    },
    "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]]
    }
  }
```

CloudWatch 에이전트 구성 파일: 로그 섹션

`logs` 섹션에 포함되는 필드는 다음과 같습니다.

- `logs_collected` – `logs` 섹션이 포함되어 있는 경우 필수입니다. 서버로부터 수집될 로그 파일 및 Windows 서버 로그를 지정합니다. 두 필드인 `files` 및 `windows_events`를 포함할 수 있습니다.
  - `files` – CloudWatch 에이전트가 수집할 일반 로그 파일을 지정합니다. 여기에는 `collect_list`라는 필드 하나가 포함되는데, 이 필드는 이러한 파일을 추가로 정의합니다.
    - `collect_list` – `files`가 포함되어 있는 경우 필수입니다. 항목 어레이를 포함하며, 항목 각각이 수집될 하나의 로그 파일을 지정합니다. 이러한 항목 각각에는 다음 필드를 포함할 수 있습니다.
      - `file_path` – CloudWatch Logs에 업로드할 로그 파일의 경로를 지정합니다. 수퍼 별표로 `**`를 추가한, 규칙과 일치하는 표준 Unix glob는 허용됩니다. 예를 들어 `/var/log/**.log`를 지정하면 `/var/log` 디렉터리 트리의 모든 `.log` 파일이 수집됩니다. 보다 자세한 예는 Glob Library를 참조하십시오.

        표준 별표를 표준 와일드카드로 사용할 수도 있습니다. 예를 들어, `/var/log/system.log*`는 `/var/log`에서 `system.log_1111`, `system.log_2222` 등의 파일과 일치합니다.

        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.2018-06-01-01` 및 `access_log.2018-06-01-02`)을 지정할 때 사용하는 것이 좋습니다. 여러 종류의 파일을 지정하려면 로그 파일이 종류별로 서로 다른 로그 스트림에 들어가도록 에이전트 구성 파일에 다른 로그 스트림 항목 하나를 추가합니다.
      - `auto_removal` – 선택 사항. true인 경우 CloudWatch 에이전트는 CloudWatch Logs에 이전 로그 파일을 업로드한 후 자동으로 해당 파일을 제거합니다. 에이전트는 각 날짜에 대해 별도의 파일을 만드는 로그와 같이 여러 파일을 생성하는 로그에서만 전체 파일을 제거합니다. 로그가 단일 파일에 연속적으로 기록되는 경우에는 제거되지 않습니다.

        로그 파일 순환 또는 제거 방법이 이미 있는 경우 이 필드를 생략하거나 `false`로 설정하는 것이 좋습니다.

        이 필드를 생략하면 기본값인 `false`가 사용됩니다.
      - `log_group_name` – 선택 사항. CloudWatch Logs에서 로그 그룹 이름으로 사용할 항목을 지정합니다. 이름의 일부로 `{instance_id}`, `{hostname}`, `{local_hostname}`, `{ip_address}`를 이름 안의 변수로 사용할 수 있습니다. `{hostname}`은 EC2 메타데이터에서 호스트 이름을 가져오고, `{local_hostname}`은 네트워크 구성 파일에 있는 호스트 이름을 사용합니다.

        이러한 변수를 사용하여 많은 로그 그룹을 생성하는 경우 리전별 계정당 5,000개 로그 그룹이 한도라는 점에 유의하십시오.

        허용되는 문자에는 a–z, A–Z, 0–9, '_'(밑줄), '- (하이픈), '/'(슬래시) 및 '.'(마침표)가 포함됩니다.

        혼동하지 않도록 이 필드를 지정하는 것이 좋습니다. 이 필드를 생략할 경우 로그 그룹 이름으로 마지막 점까지의 파일 경로가 사용됩니다. 예를 들어 파일 경로가 `/tmp/TestLogFile.log.2017-07-11-14`이면 로그 그룹 이름은 `/tmp/TestLogFile.log`입니다.
      - `log_stream_name` – 선택 사항. CloudWatch Logs에서 로그 스트림 이름으로 사용할 항목을 지정합니다. 이름의 일부로 `{instance_id}`, `{hostname}`, `{local_hostname}`, `{ip_address}`를 이름 안의 변수로 사용할 수 있습니다. `{hostname}`은 EC2 메타데이터에서 호스트 이름을 가져오고, `{local_hostname}`은 네트워크 구성 파일에 있는 호스트 이름을 사용합니다.

        이 필드를 생략하면 기본값인 `{instance_id}`가 사용됩니다. 로그 스트림이 아직 없는 경우 로그 스트림이 자동으로 생성됩니다.
      - `timezone` – 선택 사항. 로그 이벤트에 타임스탬프를 표시할 때 사용할 시간대를 지정합니다. 유효 값은 `UTC`와 `Local`입니다. 기본값은 `Local`입니다.

        `timestamp_format`에 값을 지정하지 않으면 이 파라미터는 무시됩니다.

- `timestamp_format` – 선택 사항. 일반 텍스트와 %로 시작하는 특수 기호를 사용하여 타임스탬프 형식을 지정합니다. 이 필드를 생략한 경우 현재 시간이 사용됩니다. 이 필드를 사용하면 다음 목록에 있는 기호를 형식의 일부로 사용할 수 있습니다.

  단일 로그 항목에 형식과 일치하는 두 개의 타임스탬프가 포함된 경우 첫 번째 타임스탬프가 사용됩니다.

  이 기호 목록은 이전 CloudWatch Logs 에이전트에서 사용되는 목록과 다릅니다. 이러한 차이점에 대한 요약은 통합 CloudWatch 에이전트와 이전 CloudWatch Logs 에이전트 간 타임스탬프 차이 (p. 269) 단원을 참조하십시오.

  `%y`

  제로 패딩된 10진수 형태의 세기를 제외한 연도 예를 들어 `19`는 2019년을 나타냅니다.

  `%Y`

  10진수 형태로 세기가 포함된 연도 예: `2019`.

  `%b`

  로캘 약어 형태의 월

  `%B`

  로캘 전체 이름 형태의 월

  `%m`

  제로 패딩된 10진수 형태의 월

  `%-m`

  10진수 형태의 월(제로 패딩되지 않음)

  `%d`

  제로 패딩된 10진수 형태의 월 날짜

  `%-d`

  10진수 형태의 월 날짜(제로 패딩되지 않음)

  `%A`

  평일의 전체 이름(예: `Monday`)

  `%a`

  평일의 약어(예: `Mon`)

  `%H`

  제로 패딩된 10진수 형태의 시간(24시간 방식)

  `%I`

  제로 패딩된 10진수 형태의 시간(12시간 방식)

  `%-I`

  10진수 형태의 시간(12시간 방식, 제로 패딩되지 않음)

  `%p`

  AM 또는 PM

  `%M`

  제로 패딩된 10진수 형태의 분

`%-M`

10진수 형태의 분(제로 패딩되지 않음)

`%S`

제로 패딩된 10진수 형태의 초

`%-S`

10진수 형태의 초(제로 패딩되지 않음)

`%f`

10진수 형태(1-9의 숫자)의 소수부 초(왼쪽에 제로 패딩됨)

`%Z`

시간대(예: `PST`)

`%z`

현지 시간대와 UTC 간의 오프셋으로 표시되는 시간대입니다. 예, `-0700`. 이 형식만 지원됩니다. 예를 들어, `-07:00`은 유효한 형식이 아닙니다.

- `multi_line_start_pattern` – 로그 메시지의 시작을 식별하기 위한 패턴을 지정합니다. 로그 메시지는 패턴과 일치하는 하나의 줄 및 패턴과 일치하지 않는 후속 줄로 이루어져 있습니다.

  이 필드를 생략할 경우, 여러 줄 모드가 비활성화되고 기본적으로 공백이 아닌 문자로 시작되는 줄에서 이전의 로그 메시지가 종료되고 새로운 로그 메시지가 시작됩니다.

  이 필드를 포함하는 경우, 타임스탬프 형식과 동일한 정규식을 사용하도록 `{timestamp_format}`을 지정할 수 있습니다. 그렇지 않으면, CloudWatch Logs에 다른 정규식을 지정하여 여러 줄 항목의 시작 줄을 결정하는 데 사용할 수 있습니다.

- `encoding` – 로그 파일을 정확하게 읽을 수 있도록 로그 파일의 인코딩을 지정합니다. 코딩을 잘못 지정하면 디코딩할 수 없는 문자를 다른 문자가 대체하기 때문에 데이터가 손실될 수 있습니다.

  기본값은 `utf-8`입니다. 가능한 모든 값은 다음과 같습니다.

  ```
  ascii, big5, euc-jp, euc-kr, gbk, gb18030, ibm866, iso2022-jp,
  iso8859-2, iso8859-3, iso8859-4, iso8859-5, iso8859-6, iso8859-7,
  iso8859-8, iso8859-8-i, iso8859-10, iso8859-13, iso8859-14, iso8859-15,
  iso8859-16, koi8-r, koi8-u, macintosh, shift_jis, utf-8, utf-16,
  windows-874, windows-1250, windows-1251, windows-1252, windows-1253,
  windows-1254, windows-1255, windows-1256, windows-1257, windows-1258, x-
  mac-cyrillic
  ```

- `windows_events` 섹션은 Windows Server가 실행되는 서버로부터 수집할 Windows 이벤트 유형을 지정합니다. 여기에는 다음 필드가 포함됩니다.
  - `collect_list` – `windows_events`가 포함되어 있는 경우 필수입니다. 수집될 Windows 이벤트의 유형 및 수준을 지정합니다. 수집될 각 로그에는 이 섹션에 있는 항목이 있으며, 여기에는 다음 필드를 포함할 수 있습니다.
    - `event_name` – 기록할 Windows 이벤트 유형을 지정합니다. 이 유형은 Windows 이벤트 로그 채널 이름과 동일합니다(예: `System`, `Security`, `Application` 등). 이 필드는 기록할 Windows 이벤트 유형마다 있어야 합니다.
    - `event_levels` – 기록할 이벤트 레벨을 지정합니다. 기록할 각 수준을 지정해야 합니다. 가능한 값은 `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` 및 `VERBOSE`입니다. 이 필드는 기록할 Windows 이벤트 유형마다 있어야 합니다.
    - `log_group_name` – 필수 사항. CloudWatch Logs에서 로그 그룹 이름으로 사용할 항목을 지정합니다.
    - `log_stream_name` – 선택 사항. CloudWatch Logs에서 로그 스트림 이름으로 사용할 항목을 지정합니다. 이름의 일부로 `{instance_id}`, `{hostname}`, `{local_hostname}`, `{ip_address}`를

이름 안의 변수로 사용할 수 있습니다. {hostname}은 EC2 메타데이터에서 호스트 이름을 가져오고, {local_hostname}은 네트워크 구성 파일에 있는 호스트 이름을 사용합니다.

이 필드를 생략하면 기본값인 {instance_id}가 사용됩니다. 로그 스트림이 아직 없는 경우 로그 스트림이 자동으로 생성됩니다.

  - event_format – 선택 사항. CloudWatch Logs에 Windows 이벤트를 저장할 때 사용할 형식을 지정합니다. xml은 Windows 이벤트 뷰어에서처럼 XML 형식을 사용합니다. text는 기존의 CloudWatch Logs 에이전트 형식을 사용합니다.

- log_stream_name – 필수 사항. collect_list의 항목에 개별 로그 스트림 이름이 정의되어 있지 않은 Windows 이벤트나 로그에 사용될 기본 로그 스트림 이름을 지정합니다.
- endpoint_override – 에이전트가 로그를 보내는 엔드포인트로 사용할 FIPS 엔드포인트 또는 프라이빗 링크를 지정합니다. 이 필드를 지정하고 프라이빗 링크를 설정하면 로그를 Amazon VPC 엔드포인트로 보낼 수 있습니다. 자세한 내용은 Amazon VPC이란 무엇입니까?단원을 참조하십시오.

  endpoint_override의 값은 URL인 문자열이어야 합니다.
- force_flush_interval – 로그가 서버로 전송되기 전에 메모리 버퍼에 남아 있을 최대 시간(초)을 지정합니다. 이 필드에 대한 설정과 상관없이 버퍼 내 로그의 크기가 1MB에 도달하면 로그가 즉시 서버로 전송됩니다. 기본값은 5입니다.
- credentials – 로그를 다른 AWS 계정에 보낼 때 사용할 IAM 역할을 지정합니다. 지정된 경우 이 필드는 1개의 파라미터 role_arn를 포함합니다.
  - role_arn – 로그를 다른 AWS 계정에 보낼 때 인증에 사용할 IAM 역할의 ARN을 지정합니다. 자세한 내용은 다른 계정에 지표 및 로그 전송 (p. 268) 단원을 참조하십시오. 여기에 지정한 경우 이를 통해 구성 파일(해당 시)의 agent 섹션에 지정되어 있는 role_arn을 재정의합니다.
- metrics_collected – 에이전트가 로그에 포함된 지표를 수집하도록 지정합니다. 현재 metrics_collected 필드에는 emf 필드만 포함될 수 있습니다.
  - emf – 에이전트가 포함된 지표 형식의 로그를 수집하도록 지정합니다. 이러한 로그에서 지표 데이터를 생성할 수 있습니다. 자세한 내용은 높은 카디널리티 로그 수집 및 CloudWatch 임베디드 지표 형식으로 지표 생성 (p. 328) 단원을 참조하십시오.

다음은 logs 섹션의 예입니다.

```
"logs":
   {
       "logs_collected": {
           "files": {
               "collect_list": [
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\amazon-cloudwatch-agent.log",
                       "log_group_name": "amazon-cloudwatch-agent.log",
                       "log_stream_name": "my_log_stream_name_1",
                       "timestamp_format": "%H: %M: %S%y%b%-d"
                   },
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\test.log",
                       "log_group_name": "test.log",
                       "log_stream_name": "my_log_stream_name_2"
                   }
               ]
           },
           "windows_events": {
               "collect_list": [
                   {
                       "event_name": "System",
                       "event_levels": [
                           "INFORMATION",
                           "ERROR"
```

```
                    ],
                    "log_group_name": "System",
                    "log_stream_name": "System"
                },
                {
                    "event_name": "CustomizedName",
                    "event_levels": [
                        "INFORMATION",
                        "ERROR"
                    ],
                    "log_group_name": "CustomizedLogGroup",
                    "log_stream_name": "CustomizedLogStream"
                }
            ]
        }
    },
    "log_stream_name": "my_log_stream_name"
}
```

CloudWatch 에이전트 구성 파일: 전체 예

다음은 Linux 서버에 대한 전체 CloudWatch 에이전트 구성 파일의 예입니다.

수집하려는 척도에 대해 `measurement` 섹션에 나열된 항목들은 완전한 척도 이름을 지정하거나, 리소스 유형에 추가되는 척도 이름의 일부만 지정할 수 있습니다. 예를 들어 `diskio` 섹션의 `measurement` 섹션에 `reads` 또는 `diskio_reads`을 지정하면 `diskio_reads` 지표가 수집될 수 있습니다.

이 예에는 `measurement` 섹션에서 지표를 지정하는 두 가지 방법이 모두 포함되어 있습니다.

```
{
  "agent": {
    "metrics_collection_interval": 10,
    "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
  },
  "metrics": {
    "metrics_collected": {
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit": "Percent"},
          {"name": "cpu_usage_nice", "unit": "Percent"},
          "cpu_usage_guest"
        ],
        "totalcpu": false,
        "metrics_collection_interval": 10,
        "append_dimensions": {
          "customized_dimension_key_1": "customized_dimension_value_1",
          "customized_dimension_key_2": "customized_dimension_value_2"
        }
      },
      "disk": {
        "resources": [
          "/",
          "/tmp"
        ],
        "measurement": [
          {"name": "free", "rename": "DISK_FREE", "unit": "Gigabytes"},
          "total",
          "used"
        ],
         "ignore_file_system_types": [
```

```
            "sysfs", "devtmpfs"
          ],
          "metrics_collection_interval": 60,
          "append_dimensions": {
            "customized_dimension_key_3": "customized_dimension_value_3",
            "customized_dimension_key_4": "customized_dimension_value_4"
          }
        },
        "diskio": {
          "resources": [
            "*"
          ],
          "measurement": [
            "reads",
            "writes",
            "read_time",
            "write_time",
            "io_time"
          ],
          "metrics_collection_interval": 60
        },
        "swap": {
          "measurement": [
            "swap_used",
            "swap_free",
            "swap_used_percent"
          ]
        },
        "mem": {
          "measurement": [
            "mem_used",
            "mem_cached",
            "mem_total"
          ],
          "metrics_collection_interval": 1
        },
        "net": {
          "resources": [
            "eth0"
          ],
          "measurement": [
            "bytes_sent",
            "bytes_recv",
            "drop_in",
            "drop_out"
          ]
        },
        "netstat": {
          "measurement": [
            "tcp_established",
            "tcp_syn_sent",
            "tcp_close"
          ],
          "metrics_collection_interval": 60
        },
        "processes": {
          "measurement": [
            "running",
            "sleeping",
            "dead"
          ]
        }
      },
      "append_dimensions": {
        "ImageId": "${aws:ImageId}",
        "InstanceId": "${aws:InstanceId}",
```

```
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],
[]],
        "force_flush_interval" : 30
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "log_stream_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
                "log_group_name": "test.log",
                "log_stream_name": "test.log",
                "timezone": "Local"
              }
            ]
          }
        },
        "log_stream_name": "my_log_stream_name",
        "force_flush_interval" : 15
      }
    }
```

다음은 Windows Server가 실행되는 서버에 대한 전체 CloudWatch 에이전트 구성 파일의 예입니다.

```
{
      "agent": {
        "metrics_collection_interval": 60,
        "logfile": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\\amazon-
cloudwatch-agent.log"
      },
      "metrics": {
        "metrics_collected": {
          "Processor": {
            "measurement": [
              {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
              "% Interrupt Time",
              "% User Time",
              "% Processor Time"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
              "customized_dimension_key_1": "customized_dimension_value_1",
              "customized_dimension_key_2": "customized_dimension_value_2"
            }
          },
          "LogicalDisk": {
            "measurement": [
              {"name": "% Idle Time", "unit": "Percent"},
              {"name": "% Disk Read Time", "rename": "DISK_READ"},
              "% Disk Write Time"
            ],
            "resources": [
              "*"
```

```
            ]
          },
          "customizedObjectName": {
            "metrics_collection_interval": 60,
            "customizedCounterName": [
              "metric1",
              "metric2"
            ],
            "resources": [
              "customizedInstaces"
            ]
          },
          "Memory": {
            "metrics_collection_interval": 5,
            "measurement": [
              "Available Bytes",
              "Cache Faults/sec",
              "Page Faults/sec",
              "Pages/sec"
            ]
          },
          "Network Interface": {
            "metrics_collection_interval": 5,
            "measurement": [
              "Bytes Received/sec",
              "Bytes Sent/sec",
              "Packets Received/sec",
              "Packets Sent/sec"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
              "customized_dimension_key_3": "customized_dimension_value_3"
            }
          },
          "System": {
            "measurement": [
              "Context Switches/sec",
              "System Calls/sec",
              "Processor Queue Length"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]]
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\\amazon-
cloudwatch-agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\test.log",
                "log_group_name": "test.log",
```

```
            "timezone": "Local"
          }
        ]
      },
      "windows_events": {
        "collect_list": [
          {
            "event_name": "System",
            "event_levels": [
              "INFORMATION",
              "ERROR"
            ],
            "log_group_name": "System",
            "log_stream_name": "System",
            "event_format": "xml"
          },
          {
            "event_name": "CustomizedName",
            "event_levels": [
              "WARNING",
              "ERROR"
            ],
            "log_group_name": "CustomizedLogGroup",
            "log_stream_name": "CustomizedLogStream",
            "event_format": "xml"
          }
        ]
      }
    },
    "log_stream_name": "example_log_stream_name"
  }
}
```

수동으로 CloudWatch 에이전트 구성 파일 저장

CloudWatch 에이전트 구성 파일을 수동으로 생성하거나 편집할 경우 이름을 지정할 수 있습니다. 문제를 간단하게 해결할 수 있도록 이름을 Linux 서버에서는 `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json`로 지정하고, Windows 서버를 실행하는 서버에서는 `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`로 지정하는 것이 좋습니다. 파일을 생성한 후에는 해당 파일을 에이전트를 실행하려고 하는 다른 서버에 복사할 수 있습니다.

CloudWatch 에이전트 구성 파일을 시스템 관리자 Parameter Store에 업로드

SSM Agent를 사용하여 서버에 CloudWatch 에이전트를 설치하려는 경우 CloudWatch 에이전트 구성 파일을 수동으로 편집한 후 시스템 관리자 Parameter Store에 업로드할 수 있습니다. 이렇게 하려면 시스템 관리자 `put-parameter` 명령을 사용합니다.

파일을 Parameter Store에 저장할 수 있도록 하려면 충분한 권한이 있는 IAM 역할을 사용해야 합니다. 자세한 내용은 CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 (p. 201) 단원을 참조하십시오.

다음 명령을 사용합니다. 여기서 *`parameter name`*은 Parameter Store의 이 파일에 대해 사용될 이름이며, *`configuration_file_pathname`*은 편집한 구성 파일의 경로 및 파일 이름입니다.

```
aws ssm put-parameter --name "parameter name" --type "String" --value
 file://configuration_file_pathname
```

procstat 플러그인을 사용하여 프로세스 지표 수집

procstat 플러그인을 사용하면 개별 프로세스에서 지표를 수집할 수 있습니다. 이 플러그인은 Windows Server 2008 R2 이상이 실행되는 서버와 Linux 서버에서 지원됩니다.

주제

- procstat를 사용하도록 CloudWatch 에이전트 구성 (p. 246)
- Procstat로 수집한 지표 (p. 248)

procstat를 사용하도록 CloudWatch 에이전트 구성

procstat 플러그인을 사용하려면 CloudWatch 에이전트 구성 파일의 `metrics_collected` 섹션에 `procstat` 섹션을 추가합니다. 모니터링할 프로세스를 지정하는 방법은 세 가지가 있습니다. 이러한 메서드는 하나만 사용할 수 있지만 해당 메서드를 사용해 모니터링할 프로세스를 여러 개 지정할 수 있습니다.

- `pid_file`: 프로세스를 통해 생성된 프로세스 식별 번호(PID)의 이름으로 프로세스를 선택합니다.
- `exe`: 정규식 일치 규칙을 사용하여 프로세스 이름이 지정한 문자열과 일치하는 프로세스를 선택합니다. 자세한 내용은 구문을 참조하십시오.
- `pattern`: 프로세스를 시작하는 데 사용된 명령줄로 프로세스를 선택합니다. 정규식 일치 규칙을 사용하여 명령줄이 지정한 문자열과 일치하는 프로세스가 모두 선택됩니다. 명령과 함께 사용된 파라미터 및 옵션을 포함하여 전체 명령줄을 확인합니다.

위 섹션 중 두 개 이상을 포함했더라도 CloudWatch 에이전트는 이러한 방법 중 하나만 사용합니다. 섹션을 두 개 이상 지정한 경우 CloudWatch 에이전트는 `pid_file` 섹션을 사용합니다(있는 경우). 없는 경우에는 `exe` 섹션을 사용합니다.

Linux 서버의 경우 `exe` 또는 `pattern` 섹션에서 지정한 문자열은 정규식으로 평가됩니다. Windows Server를 실행하는 서버에서 이러한 문자열은 WMI 쿼리로 평가됩니다. 자세한 내용은 LIKE 연산자를 참조하십시오.

어떤 방법을 사용하든 간에 지표 수집 빈도(초)를 지정하는 선택적 `metrics_collection_interval` 파라미터를 포함할 수 있습니다. 이 파라미터를 생략하면 기본값인 60초가 사용됩니다.

다음 단원의 예제에서 `procstat` 섹션은 에이전트 구성 파일의 `metrics_collected` 섹션에 포함된 유일한 섹션입니다. 실제 구성 파일에서는 `metrics_collected`에 다른 섹션을 포함할 수도 있습니다. 자세한 내용은 CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 (p. 227) 단원을 참조하십시오.

Pid_file로 구성

다음 예제 `procstat` 섹션에서는 PID 파일 `example1.pid` 및 `example2.pid`를 생성하는 프로세스를 모니터링합니다. 각 프로세스에서는 여러 지표가 수집됩니다. `example2.pid`를 생성하는 프로세스에서 수집되는 지표는 10초마다 수집되고, `example1.pid` 프로세스에서 수집되는 지표는 60초마다 수집됩니다(기본값).

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pid_file": "/var/run/example1.pid",
                    "measurement": [
                        "cpu_usage",
                        "memory_rss"
                    ]
                },
                {
```

```
                    "pid_file": "/var/run/example2.pid",
                    "measurement": [
                        "read_bytes",
                        "read_count",
                        "write_bytes"
                    ],
                    "metrics_collection_interval": 10
                }
            ]
        }
    }
}
```

Exe로 구성

다음 예제 `procstat` 섹션은 문자열 `agent` 또는 `plugin`과 이름이 일치하는 모든 프로세스를 모니터링합니다. 각 프로세스에서 동일한 지표가 수집됩니다.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "exe": "agent",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                },
                {
                    "exe": "plugin",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                }
            ]
        }
    }
}
```

Pattern으로 구성

다음 예제 `procstat` 섹션은 문자열 `config` 또는 `-c`와 명령줄이 일치하는 모든 프로세스를 모니터링합니다. 각 프로세스에서 동일한 지표가 수집됩니다.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pattern": "config",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                },
                {
```

```
                    "pattern": "-c",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                }
            ]
        }
    }
}
```

Procstat로 수집한 지표

다음 표에는 `procstat` 플러그인을 사용하여 수집할 수 있는 지표가 나열되어 있습니다.

CloudWatch 에이전트는 다음 지표 이름의 시작 부분에 `procstat`를 추가합니다. Linux 서버 또는 Windows Server를 실행하는 서버에서 수집되었는지 여부에 따라 구문이 다릅니다. 예를 들어, `cpu_time` 지표는 Linux에서 수집된 경우 `procstat_cpu_time`으로 표시되고, Windows Server에서 수집된 경우에는 `procstat cpu_time`으로 표시됩니다.

지표 이름	제공 위치	설명
`cpu_time`	리눅스	프로세스가 CPU를 사용하는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 수
`cpu_time_system`	Linux, Windows Server	프로세스가 시스템 모드에 있는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 유형: Float 단위: 수
`cpu_time_user`	Linux, Windows Server	프로세스가 사용자 모드에 있는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 수
`cpu_usage`	Linux, Windows Server	어떠한 용량에서든 프로세스가 활성화되어 있는 시간(백분율)입니다. 단위: 백분율

지표 이름	제공 위치	설명
memory_data	리눅스	프로세스가 데이터에 사용하는 메모리의 양입니다. 단위: 바이트
memory_locked	리눅스	프로세스가 잠근 메모리의 양입니다. 단위: 바이트
memory_rss	Linux, Windows Server	프로세스에서 사용 중인 실제 메모리의 크기(실제 상주 메모리)입니다. 단위: 바이트
memory_stack	리눅스	프로세스에서 사용 중인 스택 메모리의 양입니다. 단위: 바이트
memory_swap	리눅스	프로세스에서 사용 중인 스왑 메모리의 양입니다. 단위: 바이트
memory_vms	Linux, Windows Server	프로세스에서 사용 중인 가상 메모리의 양입니다. 단위: 바이트
pid	Linux, Windows Server	프로세스 식별자(ID)입니다. 단위: 수

지표 이름	제공 위치	설명
pid_count	Linux, Windows Server	프로세스와 관련된 프로세스 ID의 수입니다. Linux 서버에서 이 지표의 전체 이름은 `procstat_lookup_pid_cour` 고 Windows Server에서 해당 이름은 `procstat_lookup pid_count`입니다. 단위: 수
`read_bytes`	Linux, Windows Server	프로세스가 디스크에서 읽은 바이트 수입니다. 단위: 바이트
`write_bytes`	Linux, Windows Server	프로세스가 디스크에 기록한 바이트 수입니다. 단위: 바이트
`read_count`	Linux, Windows Server	프로세스에서 실행한 디스크 읽기 작업의 수입니다. 단위: 수
`write_count`	Linux, Windows Server	프로세스에서 실행한 디스크 쓰기 작업의 수입니다. 단위: 수
`involuntary_context_switches`	리눅스	프로세스의 컨텍스트가 강제로 전환된 횟수입니다. 단위: 수
`voluntary_context_switches`	리눅스	프로세스의 컨텍스트가 강제로 전환된 횟수입니다. 단위: 수

지표 이름	제공 위치	설명
realtime_priority	리눅스	프로세스에 대한 실시간 우선 순위의 현재 사용량입니다. 단위: 수
nice_priority	리눅스	프로세스에 대한 nice priority의 현재 사용입니다. 단위: 수
signals_pending	리눅스	프로세스에서 처리하도록 대기 중인 신호 수입니다. 단위: 수
rlimit_cpu_time_hard	리눅스	프로세스에 대한 하드 CPU 시간 리소스 제한입니다. 단위: 수
rlimit_cpu_time_soft	리눅스	프로세스에 대한 소프트 CPU 시간 리소스 제한입니다. 단위: 수
rlimit_file_locks_hard	리눅스	프로세스에 대한 하드 파일 잠금 리소스 제한입니다. 단위: 수
rlimit_file_locks_soft	리눅스	프로세스에 대한 소프트 파일 잠금 리소스 제한입니다. 단위: 수
rlimit_memory_data_hard	리눅스	데이터에 사용되는 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다. 단위: 바이트

지표 이름	제공 위치	설명
rlimit_memory_data_soft	리눅스	데이터에 사용되는 메모리와 관련해 프로세스에 대한 소프트 리소스 제한입니다. 단위: 바이트
rlimit_memory_locked_hard	리눅스	잠긴 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다. 단위: 바이트
rlimit_memory_locked_soft	리눅스	잠긴 메모리와 관련해 프로세스에 대한 소프트 리소스 제한입니다. 단위: 바이트
rlimit_memory_rss_hard	리눅스	물리적 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다. 단위: 바이트
rlimit_memory_rss_soft	리눅스	물리적 메모리와 관련해 프로세스에 대한 소프트 리소스 제한입니다. 단위: 바이트
rlimit_memory_stack_hard	리눅스	프로세스 스택에 대한 하드 리소스 제한입니다. 단위: 바이트
rlimit_memory_stack_soft	리눅스	프로세스 스택에 대한 소프트 리소스 제한입니다. 단위: 바이트
rlimit_memory_vms_hard	리눅스	가상 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다. 단위: 바이트

StatsD로 시작하는 사용자 지정 지표 검색

CloudWatch 에이전트를 StatsD 프로토콜과 함께 사용하여 애플리케이션 또는 서비스에서 사용자 지정 지표를 검색할 수 있습니다. StatsD는 Windows Server가 실행되는 서버와 Linux 서버에서 모두 지원되며, CloudWatch는 다음 StatsD 형식을 지원합니다.

```
MetricName:value|type|@sample_rate|#tag1:
  value,tag1...
```

- MetricName – 콜론, 막대, # 문자 또는 @ 문자를 포함하는 문자열.
- value – 이 값은 정수 또는 부동 소수점일 수 있습니다.
- type – c 카운터, g 게이지, ms 타이머, h 히스토그램 또는 s 세트에 지정하십시오.
- sample_rate – (선택 사항) 0과 1(포함) 사이의 부동 소수점. 카운터, 히스토그램 및 타이머 지표에만 사용하십시오. 기본값은 1(시간의 샘플링 100%)입니다.
- tags – (선택 사항) 쉼표로 구분된 태그 목록입니다. StatsD 태그는 CloudWatch의 차원과 유사합니다. env:prod와 같이 키/값 태그에 콜론을 사용하십시오.

이 형식을 따르는 모든 StatsD 클라이언트를 사용하여 지표를 CloudWatch 에이전트에 보낼 수 있습니다. 사용 가능한 일부 StatsD 클라이언트에 대한 자세한 내용은 GitHub의 StatsD 클라이언트 페이지를 참조하십시오.

이러한 사용자 지정 지표를 수집하려면 에이전트 구성 파일의 metrics_collected 섹션에 "statsd":{} 줄을 추가합니다. 이 줄을 수동으로 추가할 수 있습니다. 마법사를 사용하여 구성 파일을 생성하는 경우 자동으로 수행됩니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.

StatsD 기본 구성은 대부분의 사용자에게 작동합니다. 필요에 따라 에이전트 구성 파일의 statsd 섹션에 추가할 수 있는 선택 사항 필드는 3개입니다.

- service_address – CloudWatch 에이전트가 수신해야 하는 서비스 주소입니다. 형식은 *ip*:*port*입니다. IP 주소를 생략하면 에이전트가 유효한 인터페이스를 모두 수신합니다. UDP 형식만 지원되므로 UDP 접두사를 지정할 필요가 없습니다.

  기본값은 :8125입니다.
- metrics_collection_interval – StatsD 플러그인에서 지표를 실행하고 수집하는 간격(초)입니다. 기본값은 10초입니다. 범위는 1–172,000입니다.
- metrics_aggregation_interval – CloudWatch에서 지표를 단일 데이터 포인트로 집계하는 간격(초)입니다. 기본 값은 60초입니다.

  예를 들어 metrics_collection_interval이 10이고 metrics_aggregation_interval이 60이면, CloudWatch에서는 10초마다 데이터를 수집합니다. 매분 후 이 1분의 6개의 데이터 표시값이 단일 데이터 포인트에 집계되어 CloudWatch에 전송됩니다.

  범위는 0–172,000입니다. metrics_aggregation_interval을 0으로 설정하면 StatsD 지표를 집계할 수 없습니다.

다음은 기본값 포트와 사용자 지정 수집 및 집계 간격을 사용하는 에이전트 구성 파일의 statsd 섹션의 예입니다.

```
{
   "metrics":{
      "metrics_collected":{
         "statsd":{
            "service_address":":8125",
            "metrics_collection_interval":60,
            "metrics_aggregation_interval":300
         }
```

```
        }
    }
}
```

collectd로 사용자 지정 지표 검색

CloudWatch 에이전트와 collectd 프로토콜을 함께 사용하여 애플리케이션 또는 서비스에서 사용자 지정 지표를 검색할 수 있습니다. 이 프로토콜은 Linux 서버에서만 지원됩니다. collectd 소프트웨어를 사용하여 지표를 CloudWatch 에이전트에 보냅니다. collectd 지표의 경우 CloudWatch 에이전트가 서버 역할을 하며, collectd 플러그인이 클라이언트 역할을 합니다.

collectd 소프트웨어는 모든 서버에 자동으로 설치되지 않습니다. 자세한 내용은 collectd 다운로드 페이지를 참조하십시오.

이러한 사용자 지정 지표를 수집하려면 에이전트 구성 파일의 metrics_collected 섹션에 "collectd": {} 줄을 추가합니다. 이 줄을 수동으로 추가할 수 있습니다. 마법사를 사용하여 이 구성 파일을 생성하는 경우, 사용자를 위해 이루어집니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.

추가 선택적 파라미터도 사용 가능합니다. collectd를 사용하고 있고 `/etc/collectd/auth_file`를 collectd_auth_file로 사용하지 않는 경우 이러한 선택 사항을 일부 설정해야 합니다.

- service_address: CloudWatch 에이전트가 들어야 하는 서비스 주소. 형식은 `"udp://`*ip*`:`*port*입니다. 기본값은 `udp://127.0.0.1:25826`입니다.
- name_prefix: 각 collectd 지표의 이름 시작 부분에 부착하는 접두사. 기본값은 `collectd_`입니다. 최대 길이는 255자입니다.
- collectd_security_level: 네트워크 구성의 보안 수준을 설정합니다. 기본값은 encrypt입니다.

  encrypt는 암호화된 데이터만 수락하도록 지정합니다. sign은 서명되고 암호화된 데이터만 수락하도록 지정합니다. none은 모든 데이터를 수락하도록 지정합니다. collectd_auth_file에 값을 지정하는 경우 가능하면 암호화된 데이터가 복호화됩니다.

  자세한 내용은 collectd Wiki의 클라이언트 설정 및 가능한 상호작용을 참조하십시오.
- collectd_auth_file 사용자 이름이 비밀번호에 매핑되는 파일을 설정합니다. 이 비밀번호는 서명을 확인하고 암호화된 네트워크 패킷을 복호화하는 데 사용합니다. 제공된 경우 서명된 데이터를 확인하고 암호화된 패킷을 복호화합니다. 그렇지 않은 경우 서명을 확인하지 않아도 서명된 데이터가 수락 완료되어 암호화된 데이터를 복호화할 수 없습니다.

  기본값은 `/etc/collectd/auth_file`입니다.

  collectd_security_level이 none으로 설정된 경우 이것은 선택 사항입니다. collectd_security_level을 `encrypt` 또는 sign으로 설정한 경우 collectd_auth_file을 지정해야 합니다.

  auth 파일 형식은 각 줄에 사용자 이름 다음에 콜론이 나오고 스페이스 다음에 비밀번호가 나옵니다. 예:

  ```
  user1: user1_password

  user2: user2_password
  ```
- collectd_typesdb: 데이터 세트 설명을 포함하는 1개 이상의 파일 목록. 이 목록은 목록에 항목이 1개만 있어도 괄호로 묶어야 합니다. 이 목록의 각 항목은 큰따옴표로 묶어야 합니다. 여러 항목이 있는 경우 각각 쉼표로 구분하십시오. 기본값은 `["/usr/share/collectd/types.db"]`입니다. 자세한 내용은 https://collectd.org/documentation/manpages/types.db.5.shtml 단원을 참조하십시오.
- metrics_aggregation_interval: CloudWatch에서 지표를 단일 데이터 포인트로 집계하는 간격(초). 기본값은 60초입니다. 범위는 0 ~ 172,000입니다. 이것을 0에 설정하면 collectd 지표를 집계할 수 없습니다.

다음은 에이전트 구성 파일의 collectd 섹션의 예입니다.

```
{
```

```
   "metrics":{
      "metrics_collected":{
         "collectd":{
            "name_prefix":"My_collectd_metrics_",
            "metrics_aggregation_interval":120
         }
      }
   }
}
```

CloudWatch 에이전트가 수집하는 지표

서버에 CloudWatch 에이전트를 설치하면 서버로부터 지표를 수집할 수 있습니다. Amazon EC2 인스턴스 및 온프레미스 서버 둘 다에 및 Linux나 Windows 서버 중 하나를 실행하는 서버에 에이전트를 설치할 수 있습니다. Amazon EC2 인스턴스에 에이전트를 설치하는 경우, 수집하는 지표가 Amazon EC2 인스턴스에서 기본적으로 활성화되어 있는 지표에 추가됩니다.

인스턴스에 CloudWatch 에이전트를 설치하는 방법에 대한 내용은 CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집 (p. 189) 단원을 참조하십시오.

Windows 서버 인스턴스에서 CloudWatch 에이전트가 수집하는 지표

Windows 서버를 실행하는 서버에 CloudWatch 에이전트를 설치하면 Windows 성능 모니터의 카운터와 연결된 지표를 수집할 수 있습니다. 이러한 카운터의 CloudWatch 지표 이름은 객체 이름과 카운터 이름 사이에 공백을 넣어 생성됩니다. 예를 들어, `Processor` 객체의 `% Interrupt Time` 카운터의 이름은 CloudWatch에서 `Processor % Interrupt Time`으로 지표 이름이 지정됩니다. Windows 성능 모니터 카운터에 대한 자세한 내용은 Microsoft Windows 서버 설명서를 참조하십시오.

CloudWatch 에이전트가 수집하는 지표의 기본 네임스페이스는 `CWAgent`이지만, 에이전트를 구성할 때 다른 네임스페이스를 지정할 수도 있습니다.

Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표

다음 표에는 Linux 인스턴스에서 CloudWatch 에이전트를 사용하여 수집할 수 있는 지표가 나열되어 있습니다.

지표	설명
`cpu_time_active`	어떠한 용량에서든 CPU가 활성화되어 있는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
`cpu_time_guest`	CPU가 게스트 운영 체제에 대해 가상 CPU를 실행하는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
`cpu_time_guest_nice`	CPU가 우선 순위가 낮으며, 다른 프로세스에 의해 중단될 수 있는 게스트 운영 체제에 대해 가상 CPU를 실행하는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음

지표	설명
cpu_time_idle	CPU가 유휴 상태인 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_time_iowait	CPU가 I/O 작업이 완료될 때까지 기다리는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_time_irq	CPU가 인터럽트를 제공하는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_time_nice	우선 순위가 높은 프로세스에 의해 쉽게 중단될 수 있는 우선 순위가 낮은 프로세스가 있는 사용자 모드에 CPU가 있는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_time_softirq	CPU가 소프트웨어 인터럽트를 제공하는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_time_steal	CPU가 도용 시간에 있는 시간입니다. 도용 시간은 가상화 환경에서 다른 운영 체제에서 사용된 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_time_system	CPU가 시스템 모드에 있는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_time_user	CPU가 사용자 모드에 있는 시간입니다. 이 지표는 수백 초 단위로 측정됩니다. 단위: 없음
cpu_usage_active	어떠한 용량에서든 CPU가 활성화되어 있는 시간(백분율)입니다. 단위: 백분율
cpu_usage_guest	CPU가 게스트 운영 체제에 대해 가상 CPU를 실행하는 시간 비율입니다. 단위: 백분율
cpu_usage_guest_nice	CPU가 우선 순위가 낮으며, 다른 프로세스에 의해 중단될 수 있는 게스트 운영 체제에 대해 가상 CPU를 실행하는 시간 비율입니다. 단위: 백분율

지표	설명
cpu_usage_idle	CPU가 유휴 상태인 시간 비율 단위: 백분율
cpu_usage_iowait	CPU가 I/O 작업이 완료될 때까지 기다리는 시간 비율입니다. 단위: 백분율
cpu_usage_irq	CPU가 인터럽트를 제공하는 시간 비율입니다. 단위: 백분율
cpu_usage_nice	우선 순위가 높은 프로세스에 의해 쉽게 중단될 수 있는 우선 순위가 낮은 프로세스가 있는 사용자 모드에 CPU가 있는 시간 비율입니다. 단위: 백분율
cpu_usage_softirq	CPU가 소프트웨어 인터럽트를 제공하는 시간 비율입니다. 단위: 백분율
cpu_usage_steal	CPU가 도용 시간에 있는 시간 또는 가상화 환경에서 다른 운영 체제에서 사용된 시간의 비율입니다. 단위: 백분율
cpu_usage_system	CPU가 시스템 모드에 있는 시간 비율입니다. 단위: 백분율
cpu_usage_user	CPU가 사용자 모드에 있는 시간 비율입니다. 단위: 백분율
disk_free	디스크의 사용 가능한 공간입니다. 단위: 바이트
disk_inodes_free	디스크에서 사용 가능한 인덱스 노드 수입니다. 단위: 수
disk_inodes_total	디스크에서 예약되어 있는 총 인덱스 노드 수입니다. 단위: 수
disk_inodes_used	디스크에서 사용된 인덱스 노드 수입니다. 단위: 수
disk_total	사용된 공간이나 사용 가능한 공간을 포함한 디스크의 총 공간입니다. 단위: 바이트

지표	설명
disk_used	디스크의 사용된 공간입니다. 단위: 바이트
disk_used_percent	사용된 총 디스크 공간 비율입니다. 단위: 백분율
diskio_iops_in_progress	디바이스 드라이버에 발행되었지만 아직 완료되지는 않은 I/O 요청 수입니다. 단위: 수
diskio_io_time	디스크가 I/O 요청이 대기하도록 한 시간입니다. 단위: 밀리초 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
diskio_reads	디스크 읽기 작업 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
diskio_read_bytes	디스크에서 읽어온 바이트 수입니다. 단위: 바이트 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
diskio_read_time	읽기 요청이 디스크에서 대기하고 있는 시간입니다. 동시에 여러 개의 읽기 요청이 대기하게 되면 숫자가 증가합니다. 예를 들어, 5개의 요청 모두가 평균 100밀리초 동안 대기하는 경우 500이 보고됩니다. 단위: 밀리초 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
diskio_writes	디스크 쓰기 작업 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
diskio_write_bytes	디스크에 기록한 바이트 수입니다. 단위: 바이트 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.

지표	설명
diskio_write_time	쓰기 요청이 디스크에서 대기하고 있는 시간입니다. 동시에 여러 개의 쓰기 요청이 대기하게 되면 숫자가 증가합니다. 예를 들어, 8개의 요청 모두가 평균 1000밀리초 동안 대기하는 경우 8000이 보고됩니다. 단위: 밀리초 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
mem_active	마지막 샘플 기간 동안 몇 가지 방식으로 사용된 메모리 양입니다. 단위: 바이트
mem_available	사용 가능하며 즉시 프로세스에 제공할 수 있는 메모리 양입니다. 단위: 바이트
mem_available_percent	사용 가능하며 즉시 프로세스에 제공할 수 있는 메모리 비율입니다. 단위: 백분율
mem_buffered	버퍼에 사용되고 있는 메모리 양입니다. 단위: 바이트
mem_cached	파일 캐시에 사용되고 있는 메모리 양입니다. 단위: 바이트
mem_free	사용되지 않는 메모리 양입니다. 단위: 바이트
mem_inactive	마지막 샘플 기간 동안 몇 가지 방식으로 사용되지 않은 메모리 양입니다. 단위: 바이트
mem_total	총 메모리 크기 단위: 바이트
mem_used	현재 사용 중인 메모리 양입니다. 단위: 바이트
mem_used_percent	현재 사용 중인 메모리 비율입니다. 단위: 백분율

지표	설명
net_bytes_recv	네트워크 인터페이스에서 받은 바이트 수입니다. 단위: 바이트 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
net_bytes_sent	네트워크 인터페이스가 보낸 바이트 수입니다. 단위: 바이트 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
net_drop_in	이 네트워크 인터페이스에서 받았지만 삭제된 패킷 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
net_drop_out	이 네트워크 인터페이스가 전송했지만 삭제된 패킷 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
net_err_in	이 네트워크 인터페이스가 탐지한 수신 오류 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
net_err_out	이 네트워크 인터페이스가 탐지한 전송 오류 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
net_packets_sent	이 네트워크 인터페이스가 보낸 패킷 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.
net_packets_recv	이 네트워크 인터페이스에서 받은 패킷 수입니다. 단위: 수 이 지표에는 Sum 통계만 사용해야 합니다. Average는 사용하지 마십시오.

지표	설명
netstat_tcp_close	상태가 없는 TCP 연결 수입니다. 단위: 수
netstat_tcp_close_wait	클라이언트로부터 종료 요청을 기다리고 있는 TCP 연결 수입니다. 단위: 수
netstat_tcp_closing	클라이언트로부터 승인과 함께 종료 요청을 기다리고 있는 TCP 연결 수입니다. 단위: 수
netstat_tcp_established	설정된 TCP 연결 수입니다. 단위: 수
netstat_tcp_fin_wait1	연결 종료 프로세스 중에 FIN_WAIT1 상태에 있는 TCP 연결 수입니다. 단위: 수
netstat_tcp_fin_wait2	연결 종료 프로세스 중에 FIN_WAIT2 상태에 있는 TCP 연결 수입니다. 단위: 수
netstat_tcp_last_ack	클라이언트가 연결 종료 메시지 승인을 보내길 기다리고 있는 TCP 연결 수입니다. 연결이 종료되기 직전의 마지막 단계입니다. 단위: 수
netstat_tcp_listen	현재 연결 요청을 수신하고 있는 TCP 포트 수입니다. 단위: 수
netstat_tcp_none	비활성 클라이언트가 있는 TCP 연결 수입니다. 단위: 수
netstat_tcp_syn_sent	연결 요청을 보낸 후 일치하는 연결 요청을 기다리고 있는 TCP 연결 수입니다. 단위: 수
netstat_tcp_syn_recv	연결 요청을 보내고 받은 후 연결 요청 승인을 기다리고 있는 TCP 연결 수입니다. 단위: 수
netstat_tcp_time_wait	클라이언트가 연결 종료 요청 승인을 받았는지 확인하기 위해 현재 기다리고 있는 TCP 연결 수입니다. 단위: 수

지표	설명
netstat_udp_socket	현재 UDP 연결 수입니다. 단위: 수
processes_blocked	차단된 프로세스 수입니다. 단위: 수
processes_dead	"데드(Dead)" 상태인 프로세스 수이며, Linux에서 x 상태 코드로 나타납니다. 단위: 수
processes_idle	유휴 상태(20초 이상 절전 모드)인 프로세스 수입니다. FreeBSD 인스턴스에서만 사용할 수 있습니다. 단위: 수
processes_paging	페이징되고 있는 프로세스 수이며, Linux에서 W 상태 코드로 나타납니다. 단위: 수
processes_running	실행 중인 프로세스 수이며, R 상태 코드로 나타납니다. 단위: 수
processes_sleeping	절전 모드의 프로세스 수이며, S 상태 코드로 나타납니다. 단위: 수
processes_stopped	중지된 프로세스 수이며, T 상태 코드로 나타납니다. 단위: 수
processes_total	인스턴스에 있는 총 프로세스 수입니다. 단위: 수
processes_total_threads	프로세스를 구성하는 총 스레드 수입니다. 이 지표는 Linux 인스턴스에서만 사용할 수 있습니다. 단위: 수
processes_wait	페이징되고 있는 프로세스 수이며, FreeBSD 인스턴스에서 W 상태 코드로 나타납니다. 이 지표는 FreeBSD 인스턴스에서만 사용할 수 있습니다. 단위: 수
processes_zombies	좀비 프로세스 수이며, Z 상태 코드로 나타납니다. 단위: 수
swap_free	사용되지 않는 스왑 공간 크기입니다. 단위: 바이트

지표	설명
swap_used	현재 사용 중인 스왑 공간 크기입니다. 단위: 바이트
swap_used_percent	현재 사용 중인 스왑 공간 비율입니다. 단위: 백분율

CloudWatch 에이전트를 사용하는 일반적인 시나리오

다음 단원에서는 CloudWatch 에이전트를 사용할 때의 몇 가지 일반적인 구성 및 사용자 지정 작업을 완료하는 방법을 설명합니다.

주제

- 다른 사용자로 CloudWatch 에이전트 실행 (p. 263)
- CloudWatch 에이전트에 의해 수집되는 지표에 사용자 지정 차원 추가 (p. 264)
- 여러 CloudWatch 에이전트 구성 파일 (p. 265)
- CloudWatch 에이전트에 의해 수집되는 지표 집계 또는 롤업 (p. 266)
- CloudWatch 에이전트로 고분해능 지표 수집 (p. 267)
- 다른 계정에 지표 및 로그 전송 (p. 268)
- 통합 CloudWatch 에이전트와 이전 CloudWatch Logs 에이전트 간 타임스탬프 차이 (p. 269)

다른 사용자로 CloudWatch 에이전트 실행

Linux 서버에서 CloudWatch는 기본적으로 루트 사용자로 실행됩니다. 다른 사용자로 에이전트를 실행하려면 CloudWatch 에이전트 구성 파일의 `agent` 섹션에 있는 `run_as_user` 파라미터를 사용합니다. 이 옵션은 Linux 서버에서만 사용할 수 있습니다.

이미 루트 사용자로 에이전트를 실행 중인데 다른 사용자를 사용하도록 변경하려면 다음 절차 중 하나를 사용하십시오.

Linux를 실행하는 EC2 인스턴스에서 다른 사용자로 CloudWatch 에이전트를 실행하려면

1. 새 CloudWatch 에이전트 패키지를 다운로드하여 설치합니다. 자세한 내용은 S3 다운로드 링크를 사용하여 CloudWatch 에이전트 패키지 다운로드 (p. 191) 단원을 참조하십시오.
2. 새 Linux 사용자를 생성하거나 RPM 또는 DEB 파일이 생성한 기본 사용자 `cwagent`를 사용합니다.
3. 다음 중 한 가지 방법으로 이 사용자에 대한 자격 증명을 제공합니다.
   - `/home/root/.aws/credentials` 파일이 있으면 CloudWatch 에이전트를 실행하는 데 사용할 사용자의 자격 증명 파일을 생성해야 합니다. 이 자격 증명 파일은 `/home/username/.aws/credentials`입니다. 그런 다음 `common-config.toml`에 있는 `shared_credential_file` 파라미터 값을 자격 증명 파일의 경로 이름으로 설정합니다. 자세한 내용은 (선택 사항) 프록시 또는 리전 정보에 대한 일반 구성 수정 (p. 199) 단원을 참조하십시오.
   - `/home/root/.aws/credentials` 파일이 없으면 다음 중 하나를 수행할 수 있습니다.
     - CloudWatch 에이전트를 실행하는 데 사용할 사용자의 자격 증명 파일을 생성합니다. 이 자격 증명 파일은 `/home/username/.aws/credentials`입니다. 그런 다음 `common-config.toml`에 있는 `shared_credential_file` 파라미터 값을 자격 증명 파일의 경로 이름으로 설정합니다. 자세한 내용은 (선택 사항) 프록시 또는 리전 정보에 대한 일반 구성 수정 (p. 199) 단원을 참조하십시오.

- 자격 증명 파일을 만드는 대신 인스턴스에 IAM 역할을 연결합니다. 에이전트는 이 역할을 자격 증명 공급자로 사용합니다.

4. CloudWatch 에이전트 구성 파일에서 `agent` 섹션에 다음 행을 추가합니다.

```
"run_as_user": "username"
```

   필요에 따라 구성 파일을 추가로 수정합니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.
5. 사용자에게 필요한 권한을 부여합니다. 사용자는 수집할 로그 파일에 대한 읽기(r) 권한이 있어야 하며 로그 파일 경로의 모든 디렉터리에 대해 실행(x) 권한이 있어야 합니다.
6. 방금 수정한 구성 파일로 에이전트를 시작합니다.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:configuration-file-path -s
```

Linux를 실행하는 온프레미스 서버에서 다른 사용자로 CloudWatch 에이전트를 실행하려면

1. 새 CloudWatch 에이전트 패키지를 다운로드하여 설치합니다. 자세한 내용은 S3 다운로드 링크를 사용하여 CloudWatch 에이전트 패키지 다운로드 (p. 191) 단원을 참조하십시오.
2. 새 Linux 사용자를 생성하거나 RPM 또는 DEB 파일이 생성한 기본 사용자 `cwagent`를 사용합니다.
3. 이 사용자의 자격 증명을 사용자가 액세스할 수 있는 경로(예: `/home/username/.aws/credentials`)에 저장합니다.
4. `common-config.toml`에 있는 `shared_credential_file` 파라미터 값을 자격 증명 파일의 경로 이름으로 설정합니다. 자세한 내용은 (선택 사항) 프록시 또는 리전 정보에 대한 일반 구성 수정 (p. 199) 단원을 참조하십시오.
5. CloudWatch 에이전트 구성 파일에서 `agent` 섹션에 다음 행을 추가합니다.

```
"run_as_user": "username"
```

   필요에 따라 구성 파일을 추가로 수정합니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.
6. 사용자에게 필요한 권한을 부여합니다. 사용자는 수집할 로그 파일에 대한 읽기(r) 권한이 있어야 하며 로그 파일 경로의 모든 디렉터리에 대해 실행(x) 권한이 있어야 합니다.
7. 방금 수정한 구성 파일로 에이전트를 시작합니다.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:configuration-file-path -s
```

CloudWatch 에이전트에 의해 수집되는 지표에 사용자 지정 차원 추가

에이전트에 의해 수집되는 지표에 태그와 같은 사용자 지정 차원을 추가하려면 해당 지표를 나열하는 에이전트 구성 파일의 섹션에 `append_dimensions` 필드를 추가하십시오.

예를 들어, 구성 파일의 다음 예제 섹션에서는 `Prod`의 값이 포함된 `stackName`라는 사용자 지정 차원을 에이전트에 의해 수집된 `cpu` 및 `disk` 지표에 추가합니다.

```
"cpu":{
  "resources":[
    "*"
```

```
  ],
  "measurement":[
    "cpu_usage_guest",
    "cpu_usage_nice",
    "cpu_usage_idle"
  ],
  "totalcpu":false,
  "append_dimensions":{
    "stackName":"Prod"
  }
},
"disk":{
  "resources":[
    "/",
    "/tmp"
  ],
  "measurement":[
    "total",
    "used"
  ],
  "append_dimensions":{
    "stackName":"Prod"
  }
}
```

에이전트 구성 파일을 변경할 때마다 에이전트를 다시 시작하여 변경 사항을 적용해야 합니다.

여러 CloudWatch 에이전트 구성 파일

여러 개의 구성 파일을 사용하도록 CloudWatch 에이전트를 설정할 수 있습니다. 예를 들어 인프라의 모든 서버에서 항상 수집하려는 일련의 지표와 로그를 수집하는 공통 구성 파일을 사용할 수 있습니다. 그런 다음 특정 애플리케이션이나 특정 상황에서 지표를 수집하는 추가 구성 파일을 사용할 수 있습니다.

이렇게 설정하려면 먼저 사용하려는 구성 파일을 생성합니다. 동일한 서버에서 함께 사용할 구성 파일은 파일 이름이 서로 달라야 합니다. 구성 파일을 서버나 Parameter Store에 저장할 수 있습니다.

`fetch-config` 옵션을 사용하여 CloudWatch 에이전트를 시작하고 첫 번째 구성 파일을 지정합니다. 실행 중인 에이전트에 두 번째 구성 파일을 추가하려면, 동일한 명령에 `append-config` 옵션을 사용합니다. 구성 파일에 나열된 모든 지표와 로그가 수집됩니다. 다음 Linux 명령 예제는 파일로 저장된 구성을 사용하는 이 시나리오를 보여줍니다. 첫 행은 `infrastructure.json` 구성 파일을 사용하여 에이전트를 시작하고, 둘째 행은 `app.json` 구성 파일을 추가합니다.

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -
c file:/tmp/infrastructure.json -s
```

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a append-config -m ec2 -
c file:/tmp/app.json -s
```

다음 예제 구성 파일은 이 기능의 사용을 보여줍니다. 첫 번째 구성 파일은 인프라의 모든 서버에 사용되며, 두 번째 구성 파일은 특정 애플리케이션의 로그만 수집하며, 해당 애플리케이션을 실행하는 서버에 추가됩니다.

infrastructure.json

```
{
  "metrics": {
    "metrics_collected": {
      "cpu": {
        "resources": [
          "*"
```

```
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true
      },
      "mem": {
         "measurement": [
           "used_percent"
        ]
      }
}
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
            "log_group_name": "amazon-cloudwatch-agent.log"
          },
          {
            "file_path": "/var/log/messages",
            "log_group_name": "/var/log/messages"
          }
        ]
      }
    }
  }
}
```

app.json

```
{
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/app/app.log*",
            "log_group_name": "/app/app.log"
          }
        ]
      }
    }
  }
```

구성에 추가된 모든 구성 파일은 파일 이름이 서로 달라야 하며 초기 구성 파일의 이름과도 달라야 합니다. 파일 이름이 동일한 구성 파일이 있는 `append-config`를 에이전트에서 이미 사용 중인 구성 파일로 사용할 경우 추가 명령은 첫 번째 구성 파일에 추가되는 것이 아니라 해당 파일의 정보를 덮어씁니다. 파일 이름이 동일한 두 개의 구성 파일이 서로 다른 파일 경로에 있는 경우에도 마찬가지입니다.

앞의 예는 두 개의 구성 파일을 사용하는 것을 보여주지만, 에이전트 구성에 추가할 수 있는 구성 파일의 수는 제한되지 않습니다. 서버에 있는 구성 파일과 Parameter Store에 있는 구성을 함께 사용할 수도 있습니다.

CloudWatch 에이전트에 의해 수집되는 지표 집계 또는 롤업

에이전트에 의해 수집되는 지표를 집계하거나 롤업하려면 에이전트 구성 파일의 해당 지표에 대한 섹션에 `aggregation_dimensions` 필드를 추가하십시오.

예를 들어, 다음 구성 파일 조각은 `AutoScalingGroupName` 차원에서 지표를 롤업합니다. 각 Auto Scaling 그룹의 모든 인스턴스의 지표가 집계되고 전체적으로 표시될 수 있습니다.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"]]
}
```

Auto Scaling 그룹 이름에서 롤업할 뿐 아니라 `InstanceId` 및 `InstanceType` 차원 각각의 조합을 따라서도 롤업하려는 경우 다음을 추가합니다.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId", "InstanceType"]]
}
```

대신 지표를 하나의 모음에 롤업하려면 `[]`를 사용합니다.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [[]]
}
```

에이전트 구성 파일을 변경할 때마다 에이전트를 다시 시작하여 변경 사항을 적용해야 합니다.

CloudWatch 에이전트로 고분해능 지표 수집

`metrics_collection_interval` 필드는 수집되는 지표의 시간 간격을 초 단위로 지정합니다. 이 필드에 60 미만의 값을 지정하면 지표가 고분해능 지표로 수집됩니다.

예를 들어, 모든 지표가 고해상도 지표이며 10초마다 수집되어야 하는 경우 `agent` 섹션의 `metrics_collection_interval`에서 글로벌 지표 수집 간격 값으로 10을 지정합니다.

```
"agent": {
  "metrics_collection_interval": 10
}
```

또는 다음 예에서는 `cpu` 지표는 1초마다 수집되도록 설정하고 다른 모든 지표는 1분마다 수집되도록 설정합니다.

```
"agent":{
  "metrics_collection_interval": 60
},
"metrics":{
  "metrics_collected":{
    "cpu":{
      "resources":[
        "*"
      ],
      "measurement":[
        "cpu_usage_guest"
      ],
      "totalcpu":false,
      "metrics_collection_interval": 1
```

```
    },
    "disk":{
      "resources":[
        "/",
        "/tmp"
      ],
      "measurement":[
        "total",
        "used"
      ]
    }
  }
}
```

에이전트 구성 파일을 변경할 때마다 에이전트를 다시 시작하여 변경 사항을 적용해야 합니다.

다른 계정에 지표 및 로그 전송

CloudWatch 에이전트가 지표와 로그 중 하나 또는 둘 다 다른 계정에 전송하도록 하려면 전송 서버의 에이전트 구성 파일에 `role_arn` 파라미터를 지정하십시오. `role_arn` 값은 데이터를 대상 계정에 전송할 때 에이전트가 사용하는 대상 계정의 IAM 역할을 지정합니다. 지표 또는 로그를 대상 계정에 전달할 때는 이 역할을 통해 전송 계정이 대상 계정의 해당 역할을 맡을 수 있습니다.

또한 에이전트 구성 파일에 별도의 `role_arn` 문자열 두 개를 지정할 수 있는데, 하나는 지표를 보낼 때 사용할 문자열이고 다른 하나는 로그를 보낼 때 사용할 문자열입니다.

구성 파일의 `agent` 섹션 부분에 대한 다음의 예에서는 지표와 로그를 다른 계정에 보낼 때 `CrossAccountAgentRole`을 사용하도록 에이전트를 설정합니다.

```
{
  "agent": {
    "credentials": {
      "role_arn": "arn:aws:iam::123456789012:role/CrossAccountAgentRole"
    }
  },
  .....
}
```

또는 다음의 예에서는 지표와 로그를 전송하기 위해 사용할 전송 계정에 다른 역할을 설정합니다.

```
"metrics": {
    "credentials": {
     "role_arn": "RoleToSendMetrics"
    },
    "metrics_collected": {....
```

```
"logs": {
    "credentials": {
    "role_arn": "RoleToSendLogs"
    },
    ....
```

필요한 정책

에이전트 구성 파일에 `role_arn`를 지정할 때는 전송 및 대상 계정의 IAM 역할에 특정 정책도 포함되는지 확인해야 합니다. 전송 계정과 대상 계정의 역할에는 모두 `CloudWatchAgentServerPolicy`가 있어야 합

니다. 이 정책을 역할에 부여하는 방법에 대한 자세한 내용은 Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 (p. 202) 단원을 참조하십시오.

전송 계정의 역할은 다음의 정책을 포함해야 합니다. 역할을 편집할 때 IAM 콘솔의 권한 탭에 이 정책을 추가합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::target-account-ID:role/agent-role-in-target-account"
            ]
        }
    ]
}
```

대상 계정의 역할에는 전송 계정에서 사용하는 IAM 역할을 인식하도록 다음 정책이 포함되어야 합니다. 역할을 편집할 때 IAM 콘솔의 신뢰 관계 탭에 이 정책을 추가합니다. 이 정책을 추가하는 대상 계정의 역할은 CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 (p. 194)에서 생성한 역할입니다. 이 역할은 전송 계정에서 사용한 정책의 *agent-role-in-target-account*에 지정된 역할입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::sending-account-ID:role/role-specified-in-role_arn"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

통합 CloudWatch 에이전트와 이전 CloudWatch Logs 에이전트 간 타임스탬프 차이

CloudWatch 에이전트는 이전 CloudWatch Logs 에이전트와 비교할 때 타임스탬프 형식에 대한 다른 기호 집합을 지원합니다. 이러한 차이는 다음 표에 표시됩니다.

두 에이전트에서 지원하는 기호	통합 CloudWatch 에이전트에서만 지원하는 기호	기존 CloudWatch Logs 에이전트에서만 지원하는 기호
%A, %a, %b, %B, %d, %f, %H, %I, %m, %M, %p, %S, %y, %Y, %Z, %z	%-d, %-I, %-m, %-M, %-S	%c,%j, %U, %W, %w

새 CloudWatch 에이전트에서 지원하는 기호의 의미에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트 구성 파일: 로그 섹션을 참조하십시오. CloudWatch Logs 에이전트에서 지원되

는 기호에 대한 자세한 내용은 Amazon CloudWatch Logs User Guide의 에이전트 구성 파일을 참조하십시오.

CloudWatch 에이전트의 문제 해결

다음 정보를 사용하면 CloudWatch 에이전트의 문제를 해결하는 데 도움이 됩니다.

주제

- CloudWatch 에이전트 명령줄 파라미터 (p. 270)
- Run Command를 사용한 CloudWatch 에이전트 설치 실패 (p. 270)
- CloudWatch 에이전트가 시작되지 않음 (p. 270)
- CloudWatch 에이전트가 실행 중인지 확인 (p. 271)
- 지표가 저장되는 위치 (p. 271)
- CloudWatch 에이전트가 시작되지 않고 오류에 Amazon EC2 리전이 언급되어 있음 (p. 272)
- Windows Server에서 자격 증명을 찾을 수 없음 (p. 272)
- CloudWatch 에이전트 파일 및 위치 (p. 272)
- CloudWatch 에이전트에 의해 생성되는 로그 (p. 273)
- CloudWatch 에이전트 중지 및 다시 시작 (p. 273)

CloudWatch 에이전트 명령줄 파라미터

CloudWatch에서 지원되는 전체 파라미터 목록을 보려면 설치되어 있는 컴퓨터에서 명령줄에 다음과 같이 입력합니다.

```
amazon-cloudwatch-agent-ctl -help
```

Run Command를 사용한 CloudWatch 에이전트 설치 실패

시스템 관리자 Run Command를 사용하여 CloudWatch 에이전트를 설치하려면 대상 서버의 SSM Agent가 2.2.93.0 이상 버전이어야 합니다. SSM Agent가 올바른 버전이 아닌 경우 다음 메시지를 포함한 오류가 표시될 수 있습니다.

```
no latest version found for package AmazonCloudWatchAgent on platform linux
```

```
failed to download installation package reliably
```

SSM Agent 버전 업데이트에 대한 자세한 내용은 AWS 시스템 관리자 사용 설명서의 SSM Agent 설치 및 구성을 참조하십시오.

CloudWatch 에이전트가 시작되지 않음

CloudWatch 에이전트가 시작되지 않는 경우 구성에 문제가 있을 수 있습니다. 구성 정보는 `configuration-validation.log` 파일에 기록됩니다. 이 파일은 Linux 서버의 `/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` 및 Windows Server가 실행되는 서버의 `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`에 있습니다.

CloudWatch 에이전트가 실행 중인지 확인

CloudWatch 에이전트를 쿼리하여 해당 에이전트가 실행 중인지 아니면 중지되었는지를 확인할 수 있습니다. AWS 시스템 관리자를 사용하면 이 작업을 원격으로 수행할 수 있습니다. 명령줄을 사용할 수도 있지만, 로컬 서버를 확인하기 위해서만 사용해야 합니다.

Run Command을 사용하여 CloudWatch 에이전트의 상태를 쿼리하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose Run Command.

   -or-

   If the AWS 시스템 관리자 home page opens, scroll down and choose Explore Run Command.
3. Run 명령을 선택합니다.
4. 명령 문서 목록에서 AmazonCloudWatch-ManageAgent를 선택합니다.
5. 대상 영역에서 확인할 인스턴스를 선택합니다.
6. 작업 목록에서 상태를 선택합니다.
7. Optional Configuration Source(구성 소스(선택 사항)) 및 Optional Configuration Location(구성 위치(선택 사항))을 비워둡니다.
8. 실행을 선택합니다.

에이전트가 실행 중인 경우 결과가 다음과 유사하게 표시됩니다.

```
{
       "status": "running",
       "starttime": "2017-12-12T18:41:18",
       "version": "1.73.4"
}
```

에이전트가 중지된 경우 `"status"` 필드에 `"stopped"`가 표시됩니다.

명령줄을 사용하여 CloudWatch 에이전트의 상태를 로컬에서 쿼리하려면

- Linux 서버의 경우 다음을 입력합니다.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status
  ```

  Windows Server가 실행되는 서버의 경우 관리자로서 PowerShell에 다음을 입력합니다.

  ```
  & $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2
   -a status
  ```

지표가 저장되는 위치

CloudWatch 에이전트가 실행 중이지만 AWS Management 콘솔 또는 AWS CLI에서 이 에이전트에 의해 수집된 지표를 찾을 수 없는 경우 올바른 네임스페이스를 사용하고 있는지 확인하십시오. 기본적으로 에이전트에 의해 수집되는 지표의 네임스페이스는 `CWAgent`입니다. 에이전트 구성 파일의 `metrics` 섹션에 있는 `namespace` 필드를 사용하여 이 네임스페이스를 사용자 지정할 수 있습니다. 원하는 지표가 표시되지 않는 경우 구성 파일을 점검하여 사용 중인 네임스페이스를 확인하십시오.

CloudWatch 에이전트 패키지를 처음 다운로드하는 경우 에이전트 구성 파일은 `amazon-cloudwatch-agent.json`입니다. 이 파일은 구성 마법사를 실행한 디렉터리에 있습니다. 그렇지 않으면 이 파일을 다

른 디렉터리로 옮긴 것일 수 있습니다. 구성 마법사를 사용하는 경우, 마법사의 에이전트 구성 파일 출력이 `config.json`으로 명명됩니다. `namespace` 필드를 포함한 구성 파일에 대한 자세한 내용은 CloudWatch 에이전트 구성 파일: 지표 섹션 (p. 228) 단원을 참조하십시오.

CloudWatch 에이전트가 시작되지 않고 오류에 Amazon EC2 리전이 언급되어 있음

에이전트가 시작되지 않고 오류 메시지에 Amazon EC2 리전 엔드포인트가 언급되어 있으면 액세스 권한을 부여하지 않고 Amazon EC2 엔드포인트에 액세스해야 하도록 에이전트를 구성했을 수 있습니다.

예를 들어, 에이전트 구성 파일에 Amazon EC2 메타데이터에 따라 좌우되는 `append_dimensions` 파라미터의 값을 지정하고 프록시를 사용할 경우 서버에서 Amazon EC2의 엔드포인트에 액세스할 수 있어야 합니다. 이러한 엔드포인트에 대한 자세한 내용은 Amazon Web Services 일반 참조의 Amazon Elastic Compute Cloud(Amazon EC2)를 참조하십시오.

Windows Server에서 자격 증명을 찾을 수 없음

Windows Server에서 자격 증명이 `$SystemDrive\\Users\\Administrator\\.aws`에 있지 않거나 (Windows Server 2008 R2 또는 Windows Server 2012의 경우) "`$SystemDrive\\Documents and Settings\\Administrator\\.aws`에 있지 않으면(Windows Server 2003의 경우) `common.toml`에서 `shared_credential_file` 옵션을 사용하여 자격 증명 경로를 직접 지정할 수 있습니다.

자격 증명 파일이 없는 경우 만들어야 합니다. 자세한 내용은 (선택 사항) 프록시 또는 리전 정보에 대한 일반 구성 수정 (p. 199) 단원을 참조하십시오.

CloudWatch 에이전트 파일 및 위치

다음 표에는 CloudWatch 에이전트가 설치하고 사용한 파일이 Linux 또는 Windows Server를 실행하는 서버에서 해당 에이전트의 위치와 함께 나열되어 있습니다.

File	Linux 위치	Windows Server 위치
에이전트의 시작, 중지 및 재시작을 제어하는 제어 스크립트입니다.	`/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl` 또는 `/usr/bin/amazon-cloudwatch-agent-ctl`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1`
에이전트가 작성하는 로그 파일입니다. AWS Support에 문의할 경우 이 파일을 첨부해야 할 수도 있습니다.	`/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` 또는 `/var/log/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`
에이전트 구성 검증 파일입니다.	`/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` 또는 `/var/log/amazon/amazon-cloudwatch-agent/configuration-validation.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`

File	Linux 위치	Windows Server 위치
마법사가 에이전트를 생성한 직후에 에이전트를 구성할 때 사용되는 JSON 파일입니다. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.	`/opt/aws/amazon-cloudwatch-agent/bin/config.json`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\config.json`
이 구성 파일을 Parameter Store에서 다운로드한 경우 에이전트를 구성하는 데 사용되는 JSON 파일입니다.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` 또는 `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`
에이전트가 사용할 리전 및 자격 증명 정보를 지정하는 데 사용되는 TOML 파일이며 시스템 기본값은 무시됩니다.	`/opt/aws/amazon-cloudwatch-agent/etc/common-config.toml` 또는 `/etc/amazon/amazon-cloudwatch-agent/common-config.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\common-config.toml`

CloudWatch 에이전트에 의해 생성되는 로그

에이전트는 실행 중에 로그를 생성합니다. 이 로그에는 문제 해결 정보가 들어 있습니다. 이 로그는 `amazon-cloudwatch-agent.log` 파일입니다. 이 파일은 Linux 서버의 `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` 및 Windows Server가 실행되는 서버의 `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`에 있습니다.

`amazon-cloudwatch-agent.log` 파일에 추가 세부 정보를 기록하도록 에이전트를 구성할 수 있습니다. 에이전트 구성 파일의 `agent` 섹션에서 `debug` 필드를 `true`로 설정한 후 CloudWatch 에이전트를 다시 구성하고 다시 시작합니다. 이 추가 정보 기록을 비활성화하려면 `debug` 필드를 `false`로 설정하고 에이전트를 다시 구성하고 다시 시작합니다. 자세한 내용은 CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 (p. 227) 단원을 참조하십시오.

CloudWatch 에이전트 중지 및 다시 시작

AWS 시스템 관리자 또는 명령줄을 사용하여 CloudWatch 에이전트를 수동으로 중지할 수 있습니다.

Run Command를 사용하여 CloudWatch 에이전트를 중지하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, choose Run Command.

   -or-

   If the AWS 시스템 관리자 home page opens, scroll down and choose Explore Run Command.
3. Run 명령을 선택합니다.
4. 명령 문서 목록에서 AmazonCloudWatch-ManageAgent를 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치한 인스턴스를 선택합니다.
6. 작업 목록에서 중지를 선택합니다.

7. Optional Configuration Source(구성 소스(선택 사항)) 및 Optional Configuration Location(구성 위치(선택 사항))을 비워둡니다.
8. 실행을 선택합니다.

명령줄을 사용하여 CloudWatch 에이전트를 로컬에서 중지하려면

- Linux 서버의 경우 다음을 입력합니다.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a stop
```

  Windows Server가 실행되는 서버의 경우 관리자로서 PowerShell에 다음을 입력합니다.

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2
 -a stop
```

에이전트를 다시 시작하려면 CloudWatch 에이전트 시작 (p. 208) 섹션의 지침을 따르십시오.

Amazon CloudWatch Application Insights for .NET and SQL Server

Amazon CloudWatch Application Insights for .NET and SQL Server는 .NET 및 SQL Server 애플리케이션에 대한 관찰을 용이하게 합니다. 애플리케이션 리소스에 대한 최상의 모니터를 설정하고 데이터를 분석하여 애플리케이션 문제의 징후가 있는지 지속적으로 확인할 수 있습니다. Sagemaker 및 기타 AWS 기술을 기반으로 하는 Application Insights는 모니터링되는 애플리케이션의 잠재적인 문제를 보여주는 자동화된 대시보드를 제공하므로 애플리케이션 및 인프라와 관련된 지속적인 문제를 신속하게 격리할 수 있습니다. Application Insights가 제공하는 애플리케이션 상태에 대한 향상된 가시성을 통해 MTTR(Mean Time to Repair)을 줄일 수 있으므로 여러 팀과 전문가의 도움 없이도 애플리케이션 문제를 해결할 수 있습니다.

애플리케이션을 Amazon CloudWatch Application Insights for .NET and SQL Server에 추가하면 애플리케이션의 리소스를 검색하고 CloudWatch에서 애플리케이션 구성 요소에 대한 지표 및 로그를 권장 및 구성합니다. 애플리케이션 구성 요소에는 SQL Server 백엔드 데이터베이스와 Microsoft IIS/웹 티어가 포함될 수 있습니다. Application Insights는 기록 데이터를 사용해 지표 패턴을 분석하여 이상을 탐지하고 .NET, SQL Server 오류 로그 및 IIS를 비롯한 애플리케이션 로그에서 오류 및 예외를 지속적으로 감지합니다. 이 알고리즘은 분류 알고리즘과 기본 규칙을 조합하여 이러한 관찰 결과를 연결합니다. 그런 다음 관련 관찰 및 문제 심각도 정보를 표시하는 대시보드를 자동으로 생성하여 작업 우선 순위를 지정합니다. 애플리케이션 지연 시간, SQL Server 백업 실패, 메모리 누수, 대용량 HTTP 요청 및 중단된 I/O 작업과 같은 일반적인 .NET 및 SQL 애플리케이션 스택 문제의 경우 가능한 근본 원인 및 해결 단계를 가리키는 추가 통찰력을 제공합니다. AWS Systems Manager OpsCenter와의 기본 통합 기능을 통해 관련 Systems Manager Automation 문서를 실행하여 문제를 해결할 수 있습니다.

주제

- Amazon CloudWatch Application Insights for .NET and SQL Server란 무엇입니까? (p. 275)
- Amazon CloudWatch Application Insights for .NET and SQL Server 시작하기 (p. 278)
- Amazon CloudWatch Application Insights for .NET and SQL Server에 의해 감지된 문제 보기 및 해결 (p. 289)
- Amazon CloudWatch Application Insights for .NET and SQL Server 작동 방식 (p. 292)
- Amazon CloudWatch Application Insights for .NET and SQL Server에서 지원되는 로그 및 지표 (p. 300)

Amazon CloudWatch Application Insights for .NET and SQL Server란 무엇입니까?

CloudWatch Application Insights for .NET and SQL Server를 사용하면 .NET 및 SQL Server 애플리케이션을 모니터링할 수 있습니다. 이 기능은 애플리케이션 리소스 및 기술 스택(예: 데이터베이스, 웹(IIS) 및 애플리케이션 서버, OS, 로드 밸런서, 대기열 등) 전반에서 주요 지표 및 로그를 파악하고 설정합니다. 지표 및 로그를 지속적으로 모니터링하여 이상 및 오류를 감지하고 연결합니다. 오류 및 이상이 감지되면 Application Insights는 CloudWatch 이벤트를 생성합니다. 문제 해결을 돕기 위해 감지한 문제에 대한 자동 대시보드를 생성함으로써 상관관계가 있는 지표 이상 항목 및 로그 오류와 함께 잠재적인 근본 원인을 알려주는 추가 통찰력을 제공합니다. 자동화된 대시보드를 사용하면 애플리케이션의 상태를 정상으로 유지하고 애플리케이션의 최종 사용자에게 미치는 영향을 방지하기 위해 신속한 조치를 취할 수 있습니다. 또한 Systems Manager OpsCenter를 사용하여 문제를 해결할 수 있도록 OpsItem을 생성합니다.

애플리케이션 리소스 모니터 자동 설정

CloudWatch Application Insights for .NET and SQL Server는 애플리케이션에 대한 모니터링을 설정하는 데 걸리는 시간을 줄여줍니다. 그 방법은 애플리케이션 리소스를 검색하고, 권장되는 지표 및 로그에 대한 사용

자 지정 목록을 제공하고, 이를 CloudWatch에서 설정하여 Amazon EC2, 탄력적 로드 밸런서(ELB)와 같은 애플리케이션 리소스에 필요한 가시성을 제공하는 것입니다. 또한 모니터링된 지표에 대해 동적 경보를 설정합니다. 경보는 지난 2주 동안 감지된 이상을 기반으로 자동 업데이트됩니다.

문제 감지 및 알림

CloudWatch Application Insights for .NET and SQL Server는 지표 이상 및 로그 오류와 같은 애플리케이션의 잠재적 문제의 징후를 감지합니다. 이 관찰 결과를 통해 애플리케이션의 잠재적 문제를 표면화할 수 있습니다. 그런 다음 알림을 받거나 작업을 수행하도록 구성할 수 있는 (p. 291) CloudWatch 이벤트를 생성합니다. 따라서 지표나 로그 오류에 대한 개별 경보를 생성할 필요가 없습니다.

신속한 문제 해결

CloudWatch Application Insights for .NET and SQL Server는 감지된 문제에 대해 CloudWatch 자동 대시보드를 생성합니다. 대시보드에는 문제 해결에 도움이 되는 관련 지표 이상 및 로그 오류를 포함하여 문제에 대한 세부 정보가 표시됩니다. 또한 이상 및 오류의 잠재적 근본 원인을 파악하는 추가 통찰력을 제공합니다.

콘텐츠

- 개념 (p. 276)
- 요금 (p. 276)
- 관련 AWS 제품 (p. 276)
- 지원되는 애플리케이션 구성 요소 (p. 277)
- 지원되는 기술 스택 (p. 278)

개념

다음은 Application Insights가 애플리케이션을 모니터링하는 방법을 이해하는 데 있어 중요한 개념입니다.

구성 요소

애플리케이션을 구성하는 유사한 리소스의 자동 그룹형, 독립 실행형 또는 사용자 지정 그룹. 모니터링 기능을 향상시키기 위해서는 유사한 리소스를 사용자 지정 구성 요소로 그룹화하는 것이 좋습니다.

관찰

애플리케이션 또는 애플리케이션 리소스에서 감지된 개별 이벤트(지표 이상, 로그 오류 또는 예외)

문제

문제는 관련 관찰을 연결, 분류 및 그룹화하여 감지됩니다.

CloudWatch Application Insights for .NET and SQL Server의 다른 주요 개념에 대한 정의는 Amazon CloudWatch 개념을 참조하십시오.

요금

CloudWatch Application Insights for .NET and SQL Server는 감지된 문제에 대한 알림을 위해 CloudWatch 지표, CloudWatch Logs 및 CloudWatch 이벤트를 사용하여 선택한 애플리케이션 리소스에 대한 권장 지표 및 로그를 설정합니다. 이러한 기능은 CloudWatch 요금에 따라 AWS 계정으로 청구됩니다. 감지된 문제에 대해 CloudWatch 이벤트 및 자동 대시보드를 생성합니다. 설정 지원, 데이터 분석 모니터링 또는 문제 감지에 대해서는 비용이 청구되지 않습니다.

관련 AWS 제품

다음 서비스가 CloudWatch Application Insights for .NET and SQL Server와 함께 사용됩니다.

- Amazon CloudWatch를 사용하여 시스템 전체의 리소스 사용률, 애플리케이션 성능 및 운영 상태를 파악할 수 있습니다. 또한 지표를 수집 및 추적하고, 경보 알림을 전송하며, 정의한 규칙에 따라 모니터링 중인 리소스를 자동으로 업데이트하며, 고유한 사용자 지정 지표를 모니터링할 수 있습니다. CloudWatch Application Insights for .NET and SQL Server는 특히 CloudWatch 기본 운영 대시보드 내에서 CloudWatch를 통해 시작됩니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.
- AWS 리소스 그룹을 사용하면 애플리케이션을 구성하는 리소스를 구성할 수 있습니다. 리소스 그룹을 사용하여 많은 리소스에 대한 작업을 한 번에 관리하고 자동화할 수 있습니다. 단일 애플리케이션에 대해 하나의 리소스 그룹만 등록할 수 있습니다. 자세한 내용은 AWS 리소스 그룹 사용 설명서를 참조하십시오.
- IAM은 사용자를 위해 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있는 웹 서비스입니다. IAM을 사용하여 AWS 리소스를 사용할 수 있는 사람을 제어(인증)하고 해당 사용자가 사용할 수 있는 리소스 및 사용 방법을 제어(권한 부여)할 수 있습니다. 자세한 내용은 Amazon CloudWatch에 대한 인증 및 액세스 제어를 참조하십시오.
- Amazon EC2는 AWS 클라우드에서 확장 가능한 컴퓨팅 파워를 제공합니다. Amazon EC2를 통해 원하는 만큼 가상 서버를 구축하고 보안 및 네트워크 구성과 스토리지 관리가 가능합니다. 요건이나 갑작스러운 인기 증대 등 변동 사항에 따라 확장하거나 축소할 수 있어 트래픽 예측 필요성이 줄어듭니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서 또는 Windows 인스턴스용 Amazon EC2 설명서를 참조하십시오.
- Elastic Load Balancing은 여러 가용 영역에서 EC2 인스턴스, 컨테이너 및 IP 주소 같은 여러 대상에 수신 애플리케이션 또는 네트워크 트래픽을 분산합니다. 자세한 내용은 Elastic Load Balancing 사용 설명서를 참조하십시오.
- Amazon EC2 Auto Scaling을 사용하면 애플리케이션의 로드를 처리할 수 있는 정확한 수의 EC2 인스턴스를 유지할 수 있습니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서를 참조하십시오.
- Amazon SQS는 내구력 있고 가용성이 뛰어난 보안 호스팅 대기열을 제공하며 이를 통해 분산 소프트웨어 시스템과 구성 요소를 통합 및 분리할 수 있습니다. 자세한 내용은 Amazon SQS 사용 설명서를 참조하십시오.
- AWS Systems Manager OpsCenter는 각 OpsItem, 관련 OpsItem 및 관련 리소스에 대한 상황별 조사 데이터를 제공하면서 서비스 전반에 걸쳐 OpsItem을 집계하고 표준화합니다. OpsCenter는 문제를 신속하게 해결하는 데 사용할 수 있는 Systems Manager Automation 문서(실행서)도 제공합니다. 각 OpsItem에 대해 검색 가능한 사용자 지정 데이터를 지정할 수 있습니다. 상태 및 소스별로 OpsItem에 대한 자동 생성 요약 보고서를 볼 수도 있습니다. 자세한 내용은 AWS 시스템 관리자 사용 설명서를 참조하십시오.

지원되는 애플리케이션 구성 요소

CloudWatch Application Insights for .NET and SQL Server는 리소스 그룹(이 경우 .NET 애플리케이션에서 사용하는 리소스 그룹)을 검색하여 애플리케이션 구성 요소를 식별합니다. 구성 요소는 독립 실행형, 자동 그룹형(예: Auto Scaling 그룹의 인스턴스 또는 로드 밸런서 뒤의 인스턴스) 또는 사용자 지정(개별 EC2 인스턴스 그룹화) 형식일 수 있습니다. CloudWatch Application Insights for .NET and SQL Server에서 지원하는 구성 요소는 다음과 같습니다.

- Amazon EC2
- Amazon RDS
- Elastic Load Balancing: Application Load Balancer 및 Classic Load Balancer(이러한 로드 밸런서의 모든 대상 인스턴스가 식별되고 구성됨)
- Amazon EC2 Auto Scaling 그룹: AWS Auto Scaling(Auto Scaling 그룹은 모든 대상 인스턴스에 대해 동적으로 구성되며 애플리케이션이 확장되면 CloudWatch Application Insights for .NET and SQL Server에서 자동으로 새 인스턴스 구성) Auto Scaling 그룹은 CloudFormation 기반 리소스 그룹에서 지원되지 않습니다.
- Amazon Simple Queue Service(Amazon SQS)

  다른 구성 요소 유형의 리소스는 .NET 및 SQL Server용 CloudWatch Application Insights에서 무시됩니다. 지원되는 구성 요소 유형이 Application Insights 애플리케이션에 나타나지 않으면 해당 구성 요소가 이미 Application Insights에서 모니터링 중인 다른 애플리케이션에서 등록되고 관리되는 것일 수 있습니다.

지원되는 기술 스택

CloudWatch Application Insights for .NET and SQL Server는 다음을 지원합니다.

- 프런트 엔드: Microsoft 인터넷 정보 서비스(IIS) 웹 서버
- 작업자 티어: .NET Framework
- 데이터베이스: RDS 또는 EC2에서 실행 중인 Microsoft SQL Server

Amazon CloudWatch Application Insights for .NET and SQL Server 시작하기

CloudWatch Application Insights for .NET and SQL Server를 시작하려면 아래에 설명된 사전 조건을 충족하고 IAM 정책을 생성했는지 확인합니다. 그런 다음 콘솔 링크를 사용하여 CloudWatch Application Insights for .NET and SQL Server를 활성화할 수 있습니다. 애플리케이션 리소스를 구성하려면 애플리케이션 설정 (p. 280) 단원의 단계를 따르십시오.

목차

- CloudWatch Application Insights for .NET and SQL Server에 액세스 (p. 278)
- 사전 조건 (p. 278)
- IAM 정책 (p. 279)
- 애플리케이션 설정 (p. 280)

CloudWatch Application Insights for .NET and SQL Server에 액세스

CloudWatch Application Insights for .NET and SQL Server에 대한 액세스 권한이 있는 경우 다음 인터페이스 중 하나를 통해 관리할 수 있습니다.

- CloudWatch 콘솔: 애플리케이션에 대한 모니터링을 추가하려면 CloudWatch 콘솔의 왼쪽 탐색 창에서 설정을 선택합니다. 설정 페이지에서 Application Insights for .NET and SQL Server(.NET 및 SQL Server용 Application Insights)를 선택합니다. 애플리케이션을 구성한 후 CloudWatch 콘솔을 사용하여 감지된 문제점을 보고 분석할 수 있습니다.
- AWS 명령줄 인터페이스(AWS CLI): AWS CLI를 사용하여 AWS API 작업에 액세스할 수 있습니다. 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 AWS 명령줄 인터페이스 설치를 참조하십시오. API 참조는 Amazon CloudWatch Application Insights for .NET and SQL Server API 참조를 참조하십시오.

사전 조건

CloudWatch Application Insights를 사용하여 애플리케이션을 구성하려면 다음 사전 조건을 완료해야 합니다.

- AWS Systems Manager 활성화: Systems Manager 에이전트(SSM 에이전트)를 설치해야 하며 인스턴스가 SSM을 사용할 수 있어야 합니다. SSM 에이전트 설치 방법에 대한 단계는 AWS Systems Manager 설정을 참조하십시오.
- EC2 인스턴스 역할: Systems Manager를 활성화하려면 `AmazonEC2RoleforSSM`에 연결하고(AWS Systems Manager에 대한 자격 증명 기반 정책(IAM 정책) 사용 참조) CloudWatch를 통해 인스턴스 지표 및 로그를 활성화하려면 `CloudWatchAgentServerPolicy`에 연결해야 합니다. 자세한 내용은 CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성을 참조하십시오.

- AWS 리소스 그룹: .NET 및 SQL Server 애플리케이션을 CloudWatch Application Insights for .NET and SQL Server에 온보딩하려면 애플리케이션 스택에 사용되는 모든 관련 AWS 리소스를 포함하는 리소스 그룹을 생성해야 합니다. 여기에는 Application Load Balancer, IIS 및 웹 프런트 엔드를 실행하는 EC2 인스턴스, .NET 작업자 티어 및 SQL Server 데이터베이스가 포함됩니다. CloudWatch Application Insights for .NET and SQL Server에는 리소스 그룹과 동일한 태그 또는 CloudFormation 스택을 사용하는 Auto Scaling 그룹이 자동으로 포함됩니다. 이는 Auto Scaling 그룹이 현재 리소스 그룹에서 지원되지 않기 때문입니다. 자세한 내용은 AWS 리소스 그룹 시작하기를 참조하십시오.
- IAM 권한: 관리자가 아닌 사용자의 경우 AWS Identity and Access Management(IAM) 정책을 만들어 사용자 자격 증명에 연결해야 합니다. IAM 정책 (p. 279) 단원을 참조하십시오.
- 서비스 연결 역할: CloudWatch Application Insights for .NET and SQL Server는 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 수동으로 생성할 필요가 없습니다. 대신 AWS Management Console에서 첫 번째 CloudWatch Application Insights for .NET and SQL Server 애플리케이션을 생성할 때 자동으로 생성됩니다. 자세한 내용은 CloudWatch Application Insights for .NET and SQL Server에 서비스 연결 역할 사용 (p. 372) 단원을 참조하십시오.
- EC2 Windows 인스턴스용 성능 카운터 지표 지원: EC2 Windows 인스턴스에서 성능 카운터 지표를 모니터링하려면 인스턴스에 성능 카운터를 설치해야 합니다. 성능 카운터 지표와, 이에 해당하는 성능 카운터 이름은 성능 카운터 지표 (p. 315)를 참조하십시오. 성능 카운터에 대한 자세한 내용은 Performance Counters(성능 카운터)를 참조하십시오.

IAM 정책

CloudWatch Application Insights for .NET and SQL Server를 사용하려면 Identity and Access Management(IAM) 정책을 생성하여 IAM 사용자 자격 증명에 연결해야 합니다. IAM 정책은 사용자 권한을 정의합니다.

콘솔을 사용해 IAM 정책을 생성하려면

IAM 콘솔을 사용해 IAM 정책을 생성하려면 다음 절차를 따르십시오.

1. IAM 콘솔로 이동합니다. 왼쪽 탐색 창에서 정책을 선택합니다.
2. 페이지 상단에서 정책 생성을 선택합니다.
3. JSON 탭을 선택합니다.
4. 다음 JSON 문서를 복사하여 JSON 탭에 붙여 넣습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "applicationinsights:*",
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

5. 정책 검토를 선택합니다.
6. 정책의 이름을 입력합니다(예: “AppInsightsPolicy”). 필요한 경우 설명을 입력합니다.
7. 정책 생성을 선택합니다.
8. 왼쪽 탐색 창에서 사용자를 선택합니다.
9. 정책을 연결하려는 사용자의 사용자 이름을 선택합니다.
10. Add permissions(권한 추가)를 선택합니다.

11. Attach existing policies directly(기존 정책 직접 연결)를 선택합니다.
12. 방금 만든 정책을 검색하고 정책 이름 왼쪽에 있는 확인란을 선택합니다.
13. Next: Review(다음: 검토)를 선택합니다.
14. 올바른 정책이 나열되어 있는지 확인하고 Add permissions(권한 추가)를 선택합니다.
15. CloudWatch Application Insights for .NET and SQL Server를 사용할 때 방금 생성한 정책과 연결된 사용자로 로그인해야 합니다.

AWS CLI를 사용해 IAM 정책을 생성하려면

AWS CLI를 사용해 IAM 정책을 생성하려면 위 JSON 문서를 현재 폴더에 있는 파일로 사용하여 명령줄에서 create-policy 작업을 실행할 수 있습니다.

Windows PowerShell용 AWS 도구를 사용하여 IAM 정책을 생성하려면

Windows PowerShell용 AWS 도구를 사용해 IAM 정책을 생성하려면 위 JSON 문서를 현재 폴더에 있는 파일로 사용하여 New-IAMPolicy cmdlt를 실행합니다.

애플리케이션 설정

이 단원에서는 콘솔, AWS CLI, Windows PowerShell용 AWS 도구를 사용하여 CloudWatch Application Insights for .NET and SQL Server 애플리케이션을 설정, 구성 및 관리하는 방법을 단계별로 안내합니다.

주제

- CloudWatch 콘솔을 사용해 애플리케이션 추가 및 구성 (p. 280)
- 콘솔을 사용해 애플리케이션 비활성화 (p. 282)
- 콘솔을 사용해 애플리케이션 구성 요소에 대한 모니터링 비활성화 (p. 282)
- 콘솔을 사용해 애플리케이션 삭제 (p. 282)
- 명령줄을 사용하여 Application Insights에서 애플리케이션 추가 및 관리 (p. 282)
- 명령줄을 사용하여 모니터링 관리 및 업데이트 (p. 285)

CloudWatch 콘솔을 사용해 애플리케이션 추가 및 구성

CloudWatch 콘솔에서 애플리케이션 추가 및 구성

CloudWatch 콘솔에서 CloudWatch Application Insights for .NET and SQL Server를 시작하려면 다음 단계를 따르십시오.

1. 시작. CloudWatch 콘솔 랜딩 페이지를 엽니다. 왼쪽 탐색 창에서 설정을 선택합니다. 설정 페이지에서 Application Insights for .NET and SQL Server(.NET 및 SQL Server용 Application Insights) > View applications to get started(시작할 애플리케이션 보기)를 선택합니다.
2. 애플리케이션 추가. .NET 및 SQL Server 애플리케이션에 대한 모니터링을 설정하려면 CloudWatch Application Insights for .NET and SQL Server 페이지에서 Add an application(애플리케이션 추가)을 선택합니다. 이 페이지는 모니터링 상태와 함께 CloudWatch Application Insights for .NET and SQL Server를 사용하여 모니터링되는 애플리케이션 목록을 표시합니다. Add an application(애플리케이션 추가)을 선택하면 Add an application(애플리케이션 추가) 페이지로 이동합니다.
3. 리소스 그룹 선택. Add an application(애플리케이션 추가) 페이지에서 CloudWatch Application Insights for .NET and SQL Server에 애플리케이션을 추가하려면 드롭다운 목록에서 애플리케이션 리소스가 포함된 AWS 리소스 그룹을 선택합니다. 이러한 리소스로는 프런트 엔드 서버, 로드 밸런서, Auto Scaling 그룹, 데이터베이스 서버 등을 들 수 있습니다.

   CloudWatch Application Insights for .NET and SQL Server에서는 태그 기반 및 CloudFormation 기반 리소스 그룹을 모두 지원합니다(Auto Scaling 그룹 제외). 자세한 내용은 Tag Editor 작업을 참조하십시오.

.NET 애플리케이션에 대한 리소스 그룹을 아직 생성하지 않은 경우 지금 생성할 수 있습니다. 자세한 내용은 AWS 리소스 그룹 사용 설명서를 참조하십시오.

4. 모니터링 세부 정보 추가. 애플리케이션을 추가하면 애플리케이션 구성 요소, 해당 구성 요소의 리소스 및 모니터링 상태가 나열된 모니터링 세부 정보 페이지로 이동합니다. 구성 요소는 애플리케이션을 구성하는 유사한 리소스의 자동 그룹형, 독립 실행형 또는 사용자 지정 그룹입니다. 기본적으로 CloudWatch Application Insights for .NET and SQL Server는 Auto Scaling 그룹에 있는 인스턴스와 Elastic Load Balancer 뒤에 있는 인스턴스를 그룹화합니다. 이 페이지에서 사용자 지정 구성 요소를 구성하고 각 애플리케이션 구성 요소에 대한 모니터링을 관리할 수 있습니다. 지원되는 구성 요소는 지원되는 애플리케이션 구성 요소 (p. 277) 단원을 참조하십시오.
5. 구성 요소 구성. 리소스 그룹을 선택하면 구성 요소를 구성하라는 메시지가 표시됩니다. .NET 웹 서버 인스턴스 등 유사한 리소스를 사용자 지정 구성 요소로 그룹화하여 보다 쉽게 온보딩을 수행하고 모니터링 및 통찰력을 향상시킬 수 있습니다. 기본적으로 CloudWatch Application Insights for .NET and SQL Server는 Auto Scaling 그룹에 있는 인스턴스와 Elastic Load Balancer 뒤에 있는 인스턴스를 그룹화합니다. 지원되는 구성 요소는 지원되는 애플리케이션 구성 요소 (p. 277) 단원을 참조하십시오.

   Application components(애플리케이션 구성 요소)에서 모니터를 설정할 각 구성 요소에 대해 구성 요소를 선택하고 Manage Monitoring(모니터링 관리)을 선택합니다.
6. 모니터 활성화. 애플리케이션 구성 요소에 대한 모니터링을 설정하려면 모니터링할 구성 요소를 선택한 다음 Manage Monitoring(모니터링 관리)을 선택합니다. Enable Monitoring(모니터링 활성화) 확인란을 선택합니다. 확인란을 선택하면 드롭다운에 관련 애플리케이션 티어가 채워집니다. 선택한 구성 요소의 애플리케이션 티어를 선택합니다. 티어는 선택한 리소스에서 실행 중인 애플리케이션 스택의 일부를 나타냅니다.

   티어 선택 사항에 따라 CloudWatch Application Insights for .NET and SQL Server는 선택된 구성 요소에서 모니터링할 로그에 대한 권장 사항을 제공합니다. 이 권장 사항은 필요에 따라 사용자 지정할 수 있습니다.

   Microsoft SQL Server 오류 로그 및 IIS 로그를 포함한 애플리케이션별 로그의 경우 기본 로그 경로(있는 경우)를 확인하거나 EC2 인스턴스에 올바른 로그 위치를 입력하십시오.

   Windows 로그 및 응용 프로그램 및 서비스 로그를 포함하여 Windows 이벤트 로그를 추가하도록 선택할 수도 있습니다. 이렇게 하려면 저장하고 분석할 이벤트 유형을 입력하면 됩니다. 그런 다음 CloudWatch 계정에 저장할 모든 이벤트 수준(중요, 오류, 경고, 정보, 상세)을 지정합니다.

   각 로그를 CloudWatch 계정에 저장 및 그룹화할 수 있도록 로그 그룹을 추가할 수 있으므로 검색이 용이해 집니다.

   CloudWatch Application Insights for .NET and SQL Server는 또한 애플리케이션 리소스에 대한 관련 지표를 설정합니다. 해당 지표 임계값을 식별하기 위해 약 2주 동안 모니터링됩니다. 과거에 지표를 생성한 경우 CloudWatch Application Insights for .NET and SQL Server는 지난 2주 동안의 기록 데이터를 가져와 임계값을 식별하고 그에 따라 경보를 설정합니다. 새로 생성된 지표의 경우 경보가 생성되기까지 최대 3일이 소요될 수 있습니다. 계정에서 생성한 CloudWatch 경보를 사용하여 애플리케이션 리소스를 모니터링할 수도 있습니다.
7. 모니터 저장. 로그와 지표 선택 및 사용자 지정이 끝나면 저장을 선택하여 선택한 구성 요소에 대한 모니터를 설정합니다. 저장을 선택하면 Application Insights가 권장 지표 및 선택한 로그를 기반으로 애플리케이션의 모든 인스턴스에 대한 CloudWatch 에이전트 구성 파일을 설정합니다. 이 프로세스가 완료되는 데 최대 1시간이 소요될 수 있습니다.

   CloudWatch Application Insights for .NET and SQL Server는 또한 구성 요소에서 선택한 지표에 대한 CloudWatch 경보를 설정합니다. 경보는 지난 2주 간의 기록 지표 패턴을 모니터링하여 동적으로 업데이트됩니다.

   취소를 선택하면 Application Insights는 현재 선택 사항만 삭제합니다.

   CloudWatch Application Insights for .NET and SQL Server를 사용하여 새 애플리케이션을 생성하면 서비스 연결 역할이 자동으로 생성됩니다. 서비스 연결 역할을 삭제하려면 먼저 .NET 및 SQL Server용 CloudWatch Applications Insights에서 모든 애플리케이션을 삭제한 다음 수동으로 역할을 삭제해야 합

니다. 자세한 내용은 CloudWatch Application Insights for .NET and SQL Server에 서비스 연결 역할 사용 (p. 372) 단원을 참조하십시오.

CloudWatch Application Insights for .NET and SQL Server는 이제 애플리케이션에 대한 지표 및 로그를 모니터링하도록 설정되어 있습니다. 시스템이 의미 있는 통찰력을 얻는 데 최대 2주가 걸릴 수 있습니다.

*리소스 그룹이 이미 구성되어 있고 구성을 저장하고 싶지만 CloudWatch Application Insights for .NET and SQL Server에서 애플리케이션을 모니터링하지 않으려면 CloudWatch Application Insights for .NET and SQL Server를 비활성화합니다. 구성을 삭제할 수도 있습니다.

8. AWS Systems Manager OpsCenter 통합 추가. 선택한 애플리케이션에서 문제가 감지된 시점을 보고 알림을 받으려면 모니터링 세부 정보 페이지에서 AWS OpsCenter와 통합 확인란을 선택합니다. AWS 리소스와 관련된 운영 작업 항목(OpsItem)을 해결하기 위해 수행된 작업을 추적하려면 SNS 주제 ARN을 제공합니다.
9. 모니터링 보기(선택 사항). 모니터링을 위해 애플리케이션을 설정한 후에는 CloudWatch 콘솔의 기본 개요 페이지에서 감지된 문제와 통찰력을 보고 문제를 해결할 수 있습니다. Application Insights 랜딩 페이지 또는 CloudWatch 랜딩 페이지에서 View Insights(통찰력 보기)를 선택하여 감지된 문제, 경보 및 대시보드를 볼 수 있습니다.

콘솔을 사용해 애플리케이션 비활성화

애플리케이션을 비활성화하려면 CloudWatch 대시보드의 설정에서 비활성화할 애플리케이션을 선택합니다. 작업에서 비활성화를 선택합니다. 애플리케이션을 비활성화하면 모니터링이 비활성화되지만 Application Insights는 애플리케이션 구성 요소에 대해 저장된 모니터를 저장합니다.

콘솔을 사용해 애플리케이션 구성 요소에 대한 모니터링 비활성화

애플리케이션 구성 요소에 대한 모니터링을 비활성화하려면 애플리케이션 세부 정보 페이지에서 모니터링을 비활성화할 구성 요소를 선택합니다. Manage Monitors(모니터 관리)를 선택한 다음 Enable Monitoring(모니터링 활성화) 확인란의 선택을 취소합니다.

콘솔을 사용해 애플리케이션 삭제

애플리케이션을 삭제하려면 CloudWatch 대시보드의 설정에서 삭제할 애플리케이션을 선택합니다. 작업에서 삭제를 선택합니다. 그러면 모니터링이 삭제되고 애플리케이션 구성 요소에 대해 저장된 모니터가 모두 삭제됩니다. 애플리케이션 리소스는 삭제되지 않습니다.

명령줄을 사용하여 Application Insights에서 애플리케이션 추가 및 관리

명령줄을 사용해 Application Insights 애플리케이션에 대한 정보를 얻고 이 애플리케이션을 추가, 관리 및 구성할 수 있습니다.

주제

- 애플리케이션 추가 (p. 283)
- 애플리케이션 설명 (p. 283)
- 애플리케이션의 구성 요소 나열 (p. 283)
- 구성 요소 설명 (p. 283)
- 유사한 리소스를 사용자 지정 구성 요소로 그룹화 (p. 284)
- 사용자 지정 구성 요소 그룹화 해제 (p. 284)
- 애플리케이션 업데이트 (p. 285)
- 사용자 지정 구성 요소 업데이트 (p. 285)

애플리케이션 추가

AWS CLI를 사용하여 애플리케이션 추가

AWS CLI를 사용하여 `my-resource-group`이라는 리소스 그룹에 애플리케이션을 추가하려면 생성된 opsItem을 SNS 주제 ARN인 `arn:aws:sns:us-east-1:123456789012:MyTopic`에 제공할 수 있도록 OpsCenter가 활성화한 상태에서 다음 명령을 사용합니다.

```
aws application-insights create-application --resource-group-name my-resource-group --ops-
center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Windows PowerShell용 AWS 도구를 사용하여 애플리케이션 추가

Windows PowerShell용 AWS 도구를 사용하여 `my-resource-group`이라는 리소스 그룹에 애플리케이션을 추가하려면 생성된 opsItem을 SNS 주제 ARN인 `arn:aws:sns:us-east-1:123456789012:MyTopic`에 제공할 수 있도록 OpsCenter가 활성화한 상태에서 다음 명령을 사용합니다.

```
New-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

애플리케이션 설명

AWS CLI를 사용하여 애플리케이션 설명

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션을 AWS CLI를 사용하여 설명하려면 다음 명령을 사용합니다.

```
aws application-insights describe-application --resource-group-name my-resource-group
```

Windows PowerShell용 AWS 도구를 사용하여 애플리케이션 설명

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션을 Windows PowerShell용 AWS 도구를 사용하여 설명하려면 다음 명령을 사용합니다.

```
Get-CWAIApplication -ResourceGroupName my-resource-group
```

애플리케이션의 구성 요소 나열

AWS CLI를 사용해 애플리케이션의 구성 요소 나열

`my-resource-group`이라는 리소스 그룹에서 생성된 구성 요소를 AWS CLI를 사용하여 나열하려면 다음 명령을 사용합니다.

```
aws application-insights list-components --resource-group-name my-resource-group
```

Windows PowerShell용 AWS 도구를 사용하여 애플리케이션의 구성 요소 나열

`my-resource-group`이라는 리소스 그룹에서 생성된 구성 요소를 Windows PowerShell용 AWS 도구를 사용하여 나열하려면 다음 명령을 사용합니다.

```
Get-CWAIComponentList -ResourceGroupName my-resource-group
```

구성 요소 설명

AWS CLI를 사용하여 구성 요소 설명

my-resource-group이라는 리소스 그룹에서 생성되고 애플리케이션에 속해 있는 my-component라는 구성 요소를 설명하려면 다음 AWS CLI 명령을 사용하면 됩니다.

```
aws application-insights describe-component --resource-group-name my-resource-group --
component-name my-component
```

Windows PowerShell용 AWS 도구를 사용하여 구성 요소 설명

my-resource-group이라는 리소스 그룹에서 생성되고 애플리케이션에 속해 있는 my-component라는 구성 요소를 설명하려면 다음과 같은 Windows PowerShell용 AWS 도구 명령을 사용하면 됩니다.

```
Get-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

유사한 리소스를 사용자 지정 구성 요소로 그룹화

.NET 웹 서버 인스턴스 등 유사한 리소스를 사용자 지정 구성 요소로 그룹화하여 보다 쉽게 온보딩을 수행하고 모니터링 및 통찰력을 향상시킬 수 있습니다. 현재 CloudWatch Application Insights for .NET and SQL Server는 EC2 인스턴스에 대한 사용자 지정 그룹을 지원합니다.

AWS CLI를 사용해 리소스를 사용자 지정 구성 요소로 그룹화하려면

AWS CLI를 사용해 세 가지 인스턴스(arn:aws:ec2:us-east-1:123456789012:instance/i-11111, arn:aws:ec2:us-east-1:123456789012:instance/i-22222, arn:aws:ec2:us-east-1:123456789012:instance/i-33333)를 my-resource-group이라는 리소스 그룹에 대해 생성된 애플리케이션용 사용자 지정 구성 요소(my-component)로 그룹화하려면 다음 명령을 사용하십시오.

```
aws application-insights create-component --resource-group-name my-resource-group --
component-name my-component --resource-list arn:aws:ec2:us-east-1:123456789012:instance/
i-11111 arn:aws:ec2:us-east-1:123456789012:instance/i-22222 arn:aws:ec2:us-
east-1:123456789012:instance/i-33333
```

Windows PowerShell용 AWS 도구를 사용해 리소스를 사용자 지정 구성 요소로 그룹화하려면

Windows PowerShell용 AWS 도구를 사용해 세 가지 인스턴스(arn:aws:ec2:us-east-1:123456789012:instance/i-11111, arn:aws:ec2:us-east-1:123456789012:instance/i-22222, arn:aws:ec2:us-east-1:123456789012:instance/i-33333)를 my-resource-group이라는 리소스 그룹에 대해 생성된 애플리케이션용 사용자 지정 구성 요소(my-component)로 그룹화하려면 다음 명령을 사용하십시오.

```
New-CWAIComponent -ResourceGroupName my-resource-group -ComponentName my-component
 -ResourceList arn:aws:ec2:us-east-1:123456789012:instance/i-11111,arn:aws:ec2:us-
east-1:123456789012:instance/i-22222,arn:aws:ec2:us-east-1:123456789012:instance/i-33333
```

사용자 지정 구성 요소 그룹화 해제

AWS CLI를 사용해 사용자 지정 구성 요소의 그룹화를 해제하려면

my-resource-group 리소스 그룹에서 생성된 애플리케이션의 my-component라는 사용자 지정 구성 요소를 AWS CLI를 사용해 그룹화 해제하려면 다음 명령을 사용하십시오.

```
aws application-insights delete-component --resource-group-name my-resource-group --
component-name my-new-component
```

Windows PowerShell용 AWS 도구를 사용해 사용자 지정 구성 요소의 그룹화를 해제하려면

my-resource-group 리소스 그룹에서 생성된 애플리케이션의 my-component라는 사용자 지정 구성 요소를 Windows PowerShell용 AWS 도구를 사용해 그룹화 해제하려면 다음 명령을 사용하십시오.

```
Remove-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

애플리케이션 업데이트

AWS CLI를 사용하여 애플리케이션 업데이트

AWS CLI에서 다음 명령을 사용하여 애플리케이션을 업데이트함으로써 애플리케이션에서 탐지된 문제에 대해 AWS Systems Manager OpsCenter OpsItems를 생성하고 생성된 OpsItems를 SNS 주제인 arn:aws:sns:us-east-1:123456789012:MyTopic에 연결할 수 있습니다.

```
aws application-insights update-application --resource-group-name my-resource-group --ops-
center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Windows PowerShell용 AWS 도구를 사용하여 애플리케이션 업데이트

Windows PowerShell용 AWS 도구에서 다음 명령을 사용하여 애플리케이션을 업데이트함으로써 애플리케이션에서 탐지된 문제에 대해 AWS Systems Manager OpsCenter OpsItems를 생성하고 생성된 OpsItems를 SNS 주제인 arn:aws:sns:us-east-1:123456789012:MyTopic에 연결할 수 있습니다.

```
Update-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

사용자 지정 구성 요소 업데이트

AWS CLI를 사용해 사용자 지정 구성 요소 업데이트

AWS CLI에서 다음 명령을 사용하여 my-component라는 사용자 지정 구성 요소를 새로운 구성 요소 이름인 my-new-component와 업데이트된 인스턴스 그룹으로 업데이트할 수 있습니다.

```
aws application-insights update-component --resource-group-name my-resource-
group --component-name my-component --new-component-name my-new-component --
resource-list arn:aws:ec2:us-east-1:123456789012:instance/i-44444 arn:aws:ec2:us-
east-1:123456789012:instance/i-55555
```

Windows PowerShell용 AWS 도구를 사용하여 사용자 지정 구성 요소 업데이트

Windows PowerShell용 AWS 도구에서 다음 명령을 사용하여 my-component라는 사용자 지정 구성 요소를 새로운 구성 요소 이름인 my-new-component와 업데이트된 인스턴스 그룹으로 업데이트할 수 있습니다.

```
Update-CWAIComponent -ComponentName my-component -NewComponentName my-new-
component -ResourceGroupName my-resource-group -ResourceList arn:aws:ec2:us-
east-1:123456789012:instance/i-44444,arn:aws:ec2:us-east-1:123456789012:instance/i-55555
```

명령줄을 사용하여 모니터링 관리 및 업데이트

명령줄을 사용하여 Application Insights 애플리케이션에 대한 모니터링을 관리 및 업데이트할 수 있습니다.

주제

- 애플리케이션에서 발생하는 문제 나열 (p. 286)
- 애플리케이션 문제 설명 (p. 286)

- 문제와 관련된 이상 또는 오류 설명 (p. 286)
- 애플리케이션의 이상 또는 오류 설명 (p. 286)
- 구성 요소의 모니터링 구성 설명 (p. 287)
- 구성 요소의 권장 모니터링 구성 설명 (p. 287)
- 구성 요소의 모니터링 구성 업데이트 (p. 288)
- Application Insights 모니터링에서 지정된 리소스 그룹 제거 (p. 288)

애플리케이션에서 발생하는 문제 나열

AWS CLI를 사용하여 애플리케이션에서 발생하는 문제 나열

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션용 Unix Epoch 이후 1,000~10,000밀리초 사이에 탐지된 애플리케이션 문제를 AWS CLI를 사용해 나열하려면 다음 명령을 사용하십시오.

```
 aws application-insights list-problems --resource-group-name my-resource-group --start-
time 1000 --end-time 10000
```

Windows PowerShell용 AWS 도구를 사용하여 애플리케이션의 문제 나열

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션용 Unix Epoch 이후 1,000~10,000밀리초 사이에 탐지된 애플리케이션 문제를 Windows PowerShell용 AWS 도구를 사용해 나열하려면 다음 명령을 사용하십시오.

```
$startDate = "8/6/2019 3:33:00"
$endDate = "8/6/2019 3:34:00"
Get-CWAIProblemList -ResourceGroupName my-resource-group -StartTime $startDate -
EndTime $endDate
```

애플리케이션 문제 설명

AWS CLI를 사용하여 애플리케이션 문제 설명

AWS CLI를 사용하여 문제 ID가 `p-1234567890`인 문제를 설명하려면 다음 명령을 사용하십시오.

```
aws application-insights describe-problem –problem-id p-1234567890
```

Windows PowerShell용 AWS 도구를 사용하여 애플리케이션 문제 설명

Windows PowerShell용 AWS 도구를 사용하여 문제 ID가 `p-1234567890`인 문제를 설명하려면 다음 명령을 사용하십시오.

```
Get-CWAIProblem -ProblemId p-1234567890
```

문제와 관련된 이상 또는 오류 설명

AWS CLI를 사용하여 문제와 관련된 이상 또는 오류 설명

AWS CLI를 사용하여 문제 ID가 `p-1234567890`인 문제와 관련된 이상 또는 오류를 설명하려면 다음 명령을 사용하십시오.

```
aws application-insights describe-problem-observations --problem-id -1234567890
```

Windows PowerShell용 AWS 도구를 사용하여 문제와 관련된 이상 또는 오류 설명

Windows PowerShell용 AWS 도구를 사용하여 문제 ID가 p-1234567890인 문제와 관련된 이상 또는 오류를 설명하려면 다음 명령을 사용하십시오.

```
Get-CWAIProblemObservation -ProblemId p-1234567890
```

애플리케이션의 이상 또는 오류 설명

AWS CLI를 사용하여 애플리케이션의 이상 또는 오류 설명

AWS CLI를 사용하여 관찰 ID가 o-1234567890인 애플리케이션의 이상 또는 오류를 설명하려면 다음 명령을 사용하십시오.

```
aws application-insights describe-observation –observation-id o-1234567890
```

Windows PowerShell용 AWS 도구를 사용하여 애플리케이션의 이상 또는 오류 설명

Windows PowerShell용 AWS 도구를 사용하여 관찰 ID가 o-1234567890인 애플리케이션의 이상 또는 오류를 설명하려면 다음 명령을 사용하십시오.

```
Get-CWAIObservation -ObservationId o-1234567890
```

구성 요소의 모니터링 구성 설명

AWS CLI를 사용하여 구성 요소의 모니터링 구성 설명

my-resource-group 리소스 그룹에서 생성된 애플리케이션에 있는 my-component라는 구성 요소의 모니터링 구성을 AWS CLI를 사용해 설명하려면 다음 명령을 사용하십시오.

```
aws application-insights describe-component-configuration –resource-group-name my-resource-group –component-name my-component
```

Windows PowerShell용 AWS 도구를 사용하여 구성 요소의 모니터링 구성 설명

my-resource-group 리소스 그룹에서 생성된 애플리케이션에서 my-component라는 구성 요소의 모니터링 구성을 AWS CLI를 사용해 설명하려면 다음 명령을 사용하십시오.

```
Get-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-resource-group
```

구성 요소 구성에 관한 자세한 내용과 예시 JSON 파일은 구성 요소 구성 (p. 293) 단원을 참조하십시오.

구성 요소의 권장 모니터링 구성 설명

AWS CLI를 사용하여 구성 요소의 권장 모니터링 구성 설명

구성 요소가 .NET Worker 애플리케이션의 일부인 경우 my-component 리소스 그룹에서 생성된 애플리케이션에 있는 my-resource-group라는 구성 요소의 권장 모니터링 구성을 AWS CLI에서 다음 명령을 사용해 설명할 수 있습니다.

```
aws application-insights describe-component-configuration-recommendation --resource-group-name my-resource-group --component-name my-component --tier DOT_NET_WORKER
```

Windows PowerShell용 AWS 도구를 사용하여 구성 요소의 권장 모니터링 구성 설명

구성 요소가 .NET Worker 애플리케이션의 일부인 경우 `my-component` 리소스 그룹에서 생성된 애플리케이션에 있는 `my-resource-group`라는 구성 요소의 권장 모니터링 구성을 Windows PowerShell용 AWS 도구에서 다음 명령을 사용해 설명할 수 있습니다.

```
Get-CWAIComponentConfigurationRecommendation -ComponentName my-component -
ResourceGroupName my-resource-group -Tier DOT_NET_WORKER
```

구성 요소 구성에 관한 자세한 내용과 예시 JSON 파일은 구성 요소 구성 (p. 293) 단원을 참조하십시오.

구성 요소의 모니터링 구성 업데이트

AWS CLI를 사용하여 구성 요소의 모니터링 구성 업데이트

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션의 `my-component`라는 구성 요소를 AWS CLI를 사용해 업데이트하려면 다음 명령을 사용하십시오. 이 명령은 다음과 같은 작업을 포함합니다.

1. 구성 요소에 대한 모니터링을 활성화합니다.
2. 구성 요소의 티어를 `.NET Worker`로 설정합니다.
3. `configuration.txt`라는 로컬 파일을 읽도록 구성 요소의 JSON 구성을 업데이트합니다.

```
aws application-insights update-component-configuration --resource-group-name my-
resource-group --component-name my-component --tier DOT_NET_WORKER --monitor --component-
configuration "file://configuration.txt"
```

Windows PowerShell용 AWS 도구를 사용하여 구성 요소의 모니터링 구성 업데이트

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션의 `my-component`라는 구성 요소를 Windows PowerShell용 AWS 도구를 사용해 업데이트하려면 다음 명령을 사용하십시오. 이 명령은 다음과 같은 작업을 포함합니다.

1. 구성 요소에 대한 모니터링을 활성화합니다.
2. 구성 요소의 티어를 `.NET Worker`로 설정합니다.
3. `configuration.txt`라는 로컬 파일을 읽도록 구성 요소의 JSON 구성을 업데이트합니다.

```
[string]$config = Get-Content -Path configuration.txt
Update-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-
resource-group -Tier DOT_NET_WORKER -Monitor 1 -ComponentConfiguration $config
```

구성 요소 구성에 관한 자세한 내용과 예시 JSON 파일은 구성 요소 구성 (p. 293) 단원을 참조하십시오.

Application Insights 모니터링에서 지정된 리소스 그룹 제거

AWS CLI를 사용하여 Application Insights 모니터링에서 지정된 리소스 그룹 제거

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션을 AWS CLI를 사용해 모니터링에서 제거하려면 다음 명령을 사용하십시오.

```
aws application-insights delete-application --resource-group-name my-resource-group
```

Windows PowerShell용 AWS 도구를 사용하여 Application Insights 모니터링에서 지정된 리소스 그룹 제거

`my-resource-group`이라는 리소스 그룹에서 생성된 애플리케이션을 Windows PowerShell용 AWS 도구를 사용해 모니터링에서 제거하려면 다음 명령을 사용하십시오.

```
Remove-CWAIApplication -ResourceGroupName my-resource-group
```

Amazon CloudWatch Application Insights for .NET and SQL Server에 의해 감지된 문제 보기 및 해결

.NET 및 SQL Server 애플리케이션에 영향을 주는 문제에 대한 개요는 CloudWatch 콘솔의 기본 개요 페이지에 있는 CloudWatch Application Insights for .NET and SQL Server 위젯 아래에 나열됩니다. 자세한 내용은 Amazon CloudWatch Application Insights for .NET and SQL Server 시작하기 (p. 278) 단원을 참조하십시오.

CloudWatch Application Insights for .NET and SQL Server 위젯에는 다음이 표시됩니다.

- 감지된 문제의 심각도
- 문제의 요약
- 문제의 가능한 근본 원인
- 문제 시작 시간
- 문제의 해결 상태
- 영향 받는 리소스 그룹

특정 문제의 세부 정보를 드릴다운하려면 문제 요약에서 문제에 대한 설명을 선택합니다. 세부 대시보드에는 문제에 대한 통찰력과 관련 지표 이상 및 로그 오류의 조각이 표시됩니다. 여기에서 유용한지 여부를 선택하여 통찰력의 관련성에 대한 피드백을 제공할 수 있습니다.

구성되지 않은 새 리소스가 감지된 경우 문제점 요약 설명은 구성 편집 마법사로 이동하여 새 리소스를 구성합니다. 필요한 경우 세부 대시보드의 오른쪽 상단에 있는 View/edit configuration(구성 보기/편집)을 선택하여 리소스 그룹 구성을 보거나 편집할 수 있습니다.

개요로 돌아가려면 .NET 및 SQL Server용 CloudWatch Application Insights 세부 대시보드 머리글 옆에 있는 Back to overview(개요로 돌아가기)를 선택합니다.

감지된 문제에 대해 제공되는 정보

CloudWatch Application Insights for .NET and SQL Server는 감지된 문제에 대한 다음 정보를 제공합니다.

- 문제의 간략한 요약
- 문제의 시작 시간 및 날짜
- 문제의 심각도: 높음/중간/낮음
- 감지된 문제의 상태: 진행 중/해결됨
- 통찰력: 감지된 문제 및 가능한 근본 원인에 대해 자동으로 생성된 통찰력
- 인사이트에 대한 피드백: CloudWatch Application Insights for .NET and SQL Server에 의해 생성된 인사이트 유용성에 대해 사용자가 제공한 피드백
- 관련 관찰: 다양한 애플리케이션 구성 요소에서 발생한 문제와 관련된 로그의 오류 조각 및 지표 이상의 상세 보기

피드백

감지된 문제에 대해 자동으로 생성된 통찰력에 대한 유용성을 평가하여 피드백을 제공할 수 있습니다. 통찰력에 대한 피드백은 애플리케이션 진단(지표 이상 및 로그 예외)과 함께 향후 유사한 문제의 탐지를 개선하는 데 사용됩니다.

구성 오류

CloudWatch Application Insights for .NET and SQL Server는 구성을 사용하여 구성 요소에 대한 모니터링 원격 측정을 생성합니다. Application Insights가 계정 또는 구성과 관련된 문제를 발견하면 애플리케이션의 구성 문제를 해결하는 방법에 대한 정보를 설명 필드에 제공합니다.

다음 표는 특정 설명에 대해 제안된 해결 방법을 보여줍니다.

설명	제안된 해결 방법	추가 참고 사항
경보 할당량에 이미 도달했습니다.	기본적으로 각 AWS 계정에는 AWS 리전당 5,000개의 CloudWatch 경보가 있을 수 있습니다. CloudWatch 제한을 참조하십시오. 이 제한의 구속을 받는 경우 CloudWatch Application Insights for .NET and SQL Server는 애플리케이션을 모니터링하는 데 필요한 모든 경보를 생성할 수 없습니다. 이 문제를 해결하려면 CloudWatch 경보에 대한 계정 제한을 높이십시오.	해당 사항 없음
CloudFormation 할당량에 이미 도달했습니다.	Application Insights는 모든 애플리케이션 구성 요소에 대한 CloudWatch 에이전트 설치 및 구성을 관리하기 위해 각 애플리케이션에 대해 하나의 CloudFormation 스택을 만듭니다. 기본적으로 각 AWS 계정에는 200개의 스택이 포함될 수 있습니다. AWS CloudFormation 한도를 참조하십시오. 이를 해결하려면 CloudFormation 스택의 제한을 높이십시오.	해당 사항 없음
다음 인스턴스에 SSM 인스턴스 역할이 없습니다.	Application Insights가 애플리케이션 인스턴스에 CloudWatch 에이전트를 설치 및 구성하려면 AmazonEC2RoleforSSM 및 CloudWatchAgentServerPolicy 정책을 인스턴스 역할에 연결해야 합니다.	ApplicationInsights는 SSM DescribeInstanceInformation API를 호출하여 SSM 권한이 있는 인스턴스 목록을 가져옵니다. 인스턴스에 역할을 연결한 후 SSM이 인스턴스를 DescribeInstanceInformation 결과에 포함시키는 데는 약간의 시간이 필요합니다. SSM이 결과에 인스턴스를 포함할 때까지는 NO_SSM_INSTANCE_ROLE 오류가 애플리케이션에 남아 있습니다.
새 구성 요소에 구성이 필요할 수 있습니다.	Application Insights는 애플리케이션 리소스 그룹에 새 구성 요소가 있음을 감지합니다. 이 문제를 해	해당 사항 없음

설명	제안된 해결 방법	추가 참고 사항
	결하려면 새 구성 요소를 적절히 구성하십시오.	

감지된 문제에 대한 알림 및 작업 설정

CloudWatch Application Insights for .NET and SQL Server에 추가되는 각 애플리케이션의 경우 CloudWatch 이벤트가 다음 이벤트에 대해 게시됩니다.

- 문제 생성. .NET 및 SQL Server용 CloudWatch Application Insights가 새로운 문제를 감지할 때 발생합니다.
  - 세부 유형: "Application Insights 문제 감지됨"
  - 세부 정보:
    - `problemId`: 감지된 문제 ID
    - `region`: 문제가 생성된 AWS 리전
    - `resourceGroupName`: 문제가 감지된 등록된 애플리케이션의 리소스 그룹
    - `status`: 문제의 상태
    - `severity`: 문제의 심각도
    - `problemUrl`: 문제의 콘솔 URL

- 문제 업데이트. 새로운 관찰로 문제가 업데이트되거나 기존의 관찰이 업데이트되어 문제가 업데이트될 때 발생합니다. 업데이트에는 문제의 해결 또는 종료가 포함됩니다.
  - 세부 유형: "Application Insights 문제 업데이트됨"
  - 세부 정보:
    - `problemId`: 생성된 문제 ID
    - `region`: 문제가 생성된 AWS 리전
    - `resourceGroupName`: 문제가 감지된 등록된 애플리케이션의 리소스 그룹
    - `status`: 문제의 상태
    - `severity`: 문제의 심각도
    - `problemUrl`: 문제의 콘솔 URL

예제: 애플리케이션에서 생성된 문제 이벤트에 대한 알림 수신 방법

CloudWatch Application Insights for .NET and SQL Server에서 생성된 문제 이벤트에 대한 알림을 받으려면 CloudWatch Application Insights가 특정 애플리케이션에 대한 문제를 수신하도록 하는 규칙을 생성합니다.

```
aws events put-rule \
--name cwe-test \
--event-pattern \
'{
  "source": ["aws.applicationinsights"],
  "detail-type": ["Application Insights Problem Detected"],
  "detail": {
        "resourceGroupName": ["<RESOURCE_GROUP_NAME>"]
   }
  }'
```

그런 다음 이벤트를 수신해야 하는 적절한 CloudWatch 이벤트 대상을 추가합니다.

```
aws events put-targets \
```

```
--rule cwe-test \
--targets Id=1,Arn=<TARGET_ARN>
```

Note

CloudWatch 콘솔의 왼쪽 탐색 창에서 이벤트 아래의 규칙을 선택하여 규칙을 생성할 수 있습니다. 그런 다음 규칙 생성 > 이벤트 패턴을 선택한 다음 서비스별 이벤트와 일치시킬 이벤트 패턴을 작성 드롭다운에서 사용자 지정 이벤트 패턴을 선택합니다.

AWS 시스템 관리자를 통한 작업. CloudWatch Application Insights for .NET and SQL Server는 Systems Manager OpsCenter와의 통합 기능을 기본으로 제공합니다. 애플리케이션에 이 통합을 사용하도록 선택하면 애플리케이션에서 감지된 모든 문제에 대한 OpsItem이 OpsCenter 콘솔에 생성됩니다. OpsCenter 콘솔에서 CloudWatch Application Insights에 의해 감지된 문제에 대한 요약된 정보를 보고 Systems Manager Automation 실행서를 선택하여 수정 작업을 수행하거나 애플리케이션에서 리소스 문제를 일으키는 Windows 프로세스를 추가로 식별할 수 있습니다.

Amazon CloudWatch Application Insights for .NET and SQL Server 작동 방식

이 단원에서는 다음을 포함하여 CloudWatch Application Insights for .NET and SQL Server 작동 방법에 대해 설명합니다.

주제

- 데이터 보존 (p. 293)
- 구성 요소 구성 (p. 293)
- 할당량 (p. 300)

애플리케이션 검색 및 구성

애플리케이션이 CloudWatch Application Insights for .NET and SQL Server에 처음 추가되면 애플리케이션 구성 요소를 검색하여 주요 지표, 로그 및 기타 데이터 원본을 권장하고 애플리케이션을 모니터링합니다. 그러면 이러한 권장 사항을 기반으로 애플리케이션을 구성할 수 있습니다.

데이터 사전 처리

CloudWatch Application Insights for .NET and SQL Server는 애플리케이션 리소스 전반에서 모니터링되는 데이터 원본을 지속적으로 분석하여 지표 이상 및 로그 오류를 발견합니다(관찰).

지능형 문제 감지

CloudWatch Application Insights 엔진은 분류 알고리즘과 기본 제공 규칙을 사용하여 관찰을 상호 연결함으로써 애플리케이션의 문제를 감지합니다. 문제 해결을 돕기 위해 문제에 대한 상황별 정보가 포함된 자동 CloudWatch 대시보드를 만듭니다.

알림 및 작업

CloudWatch Application Insights에서 애플리케이션 문제를 감지하면 CloudWatch 이벤트를 생성하여 문제를 알려줍니다. 이러한 이벤트를 설정하는 방법에 대한 자세한 내용은 감지된 문제에 대한 알림 및 작업 설정 (p. 291) 단원을 참조하십시오.

예제 시나리오

SQL Server 데이터베이스가 지원하는 ASP .NET 애플리케이션이 있습니다. 갑자기 메모리 부족 문제가 발생하여 데이터베이스가 오작동하기 시작합니다. 이로 인해 애플리케이션 성능이 저하되고 웹 서버 및 로드 밸런서에서 HTTP 500 오류가 발생할 수 있습니다.

CloudWatch Application Insights for .NET and SQL Server 및 지능형 분석을 사용하면 동적으로 생성된 대시보드에서 관련 지표 및 로그 파일 조각을 확인하여 문제를 일으키는 애플리케이션 계층을 식별할 수 있습니다. 이 경우 SQL 데이터베이스 계층에 문제가 있을 수 있습니다.

데이터 보존

CloudWatch Application Insights for .NET and SQL Server는 문제를 55일 동안 유지하고 관찰을 60일 동안 유지합니다.

구성 요소 구성

구성 요소 구성은 구성 요소의 구성 설정을 설명하는 JSON 형식의 텍스트 파일입니다. 다음 예는 구성 요소 구성 구조 및 해당 섹션을 보여줍니다.

JSON

다음 예는 JSON 형식의 템플릿 조각을 보여줍니다.

```
{
  "alarmMetrics" : [
    list of alarm metrics
  ],
  "logs" : [
    list of logs
  ],
  "instances" : [
    component nested instances configuration
  ],
  "windowsEvents" : [
    list of windows events channels configurations
  ],
  "alarms" : [
    list of CloudWatch alarms
  ]
}
```

구성 요소 구성 섹션

구성 요소 구성은 여러 개의 주요 섹션으로 구성되어 있습니다. 구성 요소 구성 섹션의 순서는 임의 순서일 수 있습니다.

- alarmMetrics(선택 사항)

  구성 요소에 대해 모니터링할 지표 (p. 294) 목록. alarmMetrics 섹션은 모든 구성 요소 유형에 포함될 수 있습니다.
- 로그(선택 사항)

  구성 요소에 대해 모니터링할 로그 (p. 294) 목록. 로그 섹션은 EC2 인스턴스에만 있습니다.
- 인스턴스(선택 사항)

  구성 요소의 중첩 인스턴스 구성. 중첩 인스턴스와 인스턴스 섹션은 ELB, ASG 및 사용자 지정 그룹화 EC2 인스턴스와 같은 구성 요소 유형에 포함될 수 있습니다.
- 경보(선택 사항)

  구성 요소에 대해 모니터링할 경보 (p. 295) 목록입니다. 경보 섹션은 모든 구성 요소 유형에 포함될 수 있습니다.

다음 예는 JSON 형식의 인스턴스 섹션 조각을 보여줍니다.

```
{
  "alarmMetrics" : [
    list of alarm metrics
  ],
  "logs" : [
    list of logs
  ],
  "windowsEvents" : [
    list of windows events channels configurations
  ]
}
```

지표

구성 요소에 대해 모니터링할 지표를 정의합니다.

JSON

```
{
  "alarmMetricName" : "monitoredMetricName",
  "monitor" : true/false
}
```

속성

- alarmMetricName(필수 사항)

  구성 요소에 대해 모니터링할 지표의 이름. Application Insights에서 지원하는 지표는 Amazon CloudWatch Application Insights for .NET and SQL Server에서 지원되는 로그 및 지표 (p. 300) 단원을 참조하십시오.
- monitor(선택 사항)

  지표를 모니터링할지 여부를 나타내는 부울. 기본값은 `true`입니다.

로그

구성 요소에 대해 모니터링할 로그를 정의합니다.

JSON

```
{
  "logGroupName" : "logGroupName",
  "logPath" : "logPath",
  "logType" : "logType",
  "encoding" : "encodingType",
  "monitor" : true/false
}
```

속성

- logGroupName(필수 사항)

  모니터링된 로그에 연결할 CloudWatch 로그 그룹 이름. 로그 그룹 이름 제약 조건에 대해서는 CreateLogGroup을 참조하십시오.
- logPath(필수 사항)

모니터링할 로그의 경로. 로그 경로는 절대 Windows 시스템 파일 경로여야 합니다. 자세한 내용은 CloudWatch 에이전트 구성 파일: 로그 섹션을 참조하십시오.

- logType(필수 사항)

  로그 유형에 따라 Application Insights가 로그를 분석하는 로그 패턴이 결정됩니다. 로그 유형은 SQL_SERVER/IIS/APPLICATION/DEFAULT에서 선택됩니다.

- encoding(선택 사항)

  모니터링할 로그의 인코딩 유형. 지정된 인코딩은 CloudWatch 에이전트 지원 인코딩 목록에 포함되어야 합니다. 제공되지 않은 경우 CloudWatch Application Insights는 로그 유형에 대한 기본 인코딩 유형을 사용합니다.

  - APPLICATION/DEFAULT: utf-8 인코딩
  - SQL_SERVER: utf-16 인코딩
  - IIS: ascii 인코딩

- monitor(선택 사항)

  로그를 모니터링할지 여부를 나타내는 부울. 기본값은 `true`입니다.

Windows 이벤트

기록할 Windows 이벤트를 정의합니다.

JSON

```
{
  "logGroupName" : "logGroupName",
  "eventName" : "eventName",
  "eventLevels" : ["ERROR","WARNING","CRITICAL","INFORMATION","VERBOSE"],
  "monitor" : true/false
}
```

속성

- logGroupName(필수 사항)

  모니터링된 로그에 연결할 CloudWatch 로그 그룹 이름. 로그 그룹 이름 제약 조건에 대해서는 CreateLogGroup을 참조하십시오.

- eventName(필수)

  기록할 Windows 이벤트 유형입니다. 이것은 Windows 이벤트 로그 채널 이름과 동일합니다. 예를 들면 System, Security, CustomEventName 등이 있습니다. 이 필드는 기록할 Windows 이벤트 유형마다 있어야 합니다.

- eventLevels(필수)

  기록할 이벤트 수준을 지정합니다. 기록할 각 수준을 지정해야 합니다. 가능한 값은 `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` 및 `VERBOSE`입니다. 이 필드는 기록할 Windows 이벤트 유형마다 있어야 합니다.

- monitor(선택 사항)

  로그를 모니터링할지 여부를 나타내는 부울. 기본값은 `true`입니다.

경보

구성 요소에 대해 모니터링할 CloudWatch 경보를 정의합니다.

JSON

```
{
  "alarmName" : "monitoredAlarmName",
  "severity" : HIGH/MEDIUM/LOW
}
```

속성

- alarmName(필수)

  구성 요소에 대해 모니터링할 CloudWatch 경보의 이름입니다.
- severity(선택 사항)

  경보가 울릴 때의 중단 정도를 나타냅니다.

구성 요소 구성 예제

다음 예는 관련 AWS 서비스에 대한 구성 요소 구성을 JSON 형식으로 보여줍니다.

Amazon Elastic Compute Cloud(EC2) 인스턴스

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed"
    }
  ],
  "logs" : [
    {
      "logGroupName" : "my_log_group",
      "logPath" : "C:\\LogFolder\\*",
      "logType" : "APPLICATION",
      "monitor" : true
    },
    {
      "logGroupName" : "my_log_group_2",
      "logPath" : "C:\\LogFolder2\\*",
      "logType" : "IIS",
      "encoding" : "utf-8"
    }
  ],
  "windowsEvents" : [
    {
      "logGroupName" : "my_log_group_3",
      "eventName" : "Application",
      "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "my_log_group_4",
      "eventName" : "System",
      "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
      "monitor" : true
  }],
   "alarms" : [
    {
      "alarmName" : "my_instance_alarm_1",
      "severity" : "HIGH"
```

```
    },
    {
      "alarmName" : "my_instance_alarm_2",
      "severity" : "LOW"
    }
  ]
}
```

Amazon Relational Database(RDS) 인스턴스

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "BurstBalance",
      "monitor" : true
    }, {
      "alarmMetricName" : "WriteThroughput",
      "monitor" : false
    }
  ],

  "alarms" : [
    {
      "alarmName" : "my_rds_instance_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Elastic Load Balancing(ELB)

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "EstimatedALBActiveConnectionCount",
    }, {
      "alarmMetricName" : "HTTPCode_Backend_5XX"
    }
  ],
  "instances" : {
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
      }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_2",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
      }
    ]
  },
```

```
  "alarms" : [
    {
      "alarmName" : "my_elb_alarm",
      "severity" : "HIGH"
    }
  ]
}
```

애플리케이션 ELB

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "ActiveConnectionCount",
    }, {
      "alarmMetricName" : "TargetResponseTime"
    }
  ],
  "instances" : {
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
      }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_2",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
      }
    ]
  },

  "alarms" : [
    {
      "alarmName" : "my_alb_alarm",
      "severity" : "LOW"
    }
  ]
}
```

Amazon EC2 Auto Scaling(ASG)

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "CPUCreditBalance",
    }, {
      "alarmMetricName" : "EBSIOBalance%"
    }
  ],
  "instances" : {
    "alarmMetrics" : [
```

```
      {
        "alarmMetricName" : "CPUUtilization",
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
      }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_2",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
      }
    ]
  },
  "alarms" : [
    {
      "alarmName" : "my_asg_alarm",
      "severity" : "LOW"
    }
  ]
}
```

Amazon Simple Queue Service(SQS)

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "ApproximateAgeOfOldestMessage"
    }, {
      "alarmMetricName" : "NumberOfEmptyReceives"
    }
  ],
  "alarms" : [
    {
      "alarmName" : "my_sqs_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

고객이 그룹화한 EC2 인스턴스

```
{
  "instances" : {
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
```

```
      }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_2",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
      }
    ]
  },
  "alarms" : [
    {
      "alarmName" : "my_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

할당량

다음 표에는 CloudWatch Application Insights for .NET and SQL Server에 대한 기본 할당량이 나와 있습니다. 별도로 표시되지 않는 한, AWS 리전별로 각 할당량이 적용됩니다. 서비스 할당량을 늘리도록 요청하려면 AWS Support에 문의하십시오. 많은 서비스에는 변경할 수 없는 할당량이 포함되어 있습니다. 특정 서비스의 할당량에 대한 자세한 내용은 해당 서비스 설명서를 참조하십시오.

리소스	기본 할당량
API 요청	모든 API 작업이 5TPS로 조절됨
애플리케이션	계정당 10
로그 스트림	리소스당 5
문제별 관찰	대시보드당 20 DescribeProblemObservations 작업당 40
지표	리소스당 10
리소스	애플리케이션당 30

Amazon CloudWatch Application Insights for .NET and SQL Server에서 지원되는 로그 및 지표

다음 목록에서는 .NET 및 SQL Server용 CloudWatch Application Insights에서 지원되는 로그 및 지표를 보여줍니다.

CloudWatch Application Insights for .NET and SQL Server는 다음 로그를 지원합니다.

- Microsoft 인터넷 정보 서비스(IIS) 로그
- EC2상의 SQL Server에 대한 오류 로그
- 사용자 지정 .NET 애플리케이션 로그(예: Log4Net)
- Windows 로그(시스템, 응용 프로그램 및 보안)와, 응용 프로그램 및 서비스 로그가 포함된 Windows 이벤트 로그

CloudWatch Application Insights for .NET and SQL Server는 다음 애플리케이션 구성 요소에 대한 지표를 지원합니다.

- Amazon Elastic Compute Cloud(EC2) (p. 301)
- Elastic Load Balancer(ELB) (p. 307)
- 애플리케이션 ELB (p. 307)
- Amazon EC2 Auto Scaling 그룹 (p. 307)
- Amazon Simple Queue Server(SQS) (p. 308)
- Amazon Relational Database Service(RDS) (p. 308)
- 데이터 포인트 요구 사항이 있는 지표 (p. 309)
- 권장 지표 (p. 313)
- 성능 카운터 지표 (p. 315)

Amazon Elastic Compute Cloud(EC2)

CloudWatch 기본 지표

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_System

CloudWatch 에이전트 지표(Windows Server)

.NET CLR 예외 발생된 예외 수

.NET CLR 예외 발생된 예외 수/초

.NET CLR 예외 필터 수/초

.NET CLR 예외 최종 수/초

.NET CLR 예외 발생 깊이 파악/초

.NET CLR Interop CCW 수

.NET CLR Interop 스텁 수

.NET CLR Interop TLB 내보내기 횟수/초

.NET CLR Interop TLB 가져오기 횟수/초

.NET CLR Interop 마샬링 수

.NET CLR Jit Jit 시간(%)

.NET CLR Jit 표준 Jit 실패

.NET CLR 로딩 로딩 시간(%)

.NET CLR 로딩 로드 실패 비율

.NET CLR LocksAndThreads 경합 속도/초

.NET CLR LocksAndThreads 대기열 길이/초

.NET CLR 메모리 수 총 커밋된 바이트 수

.NET CLR 메모리 GC의 시간(%)

.NET CLR 네트워킹 4.0.0.0 HttpWebRequest 평균 대기열 시간

.NET CLR 네트워킹 4.0.0.0 중단된 HttpWebRequests/초

.NET CLR 네트워킹 4.0.0.0 실패한 HttpWebRequests/초

.NET CLR 네트워킹 4.0.0.0 대기 중인 HttpWebRequests/초

ASP.NET 응용 프로그램 재시작

ASP.NET 응용 프로그램 오류 합계/초

ASP.NET 응용 프로그램 오류 실행 중 처리되지 않은 오류/초

ASP.NET 응용 프로그램 응용 프로그램 대기열의 요청

ASP.NET 응용 프로그램 요청/초

ASP.NET 요청 대기 시간

ASP.NET 대기 중인 요청

HTTP 서비스 요청 대기열 CurrentQueueSize

LogicalDisk % 사용 가능한 공간

사용 중인 메모리 % 커밋된 바이트

사용 가능한 메모리(MB)

메모리 페이지/초

네트워크 인터페이스 바이트 합계/초

페이징된 파일 % 사용량

PhysicalDisk % 디스크 시간

PhysicalDisk 평균 디스크 대기열 길이

PhysicalDisk 평균 디스크 초/읽기

PhysicalDisk 평균 디스크 초/쓰기

PhysicalDisk 디스크 읽기 바이트/초

PhysicalDisk 디스크 읽기/초

PhysicalDisk 디스크 쓰기 바이트/초

PhysicalDisk 디스크 쓰기/초

프로세서 % 유휴 시간

프로세서 % 중단 시간

프로세서 % 프로세서 시간

프로세서 % 사용자 시간

SQLServer:액세스 방법 전달된 레코드/초

SQL Server:액세스 방법 전체 스캔/초

SQLServer:액세스 방법 페이지 분할/초

SQLServer:버퍼 관리자 버퍼 캐시 적중률

SQLServer:버퍼 관리자 페이지 예상 수명

SQLServer:차단된 일반 통계 프로세스

SQLServer:일반 통계 사용자 연결

SQLServer:래치 평균 래치 대기 시간(ms)

SQLServer:잠금 평균 대기 시간(ms)

SQLServer:잠금 잠금 제한 시간/초

SQLServer:잠금 잠금 대기/초

SQLServer:잠금 교착 상태 수/초

SQLServer:메모리 관리자 보류 중인 메모리 부여

SQLServer:SQL 통계 배치 요청/초

SQLServer:SQL 통계 SQL 컴파일/초

SQLServer:SQL 통계 SQL 재컴파일/초

시스템 프로세서 대기열 길이

TCPv4 연결 설정

TCPv6 연결 설정

W3SVC_W3WP 파일 캐시 플러시

W3SVC_W3WP 파일 캐시 누락

W3SVC_W3WP 요청/초

W3SVC_W3WP URI 캐시 플러시

W3SVC_W3WP URI 캐시 누락

웹 서비스 수신된 바이트/초

웹 서비스 전송된 바이트/초

웹 서비스 연결 시도/초

웹 서비스 현재 연결

웹 서비스 요청 가져오기/초

웹 서비스 POST 요청/초

CloudWatch 에이전트 지표(Linux 서버)

cpu_time_active

cpu_time_guest

cpu_time_guest_nice

cpu_time_idle

cpu_time_iowait

cpu_time_irq

cpu_time_nice

cpu_time_softirq

cpu_time_steal

cpu_time_system

cpu_time_user

cpu_usage_active

cpu_usage_guest

cpu_usage_guest_nice

cpu_usage_idle

cpu_usage_iowait

cpu_usage_irq

cpu_usage_nice

cpu_usage_softirq

cpu_usage_steal

cpu_usage_system

cpu_usage_user

disk_free

disk_inodes_free

disk_inodes_used

disk_used

disk_used_percent

diskio_io_time

diskio_iops_in_progress

diskio_read_bytes

diskio_read_time

diskio_reads

diskio_write_bytes

diskio_write_time

diskio_writes

mem_active

mem_available

mem_available_percent

mem_buffered

mem_cached

mem_free

mem_inactive

mem_used

mem_used_percent

net_bytes_recv

net_bytes_sent

net_drop_in

net_drop_out

net_err_in

net_err_out

net_packets_recv

net_packets_sent

netstat_tcp_close

netstat_tcp_close_wait

netstat_tcp_closing

netstat_tcp_established

netstat_tcp_fin_wait1

netstat_tcp_fin_wait2

netstat_tcp_last_ack

netstat_tcp_listen

netstat_tcp_none

netstat_tcp_syn_recv

netstat_tcp_syn_sent

netstat_tcp_time_wait

netstat_udp_socket

processes_blocked

processes_dead

processes_idle

processes_paging

processes_running

processes_sleeping

processes_stopped

processes_total

processes_total_threads

processes_wait

processes_zombies

swap_free

swap_used

swap_used_percent

Elastic Load Balancer(ELB)

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

RequestCount

UnHealthyHostCount

애플리케이션 ELB

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

지연 시간

RequestCount

SurgeQueueLength

UnHealthyHostCount

Amazon EC2 Auto Scaling 그룹

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_System

Amazon Simple Queue Server(SQS)

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

Amazon Relational Database Service(RDS)

BurstBalance

CPUCreditBalance

CPUUtilization

DatabaseConnections

DiskQueueDepth

FailedSQLServerAgentJobsCount

FreeStorageSpace

FreeableMemory

NetworkReceiveThroughput

NetworkTransmitThroughput

ReadIOPS

ReadLatency

ReadThroughput

WriteIOPS

WriteLatency

WriteThroughput

데이터 포인트 요구 사항이 있는 지표

경보가 발생하는 명확한 기본 임계값이 없는 지표의 경우, Application Insights는 경보가 발생하는 데 필요한 임계값을 예측할 수 있을 만큼의 충분한 데이터 포인트가 지표에 확보될 때까지 기다립니다. 경보가 생성되기 전에 CloudWatch Application Insights가 확인하는 지표 데이터 포인트 요구 사항은 다음과 같습니다.

- 지표에 지난 15 ~ 2일 동안의 데이터 포인트가 100개 이상 있는 경우
- 지표에 마지막 날 이후 데이터 포인트가 100개 이상 있는 경우

다음 지표는 이러한 데이터 포인트 요구 사항을 따릅니다. CloudWatch 에이전트 지표는 경보를 생성하는 데 최대 1시간이 걸립니다.

AWS/ApplicationELB

ActiveConnectionCount

ConsumedLCUs

HTTPCode_ELB_4XX_Count

HTTPCode_Target_2XX_Count

HTTPCode_Target_3XX_Count

HTTPCode_Target_4XX_Count

HTTPCode_Target_5XX_Count

NewConnectionCount

ProcessedBytes

TargetResponseTime

UnHealthyHostCount

AWS/AutoScaling

GroupDesiredCapacity

GroupInServiceInstances

GroupMaxSize

GroupMinSize

GroupPendingInstances

GroupStandbyInstances

GroupTerminatingInstances

GroupTotalInstances

AWS/EC2

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

AWS/ELB

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

지연 시간

RequestCount

SurgeQueueLength

UnHealthyHostCount

AWS/RDS

CPUCreditBalance

DatabaseConnections

DiskQueueDepth

FreeStorageSpace

FreeableMemory

NetworkReceiveThroughput

NetworkTransmitThroughput

ReadIOPS

ReadThroughput

WriteIOPS

WriteThroughput

AWS/SQS

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

CWAgent

LogicalDisk % 사용 가능한 공간

사용 중인 메모리 % 커밋된 바이트

사용 가능한 메모리(MB)

네트워크 인터페이스 바이트 합계/초

페이징된 파일 % 사용량

PhysicalDisk % 디스크 시간

PhysicalDisk 평균 디스크 초/읽기

PhysicalDisk 평균 디스크 초/쓰기

PhysicalDisk 디스크 읽기 바이트/초

PhysicalDisk 디스크 읽기/초

PhysicalDisk 디스크 쓰기 바이트/초

PhysicalDisk 디스크 쓰기/초

프로세서 % 유휴 시간

프로세서 % 중단 시간

프로세서 % 프로세서 시간

프로세서 % 사용자 시간

SQLServer:액세스 방법 전달된 레코드/초

SQLServer:액세스 방법 페이지 분할/초

SQLServer:버퍼 관리자 버퍼 캐시 적중률

SQLServer:버퍼 관리자 페이지 예상 수명

SQLServer:차단된 일반 통계 프로세스

SQLServer:SQL 통계 배치 요청/초

SQLServer:SQL 통계 SQL 컴파일/초

SQLServer:SQL 통계 SQL 재컴파일/초

시스템 프로세서 대기열 길이

TCPv4 연결 설정

TCPv6 연결 설정

권장 지표

다음 표에는 각 구성 요소 유형에 대해 권장되는 지표가 나와 있습니다.

구성 요소 유형	워크로드 유형	권장 지표
EC2 인스턴스(Windows Server)	Microsoft IIS/.NET 웹 프런트 엔드	CPUUtilization StatusCheckFailed 프로세서 % 프로세서 시간 사용 중인 메모리 % 커밋된 바이트 사용 가능한 메모리(MB) .NET CLR 예외 발생된 예외 수/초 .NET CLR 메모리 수 총 커밋된 바이트 수 .NET CLR 메모리 GC의 시간(%) ASP.NET 응용 프로그램 응용 프로그램 대기열의 요청 ASP.NET 대기 중인 요청 ASP.NET 응용 프로그램 재시작
	Microsoft SQL Server 데이터베이스 티어	CPUUtilization StatusCheckFailed 프로세서 % 프로세서 시간 사용 중인 메모리 % 커밋된 바이트 사용 가능한 메모리(MB) 페이징된 파일 % 사용량 시스템 프로세서 대기열 길이 네트워크 인터페이스 바이트 합계/초 PhysicalDisk % 디스크 시간 SQLServer:버퍼 관리자 버퍼 캐시 적중률 SQLServer:버퍼 관리자 페이지 예상 수명 SQLServer:차단된 일반 통계 프로세스

구성 요소 유형	워크로드 유형	권장 지표
		SQLServer:일반 통계 사용자 연결 SQLServer:잠금 교착 상태 수/초 SQLServer:SQL 통계 배치 요청/초
	.NET 작업자 풀/중간 티어	CPUUtilization StatusCheckFailed 프로세서 % 프로세서 시간 사용 중인 메모리 % 커밋된 바이트 사용 가능한 메모리(MB) .NET CLR 예외 발생된 예외 수/초 .NET CLR 메모리 수 총 커밋된 바이트 수 .NET CLR 메모리 GC의 시간(%)
	.NET Core 티어	CPUUtilization StatusCheckFailed 프로세서 % 프로세서 시간 사용 중인 메모리 % 커밋된 바이트 사용 가능한 메모리(MB)
EC2 인스턴스(Linux 서버)	.NET Core 티어 또는 SQL Server 데이터베이스 티어	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent
Classic ELB	모두	HTTPCode_Backend_4XX HTTPCode_Backend_5XX 지연 시간 SurgeQueueLength UnHealthyHostCount

구성 요소 유형	워크로드 유형	권장 지표
애플리케이션 ELB	모두	HTTPCode_Target_4XX_Count HTTPCode_Target_5XX_Count TargetResponseTime UnHealthyHostCount
RDS 인스턴스	모두	CPUUtilization ReadLatency WriteLatency BurstBalence FailedSQLServerAgentJobsCount
SQS 대기열	모두	ApproximateAgeOfOldestMessage ApproximateNumberOfMessagesVisible NumberOfMessagesSent

성능 카운터 지표

성능 카운터 지표는 해당 성능 카운터 세트가 Windows 인스턴스에 설치된 경우에만 인스턴스에 권장됩니다.

성능 카운터 지표 이름	성능 카운터 집합 이름
.NET CLR 예외 발생된 예외 수	.NET CLR 예외
.NET CLR 예외 발생된 예외 수/초	.NET CLR 예외
.NET CLR 예외 필터 수/초	.NET CLR 예외
.NET CLR 예외 최종 수/초	.NET CLR 예외
.NET CLR 예외 발생 깊이 파악/초	.NET CLR 예외
.NET CLR Interop CCW 수	.NET CLR Interop
.NET CLR Interop 스텁 수	.NET CLR Interop
.NET CLR Interop TLB 내보내기 횟수/초	.NET CLR Interop
.NET CLR Interop TLB 가져오기 횟수/초	.NET CLR Interop
.NET CLR Interop 마샬링 수	.NET CLR Interop
.NET CLR Jit Jit 시간(%)	.NET CLR Jit
.NET CLR Jit 표준 Jit 실패	.NET CLR Jit
.NET CLR 로딩 로딩 시간(%)	.NET CLR 로딩

성능 카운터 지표 이름	성능 카운터 집합 이름
.NET CLR 로딩 로드 실패 비율	.NET CLR 로딩
.NET CLR LocksAndThreads 경합 속도/초	.NET CLR LocksAndThreads
.NET CLR LocksAndThreads 대기열 길이/초	.NET CLR LocksAndThreads
.NET CLR 메모리 수 총 커밋된 바이트 수	.NET CLR 메모리
.NET CLR 메모리 GC의 시간(%)	.NET CLR 메모리
.NET CLR 네트워킹 4.0.0.0 HttpWebRequest 평균 대기열 시간	.NET CLR 네트워킹 4.0.0.0
.NET CLR 네트워킹 4.0.0.0 중단된 HttpWebRequests/초	.NET CLR 네트워킹 4.0.0.0
.NET CLR 네트워킹 4.0.0.0 실패한 HttpWebRequests/초	.NET CLR 네트워킹 4.0.0.0
.NET CLR 네트워킹 4.0.0.0 대기 중인 HttpWebRequests/Sec	.NET CLR 네트워킹 4.0.0.0
ASP.NET 응용 프로그램 재시작	ASP.NET
ASP.NET 응용 프로그램 오류 합계/초	ASP.NET 응용 프로그램
ASP.NET 응용 프로그램 오류 실행 중 처리되지 않은 오류/초	ASP.NET 응용 프로그램
ASP.NET 응용 프로그램 응용 프로그램 대기열의 요청	ASP.NET 응용 프로그램
ASP.NET 응용 프로그램 요청/초	ASP.NET 응용 프로그램
ASP.NET 요청 대기 시간	ASP.NET
ASP.NET 대기 중인 요청	ASP.NET
HTTP 서비스 요청 대기열 CurrentQueueSize	HTTP 서비스 요청 대기열
LogicalDisk % 사용 가능한 공간	LogicalDisk
사용 중인 메모리 % 커밋된 바이트	Memory
사용 가능한 메모리(MB)	Memory
메모리 페이지/초	Memory
네트워크 인터페이스 바이트 합계/초	네트워크 인터페이스
페이징된 파일 % 사용량	페이징된 파일
PhysicalDisk % 디스크 시간	PhysicalDisk
PhysicalDisk 평균 디스크 대기열 길이	PhysicalDisk
PhysicalDisk 평균 디스크 초/읽기	PhysicalDisk
PhysicalDisk 평균 디스크 초/쓰기	PhysicalDisk
PhysicalDisk 디스크 읽기 바이트/초	PhysicalDisk

성능 카운터 지표 이름	성능 카운터 집합 이름
PhysicalDisk 디스크 읽기/초	PhysicalDisk
PhysicalDisk 디스크 쓰기 바이트/초	PhysicalDisk
PhysicalDisk 디스크 쓰기/초	PhysicalDisk
프로세서 % 유휴 시간	Processor
프로세서 % 중단 시간	Processor
프로세서 % 프로세서 시간	Processor
프로세서 % 사용자 시간	Processor
SQLServer:액세스 방법 전달된 레코드/초	SQLServer:액세스 방법
SQLServer:액세스 방법 전체 스캔/초	SQLServer:액세스 방법
SQLServer:액세스 방법 페이지 분할/초	SQLServer:액세스 방법
SQLServer:버퍼 관리자 버퍼 캐시 적중률	SQLServer:버퍼 관리자
SQLServer:버퍼 관리자 페이지 예상 수명	SQLServer:버퍼 관리자
SQLServer:차단된 일반 통계 프로세스	SQLServer:일반 통계
SQLServer:일반 통계 사용자 연결	SQLServer:일반 통계
SQLServer:래치 평균 래치 대기 시간(ms)	SQLServer:래치
SQLServer:잠금 평균 대기 시간(ms)	SQLServer:잠금
SQLServer:잠금 잠금 제한 시간/초	SQLServer:잠금
SQLServer:잠금 잠금 대기/초	SQLServer:잠금
SQLServer:잠금 교착 상태 수/초	SQLServer:잠금
SQLServer:메모리 관리자 보류 중인 메모리 부여	SQLServer:메모리 관리자
SQLServer:SQL 통계 배치 요청/초	SQLServer:SQL 통계
SQLServer:SQL 통계 SQL 컴파일/초	SQLServer:SQL 통계
SQLServer:SQL 통계 SQL 재컴파일/초	SQLServer:SQL 통계
시스템 프로세서 대기열 길이	시스템
TCPv4 연결 설정	TCPv4
TCPv6 연결 설정	TCPv6
W3SVC_W3WP 파일 캐시 플러시	W3SVC_W3WP
W3SVC_W3WP 파일 캐시 누락	W3SVC_W3WP
W3SVC_W3WP 요청/초	W3SVC_W3WP
W3SVC_W3WP URI 캐시 플러시	W3SVC_W3WP
W3SVC_W3WP URI 캐시 누락	W3SVC_W3WP

성능 카운터 지표 이름	성능 카운터 집합 이름
웹 서비스 수신된 바이트/초	웹 서비스
웹 서비스 전송된 바이트/초	웹 서비스
웹 서비스 연결 시도/초	웹 서비스
웹 서비스 현재 연결	웹 서비스
웹 서비스 요청 가져오기/초	웹 서비스
웹 서비스 POST 요청/초	웹 서비스

CloudWatch 지표를 게시하는 AWS 서비스

다음 AWS 서비스는 지표를 CloudWatch에 게시합니다. 지표 및 차원에 대한 자세한 내용은 지정된 설명서를 참조하십시오.

서비스	네임스페이스	설명서
Amazon Managed Apache Cassandra Service	`AWS/Cassandra`	Amazon Managed Cassandra Service 지표 및 차원
Amazon API Gateway	`AWS/ApiGateway`	Amazon CloudWatch를 사용한 API 실행 모니터링
AppStream 2.0	`AWS/AppStream`	Amazon AppStream 2.0 리소스 모니터링
AWS AppSync	`AWS/AppSync`	CloudWatch 지표
Amazon Athena	`AWS/Athena`	CloudWatch 지표를 사용한 Athena 쿼리 모니터링
AWS Billing and Cost Management	`AWS/Billing`	알림을 사용한 요금 모니터링
ACM Private CA	`AWS/ACMPrivateCA`	지원되는 CloudWatch 지표
AWS Chatbot	`AWS/Chatbot`	Amazon CloudWatch를 사용한 AWS Chatbot 모니터링
Amazon CloudFront	`AWS/CloudFront`	CloudWatch를 사용한 CloudFront 활동 모니터링
AWS CloudHSM	`AWS/CloudHSM`	CloudWatch 지표 가져오기
Amazon CloudSearch	`AWS/CloudSearch`	Amazon CloudWatch를 사용한 Amazon CloudSearch 도메인 모니터링
Amazon CloudWatch Logs	`AWS/Logs`	CloudWatch 지표를 사용한 사용량 모니터링
AWS CodeBuild	`AWS/CodeBuild`	AWS CodeBuild 모니터링
Amazon Cognito	`AWS/Cognito`	Amazon Cognito 모니터링
Amazon Connect	`AWS/Connect`	Amazon CloudWatch 지표를 통해 Amazon Connect 모니터링
AWS DataSync	`AWS/DataSync`	작업 모니터링
AWS Database Migration Service	`AWS/DMS`	AWS DMS 작업 모니터링
AWS Direct Connect	`AWS/DX`	Amazon CloudWatch로 모니터링
Amazon DocumentDB	`AWS/DocDB`	Amazon DocumentDB 지표

서비스	네임스페이스	설명서
Amazon DynamoDB	`AWS/DynamoDB`	DynamoDB 지표 및 차원
Amazon EC2	`AWS/EC2`	CloudWatch를 사용한 인스턴스 모니터링
Amazon EC2 스팟 집합	`AWS/EC2Spot`	스팟 집합에 대한 CloudWatch 지표
Amazon EC2 Auto Scaling	`AWS/AutoScaling`	CloudWatch를 사용한 Auto Scaling 그룹 및 인스턴스 모니터링
AWS Elastic Beanstalk	`AWS/ElasticBeanstalk`	환경에 대한 Amazon CloudWatch 사용자 지정 지표 게시
Amazon Elastic Block Store	`AWS/EBS`	Amazon EBS에 대한 Amazon CloudWatch 지표
Amazon Elastic Container Service	`AWS/ECS`	Amazon ECS CloudWatch 지표
Amazon Elastic File System	`AWS/EFS`	CloudWatch를 사용한 모니터링
Amazon Elastic Inference	`AWS/ElasticInference`	CloudWatch 지표를 사용하여 Amazon EI 모니터링
Elastic Load Balancing	`AWS/ApplicationELB`	Application Load Balancer에 대한 CloudWatch 지표
Elastic Load Balancing	`AWS/ELB`	Classic Load Balancer에 대한 CloudWatch 지표
Elastic Load Balancing	`AWS/NetworkELB`	Network Load Balancer에 대한 CloudWatch 지표
Amazon Elastic Transcoder	`AWS/ElasticTranscoder`	Amazon CloudWatch로 모니터링
Amazon ElastiCache for Memcached	`AWS/ElastiCache`	CloudWatch 지표를 사용한 사용량 모니터링
Redis용 Amazon ElastiCache	`AWS/ElastiCache`	CloudWatch 지표를 사용한 사용량 모니터링
Amazon Elasticsearch Service	`AWS/ES`	CloudWatch를 사용한 클러스터 지표 및 통계 모니터링
Amazon EMR	`AWS/ElasticMapReduce`	CloudWatch를 사용한 지표 모니터링
AWS Elemental MediaConnect	`AWS/MediaConnect`	Amazon CloudWatch를 사용한 MediaConnect 모니터링
AWS Elemental MediaConvert	`AWS/MediaConvert`	CloudWatch 지표

서비스	네임스페이스	설명서
AWS Elemental MediaPackage	`AWS/MediaPackage`	CloudWatch 지표
AWS Elemental MediaTailor	`AWS/MediaTailor`	Amazon CloudWatch를 사용한 AWS Elemental MediaTailor 모니터링
Amazon EventBridge	`AWS/Events`	CloudWatch 지표를 사용한 사용량 모니터링
Amazon FSx for Lustre	`AWS/FSx`	Amazon FSx for Lustre 모니터링
Amazon FSx for Windows File Server	`AWS/FSx`	Amazon FSx for Windows File Server 모니터링
Amazon GameLift	`AWS/GameLift`	CloudWatch를 사용한 Amazon GameLift 모니터링
AWS Glue	`AWS/Glue`	CloudWatch 지표를 사용한 AWS Glue 모니터링
AWS Ground Station	`AWS/GroundStation`	Amazon CloudWatch를 사용한 지표
Amazon Inspector	`AWS/Inspector`	CloudWatch를 사용한 Amazon Inspector 모니터링
AWS IoT	`AWS/IoT`	AWS IoT 지표 및 차원
AWS IoT Analytics	`AWS/IoTAnalytics`	네임스페이스, 지표 및 차원
AWS IoT Things Graph	`AWS/ThingsGraph`	지표
AWS Key Management Service	`AWS/KMS`	CloudWatch를 사용한 모니터링
Amazon Kinesis Data Analytics	`AWS/KinesisAnalytics`	Kinesis Data Analytics: CloudWatch를 사용하여 모니터링 Java 애플리케이션용 Kinesis Data Analytics: Amazon Kinesis Data Analytics 지표 및 차원 보기
Amazon Kinesis Data Firehose	`AWS/Firehose`	CloudWatch 지표를 사용한 Kinesis Data Firehose 모니터링
Amazon Kinesis Data Streams	`AWS/Kinesis`	Amazon CloudWatch를 사용한 Amazon Kinesis Data Streams 모니터링
Amazon Kinesis 비디오 스트림	`AWS/KinesisVideo`	CloudWatch를 사용한 Kinesis 비디오 스트림 지표 모니터링
AWS Lambda	`AWS/Lambda`	AWS Lambda 지표
Amazon Lex	`AWS/Lex`	CloudWatch를 사용한 Amazon Lex 모니터링
Amazon Machine Learning	`AWS/ML`	CloudWatch 지표를 사용한 Amazon ML 모니터링

서비스	네임스페이스	설명서
Amazon Managed Streaming for Apache Kafka	`AWS/Kafka`	Amazon CloudWatch를 사용한 Amazon MSK 모니터링
Amazon MQ	`AWS/AmazonMQ`	Amazon CloudWatch를 사용한 Amazon MQ 브로커 모니터링
Amazon Neptune	`AWS/Neptune`	CloudWatch를 사용한 Neptune 모니터링
AWS OpsWorks	`AWS/OpsWorks`	Amazon CloudWatch를 사용한 스택 모니터링
Amazon Polly	`AWS/Polly`	CloudWatch와 Amazon Polly의 통합
Amazon QLDB	`AWS/QLDB`	Amazon QLDB 지표 및 차원
Amazon Redshift	`AWS/Redshift`	Amazon Redshift 성능 데이터
Amazon Relational Database Service	`AWS/RDS`	Amazon CloudWatch를 사용한 모니터링
AWS RoboMaker	`AWS/Robomaker`	Amazon CloudWatch로 AWS RoboMaker 모니터링
Amazon Route 53	`AWS/Route53`	Amazon Route 53 모니터링
Amazon SageMaker	`AWS/SageMaker`	CloudWatch를 사용한 Amazon SageMaker 모니터링
Enterprise Support에 대한 AWS SDK 지표	`AWS/SDKMetrics`	Enterprise Support에 대한 AWS SDK 지표에서 수집된 지표 및 데이터
AWS Service Catalog	`AWS/ServiceCatalog`	AWS Service Catalog CloudWatch 지표
AWS Shield Advanced	`AWS/DDoSProtection`	CloudWatch를 사용한 모니터링
Amazon Simple Email Service	`AWS/SES`	CloudWatch에서 Amazon SES 이벤트 데이터 가져오기
Amazon Simple Notification Service	`AWS/SNS`	CloudWatch를 사용한 Amazon SNS 모니터링
Amazon Simple Queue Service	`AWS/SQS`	CloudWatch를 사용한 Amazon SQS 대기열 모니터링
Amazon Simple Storage Service	`AWS/S3`	Amazon CloudWatch를 사용한 지표 모니터링
Amazon Simple Workflow Service	`AWS/SWF`	CloudWatch에 대한 Amazon SWF 지표
AWS Step Functions	`AWS/States`	CloudWatch를 사용한 Step Functions 모니터링
AWS Storage Gateway	`AWS/StorageGateway`	게이트웨이 및 리소스 모니터링
AWS 시스템 관리자 명령 실행	`AWS/SSM-RunCommand`	CloudWatch를 사용하여 명령 실행 지표 모니터링

서비스	네임스페이스	설명서
Amazon Textract	`AWS/Textract`	Amazon Textract용 CloudWatch 지표
SFTP를 위한 AWS 전송	`AWS/Transfer`	AWS SFTP CloudWatch 지표
Amazon Translate	`AWS/Translate`	Amazon Translate에 대한 CloudWatch 지표 및 차원
AWS Trusted Advisor	`AWS/TrustedAdvisor`	CloudWatch를 사용한 Trusted Advisor 경보 생성
Amazon VPC	`AWS/NATGateway`	CloudWatch를 사용한 NAT 게이트웨이 모니터링
Amazon VPC	`AWS/TransitGateway`	해당 Transit Gateway용 CloudWatch 지표
Amazon VPC	`AWS/VPN`	CloudWatch를 사용한 모니터링
AWS WAF	`WAF`	CloudWatch를 사용한 모니터링
Amazon WorkMail	`AWS/WorkMail`	Amazon CloudWatch를 사용한 Amazon WorkMail 모니터링
Amazon WorkSpaces	`AWS/WorkSpaces`	CloudWatch 지표를 사용하여 WorkSpaces 모니터링

경보 이벤트 및 EventBridge

CloudWatch 경보에서 경보 상태를 변경할 때마다 CloudWatch에서 Amazon EventBridge로 이벤트를 보냅니다. EventBridge 및 이러한 이벤트를 사용하여 경보 상태가 변경될 때 알림 등의 작업을 수행하는 규칙을 작성할 수 있습니다. 자세한 내용은 Amazon EventBridge란 무엇입니까? 단원을 참조하십시오.

현재 AWS Management 콘솔을 사용하여 CloudWatch에서 이벤트를 기반으로 규칙을 생성해야 합니다. 규칙에서 감시할 이벤트 텍스트에는 사용자 지정 이벤트 패턴 옵션을 사용해야 합니다. AWS SDK 또는 AWS API를 사용하여 CloudWatch에서 이벤트를 기반으로 규칙을 생성할 수도 있습니다.

CloudWatch의 샘플 이벤트

이 단원에는 CloudWatch의 예제 이벤트가 포함되어 있습니다.

단일 지표 경보에 대한 상태 변경


```
{
    "version": "0",
    "id": "c4c1c1c9-6542-e61b-6ef0-8c4d36933a92",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:04:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "configuration": {
            "description": "Goes into alarm when server CPU utilization is too high!",
            "metrics": [
                {
                    "id": "30b6c6b2-a864-43a2-4877-c09a1afc3b87",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "CPUUtilization",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ]
        },
        "previousState": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [0.0666851903306472 (01/10/19 13:46:00)] was not greater than the threshold (50.0)
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-01T13:56:40.985+0000\",\"startDate\":\"2019-10-01T13:46:00.000+0000\",\"statistic
\":\"Average\",\"period\":300,\"recentDatapoints\":[0.0666851903306472],\"threshold
\":50.0}",
```

```
            "timestamp": "2019-10-01T13:56:40.987+0000",
            "value": "OK"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints [99.50160229693434
 (02/10/19 16:59:00)] was greater than the threshold (50.0) (minimum 1 datapoint for OK ->
 ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:04:40.985+0000\",\"startDate\":\"2019-10-02T16:59:00.000+0000\",\"statistic
\":\"Average\",\"period\":300,\"recentDatapoints\":[99.50160229693434],\"threshold
\":50.0}",
            "timestamp": "2019-10-02T17:04:40.989+0000",
            "value": "ALARM"
        }
    }
}
```

지표 수식 경보에 대한 상태 변경

```
{
    "version": "0",
    "id": "2dde0eb1-528b-d2d5-9ca6-6d590caf2329",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:20:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "configuration": {
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "metrics": [
                {
                    "expression": "SUM(METRICS())",
                    "id": "e1",
                    "label": "Total Network Traffic",
                    "returnData": true
                },
                {
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkIn",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkOut",
                            "namespace": "AWS/EC2"
```

```
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                }
            ]
        },
        "previousState": {
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2019-10-02T17:20:03.642+0000",
            "value": "INSUFFICIENT_DATA"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints [45628.0 (02/10/19
 17:10:00)] was greater than the threshold (10000.0) (minimum 1 datapoint for OK -> ALARM
 transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:20:48.551+0000\",\"startDate\":\"2019-10-02T17:10:00.000+0000\",\"period
\":300,\"recentDatapoints\":[45628.0],\"threshold\":10000.0}",
            "timestamp": "2019-10-02T17:20:48.554+0000",
            "value": "ALARM"
        }
    }
}
```

이상 탐지 경보에 대한 상태 변경

```
{
    "version": "0",
    "id": "daafc9f1-bddd-c6c9-83af-74971fcfc4ef",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-03T16:00:04Z",
    "region": "us-east-1",
    "resources": ["arn:aws:cloudwatch:us-east-1:123456789012:alarm:EC2 CPU Utilization
 Anomaly"],
    "detail": {
        "alarmName": "EC2 CPU Utilization Anomaly",
        "state": {
            "value": "ALARM",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints [0.0 (03/10/19
 15:58:00)] was less than the lower thresholds [0.020599444741798756] or greater than the
 upper thresholds [0.3006915352732461] (minimum 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T16:00:04.650+0000\",\"startDate\":\"2019-10-03T15:58:00.000+0000\",\"period
\":60,\"recentDatapoints\":[0.0],\"recentLowerThresholds\":[0.020599444741798756],
\"recentUpperThresholds\":[0.3006915352732461]}",
            "timestamp": "2019-10-03T16:00:04.653+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints
 [0.166666666664241 (03/10/19 15:57:00)] was not less than the lower thresholds
 [0.0206719426210418] or not greater than the upper thresholds [0.30076870222143803]
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T15:59:04.670+0000\",\"startDate\":\"2019-10-03T15:57:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.166666666664241],\"recentLowerThresholds\":
[0.0206719426210418],\"recentUpperThresholds\":[0.30076870222143803]}",
            "timestamp": "2019-10-03T15:59:04.672+0000"
        },
        "configuration": {
            "description": "Goes into alarm if CPU Utilization is out of band",
```

```
            "metrics": [{
                "id": "m1",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 60,
                    "stat": "Average"
                },
                "returnData": true
            }, {
                "id": "ad1",
                "expression": "ANOMALY_DETECTION_BAND(m1, 0.8)",
                "label": "CPUUtilization (expected)",
                "returnData": true
            }]
        }
    }
}
```

높은 카디널리티 로그 수집 및 CloudWatch 임베디드 지표 형식으로 지표 생성

CloudWatch 임베디드 지표 형식을 사용하면 복잡하고 카디널리티가 높은 애플리케이션 데이터를 로그 형태로 수집하고 해당 로그에서 실행 가능한 지표를 생성할 수 있습니다. 세부 로그 이벤트 데이터와 함께 사용자 지정 지표를 포함할 수 있으며, CloudWatch를 통해 사용자 지정 지표가 자동으로 추출되므로 실시간 인시던트 감지를 위해 이를 시각화하고 경보를 전송할 수 있습니다. 또한 CloudWatch Logs Insights를 통해 추출된 지표와 관련된 세부 로그 이벤트를 쿼리하여 운영 이벤트의 근본 원인을 심층적으로 분석할 수 있습니다.

임베디드 지표 형식을 사용하면 Lambda 함수 및 컨테이너와 같은 휘발성 리소스에서 실행 가능한 사용자 지정 지표를 생성할 수 있습니다. 임베디드 지표 형식을 사용하여 이러한 휘발성 리소스에서 로그를 전송함으로써, 이제 별도의 코드를 구성하거나 유지 관리할 필요 없이 손쉽게 사용자 지정 지표를 생성할 수 있으며 로그 데이터에 대한 강력한 분석 기능을 얻을 수 있습니다.

임베디드 지표 형식을 사용하는 경우 클라이언트 라이브러리를 사용하여 로그를 생성할 수 있습니다. 자세한 내용은 클라이언트 라이브러리를 사용하여 임베디드 지표 형식 로그 생성 (p. 329) 단원을 참조하십시오. 또는 수동으로 로그를 생성하고 PutLogEvents API 또는 CloudWatch 에이전트를 사용하여 제출할 수 있습니다.

로그 수집 및 보관과 생성된 사용자 지정 지표에 대한 요금이 발생합니다. 자세한 내용은 Amazon CloudWatch 요금을 참조하십시오.

> Note
>
> 지표 추출을 구성할 때는 사용자 지정 지표 사용 및 해당 청구서에 영향을 미치므로 주의해야 합니다. 높은 카디널리티 차원(예: `requestId`)을 기반으로 지표를 실수로 생성하는 경우 기본적으로 임베디드 지표 형식은 각 고유 차원 조합에 해당하는 사용자 지정 지표를 생성합니다. 자세한 내용은 차원을 참조하십시오.

주제

- 임베디드 지표 형식을 사용하여 로그 생성 (p. 328)
- 콘솔에서 지표 및 로그 보기 (p. 340)

임베디드 지표 형식을 사용하여 로그 생성

다음 방법을 사용하여 임베디드 지표 형식 로그를 생성할 수 있습니다.

- 오픈 소스 클라이언트 라이브러리를 사용하여 로그 생성 및 전송
- 로그를 수동으로 생성한 다음 CloudWatch 에이전트 또는 PutLogEvents API를 사용하여 로그 전송

수동 방법 중 하나를 사용하는 경우 로그는 정의된 JSON 형식을 따라야 합니다.

주제

- 클라이언트 라이브러리를 사용하여 임베디드 지표 형식 로그 생성 (p. 329)

- 수동으로 임베디드 지표 형식 로그 생성 (p. 329)

클라이언트 라이브러리를 사용하여 임베디드 지표 형식 로그 생성

Amazon은 임베디드 지표 형식 로그를 생성하는 데 사용할 수 있는 오픈 소스 클라이언트 라이브러리를 제공합니다. 현재 이러한 라이브러리는 Node.js와 Python에서 사용할 수 있습니다. 다른 언어에 대한 지원이 계획되어 있습니다.

라이브러리 및 사용 방법에 대한 지침은 Github에 있습니다. 다음 목록의 링크를 사용하십시오.

- Node.Js
- Python

수동으로 임베디드 지표 형식 로그 생성

수동으로 생성한 임베디드 지표 형식 로그를 전송하려면 PutLogEvents API 또는 CloudWatch 에이전트를 사용합니다.

이러한 방법 중 하나를 사용하는 경우 임베디드 지표 형식 사양을 따라야 합니다.

주제

- 사양: 임베디드 지표 형식 (p. 329)
- PutLogEvents API를 사용하여 수동으로 생성된 임베디드 지표 형식 로그 전송 (p. 334)
- CloudWatch 에이전트를 사용하여 임베디드 지표 형식 로그 전송 (p. 336)

사양: 임베디드 지표 형식

CloudWatch 임베디드 지표 형식은 구조화된 로그 이벤트에 포함된 지표 값을 자동으로 추출하도록 CloudWatch Logs에 지시하는 데 사용되는 JSON 사양입니다. CloudWatch를 사용하여 추출된 지표 값을 그래프로 만들고 경보를 생성할 수 있습니다.

임베디드 지표 형식 사양 규칙

이 형식 사양의 “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY” 및 “OPTIONAL” 키워드는 키워드 RFC2119에 설명된 대로 해석됩니다.

이 형식 사양의 "JSON", "JSON text", "JSON value", "member", "element", "object", "array", "number", "string", "boolean", "true", "false" 및 "null" 용어는 JavaScript 객체 표기법 RFC8259에 정의된 대로 해석됩니다.

임베디드 지표 형식 사양 PutLogEvents 요청 형식

CloudWatch Logs PutLogEvents API를 사용하여 임베디드 지표 형식 문서를 전송할 때 클라이언트는 다음 로그 형식 헤더를 사용해야 합니다.

```
x-amzn-logs-format: json/emf
```

Lambda에서는 이 헤더를 직접 설정할 필요가 없습니다. JSON을 임베디드 지표 형식의 표준 출력으로 작성하면 충분합니다. Lambda는 타임스탬프 및 요청 ID와 같은 메타데이터를 사용하여 로그 이벤트를 앞에 추가할 수 있지만, 이 메타데이터 뒤의 유효한 임베디드 지표 형식 문서는 유효한 것으로 간주됩니다.

임베디드 지표 형식 문서 구조

이 단원에서는 로그 형식 헤더 `x-amzn-logs-format: json/emf`로 식별되는 임베디드 지표 형식 문서의 구조에 대해 설명합니다. 임베디드 지표 형식 문서는 JavaScript 객체 표기법 RFC8259에 정의되어 있습니다.

별도의 언급이 없는 한 이 사양에 의해 정의된 객체에 추가 멤버가 포함되어서는 안 됩니다. 이 사양에 의해 인식되지 않는 멤버는 무시해야 합니다. 이 사양에 정의된 멤버는 대/소문자를 구분합니다.

임베디드 지표 형식은 표준 CloudWatch Logs 이벤트와 동일한 제한이 적용되며 최대 크기가 256KB로 제한됩니다.

루트 노드

LogEvent 메시지는 LogEvent 메시지 문자열의 시작 또는 끝에 추가 데이터가 없는 유효한 JSON 객체여야 합니다. LogEvent 구조에 대한 자세한 내용은 InputLogEvent를 참조하십시오.

임베디드 지표 형식 문서는 루트 노드에 다음과 같은 최상위 멤버를 포함해야 합니다. 이는 메타데이터 객체 (p. 330) 객체입니다.

```
{
 "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

루트 노드에는 MetricDirective 객체 (p. 330)의 참조로 정의된 모든 ??? (p. 332) 멤버가 포함되어야 합니다.

루트 노드는 위의 요구 사항에 포함되지 않은 다른 멤버를 포함할 수 있습니다. 이러한 멤버의 값은 유효한 JSON 유형이어야 합니다.

메타데이터 객체

`_aws` 멤버는 다운스트림 서비스에 LogEvent를 처리하는 방법을 알려주는 페이로드에 대한 메타데이터를 나타내는 데 사용할 수 있습니다. 값은 객체여야 하며 다음 멤버를 포함해야 합니다.

- CloudWatchMetrics — LogEvent의 루트 노드에서 지표를 추출하도록 CloudWatch에 지시하는데 사용되는 MetricDirective 객체 (p. 330)의 배열입니다.

  ```
  {
    "_aws": {
      "CloudWatchMetrics": [ ... ]
    }
  }
  ```

- Timestamp — 이벤트에서 추출된 지표에 사용되는 타임스탬프를 나타내는 숫자입니다. 값은 1970년 1월 1일 00:00:00 UTC 이후 경과된 시간(밀리초)으로 표시해야 합니다.

  ```
  {
    "_aws": {
      "Timestamp": 1559748430481
    }
  }
  ```

MetricDirective 객체

MetricDirective 객체는 추출하여 CloudWatch에 게시될 지표를 LogEvent가 포함하도록 다운스트림 서비스에 지시합니다. MetricDirectives에는 다음 멤버가 포함되어야 합니다.

- Namespace — 지표의 CloudWatch 네임스페이스를 나타내는 문자열입니다.
- Dimensions — DimensionSet 배열 (p. 331)입니다.
- Metrics — MetricDefinition (p. 331) 객체의 배열입니다. 이 배열에 100개를 초과하는 MetricDefinition 객체가 포함되어서는 안 됩니다.

DimensionSet 배열

DimensionSet는 문서의 모든 지표에 적용될 차원 키를 포함하는 문자열 배열입니다. 또한 이 배열 내의 값은 대상 멤버 (p. 332)라고 하는 루트 노드의 멤버여야 합니다.

DimensionSet에는 9개를 초과하는 차원 키가 포함되어서는 안 됩니다.

대상 멤버에는 문자열 값이 있어야 합니다. 대상 멤버는 지표 ID의 일부로 게시될 차원을 정의합니다. 사용된 모든 DimensionSet는 CloudWatch에 새 지표를 생성합니다. 차원에 대한 자세한 내용은 차원 및 차원을 참조하십시오.

```
{
 "_aws": {
   "CloudWatchMetrics": [
     {
       "Dimensions": [ [ "functionVersion" ] ],
       ...
     }
   ]
 },
 "functionVersion": "$LATEST"
}
```

> **Note**
>
> 지표 추출을 구성할 때는 사용자 지정 지표 사용 및 해당 청구서에 영향을 미치므로 주의해야 합니다. 높은 카디널리티 차원(예: `requestId`)을 기반으로 지표를 실수로 생성하는 경우 기본적으로 임베디드 지표 형식은 각 고유 차원 조합에 해당하는 사용자 지정 지표를 생성합니다. 자세한 내용은 차원을 참조하십시오.

MetricDefinition 객체

MetricDefinition은 다음 멤버를 포함해야 하는 객체입니다.

- Name — 지표 대상 멤버 (p. 332)에 대한 문자열 참조 값 (p. 332)입니다. 지표 대상은 숫자 값을 가져야 합니다.

MetricDefinition 객체에는 다음 멤버가 포함될 수 있습니다.

- Unit — 해당 지표에 대한 측정 단위를 나타내는 선택적 문자열 값입니다. 값은 유효한 CloudWatch 지표 단위여야 합니다. 유효한 단위에 대한 자세한 내용은 MetricDatum을 참조하십시오. 값이 제공되지 않으면 기본값인 NONE으로 가정합니다.

```
{
  "_aws": {
    "CloudWatchMetrics": [
      {
        "Metrics": [
          {
            "Name": "Time",
            "Unit": "Milliseconds"
          }
        ],
```

```
        ...
      }
    ]
  },
  "Time": 1
}
```

참조 값

참조 값은 루트 노드의 대상 멤버 (p. 332) 멤버를 참조하는 문자열 값입니다. 이러한 참조를 RFC6901에 설명된 JSON 포인터와 혼동해서는 안 됩니다. 대상 값은 중첩될 수 없습니다.

대상 멤버

유효한 대상은 루트 노드의 멤버여야 하며 중첩된 객체일 수 없습니다. 예를 들어 `"A.a"`의 _reference_ 값이 다음 멤버와 일치해야 합니다.

```
{ "A.a" }
```

중첩된 멤버와 일치하지 않아야 합니다.

```
{ "A": { "a" } }
```

대상 멤버의 유효한 값은 해당 멤버를 참조하는 내용에 따라 달라집니다. 예를 들어 지표 대상은 숫자 값이어야 하고 차원 대상은 문자열 값이어야 합니다.

임베디드 지표 형식 예제 및 JSON 스키마

다음은 임베디드 지표 형식의 유효한 예입니다.

```
{
  "_aws": {
    "Timestamp": 1574109732004,
    "CloudWatchMetrics": [
      {
        "Namespace": "lambda-function-metrics",
        "Dimensions": [["functionVersion"]],
        "Metrics": [
          {
            "Name": "time",
            "Unit": "Milliseconds"
          }
        ]
      }
    ]
  },
  "functionVersion": "$LATEST",
  "time": 100,
  "requestId": "989ffbf8-9ace-4817-a57c-e4dd734019ee"
}
```

다음 스키마를 사용하여 임베디드 지표 형식 문서의 유효성을 검사할 수 있습니다.

```
{
    "type": "object",
    "title": "Root Node",
    "required": [
        "_aws"
    ],
    "properties": {
        "_aws": {
```

```
            "$id": "#/properties/_aws",
            "type": "object",
            "title": "Metadata",
            "required": [
                "Timestamp",
                "CloudWatchMetrics"
            ],
            "properties": {
                "Timestamp": {
                    "$id": "#/properties/_aws/properties/Timestamp",
                    "type": "integer",
                    "title": "The Timestamp Schema",
                    "examples": [
                        1565375354953
                    ]
                },
                "CloudWatchMetrics": {
                    "$id": "#/properties/_aws/properties/CloudWatchMetrics",
                    "type": "array",
                    "title": "MetricDirectives",
                    "items": {
                        "$id": "#/properties/_aws/properties/CloudWatchMetrics/items",
                        "type": "object",
                        "title": "MetricDirective",
                        "required": [
                            "Namespace",
                            "Dimensions",
                            "Metrics"
                        ],
                        "properties": {
                            "Namespace": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Namespace",
                                "type": "string",
                                "title": "CloudWatch Metrics Namespace",
                                "examples": [
                                    "MyApp"
                                ],
                                "pattern": "^(.*)$"
                            },
                            "Dimensions": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Dimensions",
                                "type": "array",
                                "title": "The Dimensions Schema",
                                "items": {
                                    "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Dimensions/items",
                                    "type": "array",
                                    "title": "DimensionSet",
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items/items",
                                        "type": "string",
                                        "title": "DimensionReference",
                                        "examples": [
                                            "Operation"
                                        ],
                                        "pattern": "^(.*)$"
                                    }
                                }
                            },
                            "Metrics": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Metrics",
                                "type": "array",
```

```
                                "title": "MetricDefinitions",
                                "items": {
                                    "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Metrics/items",
                                    "type": "object",
                                    "title": "MetricDefinition",
                                    "required": [
                                        "Name"
                                    ],
                                    "properties": {
                                        "Name": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Name",
                                            "type": "string",
                                            "title": "MetricName",
                                            "examples": [
                                                "ProcessingLatency"
                                            ],
                                            "pattern": "^(.*)$"
                                        },
                                        "Unit": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Unit",
                                            "type": "string",
                                            "title": "MetricUnit",
                                            "examples": [
                                                "Milliseconds"
                                            ],
                                            "pattern": "^(Seconds|Microseconds|
Milliseconds|Bytes|Kilobytes|Megabytes|Gigabytes|Terabytes|Bits|Kilobits|Megabits|Gigabits|
Terabits|Percent|Count|Bytes\\/Second|Kilobytes\\/Second|Megabytes\\/Second|Gigabytes\\/
Second|Terabytes\\/Second|Bits\\/Second|Kilobits\\/Second|Megabits\\/Second|Gigabits\\/
Second|Terabits\\/Second|Count\\/Second|None)$"
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

PutLogEvents API를 사용하여 수동으로 생성된 임베디드 지표 형식 로그 전송

CloudWatch Logs PutLogEvents API를 사용하여 임베디드 지표 형식 로그를 CloudWatch Logs에 전송할 수 있습니다. PutLogEvents를 호출할 때 지표를 추출해야 한다고 CloudWatch Logs에 지시하려면 다음 HTTP 헤더를 포함해야 합니다.

```
x-amzn-logs-format: json/emf
```

다음은 AWS Java용 SDK를 사용하는 전체 예제입니다.

```
package org.example.basicapp;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsRequest;
```

```
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.InputLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.PutLogEventsRequest;

import java.util.Collections;

public class EmbeddedMetricsExample {
        public static void main(String[] args) {

                final String usage = "To run this example, supply a region id (eg. us-
east-1), log group, and stream name as command line arguments"
                                + "Ex: PutLogEvents <region-id> <log-group-name> <stream-
name>";

                if (args.length != 3) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String regionId = args[0];
                String logGroupName = args[1];
                String logStreamName = args[2];

                CloudWatchLogsClient logsClient =
 CloudWatchLogsClient.builder().region(Region.of(regionId)).build();

                // A sequence token is required to put a log event in an existing stream.
                // Look up the stream to find its sequence token.
                String sequenceToken = getNextSequenceToken(logsClient, logGroupName,
 logStreamName);

                // Build a JSON log using the EmbeddedMetricFormat.
                long timestamp = System.currentTimeMillis();
                String message = "{" +
                                "  \"_aws\": {" +
                                "    \"Timestamp\": " + timestamp  + "," +
                                "    \"CloudWatchMetrics\": [" +
                                "      {" +
                                "        \"Namespace\": \"MyApp\"," +
                                "        \"Dimensions\": [[\"Operation\"], [\"Operation\",
 \"Cell\"]]," +
                                "        \"Metrics\": [{ \"Name\": \"ProcessingLatency\",
 \"Unit\": \"Milliseconds\" }]" +
                                "      }" +
                                "    ]" +
                                "  }," +
                                "  \"Operation\": \"Aggregator\"," +
                                "  \"Cell\": \"001\"," +
                                "  \"ProcessingLatency\": 100" +
                                "}";
                InputLogEvent inputLogEvent = InputLogEvent.builder()
                        .message(message)
                        .timestamp(timestamp)
                        .build();

                // Specify the request parameters.
                PutLogEventsRequest putLogEventsRequest = PutLogEventsRequest.builder()
                        .overrideConfiguration(builder ->
                                // provide the log-format header of json/emf
                                builder.headers(Collections.singletonMap("x-amzn-logs-
format",  Collections.singletonList("json/emf"))))
                        .logEvents(Collections.singletonList(inputLogEvent))
                        .logGroupName(logGroupName)
                        .logStreamName(logStreamName)
                        .sequenceToken(sequenceToken)
                        .build();
```

```
                logsClient.putLogEvents(putLogEventsRequest);

                System.out.println("Successfully put CloudWatch log event");
        }

        private static String getNextSequenceToken(CloudWatchLogsClient logsClient, String
 logGroupName, String logStreamName) {
                DescribeLogStreamsRequest logStreamRequest =
 DescribeLogStreamsRequest.builder()
                        .logGroupName(logGroupName)
                        .logStreamNamePrefix(logStreamName)
                        .build();

                DescribeLogStreamsResponse describeLogStreamsResponse =
 logsClient.describeLogStreams(logStreamRequest);

                // Assume that a single stream is returned since a specific stream name was
                // specified in the previous request.
                return
 describeLogStreamsResponse.logStreams().get(0).uploadSequenceToken();
        }
}
```

CloudWatch 에이전트를 사용하여 임베디드 지표 형식 로그 전송

이 방법을 사용하려면 먼저 전송할 임베디드 지표 형식 로그가 있는 서비스에 대한 CloudWatch 에이전트를 설치한 다음 이벤트 전송을 시작할 수 있습니다.

CloudWatch 에이전트는 1.230621.0 이상 버전이어야 합니다.

> **Note**
>
> Lambda 함수에서 로그를 전송하기 위해 CloudWatch 에이전트를 설치할 필요가 없습니다. Lambda 함수 시간 초과는 자동으로 처리되지 않습니다. 즉, 지표가 플러시되기 전에 함수가 시간 초과되면 해당 호출에 대한 지표가 캡처되지 않습니다.

CloudWatch 에이전트 설치

임베디드 지표 형식 로그를 전송할 각 서비스에 대한 CloudWatch 에이전트를 설치합니다.

EC2에 CloudWatch 에이전트 설치

먼저 인스턴스에 CloudWatch 에이전트를 설치합니다. 자세한 내용은 CloudWatch 에이전트 설치 (p. 190) 단원을 참조하십시오.

에이전트를 설치한 후에는 임베디드 지표 형식 로그에 대해 UDP 또는 TCP 포트에서 수신 대기하도록 에이전트를 구성합니다. 다음은 기본 소켓 `tcp:25888`에서 수신 대기하는 이 구성의 예입니다. 에이전트 구성에 대한 자세한 내용은 CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 (p. 227)을 참조하십시오.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Amazon ECS에 CloudWatch 에이전트 설치

CloudWatch 에이전트를 Amazon ECS에 배포하는 가장 쉬운 방법은 사이드카로 실행하여 애플리케이션과 동일한 작업 정의에서 정의하는 것입니다.

에이전트 구성 파일 생성

CloudWatch 에이전트 구성 파일을 로컬로 생성합니다. 이 예제에서 상대 파일 경로는 `amazon-cloudwatch-agent.json`입니다.

에이전트 구성에 대한 자세한 내용은 CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 (p. 227)을 참조하십시오.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

SSM 파라미터 스토어로 구성 푸시

다음 명령을 입력하여 CloudWatch 에이전트 구성 파일을 AWS Systems Manager(SSM) 파라미터 스토어로 푸시합니다.

```
aws ssm put-parameter \
    --name "cwagentconfig" \
    --type "String" \
    --value "`cat amazon-cloudwatch-agent.json`" \
    --region "{{region}}"
```

작업 정의 구성

CloudWatch 에이전트를 사용하고 TCP 또는 UDP 포트를 노출하도록 작업 정의를 구성합니다. 사용해야 하는 샘플 작업 정의는 네트워킹 모드에 따라 다릅니다.

`webapp`에서 `AWS_EMF_AGENT_ENDPOINT` 환경 변수를 지정합니다. 이 기능은 라이브러리에서 사용되며 에이전트가 수신 대기하는 엔드포인트를 가리켜야 합니다. 또한 `cwagent`에서는 `CW_CONFIG_CONTENT`를 이전 단계에서 생성한 SSM 구성을 가리키는 “valueFrom” 파라미터로 지정합니다.

다음은 브리지 모드의 예입니다. 브리지 모드 네트워킹이 활성화된 경우 에이전트는 `links` 파라미터를 사용하여 애플리케이션에 연결하고 컨테이너 이름을 사용하여 주소를 지정해야 합니다.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
              "links": [ "cwagent" ],
              "image": "my-org/web-app:latest",
              "essential": true,
              "memory": 256,
              "cpu": 256,
              "environment": [{
                "name": "AWS_EMF_AGENT_ENDPOINT",
                "value": "tcp://cwagent:25888"
              }],
          },
          {
              "name": "cwagent",
              "mountPoints": [],
              "image": "amazon/cloudwatch-agent:latest",
              "essential": true,
              "memory": 256,
              "cpu": 256,
              "portMappings": [{
                  "protocol": "tcp",
```

```
                "containerPort": 25888
            }],
            "environment": [{
              "name": "CW_CONFIG_CONTENT",
              "value": "cwagentconfig"
            }],
        }
    ],
}
```

다음은 호스트 모드 또는 awsvpc 모드의 예입니다. 이러한 네트워크 모드에서 실행할 경우 에이전트는 `localhost`를 통해 주소를 지정할 수 있습니다.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
              "image": "my-org/web-app:latest",
              "essential": true,
              "memory": 256,
              "cpu": 256,
              "environment": [{
                "name": "AWS_EMF_AGENT_ENDPOINT",
                "value": "tcp://127.0.0.1:25888"
              }],
          },
          {
              "name": "cwagent",
              "mountPoints": [],
              "image": "amazon/cloudwatch-agent:latest",
              "essential": true,
              "memory": 256,
              "cpu": 256,
              "portMappings": [{
                  "protocol": "tcp",
                  "containerPort": 25888
              }],
              "environment": [{
                "name": "CW_CONFIG_CONTENT",
                "value": "cwagentconfig"
              }],
          }
      ],
}
```

Note

awsvpc 모드에서는 VPC에 퍼블릭 IP 주소를 제공하거나(Fargate만 해당) NAT 게이트웨이를 설정하거나 CloudWatch Logs VPC 엔드포인트를 설정해야 합니다. NAT 설정에 대한 자세한 내용은 NAT 게이트웨이를 참조하십시오. CloudWatch Logs VPC 엔드포인트 설정에 대한 자세한 내용은 인터페이스 VPC 엔드포인트와 함께 CloudWatch Logs 사용을 참조하십시오.
다음은 Fargate 시작 유형을 사용하는 작업에 퍼블릭 IP 주소를 할당하는 방법의 예입니다.

```
aws ecs run-task \
--cluster {{cluster-name}} \
--task-definition cwagent-fargate \
--region {{region}} \
--launch-type FARGATE \
--network-configuration
 "awsvpcConfiguration={subnets=[{{subnetId}}],securityGroups=[{{sgId}}],assignPublicIp=ENABLED}"
```

권한 확인

작업을 실행하는 IAM 역할에 SSM 파라미터 스토어에서 읽을 권한이 있는지 확인합니다. AmazonSSMReadOnlyAccess 정책을 연결하여 이 권한을 추가할 수 있습니다. 이를 위해 다음 명령을 입력합니다.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess \
--role-name CWAgentECSExecutionRole
```

Amazon EKS에 CloudWatch 에이전트 설치

이 클러스터에 CloudWatch Container Insights를 이미 설치한 경우 이 프로세스의 일부를 건너뛸 수 있습니다.

권한

Container Insights를 아직 설치하지 않은 경우 먼저 Amazon EKS 노드에 적절한 IAM 권한이 있는지 확인합니다. CloudWatchAgentServerPolicy가 연결되어 있어야 합니다. 자세한 내용은 사전 조건 확인 (p. 145) 단원을 참조하십시오.

ConfigMap 생성

에이전트에 대한 ConfigMap을 생성합니다. ConfigMap은 또한 TCP 또는 UDP 포트에서 수신 대기하도록 에이전트에 지시합니다. 다음 ConfigMap을 사용합니다.

```
# cwagent-emf-configmap.yaml
apiVersion: v1
data:
  # Any changes here must not break the JSON format
  cwagentconfig.json: |
    {
      "agent": {
        "omit_hostname": true
      },
      "logs": {
        "metrics_collected": {
          "emf": { }
        }
      }
    }
kind: ConfigMap
metadata:
  name: cwagentemfconfig
  namespace: default
```

Container Insights를 이미 설치한 경우 기존 ConfigMap에 다음 `"emf": { }` 줄을 추가합니다.

ConfigMap 적용

다음 명령을 입력하여 ConfigMap을 적용합니다.

```
kubectl apply -f cwagent-emf-configmap.yaml
```

에이전트 배포

CloudWatch 에이전트를 사이드카로 배포하려면 다음 예제와 같이 포드 정의에 에이전트를 추가합니다.

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
```

```
  namespace: default
spec:
  containers:
    # Your container definitions go here
    - name: web-app
      image: my-org/web-app:latest
    # CloudWatch Agent configuration
    - name: cloudwatch-agent
      image: amazon/cloudwatch-agent:latest
      imagePullPolicy: Always
      resources:
        limits:
          cpu: 200m
          memory: 100Mi
        requests:
          cpu: 200m
          memory: 100Mi
      volumeMounts:
        - name: cwagentconfig
          mountPath: /etc/cwagentconfig
      ports:
  # this should match the port configured in the ConfigMap
        - protocol: TCP
          hostPort: 25888
          containerPort: 25888
  volumes:
    - name: cwagentconfig
      configMap:
        name: cwagentemfconfig
```

CloudWatch 에이전트를 사용하여 임베디드 지표 형식 로그 전송

CloudWatch 에이전트를 설치하고 실행하고 나면 TCP 또는 UDP를 통해 임베디드 지표 형식 로그를 전송할 수 있습니다. 에이전트를 통해 로그를 전송할 때 다음 두 가지 요구 사항이 있습니다.

- 로그에는 사용할 로그 그룹을 에이전트에 알려주는 `LogGroupName` 키가 있어야 합니다.
- 각 로그 이벤트는 한 줄에 있어야 합니다. 즉, 로그 이벤트에는 줄 바꿈 문자(\n)가 포함될 수 없습니다.

또한 로그 이벤트는 임베디드 지표 형식 사양을 따라야 합니다. 자세한 내용은 사양: 임베디드 지표 형식 (p. 329) 단원을 참조하십시오.

다음은 Linux bash 셸에서 수동으로 로그 이벤트를 전송하는 예제입니다. 선택한 프로그래밍 언어에서 제공하는 UDP 소켓 인터페이스를 대신 사용할 수 있습니다.

```
echo '{"_aws":{"Timestamp":1574109732004,"LogGroupName":"Foo","CloudWatchMetrics":
[{"Namespace":"MyApp","Dimensions":[["Operation"]],"Metrics":
[{"Name":"ProcessingLatency","Unit":"Milliseconds"}]}]},"Operation":"Aggregator","ProcessingLatency":10
 \
> /dev/udp/0.0.0.0/25888
```

콘솔에서 지표 및 로그 보기

지표를 추출하는 임베디드 지표 형식 로그를 생성한 후 CloudWatch 콘솔을 사용하여 지표를 볼 수 있습니다. 임베디드 지표에는 로그를 생성할 때 지정한 차원이 있습니다. 또한 클라이언트 라이브러리를 사용하여 생성한 임베디드 지표에는 다음과 같은 기본 차원이 있습니다.

- ServiceType
- ServiceName

- LogGroup

임베디드 지표 형식 로그에서 생성된 지표를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. 임베디드 지표를 생성할 때 해당 지표에 지정한 네임스페이스를 선택합니다. 클라이언트 라이브러리를 사용하여 지표를 생성하고 네임스페이스를 지정하지 않은 경우 aws-embedded-metrics를 선택합니다. 클라이언트 라이브러리를 사용하여 생성된 임베디드 지표의 기본 네임스페이스입니다.
4. 지표 차원(예: ServiceName)을 선택합니다.
5. 모든 지표 탭에 네임스페이스의 해당 차원에 대한 모든 지표가 표시됩니다. 다음을 수행할 수 있습니다.

   a. 테이블을 정렬하려면 열 머리글을 사용합니다.
   b. 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다. 모든 지표를 선택하려면 테이블의 머리글 행에 있는 확인란을 선택합니다.
   c. 리소스로 필터링하려면 리소스 ID를 선택한 후 검색에 추가를 선택합니다.
   d. 측정치로 필터링하려면 측정치 이름을 선택한 후 검색에 추가를 선택합니다.

CloudWatch Logs Insights를 사용하여 로그 쿼리

CloudWatch Logs Insights를 사용하여 운영 이벤트의 근본 원인에 대한 심층적인 통찰력을 제공함으로써 추출된 지표와 연관된 자세한 로그 이벤트를 쿼리할 수 있습니다. 로그에서 지표를 추출할 때 얻을 수 있는 이점 중 하나는 나중에 고유한 지표(지표 이름 및 고유한 차원 집합) 및 지표 값을 기준으로 로그를 필터링하여 집계된 지표 값에 기여한 이벤트에 대한 컨텍스트를 가져올 수 있다는 것입니다.

예를 들어 영향을 받는 요청 ID 또는 x-ray 추적 ID를 얻으려면 CloudWatch Logs Insights에서 다음 쿼리를 실행합니다.

```
filter Latency > 1000 and Operation = "Aggregator"
| fields RequestId, TraceId
```

이벤트의 영향을 받는 고객을 찾는 등 높은 카디널리티 키에 대해 쿼리 시간 집계를 수행할 수도 있습니다. 다음 예는 이를 보여 줍니다.

```
filter Latency > 1000 and Operation = "Aggregator"
| stats count() by CustomerId
```

자세한 내용은 CloudWatch Logs Insights를 사용하여 로그 데이터 분석을 참조하십시오.

AWS SDK 지표를 사용하여 애플리케이션 모니터링

엔터프라이즈 고객은 CloudWatch 에이전트와 AWS SDK 지표 for Enterprise Support(SDK 지표)를 함께 사용하여 해당 호스트와 클라이언트에서 AWS SDK에 대한 지표를 수집할 수 있습니다. 이러한 지표는 AWS Enterprise Support와 공유됩니다. SDK 지표는 코드에 사용자 지정 지표를 추가하지 않고 AWS 서비스에 대한 애플리케이션의 연결에 대한 관련 지표와 진단 데이터를 수집할 수 있도록 도와주므로 로그와 데이터를 AWS Support와 공유하는 데 필요한 수동 작업을 줄일 수 있습니다.

> **Important**
>
> SDK 지표는 Enterprise Support 구독 고객만 사용할 수 있습니다. 자세한 내용은 Amazon CloudWatch 지원 센터를 참조하십시오.

AWS 서비스를 직접 호출하고 다음 단원에 나열된 버전 중 하나인 AWS SDK를 사용하여 구축된 모든 애플리케이션에서 SDK 지표를 사용할 수 있습니다.

SDK 지표는 AWS SDK를 통해 생성된 호출을 모니터링하고 클라이언트 애플리케이션과 동일한 환경에서 실행되는 CloudWatch 에이전트를 사용합니다. 다음 단원에서는 SDK 지표 데이터를 생성하도록 CloudWatch 에이전트를 활성화하는 데 필요한 단계를 설명합니다. SDK에서 구성하는 필요한 항목에 대한 설명은 SDK 설명서를 참조하십시오.

주제

- Enterprise Support에 대한 AWS SDK 지표에서 수집한 지표와 데이터 (p. 343)
- SDK 지표 설정 (p. 345)

지원 버전

SDK 지표를 사용하려면 CloudWatch 에이전트 버전 1.207573.0 이상을 사용해야 합니다. CloudWatch 에이전트를 이미 실행하는 경우 다음 파일에서 해당 버전을 찾을 수 있습니다.

- Linux: `/opt/aws/amazon-cloudwatch-agent/bin/CWAGENT_VERSION`
- Windows Server: `C:\Program Files\Amazon\AmazonCloudWatchAgent\CWAGENT_VERSION`

또한 다음 AWS SDK 버전 중 하나를 사용해야 합니다.

지원되는 SDK	SDK 지표 설명서
AWS CLI 1.16.84 이상	SDK 지표 AWS CLI 설치 지침은 Python용 SDK(Boto3)의 지침과 동일합니다.
C++용 AWS SDK 1.7.47 이상	SDK 지표
Go용 AWS SDK 1.25.18 이상	Go용 AWS SDK의 SDK 지표
AWS SDK for Java 1.11.523 이상(Java 2.x용 AWS SDK가 아직 지원되지 않음)	Enterprise Support에 대한 AWS SDK 지표 활성화
AWS SDK for JavaScript in Node.js 2.387 이상	AWS SDK for JavaScript의 SDK 지표
.NET용 AWS SDK 3.3.440 이상	SDK 지표 활성화

지원되는 SDK	SDK 지표 설명서
PHP용 AWS SDK 3.85.0 이상	PHP용 AWS SDK 버전 3의 SDK 지표
Python용 AWS SDK(Boto3) 1.9.78 이상	SDK 지표
Ruby용 AWS SDK 3.45.0 이상	Ruby용 AWS SDK의 SDK 지표

Enterprise Support에 대한 AWS SDK 지표에서 수집한 지표와 데이터

SDK 지표는 애플리케이션으로부터 데이터를 수집한 후 이를 사용하여 CloudWatch에 지표를 전송합니다. 다음 표에는 SDK 지표에서 수집하는 데이터 목록이 나와 있습니다.

테스트	유형
메시지 버전	문자열
메시지 ID	문자열
서비스 엔드포인트	문자열
정규화된 서비스 ID	문자열
API 작업 이름	문자열
가용성(SDK 고객 관점)	정수(0 또는 1) 및 샘플 수
지연 시간(SDK 고객 관점)	Distribution
SDK 버전	문자열
클라이언트 언어 런타임 버전	문자열
클라이언트 운영 체제	문자열
서비스 응답 코드	키/값 페어
클라이언트 언어 런타임 버전	문자열
샘플 요청 ID	List
재시도	Distribution
조정된(throttle) 요청	Distribution
AccountID	문자열
가용 영역	문자열
인스턴스 ID	문자열
런타임 환경(Lambda/ECS)	문자열
네트워크 오류 메시지	문자열/맵
소스 IP 주소	문자열

테스트	유형
대상 IP 주소	문자열

다음 표에는 SDK 지표에서 생성되고 `AWS/SDKMetrics` 네임스페이스에 게시된 지표가 나와 있습니다. 이들 지표는 Enterprise 지원 플랜을 이용하는 고객에게만 제공됩니다. 이러한 고객은 AWS 지원 담당자 및 기술 계정 관리자를 통해 지표에 액세스할 수 있습니다.

지표는 1분 간격으로 게시됩니다.

지표	설명	--set-visible-to-all-users
CallCount	해당 코드에서 AWS 서비스에 대해 수행한 호출 중 성공하거나 실패한 총 API 호출 수. 단위: 수	`CallCount`를 기준으로 사용하여 `ServerErrorCount` 및 `ThrottleCount` 등과 같은 다른 지표와의 상관관계를 보여줍니다.
ClientErrorCount	클라이언트 오류(4xx HTTP 응답 코드)와 함께 실패한 API 호출 수. 여기에는 거부된 액세스, S3 버킷 없음, 잘못된 파라미터 값 등의 조정(throttling) 오류가 포함될 수 있습니다. 단위: 수	이 지표의 값이 크면 AWS 서비스 제한으로 인한 조정(throttling)의 결과가 아닌 한 애플리케이션의 항목을 수정해야 함을 나타냅니다. 이 경우 서비스 한도를 늘려야 합니다.
EndToEndLatency	애플리케이션이 AWS SDK를 사용하여 호출(재시도 포함)을 수행한 총 시간. 단위: 밀리초	`EndToEndLatency`를 사용하여 AWS API 호출이 애플리케이션의 전체 지연 시간에 얼마나 기여했는지를 확인할 수 있습니다. 네트워크, 방화벽 또는 기타 구성 설정 문제로 인해 예상보다 높은 지연 시간이 발생할 수 있습니다. SDK 재시도로 인해서도 지연 시간이 발생할 수 있습니다.
ConnectionErrorCount	연결 오류로 인해 실패한 API 호출 수. 이는 로드 밸런서 문제, DNS 오류 및 전송 공급자 문제를 포함하여 애플리케이션과 AWS 서비스 간의 네트워크 문제로 인해 발생할 수 있습니다. 경우에 따라 AWS 문제로 인해 이 오류가 발생할 수 있습니다. 단위: 수	이 지표를 사용하여 문제가 애플리케이션에 한정된 문제인지 아니면 인프라 또는 네트워크에 의해 발생하는지 확인하십시오. 높은 값은 API 호출에 대한 짧은 시간 초과 값을 나타낼 수도 있습니다.
ServerErrorCount	AWS 서비스의 서버 오류(5xx HTTP 응답 코드)로 인해 실패한 API 호출 수입니다. 이 실패의 원인은 일반적으로 AWS 서비스입니다. 단위: 수	이 지표를 사용하여 SDK 재시도 또는 지연 시간의 원인을 확인할 수 있습니다. 일부 AWS 팀에서는 대기 시간을 HTTP 503 응답으로 분류하기 때문에 이 지표가 항상 AWS 서비스에 결함이 있음을 나타내는 것은 아닙니다.
ThrottleCount	AWS 서비스의 조정(throttling) 때문에 실패한 API 호출의 수입니다. 단위: 수	이 지표를 사용하여 애플리케이션이 조정(throttling) 한계에 도달했는지 평가하고 재시도 및 애플리케이션 대기 시간의 원인을 확인할 수 있습니다. 높은 값이 표시되면 호출을 배치(batch)로 처리하는 대신 기간 동안 호출을 분산시켜 보십시오.

SDK 지표에서 다음 차원을 사용할 수 있습니다.

차원	설명
DestinationRegion	호출의 대상 AWS 리전입니다.
서비스	애플리케이션에서 호출하는 AWS 서비스입니다.

SDK 지표 설정

SDK 지표를 설정하려면 SDK로 일부 단계를 수행하고 일부는 CloudWatch 에이전트로 수행하십시오.

SDK 지표를 설정하려면

1. SDK 지표를 지원하는 AWS SDK를 사용하여 애플리케이션을 생성합니다.
2. Amazon EC2 인스턴스 또는 로컬 환경에서 프로젝트를 호스팅합니다.
3. SDK의 최신 버전을 설치하여 사용합니다.
4. 지표를 수신할 애플리케이션을 실행 중인 EC2 인스턴스 또는 로컬 환경에 CloudWatch 에이전트를 설치합니다. 자세한 내용은 CloudWatch 에이전트 설치 (p. 190) 단원을 참조하십시오.
5. SDK 지표에 대해 CloudWatch 에이전트 구성 자세한 내용은 SDK 지표에 대해 CloudWatch 에이전트 구성 (p. 346) 단원을 참조하십시오.
6. 지표를 수집 및 전송하도록 SDK 지표를 승인합니다. 자세한 내용은 SDK 지표에 대한 IAM 권한 설정 (p. 347) 단원을 참조하십시오.
7. SDK 지표를 활성화합니다. 사용 중인 SDK에 따라 여러 가지 방법이 있습니다. 모든 SDK에 범용으로 사용할 수 있는 하나의 방법은 다음을 환경 변수에 추가하는 것입니다.

```
export AWS_CSM_ENABLED=true
```

다른 방법에 대한 자세한 내용은 다음 표와 같이 SDK 설명서를 참조하십시오.

지원되는 SDK	SDK 지표 설명서
AWS CLI 1.16.84 이상	SDK 지표 AWS CLI 설치 지침은 Python용 SDK(Boto3)의 지침과 동일합니다.
C++용 AWS SDK 1.7.47 이상	SDK 지표
Go용 AWS SDK 1.16.18 이상	Go용 AWS SDK의 SDK 지표
AWS SDK for Java 1.11.523 이상 (Java 2.x용 AWS SDK가 아직 지원되지 않음)	Enterprise Support에 대한 AWS SDK 지표 활성화
AWS SDK for JavaScript in Node.js 2.387 이상	AWS SDK for JavaScript의 SDK 지표
.NET용 AWS SDK 3.3.440 이상	SDK 지표 활성화
PHP용 AWS SDK 3.85.0 이상	PHP용 AWS SDK 버전 3의 SDK 지표
Python용 AWS SDK(Boto3) 1.9.78 이상	SDK 지표

지원되는 SDK	SDK 지표 설명서
Ruby용 AWS SDK 3.45.0 이상	Ruby용 AWS SDK의 SDK 지표

SDK 지표에 대해 CloudWatch 에이전트 구성

CloudWatch 에이전트를 설치한 후 SDK 지표와 연동하도록 구성해야 합니다. 가장 쉬운 방법은 AWS 시스템 관리자를 사용하는 것이지만 수동으로 할 수도 있습니다.

주제

- AWS 시스템 관리자를 사용하여 SDK 지표에 대해 CloudWatch 에이전트 구성 (p. 346)
- SDK 지표에 대해 수동으로 CloudWatch 에이전트 구성 (p. 346)

AWS 시스템 관리자를 사용하여 SDK 지표에 대해 CloudWatch 에이전트 구성

이 단원에서는 SSM를 사용하여 SDK 지표와 연동하도록 CloudWatch를 구성하는 방법을 설명합니다. SSM 에이전트에 대한 자세한 내용은 SSM Agent 설치 및 구성을 참조하십시오.

SSM를 사용하여 SDK 지표를 구성하려면

1. Open the 시스템 관리자 console at https://console.aws.amazon.com/systems-manager/.
2. 탐색 창에서 Run Command를 선택합니다.
3. 명령 문서 목록에서 AWS-UpdateSSMAgent 옆의 버튼을 선택합니다.
4. 대상 영역에서 CloudWatch 에이전트를 설치한 인스턴스를 선택합니다.
5. 탐색 창에서 파라미터 스토어를 선택합니다.
6. 파라미터 생성을 선택합니다.
7. 해결 방법:
   a. 파라미터 이름을 `AmazonCSM`으로 지정합니다.
   b. *`string`* 형식을 선택합니다.
   c. 값에 `{ "csm": {"memory_limit_in_mb": 20, "port": 31000}}`를 입력합니다.
8. 파라미터 생성을 선택합니다.

구성을 완료하려면 SDK 설명서를 참조하십시오.

SDK 지표에 대해 수동으로 CloudWatch 에이전트 구성

이 단원에서는 SDK 지표와 연동하도록 CloudWatch 에이전트를 수동으로 구성하는 방법을 설명합니다.

CloudWatch 에이전트 구성 파일 생성

아직 CloudWatch 에이전트 구성 파일이 없으면 하나를 생성하십시오. 자세한 내용은 CloudWatch 에이전트 구성 파일 생성 (p. 222) 단원을 참조하십시오.

SDK 지표 구성 추가

CloudWatch 에이전트 구성 파일의 최상위 JSON 객체에 다음 `csm` 항목을 추가합니다.

```
"csm": {
    "memory_limit_in_mb": 20,
```

```
    "port": 31000
  }
```

예:

```
{
    "agent": {
      ...
    },
    "metrics": {
      ...
    },
    "csm": {
      "memory_limit_in_mb": 20,
      "port": 31000
    }
  }
```

이제 이 구성 파일을 사용하여 CloudWatch 에이전트를 시작할 수 있습니다. 자세한 내용은 CloudWatch 에이전트 시작 (p. 208) 단원을 참조하십시오.

구성을 완료하려면 SDK 설명서를 참조하십시오.

SDK 지표에 대한 IAM 권한 설정

SDK 지표를 사용하기 위해 권한을 구성하려면 SDK 지표 프로세스를 허용하는 IAM 정책을 생성한 후 CloudWatch 에이전트를 실행 중인 EC2 인스턴스에 연결된 역할 또는 사용자에게 이 정책을 연결해야 합니다.

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다.
3. 정책 생성, JSON을 선택합니다.
4. 내용 창에 다음 인라인 정책을 입력합니다. 지원되지 않는 작업에 대한 경고는 무시합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sdkmetrics:*"
            ],
            "Resource": "*"
        }
    ]
}
```

5. 정책 이름을 AmazonSDKMetrics로 지정합니다.
6. CloudWatch 에이전트를 실행 중인 EC2 인스턴스에 연결된 IAM 사용자 또는 역할에게 이 정책을 연결합니다.

서비스 할당량 통합 및 사용량 지표

CloudWatch를 사용하여 AWS 서비스 할당량을 사전 예방적으로 관리할 수 있습니다. CloudWatch 사용량 지표는 계정의 리소스 사용량 및 API 작업에 대한 가시성을 제공합니다.

이러한 지표를 사용하여 CloudWatch 그래프 및 대시보드에서 현재 서비스 사용량을 시각화할 수 있습니다. CloudWatch 지표 수학 함수를 사용하여 해당 리소스에 대한 서비스 할당량을 그래프에 표시할 수 있습니다. 사용량이 서비스 할당량에 가까워지면 경고하는 경보를 구성할 수도 있습니다.

서비스 할당량에 대한 자세한 내용은 서비스 할당량 사용 설명서의 서비스 할당량이란 무엇입니까?를 참조하십시오.

현재 다음 서비스는 서비스 할당량을 평가하는 데 사용할 수 있는 사용량 지표를 제공합니다.

- AWS CloudHSM
- Amazon CloudWatch
- Amazon DynamoDB
- Amazon EC2
- Amazon Elastic Container Registry
- AWS Key Management Service
- Amazon Kinesis Data Firehose

이러한 모든 서비스는 `AWS/Usage` 지표 네임스페이스에 사용량 지표를 게시합니다.

주제

- 서비스 할당량 시각화 및 경보 설정 (p. 348)
- CloudWatch 사용량 지표 (p. 349)

서비스 할당량 시각화 및 경보 설정

CloudWatch 콘솔을 사용하여 서비스 할당량을 시각화하고 현재 사용량 비교를 확인할 수 있습니다. 할당량에 접근하면 알림을 받도록 경보를 설정할 수도 있습니다.

서비스 할당량을 시각화하고 선택적으로 경보를 설정하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. 모든 지표 탭에서 사용량을 선택한 다음 By AWS Resource(AWS 리소스별)를 선택합니다.

   서비스 할당량 사용량 지표 목록이 나타납니다.
4. 지표 중 하나 옆에 있는 확인란을 선택합니다.

   그래프는 해당 AWS 리소스의 현재 사용량을 표시합니다.
5. 그래프에 서비스 할당량을 추가하려면 다음을 수행합니다.

   a. 그래프로 표시된 지표 탭을 선택합니다.

   b. Math expression(수학 표현식), Start with an empty expression(빈 표현식으로 시작)을 선택합니다. 그런 다음 새 행의 세부 정보에 **`SERVICE_QUOTA(m1)`**를 입력합니다.

      그래프에 새 줄이 추가되어 지표에 표시된 리소스에 대한 서비스 할당량을 표시합니다.

6. 현재 사용량을 할당량의 백분율로 보려면 새 표현식을 추가하거나 현재 SERVICE_QUOTA 표현식을 변경합니다. 새 표현식의 경우 **`m1/SERVICE_QUOTA(m1)*100`**을 사용합니다.
7. (선택 사항) 서비스 할당량에 접근하는 경우 알려주는 경보를 설정하려면 다음을 수행합니다.

   a. **`m1/SERVICE_QUOTA(m1)*100`** 행의 작업에서 경보 아이콘을 선택합니다. 이 아이콘은 종처럼 보입니다.

      경보 생성 페이지가 나타납니다.

   b. 조건에서 Threshold type(임계값 유형)이 정적이고 Whenever Expression1 is(표현식1이 해당할 때마다)가 Greater(큼)로 설정되었는지 확인합니다. than(보다)에 **`80`**을 입력합니다. 이렇게 하면 사용량이 할당량의 80%를 초과할 경우 ALARM 상태가 되는 경보가 생성됩니다.

   c. 다음을 선택합니다.

   d. 다음 페이지에서 Amazon SNS 주제를 선택하거나 새 주제를 생성합니다. 이 주제는 경보가 ALARM 상태로 전환되면 알림을 받습니다. 그런 다음 [Next]를 선택합니다.

   e. 다음 페이지에서 경보의 이름과 설명을 입력하고 다음을 선택합니다.

   f. 경보 생성을 선택합니다.

CloudWatch 사용량 지표

CloudWatch는 일부 AWS 리소스의 사용량을 추적하는 지표를 수집합니다. 이러한 지표는 AWS 서비스 할당량에 해당합니다. 이러한 지표를 추적하면 할당량을 사전 예방적으로 관리하는 데 도움이 될 수 있습니다. 자세한 내용은 서비스 할당량 통합 및 사용량 지표 (p. 348) 단원을 참조하십시오.

서비스 할당량 사용량 지표는 `AWS/Usage` 네임스페이스에 있으며 1분마다 수집됩니다.

현재 이 네임스페이스에서 CloudWatch가 게시하는 유일한 지표 이름은 `CallCount`입니다. 이 지표는 `Resource`, `Service` 및 `Type` 차원으로 게시됩니다. `Resource` 차원은 추적 중인 API 작업의 이름을 지정합니다. 예를 들어 `"Service": "CloudWatch"`, `"Type": "API"` 및 `"Resource": "PutMetricData"` 차원이 포함된 `CallCount` 지표는 계정에서 CloudWatch `PutMetricData` API 작업이 호출된 횟수를 나타냅니다.

`CallCount` 지표에는 지정된 단위가 없습니다. 지표에 가장 유용한 통계는 `SUM`이며 1분 동안의 총 작업 수를 나타냅니다.

지표

지표	설명
`CallCount`	계정에서 수행된 지정된 작업 수

Dimensions

차원	설명
`Service`	리소스가 포함된 AWS 서비스의 이름 CloudWatch 사용량 지표의 경우 이 차원의 값은 `CloudWatch`입니다.
`Class`	추적 중인 리소스의 클래스. CloudWatch API 사용량 지표는 이 차원을 `None` 값과 함께 사용합니다.
`Type`	추적 중인 리소스의 유형. 현재, `Service` 치수가 `CloudWatch`인 경우 `Type`에 대한 유일한 유효한 값은 `API`입니다.

차원	설명
`Resource`	API 작업의 이름. 유효한 값은 다음과 같습니다. `DeleteAlarms`, `DeleteDashboards`, `DescribeAlarmHistory`, `DescribeAlarms`, `GetDashboard`, `GetMetricData`, `GetMetricStatistics`, `ListMetrics`, `PutDashboard` 및 `PutMetricData`

CloudWatch 자습서

다음 시나리오에서는 Amazon CloudWatch 사용에 대해 설명합니다. 첫 번째 시나리오에서는 CloudWatch 콘솔을 사용하여 AWS 사용을 추적하고 특정 사용 임계값을 초과할 경우 이를 알리는 결제 경보를 생성합니다. 좀 더 발전한 두 번째 시나리오에서는 AWS Command Line Interface(AWS CLI)를 사용하여 GetStarted라는 가상의 애플리케이션을 위한 단일 지표를 게시합니다.

시나리오

- 예상 요금 모니터링 (p. 351)
- 지표 게시 (p. 353)

시나리오: CloudWatch를 사용하여 예상 요금 모니터링

이 시나리오에서는 예상 요금을 모니터링하기 위한 Amazon CloudWatch 경보를 생성합니다. AWS 계정에 대한 예상 요금 모니터링을 활성화하면 예상 요금이 계산되어 지표 데이터로서 하루에 여러 번 CloudWatch에 전송됩니다.

결제 지표 데이터는 미국 동부(버지니아 북부) 지역에 저장되며 전 세계 요금을 반영합니다. 결제 지표 데이터에는 사용한 AWS의 모든 서비스에 대한 예상 요금과 전반적인 총 AWS 예상 요금이 들어 있습니다.

요금이 특정 임계값을 초과한 경우 이메일로 알림을 받도록 선택할 수 있습니다. 결제 경보는 CloudWatch에 의해 트리거되며 메시지는 Amazon Simple Notification Service(Amazon SNS)를 이용해 전송됩니다.

작업

- 1단계: 결제 경보 활성화 (p. 351)
- 2단계: 결제 경보 만들기 (p. 352)
- 3단계: 경보 상태 확인 (p. 353)
- 4단계: 결제 경보 편집 (p. 353)
- 5단계: 결제 경보 삭제 (p. 353)

1단계: 결제 경보 활성화

예상 요금에 대한 경보를 생성할 수 있으려면 먼저 결제 경보를 활성화해야 합니다. 그래야만 예상되는 AWS 요금을 모니터링하고 결제 지표 데이터를 사용하여 경보를 생성할 수 있습니다. 결제 알림을 활성화한 후에는 데이터 수집을 비활성화할 수 없지만, 생성된 결제 알림은 무엇이든 삭제할 수 있습니다.

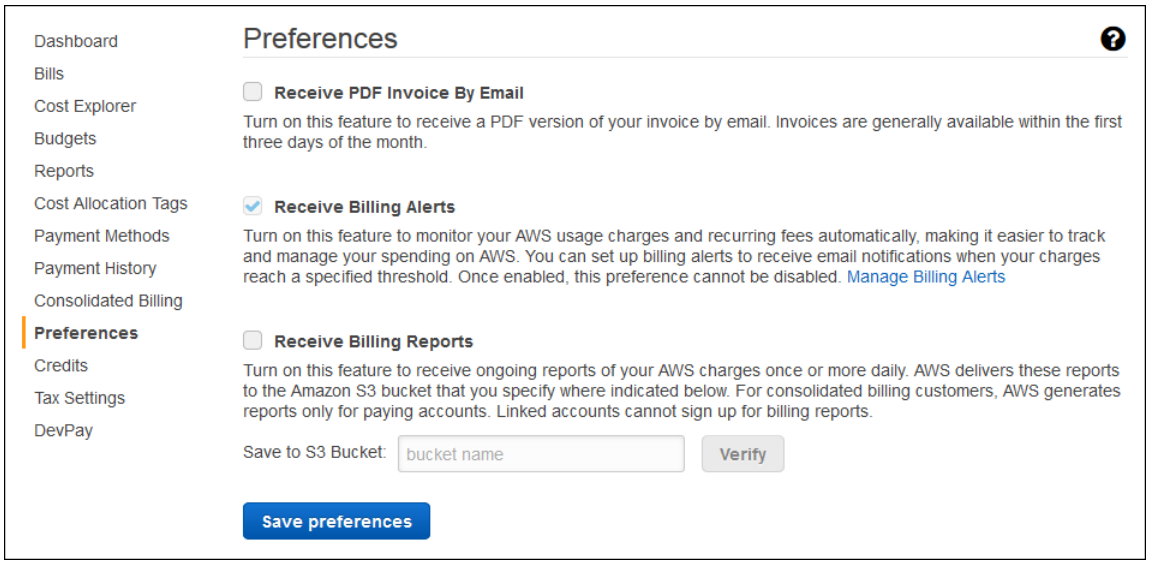
결제 경보를 처음 활성화하고 나서 결제 데이터를 확인하고 결제 경보를 설정할 수 있기까지 약 15분 정도의 시간이 걸립니다.

요구 사항

- Billing and Cost Management 콘솔에 대한 액세스 권한이 활성화된 AWS 계정 루트 사용자나 IAM 사용자로 로그인해야 합니다.
- 통합 결제 계정의 경우 결제 계정으로 로그인하면 연결된 각 계정에 대한 결제 데이터를 찾을 수 있습니다. 통합 계정에 대해서뿐만 아니라 연결된 각 계정에 대한 서비스별 총 예상 요금 및 예상 요금에 대한 결제 데이터를 볼 수 있습니다.

예상 비용 모니터링을 활성화하려면

1. https://console.aws.amazon.com/billing/에서 Billing and Cost Management 콘솔을 엽니다.
2. 탐색 창에서 기본 설정을 선택합니다.
3. 결제 알림 받기를 선택합니다.



4. Save preferences를 선택합니다.

2단계: 결제 경보 만들기

결제 경보를 활성화했으면 결제 경보를 생성할 수 있습니다. 이 시나리오에서는 AWS에 대한 예상 요금이 지정된 임계값을 초과할 때 이메일 메시지를 전송하는 경보를 생성합니다.

> Note
>
> 이 절차는 단순 옵션을 사용합니다. 고급 옵션을 사용하려면 예상 AWS 요금을 모니터링하기 위한 결제 경보 생성의 결제 경보 생성 (p. 95) 단원을 참조하십시오.

결제 경보를 만들려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 미국 동부(버지니아 북부)로 변경합니다. 결제 지표 데이터는 이 리전에 저장되며 전 세계 요금을 반영합니다.
3. 탐색 창에서 경보, 경보 생성을 선택합니다.
4. 지표 선택, 결제, 예상 요금 합계를 선택합니다.
5. EstimatedCharges 옆에 있는 확인란을 선택하고 지표 선택을 선택합니다.
6. Whenever my total AWS charges for the month exceed(해당 월 총 AWS 요금이 다음을 초과할 때마다)에서 경보를 트리거하고 이메일 알림을 전송하기 위해 초과되어야 할 금액(예: 200)을 지정합니다.

   > Tip
   >
   > 그래프에는 적정 금액을 설정하는 데 사용할 수 있는 예상 요금이 나와 있습니다.
7. 알림 보내기에서 기존 알림 목록을 선택하거나 새로 만듭니다.

   목록을 만들려면 새 목록을 선택하고 ALARM 상태로 변경될 때 알림을 보낼 이메일 주소 목록을 쉼표로 구분하여 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.

8. 경보 생성을 선택합니다.

3단계: 경보 상태 확인

이제, 방금 만든 결제 경보의 상태를 확인합니다.

경보 상태를 확인하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 미국 동부(버지니아 북부)로 변경합니다. 결제 지표 데이터는 이 리전에 저장되며 전 세계 요금을 반영합니다.
3. 탐색 창에서 Alarms를 선택합니다.
4. 경보 옆의 확인란을 선택합니다. 구독이 확인되기 전까지 "확인 보류 중"으로 표시가 됩니다. 구독 확인 후에 콘솔을 새로 고쳐 업데이트된 상태를 보여줍니다.

4단계: 결제 경보 편집

예를 들어, 매월 AWS에 사용할 수 있는 금액을 200~400달러까지 늘리고 싶은 경우, 기존 결제 경보를 편집하여 경보 트리거 전에 초과해야 하는 금액을 늘릴 수 있습니다.

결제 경보를 편집하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 미국 동부(버지니아 북부)로 변경합니다. 결제 지표 데이터는 이 리전에 저장되며 전 세계 요금을 반영합니다.
3. 탐색 창에서 Alarms를 선택합니다.
4. 경보 옆의 확인란을 선택한 후 작업과 수정을 차례로 선택합니다.
5. Whenever my total AWS charges for the month exceed(해당 월 총 AWS 요금이 다음을 초과할 때마다)에서 경보를 트리거하고 이메일 알림을 전송하기 위해 초과되어야 할 새 금액을 지정합니다.
6. 변경 사항 저장을 선택합니다.

5단계: 결제 경보 삭제

결제 경보가 더 이상 필요 없는 경우 해당 경보를 삭제할 수 있습니다.

결제 경보를 삭제하려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 미국 동부(버지니아 북부)로 변경합니다. 결제 지표 데이터는 이 리전에 저장되며 전 세계 요금을 반영합니다.
3. 탐색 창에서 Alarms를 선택합니다.
4. 경보 옆의 확인란을 선택하고 작업, 삭제를 차례로 선택합니다.
5. 확인 메시지가 나타나면 예, 삭제합니다를 선택합니다.

시나리오: CloudWatch에 지표 게시

이 시나리오에서는 AWS Command Line Interface(AWS CLI)를 사용하여 GetStarted라는 가상의 애플리케이션을 위한 단일 지표를 게시합니다. AWS CLI가 이미 설치 및 구성된 경우에는 AWS Command Line Interface 사용 설명서의 AWS Command Line Interface을 사용한 설정을 참조하십시오.

작업

- 1단계: 데이터 구성 정의 (p. 354)
- 2단계: CloudWatch에 지표 추가 (p. 354)
- 3단계: CloudWatch에서 통계 얻기 (p. 355)
- 4단계: 콘솔을 사용하여 그래프 보기 (p. 355)

1단계: 데이터 구성 정의

이 시나리오에서는 애플리케이션의 요청 지연 시간을 추적하는 데이터 요소를 게시합니다. 적절한 지표 및 네임스페이스의 이름을 선택합니다. 예를 들어 지표의 이름을 RequestLatency로 지정하고 모든 데이터 요소를 GetStarted 네임스페이스에 배치합니다.

3시간의 지연 시간 데이터를 총체적으로 나타내는 여러 데이터 요소를 게시합니다. 원시 데이터는 3시간에 걸쳐 분산된 요청 지연 시간 판독값 15개로 구성됩니다. 각 판독값은 다음과 같이 밀리초 단위입니다.

- 시간 1: 87, 51, 125, 235
- 시간 2: 121, 113, 189, 65, 89
- 시간 3: 100, 47, 133, 98, 100, 328

CloudWatch에 데이터를 단일 데이터 요소 또는 통계 세트라는 집계된 데이터 요소 세트로 게시할 수 있습니다. 지표를 1분 정도로 낮게 세분화하여 집계할 수 있습니다. 사전 정의된 키 4개(`Sum`, `Minimum`, `Maximum` 및 `SampleCount`)를 사용하여 집계된 데이터 요소를 CloudWatch에 통계 세트로 게시할 수 있습니다.

시간 1의 데이터 요소를 단일 데이터 요소로 게시합니다. 시간 2 및 시간 3의 데이터의 경우 데이터 요소를 집계하여 각 시간에 대한 통계 세트를 게시합니다. 키 값은 다음 표에 표시됩니다.

시간	원시 데이터	합계	최소	최대	SampleCount
1	87				
1	51				
1	125				
1	235				
2	121, 113, 189, 65, 89	577	65	189	5
3	100, 47, 133, 98, 100, 328	806	47	328	6

2단계: CloudWatch에 지표 추가

데이터 구성을 정의하면 데이터 추가를 시작할 준비가 된 것입니다.

CloudWatch에 데이터 요소를 게시하려면

1. 명령 프롬프트에서 put-metric-data 명령을 실행하여 첫 번째 시간에 대한 데이터를 추가합니다. 예제 타임스탬프를 UTC 기준으로 2시간 전인 타임스탬프로 변경합니다.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted \
--timestamp 2016-10-14T20:30:00Z --value 87 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted \
--timestamp 2016-10-14T20:30:00Z --value 51 --unit Milliseconds
```

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted \
--timestamp 2016-10-14T20:30:00Z --value 125 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted \
--timestamp 2016-10-14T20:30:00Z --value 235 --unit Milliseconds
```

2. 첫 번째 시간보다 1시간 늦은 타임스탬프를 사용하여 두 번째 시간에 대한 데이터를 추가합니다.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted \
--timestamp 2016-10-14T21:30:00Z --statistic-values
 Sum=577,Minimum=65,Maximum=189,SampleCount=5 --unit Milliseconds
```

3. 현재 시간에 대한 기본값에 대한 타임스탬프를 생략하고 세 번째 시간에 대한 데이터를 추가합니다.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted \
--statistic-values Sum=806,Minimum=47,Maximum=328,SampleCount=6 --unit Milliseconds
```

3단계: CloudWatch에서 통계 얻기

CloudWatch로 지표를 게시했다면 이제 다음과 같이 get-metric-statistics 명령을 사용하여 이러한 지표를 토대로 통계를 검색할 수 있습니다. 게시한 가장 빠른 타임스탬프를 포함하도록 지난 시간에서 `--start-time` 및 `--end-time`을 충분히 여유 있게 지정해야 합니다.

```
aws cloudwatch get-metric-statistics --namespace GetStarted --metric-name RequestLatency --
statistics Average \
--start-time 2016-10-14T00:00:00Z --end-time 2016-10-15T00:00:00Z --period 60
```

다음은 예제 출력입니다.

```
{
 "Datapoints": [],
 "Label": "Request:Latency"
}
```

4단계: 콘솔을 사용하여 그래프 보기

CloudWatch에 지표를 게시하면 CloudWatch 콘솔을 사용하여 통계 그래프를 볼 수 있습니다.

콘솔에서 통계 그래프를 보려면

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.
3. 모든 지표 탭의 검색 상자에RequestLatency를 입력하고 Enter 키를 누릅니다.
4. RequestLatency 지표에 대한 확인란을 선택합니다. 지표 데이터의 그래프가 위쪽 창에 표시됩니다.

자세한 내용은 지표 그래프 작성 (p. 40) 단원을 참조하십시오.

Amazon CloudWatch의 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 CloudWatch를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 보안 및 규정 준수 목적에 맞게 Amazon CloudWatch를 구성하는 방법을 보여줍니다. 또한 CloudWatch 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배웁니다.

목차

- Amazon CloudWatch의 데이터 보호 (p. 356)
- Amazon CloudWatch에 대한 ID 및 액세스 관리 (p. 357)
- Amazon CloudWatch에 대한 규정 준수 확인 (p. 381)
- Amazon CloudWatch의 복원성 (p. 382)
- Amazon CloudWatch의 인프라 보안 (p. 382)
- 인터페이스 VPC 엔드포인트와 함께 CloudWatch 사용 (p. 382)
- Synthetics 카나리아에 대한 보안 고려 사항 (p. 384)

Amazon CloudWatch의 데이터 보호

Amazon CloudWatch는 AWS 공동 책임 모델을 준수하며, 여기에는 데이터 보호 관련 규정 및 지침이 포함됩니다. AWS는 모든 AWS 서비스를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. AWS는 고객 콘텐츠 및 개인 데이터를 처리하기 위한 보안 구성 제어를 포함하여 이 인프라에서 호스팅되는 데이터에 대한 제어를 유지합니다. 데이터 컨트롤러 또는 데이터 프로세서의 역할을 담당하는 AWS 고객과 APN 파트너는 AWS 클라우드에 입력한 모든 개인 데이터에 대해 책임이 있습니다.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS Identity and Access Management (IAM)을 사용하여 개별 사용자 계정을 설정하여 각 사용자에게 직무를 수행하는 데 필요한 권한만 부여하는 것이 좋습니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 멀티 팩터 인증(MFA)을 사용합니다.
- TLS를 사용하여 AWS 리소스와 통신합니다.
- 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.

함수 이름 및 태그와 같은 메타데이터나 자유 형식 필드에 고객 계정 번호 등의 중요 식별 정보를 절대 입력하지 마십시오. 메타데이터에 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.

데이터 보호에 대한 자세한 내용은 AWS 보안 블로그에서 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하십시오.

전송 중 데이터 암호화

CloudWatch는 전송 중인 데이터의 종단 간 암호화를 사용합니다.

Amazon CloudWatch에 대한 ID 및 액세스 관리

Amazon CloudWatch에 액세스하려면 자격 증명이 필요합니다. 이 자격 증명에는 클라우드 리소스에 대한 CloudWatch 지표 데이터 검색과 같이 AWS 리소스에 액세스할 수 있는 권한이 포함되어야 합니다. 다음 단원에서는 리소스에 액세스할 수 있는지 대상을 제어하여 리소스를 보호할 수 있도록 AWS Identity and Access Management(IAM) 및 CloudWatch 사용 방법에 대한 세부 정보를 제공합니다.

- 인증 (p. 357)
- 액세스 제어 (p. 358)

인증

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 AWS 계정 사용자 자격 증명으로, 모든 AWS 리소스에 대한 전체 액세스를 제공합니다.

  Important

  보안상 관리자 사용자, 즉 AWS 계정에 대한 전체 권한이 있는 IAM 사용자를 만들 때에만 AWS 계정 사용자 자격 증명을 사용하는 것이 좋습니다. 그러면 이 관리자를 사용하여 제한된 권한이 있는 다른 IAM 사용자 및 역할을 만들 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 모범 사례 및 관리자 및 그룹 만들기를 참조하십시오.
- IAM 사용자 – IAM 사용자는 특정 사용자 지정 권한(예: CloudWatch에서 metrics를 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(CLI)를 사용하여 AWS 서비스에 프로그래밍 방식으로 액세스할 때 이러한 키를 사용할 수 있습니다. SDK 및 AWS CLI 도구는 액세스 키를 사용하여 암호화 방식으로 요청에 서명합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. CloudWatch supports는 인바운드 API 요청을 인증하기 위한 프로토콜인 서명 버전 4를 지원합니다. 요청 인증에 대한 자세한 내용은 AWS General Reference의 서명 버전 4 서명 프로세스 단원을 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 액세스

하는 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 공급자(IdP)의 기존 자격 증명을 사용할 수 있습니다. 이러한 사용자를 연합된 사용자라고 합니다. IdP를 통해 액세스를 요청하면 AWS가 연합된 사용자에게 역할을 할당합니다. 자세한 내용은 IAM 사용 설명서의 연합된 사용자 및 역할을 참조하십시오.

- 교차 계정 액세스 – 계정의 IAM 역할을 사용하여 다른 AWS 계정에 계정 리소스에 액세스할 권한을 부여할 수 있습니다. 예제는 IAM 사용 설명서의 자습서: IAM 역할을 사용한 AWS 계정 간 액세스 권한 위임을 참조하십시오.

- AWS 제품 액세스 – 계정의 IAM 역할을 사용하여 AWS 제품에 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 대한 권한을 위임할 역할 생성을 참조하십시오.

- Amazon EC2에서 실행되는 애플리케이션 – 인스턴스에서 실행되고 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 IAM 사용 설명서의 Amazon EC2에서 실행되는 애플리케이션의 역할 사용하기를 참조하십시오.

액세스 제어

요청을 인증할 수 있는 유효한 자격 증명이 있더라도 권한이 없다면 CloudWatch 리소스를 생성하거나 액세스할 수 없습니다. 예를 들어 CloudWatch 대시보드 위젯, 뷰 지표 등을 생성할 권한이 있어야 합니다.

다음 단원에서는 CloudWatch에 대한 권한을 관리하는 방법을 설명합니다. 먼저 개요를 읽어 보면 도움이 됩니다.

- CloudWatch 리소스에 대한 액세스 권한 관리 개요 (p. 359)
- CloudWatch에 대한 자격 증명 기반 정책(IAM 정책) 사용 (p. 362)
- Amazon CloudWatch 권한 참조 문서 (p. 376)

CloudWatch 대시보드 권한 업데이트

2018년 5월 1일에 AWS는 CloudWatch 대시보드에 액세스하기 위해 필요한 권한을 변경했습니다. CloudWatch 콘솔에서 대시보드에 액세스하려면 이제 대시보드 API 작업을 지원하기 위해 2017년에 소개한 다음과 같은 권한이 필요합니다.

- cloudwatch:GetDashboard
- cloudwatch:ListDashboards
- cloudwatch:PutDashboard
- cloudwatch:DeleteDashboards

CloudWatch 대시보드에 액세스하려면 다음 중 하나가 필요합니다.

- AdministratorAccess 정책
- CloudWatchFullAccess 정책
- 다음과 같은 특정 권한 중 하나 이상을 포함하는 사용자 지정 정책:
  - 대시보드를 볼 수 있는 `cloudwatch:GetDashboard` 및 `cloudwatch:ListDashboards`
  - 대시보드를 생성하거나 수정할 수 있는 `cloudwatch:PutDashboard`
  - 대시보드를 삭제할 수 있는 `cloudwatch:DeleteDashboards`

정책을 사용하여 IAM 사용자의 권한 변경에 대한 자세한 내용은 IAM 사용자의 권한 변경을 참조하십시오.

CloudWatch 권한에 대한 자세한 내용은 Amazon CloudWatch 권한 참조 문서 (p. 376) 단원을 참조하십시오.

대시보드 API 작업에 대한 자세한 내용은 Amazon CloudWatch API Reference의 PutDashboard를 참조하십시오.

CloudWatch 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 리소스 및 작업 (p. 359)
- 리소스 소유권 이해 (p. 360)
- 리소스 액세스 관리 (p. 360)
- 정책 요소 지정: 작업, 효과, 보안 주체 (p. 361)
- 정책에서 조건 지정 (p. 362)

CloudWatch 리소스 및 작업

정책에서 Amazon 리소스 이름(ARN)을 사용해 특정 경보 및 대시보드에 대한 액세스를 제한할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 Amazon CloudWatch를 위한 작업, 리소스 및 조건 키를 참조하십시오.

정책을 쓸 때 *(별표)를 리소스로 사용하여 CloudWatch 작업에 대한 액세스를 제어할 수 있습니다. 예:

```
{
  "Version": "2012-10-17",
  "Statement":[{
      "Effect":"Allow",
      "Action":["cloudwatch:GetMetricStatistics","cloudwatch:ListMetrics"],
      "Resource":"*",
      "Condition":{
         "Bool":{
            "aws:SecureTransport":"true"
```

```
            }
        }
    }
]
}
```

ARN에 대한 자세한 내용은 IAM 사용 설명서의 ARN을 참조하십시오. CloudWatch Logs ARN에 대한 자세한 내용은 Amazon Web Services 일반 참조의 Amazon 리소스 이름(ARN) 및 AWS 서비스 네임스페이스를 참조하십시오. CloudWatch 작업에 대한 정책의 예는 CloudWatch에 대한 자격 증명 기반 정책(IAM 정책) 사용 (p. 362)를 참조하십시오.

작업	ARN(지역 포함)	ARN(IAM 역할에 사용)
`Stop`	arn:aws:automate:us-east-1:ec2:stop	arn:aws:swf:us-east-1:*customer-account*:action/actions/AWS_EC2.InstanceId.Stop/1.0
`Terminate`	arn:aws:automate:us-east-1:ec2:terminate	arn:aws:swf:us-east-1:*customer-account*:action/actions/AWS_EC2.InstanceId.Terminate/1.0
`Reboot`	해당 사항 없음	arn:aws:swf:us-east-1:*customer-account*:action/actions/AWS_EC2.InstanceId.Reboot/1.0
`Recover`	arn:aws:automate:us-east-1:ec2:recover	해당 사항 없음

리소스 소유권 이해

AWS 계정은 리소스를 누가 생성했든 상관없이 계정에서 생성된 리소스를 소유합니다. 특히 리소스 소유자는 리소스 생성 요청을 인증하는 보안 주체 개체, 즉 AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할의 AWS 계정입니다. CloudWatch는 소유가 가능한 어떤 리소스도 포함하고 있지 않습니다.

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 CloudWatch의 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. CloudWatch는 자격 증명 기반 정책만 지원합니다.

주제

- 자격 증명 기반 정책(IAM 정책) (p. 361)
- 리소스 기반 정책(IAM 정책) (p. 361)

자격 증명 기반 정책(IAM 정책)

정책을 IAM 자격 증명에 연결할 수 있습니다. 예를 들면,

- 계정 내 사용자 또는 그룹에 권한 정책 연결 – 특정 사용자 또는 그 사용자가 속한 그룹에 권한 정책을 연결하여 Amazon CloudWatch 리소스(예: metrics) 생성 권한을 해당 사용자에게 부여할 수 있습니다.
- 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 주체는 AWS 서비스 보안 주체일 수도 있습니다.

  IAM을 사용하여 권한을 위임하는 방법에 대한 자세한 정보는 IAM 사용 설명서의 액세스 관리를 참조하십시오.

CloudWatch에서 자격 증명 기반 정책을 사용하는 방법에 대한 자세한 내용은 CloudWatch에 대한 자격 증명 기반 정책(IAM 정책) 사용 (p. 362) 단원을 참조하십시오. 사용자, 그룹, 역할 및 권한에 대한 자세한 내용은 IAM 사용 설명서의 자격 증명(사용자, 그룹 및 역할)을 참조하십시오.

리소스 기반 정책(IAM 정책)

Amazon S3와 같은 다른 서비스도 리소스 기반 권한 정책을 지원합니다. 예를 들어, 정책을 Amazon S3 버킷에 연결하여 해당 버킷에 대한 액세스 권한을 관리할 수 있습니다. CloudWatch는 리소스 기반 정책을 지원하지 않습니다.

정책 요소 지정: 작업, 효과, 보안 주체

각 CloudWatch 리소스에 대해 서비스는 일련의 API 작업을 정의합니다. 이러한 API 작업에 대한 권한을 부여하기 위해 CloudWatch에서는 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 리소스 및 작업 (p. 359) 및 CloudWatch 작업을 참조하십시오.

다음은 기본 정책 요소입니다.

- 리소스 – Amazon 리소스 이름(ARN)을 사용하여 정책을 적용할 리소스를 식별합니다. CloudWatch는 정책 리소스를 사용하여 제어할 수 있는 리소스를 가지고 있지 않기 때문에 IAM 정책에 와일드카드 문자(*)를 사용합니다. 자세한 내용은 CloudWatch 리소스 및 작업 (p. 359) 단원을 참조하십시오.
- 작업 – 작업 키워드를 사용하여 허용 또는 거부할 리소스 작업을 식별합니다. 예를 들어, `cloudwatch:ListMetrics` 권한은 사용자에게 `ListMetrics` 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). CloudWatch의 경우 리소스 기반 정책을 지원하지 않습니다.

IAM 정책 구문과 설명에 대한 자세한 내용은 IAM 사용 설명서의 AWS IAM JSON 정책 참조를 참조하십시오.

모든 CloudWatch API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 Amazon CloudWatch 권한 참조 문서 (p. 376) 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. 각각의 AWS 서비스에서 지원되는 콘텍스트 키 목록과 AWS 전역 정책 키 목록은 IAM 사용 설명서에서 AWS서비스 작업과 조건 콘텍스트 키 및 전역 및 IAM 조건 콘텍스트 키를 참조하십시오.

CloudWatch에 대한 자격 증명 기반 정책(IAM 정책) 사용

이 주제에서는 자격 증명 기반 정책의 예를 통해 계정 관리자가 IAM 자격 증명(사용자, 그룹, 역할)에 권한 정책을 연결함으로써 CloudWatch 리소스에 대한 작업 수행 권한을 부여하는 방법을 보여 줍니다.

> **Important**
>
> CloudWatch 리소스에 대한 액세스 관리를 위해 제공되는 기본 개념과 옵션 설명에 대한 소개 주제 부분을 먼저 읽어 보십시오. 자세한 내용은 액세스 제어 (p. 358) 단원을 참조하십시오.

이 주제의 섹션에서는 다음 내용을 학습합니다.

- CloudWatch 콘솔 사용에 필요한 권한 (p. 362)
- CloudWatch에 대한 AWS 관리형(미리 정의된) 정책 (p. 365)
- 고객 관리형 정책 예 (p. 366)

다음은 권한 정책의 예입니다.

```
{
  "Version": "2012-10-17",
  "Statement":[{
      "Effect":"Allow",
      "Action":["cloudwatch:GetMetricStatistics","cloudwatch:ListMetrics"],
      "Resource":"*",
      "Condition":{
         "Bool":{
            "aws:SecureTransport":"true"
            }
         }
      }
   ]
}
```

이 샘플 정책은 두 가지 CloudWatch 작업(`cloudwatch:GetMetricStatisticsdata` 및 `cloudwatch:ListMetrics`)에 대해 그룹에게 권한을 부여하는 명령문을 하나 가지고 있지만, 그룹이 요청에서 SSL을 사용하는 경우(`"aws:SecureTransport":"true"`)에만 지원됩니다. IAM 정책 명령문 내의 요소에 대한 자세한 내용은 IAM 사용 설명서의 정책 요소 지정: 작업, 효과, 보안 주체 (p. 361) 및 IAM 정책 요소 참조를 참조하십시오.

CloudWatch 콘솔 사용에 필요한 권한

사용자가 CloudWatch 콘솔에서 작업을 할 수 있으려면 AWS 계정에서 다른 AWS 리소스를 설명하도록 허용하는 최소한의 권한 세트가 있어야 합니다. CloudWatch 콘솔에서는 다음 서비스에 대한 권한이 필요합니다.

- Amazon EC2 Auto Scaling
- CloudTrail
- CloudWatch
- CloudWatch 이벤트
- CloudWatch Logs
- Amazon EC2
- Amazon ES
- IAM
- Kinesis
- Lambda
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon SWF
- X-Ray(ServiceLens 기능을 사용하는 경우)

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(CloudWatch에 대한 AWS 관리형 (미리 정의된) 정책 (p. 365) 참조).

AWS CLI 또는 CloudWatch API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다.

CloudWatch 콘솔에서 작업을 수행하는 데 필요한 전체 권한 세트는 아래와 같습니다.

- application-autoscaling:DescribeScalingPolicies
- autoscaling:DescribeAutoScalingGroups
- autoscaling:DescribePolicies
- cloudtrail:DescribeTrails
- cloudwatch:DeleteAlarms
- cloudwatch:DescribeAlarmHistory
- cloudwatch:DescribeAlarms
- cloudwatch:GetMetricData
- cloudwatch:GetMetricStatistics
- cloudwatch:ListMetrics
- cloudwatch:PutMetricAlarm
- cloudwatch:PutMetricData
- ec2:DescribeInstances
- ec2:DescribeTags
- ec2:DescribeVolumes
- es:DescribeElasticsearchDomain
- es:ListDomainNames
- events:DeleteRule
- events:DescribeRule

- events:DisableRule
- events:EnableRule
- events:ListRules
- events:PutRule
- iam:AttachRolePolicy
- iam:CreateRole
- iam:GetPolicy
- iam:GetPolicyVersion
- iam:GetRole
- iam:ListAttachedRolePolicies
- iam:ListRoles
- kinesis:DescribeStream
- kinesis:ListStreams
- lambda:AddPermission
- lambda:CreateFunction
- lambda:GetFunctionConfiguration
- lambda:ListAliases
- lambda:ListFunctions
- lambda:ListVersionsByFunction
- lambda:RemovePermission
- logs:CancelExportTask
- logs:CreateExportTask
- logs:CreateLogGroup
- logs:CreateLogStream
- logs:DeleteLogGroup
- logs:DeleteLogStream
- logs:DeleteMetricFilter
- logs:DeleteRetentionPolicy
- logs:DeleteSubscriptionFilter
- logs:DescribeExportTasks
- logs:DescribeLogGroups
- logs:DescribeLogStreams
- logs:DescribeMetricFilters
- logs:DescribeSubscriptionFilters
- logs:FilterLogEvents
- logs:GetLogEvents
- logs:PutMetricFilter
- logs:PutRetentionPolicy
- logs:PutSubscriptionFilter
- logs:TestMetricFilter
- s3:CreateBucket

- s3:ListBucket
- sns:CreateTopic
- sns:GetTopicAttributes
- sns:ListSubscriptions
- sns:ListTopics
- sns:SetTopicAttributes
- sns:Subscribe
- sns:Unsubscribe
- sqs:GetQueueAttributes
- sqs:GetQueueUrl
- sqs:ListQueues
- sqs:SetQueueAttributes
- swf:CreateAction
- swf:DescribeAction
- swf:ListActionTemplates
- swf:RegisterAction
- swf:RegisterDomain
- swf:UpdateAction

또한 ServiceLens에서 서비스 맵을 보려면 `AWSXrayReadOnlyAccess`가 필요합니다.

CloudWatch에 대한 AWS 관리형(미리 정의된) 정책

AWS는 AWS에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반 사용 사례를 처리합니다. 이러한 AWS 관리형 정책은 사용자가 필요한 권한을 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 단원을 참조하십시오.

계정의 사용자에게 연결할 수 있는 다음 AWS 관리형 정책은 CloudWatch에 고유합니다.

- CloudWatchFullAccess – CloudWatch에 대한 전체 액세스 권한을 부여합니다.
- CloudWatchReadOnlyAccess – CloudWatch에 대한 읽기 전용 액세스 권한을 부여합니다.
- CloudWatchActionsEC2Access – CloudWatch 경보 및 지표를 비롯해 Amazon EC2 메타데이터에 대한 읽기 전용 액세스 권한을 부여합니다. EC2 인스턴스에 대한 API 작업을 중지, 종료 및 재부팅할 수 있는 액세스 권한을 부여합니다.

  Note

  IAM 콘솔에 로그인하고 이 콘솔에서 특정 정책을 검색하여 이러한 권한 정책을 검토할 수 있습니다.

CloudWatch 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.

CloudWatch Synthetics용 AWS 관리형(미리 정의된) 정책

계정의 사용자에 연결할 수 있는 다음 AWS 관리형 정책은 CloudWatch Synthetics에 고유합니다.

- CloudWatchSyntheticsFullAccess 또는 CloudWatchSyntheticsReadOnlyAccess – 카나리아 세부 정보와 카나리아 테스트 실행의 결과를 봅니다.

- AmazonS3ReadOnlyAccess and CloudWatchReadOnlyAccess – CloudWatch 콘솔에서 모든 Synthetics 데이터를 읽을 수 있습니다.
- AWSLambdaReadOnlyAccess – 카나리아에 사용되는 소스 코드를 볼 수 있습니다.
- CloudWatchSyntheticsFullAccess – 카나리아를 생성할 수 있습니다. 또한 새 IAM 역할을 생성할 카나리아를 생성하려면 다음 인라인 정책 설명도 필요합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

Important

사용자에게 `iam:CreateRole`, `iam:CreatePolicy` 및 `iam:AttachRolePolicy` 권한을 부여하면 해당 사용자에게 AWS 계정에 대한 전체 관리 액세스 권한이 부여됩니다. 예를 들어 이러한 권한을 가진 사용자는 모든 리소스에 대한 전체 권한을 가진 정책을 생성하고 해당 정책을 모든 역할에 연결할 수 있습니다. 이러한 권한을 부여한 사람에게 매우 주의해야 합니다.

정책 연결 및 사용자에게 권한 부여에 대한 자세한 내용은 IAM 사용자의 권한 변경 및 사용자 또는 역할의 인라인 정책을 포함하려면을 참조하십시오.

고객 관리형 정책 예

이 단원에서는 다양한 CloudWatch 작업에 대한 권한을 부여하는 사용자 정책의 예를 제공합니다. 이러한 정책은 CloudWatch API, AWS SDK 또는 AWS CLI를 사용하는 경우에 적용됩니다.

예제

- 예제 1: 사용자에게 CloudWatch에 대한 전체 액세스 권한 부여 (p. 366)
- 예제 2: CloudWatch에 대한 읽기 전용 액세스 허용 (p. 367)
- 예제 3: Amazon EC2 인스턴스 중지 또는 종료 (p. 367)

예제 1: 사용자에게 CloudWatch에 대한 전체 액세스 권한 부여

다음 정책은 사용자에게 모든 CloudWatch 작업, CloudWatch Logs 작업, Amazon SNS 작업에 대한 액세스 권한과 Amazon EC2 Auto Scaling에 대한 읽기 전용 액세스 권한을 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:*",
```

```
        "logs:*",
        "sns:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

예제 2: CloudWatch에 대한 읽기 전용 액세스 허용

다음 정책은 사용자에게 CloudWatch에 대한 읽기 전용 액세스와 Amazon EC2 Auto Scaling 작업, CloudWatch 지표, CloudWatch Logs 데이터 및 경보 관련 Amazon SNS 데이터를 볼 수 있는 권한을 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "logs:Get*",
        "logs:Describe*",
        "sns:Get*",
        "sns:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

예제 3: Amazon EC2 인스턴스 중지 또는 종료

아래 정책은 CloudWatch 경보 작업이 EC2 인스턴스를 중지하거나 종료하도록 허용합니다. 아래 샘플에서 GetMetricStatistics, ListMetrics 및 DescribeAlarms 작업은 선택 사항입니다. 인스턴스를 제대로 중지 또는 종료하려면 이러한 작업을 포함하는 것이 좋습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "cloudwatch:DescribeAlarms"
      ],
      "Sid": "00000000000000",
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:StopInstances",
```

```
        "ec2:TerminateInstances"
      ],
      "Sid": "00000000000000",
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

조건 키를 사용하여 CloudWatch 네임스페이스에 대한 액세스 제한

IAM 조건 키를 사용하여 사용자가 지정된 CloudWatch 네임스페이스에만 지표를 게시하도록 제한합니다.

하나의 네임스페이스에만 게시 허용

다음 정책은 사용자가 `MyCustomNamespace`라는 네임스페이스에만 지표를 게시하도록 제한합니다.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "MyCustomNamespace"
            }
        }
    }
}
```

네임스페이스에서 게시 제외

다음 정책은 사용자가 `CustomNamespace2`를 제외한 모든 네임스페이스에 지표를 게시할 수 있도록 허용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData"
        },
        {
            "Effect": "Deny",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CustomNamespace2"
                }
            }
        }
    ]
}
```

CloudWatch에 서비스 연결 역할 사용

Amazon CloudWatch의 경우 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 CloudWatch에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 CloudWatch에서 사전 정의하며 서비스에서 다른 AWS 제품을 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다.

CloudWatch에 대한 서비스 연결 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없으므로 Amazon EC2 인스턴스를 종료, 중지 또는 재부팅할 수 있는 CloudWatch 경보를 설정할 수 있습니다. 다른 서비스 연결 역할을 사용하면 모니터링 계정에서 사용자가 지정한 다른 계정의 CloudWatch 데이터에 액세스하여 교차 계정 교차 리전 대시보드를 구축할 수 있습니다.

CloudWatch에서 이 서비스 연결 역할 권한을 정의하므로, 달리 정의되지 않은 한 CloudWatch만 해당 역할을 맡을 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 개체에 연결할 수 없습니다.

먼저 역할의 관련 리소스를 삭제해야만 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 실수로 삭제할 수 없기 때문에 CloudWatch 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 IAM으로 작업하는 AWS 서비스를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

CloudWatch 경보에 대한 서비스 연결 역할 권한

CloudWatch에서는 AWSServiceRoleForCloudWatchEvents – CloudWatch uses this service-linked role to perform Amazon EC2 alarm actions. 서비스 연결 역할을 사용합니다.

AWSServiceRoleForCloudWatchEvents 서비스 연결 역할을 맡을 CloudWatch 이벤트 서비스를 신뢰합니다. 경보에 의해 호출된 CloudWatch 이벤트는 인스턴스 작업의 종료, 중지 또는 재부팅을 호출합니다.

AWSServiceRoleForCloudWatchEvents 서비스 연결 역할 권한 정책은 CloudWatch 이벤트가 Amazon EC2 인스턴스에서 다음 작업을 완료하도록 해줍니다.

- `ec2:StopInstances`
- `ec2:TerminateInstances`
- `ec2:RecoverInstances`
- `ec2:DescribeInstanceRecoveryAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`

AWSServiceRoleForCloudWatchCrossAccount 서비스 연결 역할 권한 정책을 통해 CloudWatch가 다음 작업을 완료할 수 있습니다.

- `sts:AssumeRole`

CloudWatch 교차 계정 교차 리전에 대한 서비스 연결 역할 권한

CloudWatch은 AWSServiceRoleForCloudWatchCrossAccount라는 서비스 연결 역할을 사용합니다. CloudWatch는 이 역할을 사용하여 사용자가 지정한 다른 AWS 계정의 CloudWatch 데이터에 액세스합니다. SLR은 CloudWatch 서비스가 공유 계정에서 역할을 맡을 수 있도록 역할 수임 권한만 제공합니다. 데이터에 대한 액세스 권한을 제공하는 공유 역할입니다.

AWSServiceRoleForCloudWatchCrossAccount 서비스 연결 역할을 맡을 CloudWatch 서비스를 신뢰합니다.

CloudWatch에 대한 서비스 연결 역할 생성

이러한 서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management 콘솔, IAM CLI 또는 IAM에서 경보를 처음 생성하면 CloudWatch이 AWSServiceRoleForCloudWatchEvents을 자동으로 생성됩니다. 교차 계정 교차 리전 기능에 대한 모니터링 계정이 되도록 처음 계정을 활성화하면 CloudWatch가 AWSServiceRoleForCloudWatchCrossAccount를 자동으로 생성합니다.

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 생성을 참조하십시오.

CloudWatch에 대한 서비스 연결 역할 편집

CloudWatch에서는 AWSServiceRoleForCloudWatchEvents 또는 AWSServiceRoleForCloudWatchCrossAccount 역할을 편집할 수 없습니다. 다양한 개체가 역할을 참조할 수 있으므로 이러한 역할을 생성한 후에는 이름을 변경할 수 없습니다. 그러나 IAM을 사용하여 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할 설명 편집(IAM 콘솔)

IAM 콘솔을 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 편집하려면(콘솔 사용)

1. IAM 콘솔의 탐색 창에서 역할을 선택합니다.
2. 변경할 역할 이름을 선택합니다.
3. Role description(역할 설명)의 맨 오른쪽에서 편집을 선택합니다.
4. 상자에 새 설명을 입력하고 저장을 선택합니다.

서비스 연결 역할 설명 편집(AWS CLI)

AWS Command Line Interface에서 IAM 명령을 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 변경하려면(AWS CLI)

1. (옵션) 역할의 현재 설명을 보려면 다음 명령 중 하나를 사용합니다.

   ```
   $ aws iam get-role --role-name role-name
   ```

   AWS CLI 명령에서 역할을 참조하려면 ARN이 아니라 역할 이름을 사용해야 합니다. 예를 들어, 어떤 역할의 ARN이 `arn:aws:iam::123456789012:role/myrole`인 경우 참조할 역할은 **`myrole`**입니다.
2. 서비스 연결 역할의 설명을 업데이트하려면 다음 명령을 사용합니다.

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

서비스 연결 역할 설명 편집(IAM API)

IAM API를 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 변경하려면(API 사용)

1. (선택 사항) 역할의 현재 설명을 보려면 다음 명령을 사용합니다.

   GetRole
2. 역할 설명을 업데이트하려면 다음 명령을 사용합니다.

   UpdateRoleDescription

CloudWatch에 대한 서비스 연결 역할 삭제

자동으로 EC2 인스턴스를 중지, 종료 또는 재부팅하는 경보가 더 이상 없는 경우 AWSServiceRoleForCloudWatchEvents 역할을 삭제하는 것이 좋습니다.

따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 개체가 없도록 합니다. 단, 삭제 전에 서비스 연결 역할을 정리해야 합니다.

서비스 연결 역할 정리

IAM을 사용하여 서비스 연결 역할을 삭제하기 전에 먼저 역할에 활성 세션이 없는지 확인하고 역할에서 사용되는 리소스를 모두 제거해야 합니다.

IAM 콘솔에서 서비스 연결 역할에 활성 세션이 있는지 확인하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다. AWSServiceRoleForCloudWatchEvents 역할의 이름(확인란 아님)을 선택합니다.
3. 선택된 역할의 요약 페이지에서 Access Advisor(액세스 관리자)를 선택하고 서비스 연결 역할의 최근 활동을 검토합니다.

   Note

   CloudWatch에서 AWSServiceRoleForCloudWatchEvents 역할을 사용하는지 잘 모를 경우에는 역할을 삭제해 보십시오. 서비스에서 역할을 사용하는 경우에는 삭제가 되지 않으며, 역할이 사용되고 있는 리전을 볼 수 있습니다. 역할이 사용 중인 경우에는 세션이 종료될 때까지 기다렸다가 역할을 삭제해야 합니다. 서비스 연결 역할에 대한 세션은 취소할 수 없습니다.

서비스 연결 역할 삭제(IAM 콘솔)

IAM 콘솔을 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할을 삭제하려면(콘솔)

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택합니다. 그런 다음 삭제할 역할의 이름이나 행이 아닌 이름 옆에 있는 확인란을 선택합니다.
3. Role actions(역할 작업)에 대해 Delete role(역할 삭제)을 선택합니다.
4. 확인 대화 상자가 나타나면 서비스 마지막 액세스 데이터를 검토합니다. 이 데이터는 선택한 각 역할이 AWS 서비스를 마지막으로 액세스한 일시를 보여 줍니다. 이를 통해 역할이 현재 활동 중인지 여부를 확인할 수 있습니다. 계속하려면 예, 삭제를 선택합니다.
5. IAM 콘솔 알림을 보고 서비스 연결 역할 삭제 진행 상황을 모니터링합니다. IAM 서비스 연결 역할 삭제는 비동기이므로 삭제할 역할을 제출한 후에 삭제 작업이 성공하거나 실패할 수 있습니다. 작업에 실패할 경우 알림의 세부 정보 보기 또는 View Resources(리소스 보기)를 선택하면 삭제 실패 이유를 확인할 수 있습니다. 역할에서 사용 중인 리소스가 서비스에 있기 때문에 삭제에 실패하는 경우, 실패 원인에 리소스 목록이 포함됩니다.

서비스 연결 역할 삭제(AWS CLI)

AWS Command Line Interface에서 IAM 명령을 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할을 삭제하려면(AWS CLI)

1. 서비스 연결 역할이 사용되지 않거나 연결된 리소스가 없는 경우에는 서비스 연결 역할을 삭제할 수 없으므로 삭제 요청을 제출해야 합니다. 이러한 조건이 충족되지 않으면 요청이 거부될 수 있습니다. 삭제

작업 상태를 확인하려면 응답의 `deletion-task-id`를 캡처해야 합니다. 다음 명령을 입력하여 서비스 연결 역할 삭제 요청을 제출합니다.

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForCloudWatchEvents
```

2. 다음 명령을 입력하여 삭제 작업의 상태를 확인합니다.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

삭제 작업은 `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` 또는 `FAILED` 상태일 수 있습니다. 삭제에 실패할 경우 문제를 해결할 수 있도록 실패 이유가 호출에 반환됩니다.

서비스 연결 역할 삭제(IAM API)

IAM API를 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할(API)을 삭제하려면

1. 서비스 연결 역할 삭제 요청을 제출하려면 DeleteServiceLinkedRole를 호출합니다. 요청에서 삭제할 역할 이름을 지정합니다.

   서비스 연결 역할이 사용되지 않거나 연결된 리소스가 없는 경우에는 서비스 연결 역할을 삭제할 수 없으므로 삭제 요청을 제출해야 합니다. 이러한 조건이 충족되지 않으면 요청이 거부될 수 있습니다. 삭제 작업 상태를 확인하려면 응답의 `DeletionTaskId`를 캡처해야 합니다.
2. 삭제 상태를 확인하려면 GetServiceLinkedRoleDeletionStatus를 호출합니다. 요청에 `DeletionTaskId`를 지정합니다.

   삭제 작업은 `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` 또는 `FAILED` 상태일 수 있습니다. 삭제에 실패할 경우 문제를 해결할 수 있도록 실패 이유가 호출에 반환됩니다.

CloudWatch Application Insights for .NET and SQL Server에 서비스 연결 역할 사용

CloudWatch Application Insights의 경우 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 CloudWatch Application Insights for .NET and SQL Server에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 CloudWatch Application Insights for .NET and SQL Server에서 사전 정의하며 서비스에서 사용자 대신 다른 AWS 제품을 호출하기 위해 필요한 모든 권한을 포함합니다.

서비스 연결 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없기 때문에 CloudWatch Application Insights for .NET and SQL Server 설정이 쉬워집니다. CloudWatch Application Insights for .NET and SQL Server에서 서비스 연결 역할 권한을 정의하므로, 달리 정의되지 않은 한 CloudWatch Application Insights for .NET and SQL Server에서만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며, 이 권한 정책은 다른 IAM 개체에 연결할 수 없습니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 IAM으로 작업하는 AWS 서비스를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾으십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

CloudWatch Application Insights for .NET and SQL Server에 대한 서비스 연결 역할 권한

CloudWatch Application Insights for .NET and SQL Server에서는 이름이 ApplicationInsights-role인 서비스 연결 역할을 사용합니다. Application Insights uses this role to perform operations such as analyzing

the Resource Groups of the customer, creating CloudFormation stacks to create alarms on metrics, and configuring the CloudWatch Agent on EC2 instances.

역할 권한 정책은 CloudWatch Application Insights for .NET and SQL Server가 모든 리소스에서 다음 작업을 완료하도록 허용합니다.

- `cloudwatch:DescribeAlarmHistory`
- `cloudwatch:DescribeAlarms`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `cloudwatch:PutMetricAlarm`
- `cloudwatch:DeleteAlarms`
- `logs:GetLogEvents`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroup`
- `events:DescribeRule`
- `tag:GetResources`
- `resource-groups:ListGroupResources`
- `resource-groups:GetGroupQuery`
- `elasticloadbalancing:DescribeLoadBalancers`
- `elasticloadbalancing:DescribeTargetGroups`
- `elasticloadbalancing:DescribeTargetHealth`
- `Autoscaling:DescribeAutoScalingGroups`
- `cloudFormation:DescribeStacks`
- `cloudFormation:ListStackResources`
- `ssm:GetOpsItem`
- `ssm:CreateOpsItem`
- `ssm:DescribeOpsItems`
- `ssm:UpdateOpsItem`
- `ssm:DescribeInstanceInformation`
- `ec2:DescribeInstances`
- `rds:DescribeDBInstances`

역할 권한 정책을 사용하면 CloudWatch Application Insights for .NET and SQL Server에서 지정된 리소스에 대해 다음 작업을 완료할 수 있습니다. 이러한 정책은 계정의 .NET 및 SQL Server용 CloudWatch Application Insights에 의해 생성되거나 관련된 리소스를 관리하는 데 사용됩니다.

```
    {
          "Effect": "Allow",
          "Action": [
              "cloudFormation:CreateStack",
              "cloudFormation:UpdateStack",
              "cloudFormation:DeleteStack"
          ],
          "Resource": [
              "arn:aws:cloudformation:*:*:stack/ApplicationInsights-*"
          ]
      },
      {
```

```
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter",
                "ssm:DeleteParameter",
                "ssm:AddTagsToResource",
                "ssm:RemoveTagsFromResource"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-ApplicationInsights-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:CreateAssociation",
                "ssm:UpdateAssociation",
                "ssm:DeleteAssociation",
                "ssm:DescribeAssociation"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ssm:*:*:association/*",
                "arn:aws:ssm:*:*:managed-instance/*",
                "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure",
                "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
                "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
            ]
        },
            ]
        }
```

IAM 개체(사용자, 그룹, 역할 등)가 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 권한을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하십시오.

CloudWatch Application Insights for .NET and SQL Server에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management 콘솔에서 create a new Application Insights application 시 CloudWatch Application Insights for .NET and SQL Server에서 서비스 연결 역할을 생성합니다.

이 서비스 연결 역할을 삭제한 다음 다시 생성하려는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. create a new Application Insights application 시 CloudWatch Application Insights for .NET and SQL Server에서 서비스 연결 역할을 다시 생성합니다.

CloudWatch Application Insights for .NET and SQL Server에 대한 서비스 연결 역할 편집

CloudWatch Application Insights for .NET and SQL Server에서는 ApplicationInsights-role 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 그러나 IAM를 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스에 연결 역할 편집을 참조하십시오.

CloudWatch Application Insights for .NET and SQL Server에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 이렇게 하면 적극적으로 모니터링되거나 유지 관리되지 않는 미사용 개체를 피할 수 있습니다. 그러나 역할을 수동으로 삭제하기 전에 Application Insights에서 모든 애플리케이션을 삭제해야 합니다.

Note

리소스를 삭제하려 할 때 CloudWatch Application Insights for .NET and SQL Server 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하십시오.

ApplicationInsights-role에서 사용하는 CloudWatch Application Insights for .NET and SQL Server 리소스를 삭제하려면

- CloudWatch Application Insights for .NET and SQL Server 애플리케이션을 모두 삭제합니다. 자세한 내용은 CloudWatch Application Insights for .NET and SQL Server 사용 설명서의 "애플리케이션 삭제"를 참조하십시오.

IAM을 사용하여 서비스 연결 역할을 수동으로 삭제하려면

IAM 콘솔, AWS CLI 또는 AWS API를 사용하여 ApplicationInsights-role 서비스 연결 역할을 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 서비스에 연결 역할 삭제 단원을 참조하십시오.

CloudWatch Application Insights for .NET and SQL Server 서비스 연결 역할에 대해 지원되는 리전

CloudWatch Application Insights for .NET and SQL Server에서는 서비스를 사용할 수 있는 모든 AWS 리전에서 서비스 연결 역할 사용을 지원합니다.

리전 이름	리전 자격 증명	CloudWatch Application Insights for .NET and SQL Server의 지원
아시아 태평양(뭄바이)	ap-south-1	예
아시아 태평양(서울)	ap-northeast-2	예
아시아 태평양(싱가포르)	ap-southeast-1	예
아시아 태평양(시드니)	ap-southeast-2	예
유럽(프랑크푸르트)	eu-central-1	예
유럽(아일랜드)	eu-west-1	예
유럽(런던)	eu-west-2	예
유럽(파리)	eu-west-3	예
유럽(스톡홀름)	eu-north-1	예
남아메리카(상파울루)	sa-east-1	예
아시아 태평양(도쿄)	ap-northeast-1	예
미국 동부(버지니아 북부)	us-east-1	예
미국 동부(오하이오)	us-east-2	예
미국 서부(캘리포니아 북부 지역)	us-west-1	예
미국 서부(오레곤)	us-west-2	예
캐나다(중부)	ca-central-1	예

Amazon CloudWatch 권한 참조 문서

IAM 자격 증명에 연결할 수 있는 액세스 제어 (p. 358) 및 쓰기 권한 정책(자격 증명 기반 정책)을 설정할 때 다음 표를 참조로 사용할 수 있습니다. 표에는 각 CloudWatch API 작업과 이 작업을 수행할 수 있는 권한을 부여할 수 있는 작업이 나와 있습니다. 정책의 `Action` 필드에서 작업을 지정하고, 정책의 `Resource` 필드에서 리소스 값으로 와일드카드 문자 (*)를 지정합니다.

CloudWatch 정책에서 AWS 차원 조건 키를 사용하여 조건을 표시할 수 있습니다. AWS 차원 키의 전체 목록을 보려면 IAM 사용 설명서의 AWS글로벌 및 IAM 조건 콘텍스트 키를 참조하십시오.

> **Note**
>
> 작업을 지정하려면 `cloudwatch:` 접두사 다음에 API 작업 이름을 사용합니다. 예: `cloudwatch:GetMetricStatistics`, `cloudwatch:ListMetrics` 또는 `cloudwatch:*`(모든 CloudWatch 작업의 경우).

테이블

- CloudWatch API 작업 및 필수 권한
- CloudWatch 이벤트 API 작업 및 필수 권한
- CloudWatch Logs API 작업 및 필수 권한
- Amazon EC2 API 작업 및 필수 권한
- Amazon EC2 Auto Scaling API 작업 및 필수 권한

CloudWatch API 작업 및 작업에 필요한 권한

CloudWatchAPI 연산	필요한 권한(API 작업)
DeleteAlarms	`cloudwatch:DeleteAlarms` 경보를 삭제하는 데 필요합니다.
DeleteDashboards	`cloudwatch:DeleteDashboards` 대시보드를 삭제하는 데 필요합니다.
DescribeAlarmHistory	`cloudwatch:DescribeAlarmHistory` 경보 기록을 보는 데 필요합니다.
DescribeAlarms	`cloudwatch:DescribeAlarms` 이름으로 경보 정보를 검색하는 데 필요합니다.
DescribeAlarmsForMetric	`cloudwatch:DescribeAlarmsForMetric` 지표에 대한 경보를 보는 데 필요합니다.
DisableAlarmActions	`cloudwatch:DisableAlarmActions` 경보 작업을 비활성화하는 데 필요합니다.
EnableAlarmActions	`cloudwatch:EnableAlarmActions` 경보 작업을 활성화하는 데 필요합니다.

CloudWatchAPI 연산	필요한 권한(API 작업)
GetDashboard	`cloudwatch:GetDashboard` 기존 대시보드에 대한 데이터를 표시하는 데 필요합니다.
GetMetricData	`cloudwatch:GetMetricData` 커다란 지표 데이터 배치를 가져와 이 데이터의 지표 수식을 계산해야 합니다.
GetMetricStatistics	`cloudwatch:GetMetricStatistics` CloudWatch 콘솔의 다른 부분과 대시보드 위젯에서 그래프를 보는 데 필요합니다.
GetMetricWidgetImage	`cloudwatch:GetMetricWidgetImage` CloudWatch 지표 1개 이상의 스냅샷 그래프를 비트맵 이미지로 가져오는 데 필요합니다.
ListDashboards	`cloudwatch:ListDashboards` 계정에서 CloudWatch 대시보드의 목록을 보는 데 필요합니다.
ListMetrics	`cloudwatch:ListMetrics` CloudWatch 콘솔 내에서, 그리고 CLI에서 지표 이름을 보거나 검색하는 데 필요합니다. 대시보드 위젯에서 지표를 선택하는 데 필요합니다.
PutDashboard	`cloudwatch:PutDashboard` 대시보드를 생성하거나 기존 대시보드를 업데이트하는 데 필요합니다.
PutMetricAlarm	`cloudwatch:PutMetricAlarm` 경보를 생성 또는 업데이트하는 데 필요합니다.
PutMetricData	`cloudwatch:PutMetricData` 지표를 만드는 데 필요합니다.
SetAlarmState	`cloudwatch:SetAlarmState` 경보 상태를 수동으로 설정하는 데 필요합니다.

CloudWatch 이벤트 API 작업 및 작업에 필요한 권한

CloudWatch 이벤트API 연산	필요한 권한(API 작업)
DeleteRule	`events:DeleteRule` 규칙을 삭제하는 데 필요합니다.
DescribeRule	`events:DescribeRule`

CloudWatch 이벤트API 연산	필요한 권한(API 작업)
	규칙에 대한 세부 사항을 나열하는 데 필요합니다.
DisableRule	`events:DisableRule` 규칙을 비활성화하는 데 필요합니다.
EnableRule	`events:EnableRule` 규칙을 활성화하는 데 필요합니다.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget` 대상과 연관된 규칙을 나열하는 데 필요합니다.
ListRules	`events:ListRules` 계정에서 모든 그룹을 나열하는 데 필요합니다.
ListTargetsByRule	`events:ListTargetsByRule` 규칙과 연관된 모든 대상을 나열하는 데 필요합니다.
PutEvents	`events:PutEvents` 규칙에 일치시킬 수 있는 사용자 지정 이벤트를 추가하는 데 필요합니다.
PutRule	`events:PutRule` 규칙을 생성 또는 업데이트하는 데 필요합니다.
PutTargets	`events:PutTargets` 규칙에 대상을 추가하는 데 필요합니다.
RemoveTargets	`events:RemoveTargets` 규칙에서 대상을 제거하는 데 필요합니다.
TestEventPattern	`events:TestEventPattern` 특정 이벤트를 기준으로 이벤트 패턴을 테스트하는 데 필요합니다.

CloudWatch Logs API 작업 및 작업에 필요한 권한

CloudWatch LogsAPI 연산	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.

CloudWatch LogsAPI 연산	필요한 권한(API 작업)
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.
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.

CloudWatch LogsAPI 연산	필요한 권한(API 작업)
DescribeMetricFilters	`logs:DescribeMetricFilters` 로그 그룹과 연관된 모든 지표를 보는 데 필요합니다.
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.
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.
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.
TestMetricFilter	`logs:TestMetricFilter` 로그 이벤트 메시지 샘플을 기준으로 필터 패턴을 테스트하는 데 필요합니다.

Amazon EC2 API 작업 및 작업에 필요한 권한

Amazon EC2API 연산	필요한 권한(API 작업)
DescribeInstanceStatus	`ec2:DescribeInstanceStatus` EC2 인스턴스 상태에 대한 세부 정보를 보는 데 필요합니다.
DescribeInstances	`ec2:DescribeInstances` EC2 인스턴스에 대한 세부 정보를 보는 데 필요합니다.
RebootInstances	`ec2:RebootInstances` EC2 인스턴스를 재부팅하는 데 필요합니다.
StopInstances	`ec2:StopInstances` EC2 인스턴스를 중지하는 데 필요합니다.
TerminateInstances	`ec2:TerminateInstances` EC2 인스턴스를 종료하는 데 필요합니다.

Amazon EC2 Auto Scaling API 작업 및 작업에 필요한 권한

Amazon EC2 Auto ScalingAPI 연산	필요한 권한(API 작업)
확장	`autoscaling:Scaling` Auto Scaling 그룹을 확장하는 데 필요합니다.
트리거	`autoscaling:Trigger` Auto Scaling 작업을 트리거하는 데 필요합니다.

Amazon CloudWatch에 대한 규정 준수 확인

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon CloudWatch의 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 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 CloudWatch의 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 CloudWatch의 인프라 보안

관리형 서비스인 Amazon CloudWatch는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon CloudWatch에 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.

네트워크 격리

Virtual Private Cloud(VPC)는 AWS 클라우드에서 논리적으로 격리된 고유한 영역의 가상 네트워크입니다. 서브넷은 VPC의 IP 주소 범위입니다. VPC의 서브넷에 다양한 AWS 리소스를 배포할 수 있습니다. 예를 들어 Amazon EC2 인스턴스, EMR 클러스터 및 DynamoDB 테이블을 서브넷에 배포할 수 있습니다. 자세한 내용은 Amazon VPC 사용 설명서 단원을 참조하십시오.

퍼블릭 인터넷을 거치지 않고 VPC의 리소스와 통신할 수 있도록 CloudWatch를 활성화하려면 AWS PrivateLink를 사용합니다. 자세한 내용은 인터페이스 VPC 엔드포인트와 함께 CloudWatch 사용 (p. 382) 단원을 참조하십시오.

프라이빗 서브넷은 퍼블릭 인터넷에 대한 기본 경로가 없는 서브넷입니다. 프라이빗 서브넷에 AWS 리소스를 배포하면 Amazon CloudWatch가 리소스에서 내장된 지표를 수집하는 것을 방지할 수 없습니다.

프라이빗 서브넷에 AWS 리소스의 사용자 지정 지표를 게시해야 하는 경우 프록시 서버를 사용하여 게시할 수 있습니다. 프록시 서버는 이러한 HTTPS 요청을 CloudWatch의 퍼블릭 API 엔드포인트로 전달합니다.

인터페이스 VPC 엔드포인트와 함께 CloudWatch 사용

Amazon Virtual Private Cloud(Amazon VPC)를 사용하여 AWS 리소스를 호스팅하는 경우, VPC와 CloudWatch 간에 프라이빗 연결을 설정할 수 있습니다. 이 연결을 사용하면 CloudWatch가 퍼블릭 인터넷을 통하지 않고 VPC의 리소스와 통신하게 할 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 CloudWatch에 연결하려면 CloudWatch에 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 CloudWatch에 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 CloudWatch가 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)

CloudWatch용 VPC 종단점 생성

VPC에서 CloudWatch를 사용하기 시작하려면 CloudWatch에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 선택할 서비스 이름은 com.amazonaws.***Region***.monitoring입니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 생성을 참조하십시오.

CloudWatch에 대해 설정을 변경할 필요가 없습니다. CloudWatch는 퍼블릭 엔드포인트 또는 프라이빗 인터페이스 VPC 엔드포인트 중 사용 중인 엔드포인트를 사용하여 다른 AWS 서비스를 호출합니다. 예를 들어 CloudWatch용 인터페이스 VPC 엔드포인트를 생성할 때 VPC에 위치한 리소스에서 CloudWatch로 흐르는 지표가 이미 있는 경우에는 이 지표가 기본적으로 인터페이스 VPC 엔드포인트를 통해 흐르기 시작합니다.

CloudWatch VPC 종단점에 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 CloudWatch에 대한 엔드포인트 정책 예제입니다. 이 정책은 사용자가 VPC를 통해 CloudWatch에 연결하여 CloudWatch으로 지표 데이터를 전송할 수 있도록 허용하고, 다른 CloudWatch 작업을 수행하지 못하게 금지합니다.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

CloudWatch에 대한 VPC 종단점 정책을 수정하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 엔드포인트를 선택합니다.
3. CloudWatch에 대한 엔드포인트를 아직 생성하지 않았다면 엔드포인트 생성을 선택합니다. 그런 다음 com.amazonaws.***Region***.monitoring를 선택하고 엔드포인트 생성을 선택합니다.
4. com.amazonaws.***Region***.monitoring 엔드포인트를 선택하고 화면 하단의 정책 탭을 선택합니다.
5. 정책 편집을 선택하고 정책을 변경합니다.

Synthetics 카나리아에 대한 보안 고려 사항

CloudWatch Synthetics is in open preview. The preview is currently available in the following Regions:

- 미국 동부(버지니아 북부)
- 미국 동부(오하이오)
- 미국 서부(캘리포니아 북부 지역)
- 미국 서부(오레곤)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(스톡홀름)
- 유럽(아일랜드)
- 유럽(런던)

- 유럽(파리)
- 아시아 태평양(도쿄)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(뭄바이)
- 남아메리카(상파울루)

The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback or let us know if you would like to be informed when updates are made by emailing us at cw-synthetics@amazon.com

다음 단원에서는 Synthetices에서 카나리아를 생성 및 실행할 때 고려해야 할 보안 문제에 대해 설명합니다.

보안 연결 사용

카나리아 코드와 카나리아 테스트 실행의 결과에는 중요한 정보가 포함될 수 있으므로 암호화되지 않은 연결을 통해 엔드포인트에 카나리아를 연결하지 마십시오. `https://`로 시작하는 것과 같이 항상 암호화된 연결을 사용합니다.

카나리아 이름 지정 고려 사항

카나리아의 Amazon 리소스 이름(ARN)은 CloudWatch Synthetics 래퍼 라이브러리의 일부로 포함된 Puppeteer 기반 Chromium 브라우저에서 수행된 아웃바운드 호출의 일부로 사용자 에이전트 헤더에 포함됩니다. 이렇게 하면 CloudWatch Synthetics 카나리아 트래픽을 식별하여 호출을 수행 중인 카나리아와 다시 관련시킬 수 있습니다.

카나리아 ARN에는 카나리아 이름이 포함되어 있습니다. 독점 정보를 공개하지 않는 카나리아 이름을 선택하십시오.

또한 자신이 제어하는 웹 사이트 및 엔드포인트의 카나리아만 가리키도록 해야 합니다.

카나리아 코드의 암호

액세스 키 또는 데이터베이스 자격 증명과 같은 암호를 카나리아 소스 코드에 포함하지 않는 것이 좋습니다. AWS Secrets Manager를 사용하여 암호를 안전하게 유지하는 방법에 대한 자세한 내용은 AWS Secrets Manager란 무엇입니까?를 참조하십시오.

사용 권한 고려 사항

CloudWatch Synthetics가 만들거나 사용하는 리소스에 대한 액세스를 제한하는 것이 좋습니다. 카나리아가 테스트 실행 결과와 로그 및 스크린샷과 같은 기타 아티팩트를 저장하는 Amazon S3 버킷에 대해 엄격한 권한을 사용하십시오.

마찬가지로 카나리아 소스 코드가 저장된 위치에 대한 권한을 엄격하게 유지하여 사용자가 실수로 또는 악의적으로 카나리아에 사용된 Lambda 계층이나 Lambda 함수를 삭제하지 않도록 합니다.

원하는 카나리아 코드를 실행할 수 있도록 카나리아 코드가 저장된 Amazon S3 버킷에서 객체 버전 관리를 사용할 수 있습니다. 그런 다음 이 코드를 카나리아로 실행하도록 지정하면 다음 예제와 같이 객체 `versionId`를 경로의 일부로 포함할 수 있습니다.

```
https://bucket.s3.amazonaws.com/path/object.zip?versionId=version-id
```

```
https://s3.amazonaws.com/bucket/path/object.zip?versionId=version-id
https://bucket.s3-region.amazonaws.com/path/object.zip?versionId=version-id
```

스택 추적 및 예외 메시지

기본적으로 CloudWatch Synthetics 카나리아는 스크립트가 사용자 지정인지 아니면 블루프린트인지에 관계없이 카나리아 스크립트에서 발생하는 모든 예외를 캡처합니다. CloudWatch Synthetices는 예외 메시지와 스택 추적을 모두 세 개의 위치에 기록합니다.

- 테스트 실행을 설명할 때 디버깅 속도를 높이기 위해 CloudWatch Synthetics 서비스에 기록
- Lambda 함수가 생성되는 구성에 따라 CloudWatch Logs에 기록
- 카나리아의 `resultsLocation`에 대해 설정한 값으로 지정된 Amazon S3 위치에 업로드되는 일반 텍스트 파일인 Synthetics 로그 파일에 기록

더 적은 정보를 보내고 저장하려면 CloudWatch Synthetics 래퍼 라이브러리로 돌아가기 전에 예외를 캡처할 수 있습니다.

IAM 역할 범위를 좁게 지정

잠재적인 악성 URL 또는 엔드포인트를 방문하도록 카나리아를 구성하지 않는 것이 좋습니다. 카나리아를 신뢰할 수 없거나 알 수 없는 웹 사이트 또는 엔드포인트로 가리키면 Lambda 함수 코드가 악의적인 사용자의 스크립트에 노출될 수 있습니다. 악성 웹 사이트가 Chromium에서 벗어날 수 있다고 가정하면 인터넷 브라우저를 사용하여 웹 사이트에 연결한 경우와 비슷한 방식으로 Lambda 코드에 액세스할 수 있습니다.

범위가 축소된 권한이 있는 IAM 실행 역할로 Lambda 함수를 실행합니다. 이렇게 하면 악성 스크립트에 의해 Lambda 함수가 손상된 경우 카나리아 AWS 계정으로 실행할 때 수행할 수 있는 작업이 제한됩니다.

CloudWatch 콘솔을 사용하여 카나리아를 생성할 경우 범위가 축소된 IAM 실행 역할로 만들어집니다.

헤더 로깅

기본적으로 카나리아 헤더 로깅은 요청 URL, 응답 코드 및 응답 상태만 기록합니다. 카나리아 스크립트 내에서 더 많거나 적은 정보를 기록하도록 로깅을 사용자 지정할 수 있습니다.

AWS CloudTrail 사용을 통한 Amazon CloudWatch API 호출 로깅

Amazon CloudWatch은 CloudWatch에서 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 API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 CloudWatch에 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 CloudWatch에 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 정보 (p. 387)
- 예제: CloudWatch 로그 파일 항목 (p. 388)

CloudTrail의 CloudWatch 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. 지원되는 이벤트 활동이 CloudWatch에서 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 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

CloudWatch는 CloudTrail 로그 파일의 이벤트로 다음 작업의 로깅을 지원합니다.

- DeleteAlarms
- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DisableAlarmActions
- EnableAlarmActions

- GetDashboard
- ListDashboards
- PutDashboard
- PutMetricAlarm
- SetAlarmState

또한 CloudTrail 로그 파일에 다음과 같은 AWS SDK 지표 작업을 로깅할 수 있습니다. 이러한 작업은 SDK 지표에서만 사용할 수 있으며 해당 코드에서는 사용할 수 없습니다. 자세한 내용은 AWS SDK 지표를 사용하여 애플리케이션 모니터링 (p. 342) 단원을 참조하십시오.

- GetPublishingConfiguration – SDK 지표에서 지표 게시 방식을 지정하는 데 사용됩니다.
- GetPublishingSchema – SDK 지표에서 지표 게시 방식을 지정하는 데 사용됩니다.
- PutPublishingMetrics – SDK 지표에서 CloudWatch 에이전트 지표를 AWS Support로 전송하는 데 사용됩니다.

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.

예제: CloudWatch 로그 파일 항목

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.

다음은 PutMetricAlarm 작업을 다루는 CloudTrail 로그 항목을 보여주는 예입니다.

```
{
    "Records": [{
        "eventVersion": "1.01",
        "userIdentity": {
            "type": "Root",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID"
        },
        "eventTime": "2014-03-23T21:50:34Z",
        "eventSource": "monitoring.amazonaws.com",
        "eventName": "PutMetricAlarm",
        "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": {
            "threshold": 50.0,
            "period": 60,
            "metricName": "CloudTrail Test",
```

```
            "evaluationPeriods": 3,
            "comparisonOperator": "GreaterThanThreshold",
            "namespace": "AWS/CloudWatch",
            "alarmName": "CloudTrail Test Alarm",
            "statistic": "Sum"
        },
        "responseElements": null,
        "requestID": "29184022-b2d5-11e3-a63d-9b463e6d0ff0",
        "eventID": "b096d5b7-dcf2-4399-998b-5a53eca76a27"
    },
    ..additional entries
  ]
  }
```

아래 로그 파일 항목은 사용자가 CloudWatch 이벤트의 PutRule 작업을 호출했음을 보여줍니다.

```
{
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext":{
               "attributes":{
                  "mfaAuthenticated":"false",
                  "creationDate":"2015-11-17T23:56:15Z"
               }
            }
         },
         "eventTime":"2015-11-18T00:11:28Z",
         "eventSource":"events.amazonaws.com",
         "eventName":"PutRule",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"AWS Internal",
         "userAgent":"AWS CloudWatch Console",
         "requestParameters":{
            "description":"",
            "name":"cttest2",
            "state":"ENABLED",
            "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
            "scheduleExpression":""
         },
         "responseElements":{
            "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
         },
         "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
         "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
         "eventType":"AwsApiCall",
         "apiVersion":"2015-10-07",
         "recipientAccountId":"123456789012"
}
```

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"
}
```

Grafana 통합

Grafana 버전 6.5.0 이상을 사용하여 컨텍스트에 따라 CloudWatch 콘솔로 진행하고 와일드카드를 사용하여 동적 지표 목록을 쿼리할 수 있습니다. 이렇게 하면 Amazon Elastic Compute Cloud 인스턴스 또는 컨테이너와 같은 AWS 리소스에 대한 지표를 모니터링하는 데 도움이 됩니다. 새로운 인스턴스가 Auto Scaling 이벤트의 일환으로 생성되면 새로운 인스턴스 ID를 추적하지 않아도 그래프에 자동으로 표시됩니다. 사전 구축된 대시보드는 Amazon EC2, Amazon Elastic Block Store 및 AWS Lambda 리소스 모니터링을 위한 시작 환경을 단순화하는 데 도움이 됩니다.

시작하는 방법에 대한 자세한 내용은 Grafana 실습 설명서의 Using AWS CloudWatch in Grafana를 참조하십시오.

CloudWatch 서비스 할당량

CloudWatch에는 지표, 경보, API 요청, 이메일 알림 및 로그에 대한 다음과 같은 할당량이 있습니다.

리소스	기본 할당량
작업	5/경보. 이 할당량은 변경할 수 없습니다.
개 경보	고객당 월 10회 무료. 계정마다 리전당 5000개 할당량 증가를 요청할 수 있습니다. 지표 수학 표현식 기반 경보에는 지표가 최대 10개까지 허용됩니다.
이상 탐지 모델	계정마다 리전당 500개
API 요청	고객당 월 1,000,000회 무료.
Contributor Insights API 요청	GetInsightRuleReport에는 20개의 초당 트랜잭션(TPS) 할당량이 있습니다. 할당량 증가를 요청할 수 있습니다. 다음 API에는 1개의 초당 트랜잭션 할당량이 있습니다. 이 할당량은 변경할 수 없습니다. • DeleteInsightRules DescribeInsightRules DisableInsightRules EnableInsightRules PutInsightRule
Contributor Insights 규칙	계정당 100개의 규칙 할당량 증가를 요청할 수 있습니다.
사용자 지정 지표	할당량이 없습니다.
대시보드	대시보드 위젯당 최대 100개 지표 전체 위젯에서 대시보드당 최대 500개 지표 이 할당량에는 지표 수학 함수에서 사용하기 위해 검색한 모든 지표가 포함됩니다. 이러한 지표가 그래프에 표시되지 않는 경우에도 그렇습니다. 이러한 할당량은 변경할 수 없습니다.
DescribeAlarms	9건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 할당량 증가를 요청할 수 있습니다.
DeleteAlarms 요청 DescribeAlarmHistory 요청	이러한 각 작업에 대한 3건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 이러한 할당량은 변경할 수 없습니다.

리소스	기본 할당량
DescribeAlarmsForMetric 요청 DisableAlarmActions 요청 EnableAlarmActions 요청 SetAlarmState 요청	
DeleteDashboards 요청 GetDashboard 요청 ListDashboards 요청 PutDashboard 요청	이러한 각 작업에 대한 10건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 이러한 할당량은 변경할 수 없습니다.
PutAnomalyDetector DescribeAnomalyDetectors	10건의 초당 트랜잭션(TPS) 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
DeleteAnomalyDetector	5건의 초당 트랜잭션(TPS) 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.
차원	10/지표. 이 할당량은 변경할 수 없습니다.
GetMetricData	50건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 할당량 증가를 요청할 수 있습니다. API 요청에서 사용되는 `StartTime`이 현재 시간과 같거나 3시간 적을 경우 초당 데이터포인트(DPS)가 180,000입니다. `StartTime`이 현재 시간보다 3시간 많을 경우 DPS는 90,000입니다. 이것은 조절 없이 하나 이상의 API 호출을 사용하여 1초마다 요청할 수 있는 데이터포인트의 최대 수입니다. 이 할당량은 변경할 수 없습니다.
GetMetricData	단일 `GetMetricData` 호출에는 최대 500개의 `MetricDataQuery` 구조가 포함될 수 있습니다. 이 할당량은 변경할 수 없습니다.
GetMetricStatistics	400건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 할당량 증가를 요청할 수 있습니다.
ListMetrics	25건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 할당량 증가를 요청할 수 있습니다.
지표 데이터	15개월. 이 할당량은 변경할 수 없습니다.
MetricDatum 항목	20/PutMetricData 요청. MetricDatum 객체는 단일 값을 포함하거나 여러 값을 나타내는 StatisticSet 개체를 포함할 수 있습니다. 이 할당량은 변경할 수 없습니다.
지표	고객당 월 10회 무료.

리소스	기본 할당량
기간	최대값은 1일(86,400초)입니다. 이 할당량은 변경할 수 없습니다.
PutMetricAlarm 요청	3건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 할당량 증가를 요청할 수 있습니다.
PutMetricData 요청	HTTP POST 요청에 대하여 40KB. PutMetricData는 150건의 초당 트랜잭션(TPS)을 처리할 수 있으며, 이는 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 할당량 증가를 요청할 수 있습니다.
Amazon SNS 이메일 알림	고객당 월 1,000회 무료.

문서 기록

다음 표에는 2018년 6월부터 적용되는 Amazon CloudWatch 사용 설명서의 각 릴리스에서 변경된 중요 사항이 나와 있습니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

update-history-change	update-history-description	update-history-date
CloudWatch Synthetics 공개 미리 보기 (p. 395)	이제 CloudWatch Synthetics는 공개 미리 보기 상태입니다. Synthetics를 사용하면 카나리아를 생성하여 엔드포인트와 API를 모니터링할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 카나리아 사용을 참조하십시오.	November 25, 2019
CloudWatch Contributor Insights 공개 미리 보기 (p. 395)	이제 CloudWatch Contributor Insights는 공개 미리 보기 상태입니다. Contributor Insights를 사용하면 로그 데이터를 분석하고 기고자 데이터를 표시하는 시계열을 생성할 수 있습니다. 상위 N개의 기고자, 총 고유 기고자 수 및 사용량에 대한 지표를 볼 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Contributor Insights를 사용하여 높은 카디널리티 데이터 분석을 참조하십시오.	November 25, 2019
CloudWatch에서 ServiceLens 기능 출시 (p. 395)	ServiceLens는 추적, 지표, 로그 및 경보를 한 곳에 통합할 수 있게 하여 서비스 및 애플리케이션의 관찰 가능성을 향상시킵니다. ServiceLens는 CloudWatch와 AWS X-Ray를 통합하여 애플리케이션에 대한 종합적인 뷰를 제공합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 ServiceRens를 사용하여 애플리케이션의 상태 모니터링을 참조하십시오.	November 21, 2019
CloudWatch를 사용하여 AWS 서비스 할당량을 사전 예방적으로 관리 (p. 395)	CloudWatch를 사용하여 AWS 서비스 할당량을 사전 예방적으로 관리할 수 있습니다. CloudWatch 사용량 지표는 계정의 리소스 사용량 및 API 작업에 대한 가시성을 제공합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 서비스 할당량 통합 및 사용량 지표를 참조하십시오.	November 19, 2019

경보가 상태가 변경되면 CloudWatch에서 이벤트를 보냅니다. (p. 395)	CloudWatch 경보 상태가 변경되면 이제 CloudWatch에서 Amazon EventBridge에 이벤트를 보냅니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 경보 이벤트 및 EventBridge를 참조하십시오.	October 8, 2019
Container Insights (p. 395)	CloudWatch 이제 컨테이너 인사이트를 사용할 수 있습니다. 컨테이너 인사이트는 컨테이너 애플리케이션 및 마이크로서비스의 지표 및 로그를 수집하고 종합하며 요약합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Container Insights 사용을 참조하십시오.	August 30, 2019
Amazon EKS 및 Kubernetes에서 Container Insights 미리 보기 지표 업데이트 (p. 395)	Amazon EKS 및 Kubernetes 공개 미리 보기에서 Container Insights를 업데이트하였습니다. 이제 InstanceId가 클러스터 EC2 인스턴스에 대한 차원으로 포함되어 있습니다. 이로써 이 지표에서 생성된 경보가 중지, 종료, 재부팅 또는 복구와 같은 EC2 작업을 트리거할 수 있게 되었습니다. 뿐만 아니라 이제 Pod 및 서비스 지표를 Kubernetes 네임스페이스에서 보고하여 지표에 대한 네임스페이스의 모니터링 및 경고가 간소화되었습니다.	August 19, 2019
AWS 시스템 관리자 OpsCenter 통합에 대한 업데이트 (p. 395)	CloudWatch Application Insights가 시스템 관리자 OpsCenter와 통합되는 방법에 대한 업데이트	August 7, 2019
CloudWatch 사용량 지표 (p. 395)	CloudWatch 사용량 지표를 통해 CloudWatch 리소스 사용량을 추적하고 서비스 제한 내에서 유지할 수 있습니다. 자세한 내용은 https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Usage-Metrics.html 단원을 참조하십시오.	August 6, 2019
CloudWatch 이상 탐지 공개 미리 보기 (p. 395)	이제 CloudWatch 이상 탐지는 공개 미리 보기 상태입니다. CloudWatch에서는 머신 러닝 알고리즘을 지표의 과거 데이터에 적용하여 지표의 기대값 모델을 생성합니다. 이 모델을 시각화 및 경보 설정에 사용할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 이상 탐지 사용을 참조하십시오.	July 9, 2019

CloudWatch Container Insights 공개 미리 보기 (p. 395)	이제 CloudWatch Container Insights는 공개 미리 보기 상태입니다. 컨테이너 인사이트는 컨테이너 애플리케이션 및 마이크로서비스의 지표 및 로그를 수집하고 종합하며 요약합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Container Insights 사용을 참조하십시오.	July 9, 2019
.NET 및 SQL Server용 CloudWatch Application Insights (p. 395)	.NET 및 SQL Server용 CloudWatch Application Insights에서 .NET 및 SQL Server 애플리케이션의 상태를 확인할 수 있습니다. 애플리케이션 리소스에 대한 최상의 모니터를 설정하고 데이터를 분석하여 애플리케이션 문제의 징후가 있는지 지속적으로 확인할 수 있습니다.	June 21, 2019
CloudWatch 에이전트 단원 재구성됨 (p. 395)	에이전트를 설치하고 구성하는 데 명령줄을 사용하는 고객을 위해 CloudWatch 에이전트 설명서를 더욱 명확하게 재작성했습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집을 참조하십시오.	March 28, 2019
지표 수학 표현식에 추가된 검색 기능 (p. 395)	이제 지표 수학 표현식에서 검색 기능을 사용할 수 있습니다. 이 기능을 통해 검색 쿼리와 일치하는 새 리소스가 생성되는 대로 자동으로 업데이트되는 대시보드를 생성할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 그래프에서 검색 표현식 사용을 참조하십시오.	March 21, 2019
AWS SDK 지표 for Enterprise Support (p. 395)	SDK 지표를 사용하여 AWS 서비스의 상태를 평가하고, 계정 사용 제한에 도달하거나 서비스 중단에 의해 발생하는 지연 시간을 진단할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서에서 AWS SDK 지표를 사용하여 애플리케이션 모니터링을 참조하십시오.	December 11, 2018
수학 표현식에 대한 경보 (p. 395)	CloudWatch에서는 지표 수학 표현식을 기반으로 경보를 생성할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 수학 표현식에 대한 경보를 참조하십시오.	November 20, 2018

새 CloudWatch 콘솔 홈 페이지 (p. 395)	Amazon에서는 CloudWatch 콘솔에 사용 중인 모든 AWS 서비스에 대한 주요 지표 및 경보를 자동으로 표시하는 새로운 홈 페이지를 만들었습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 시작하기를 참조하십시오.	November 19, 2018
CloudWatch 에이전트용 AWS CloudFormation 템플릿 (p. 395)	Amazon에서는 CloudWatch 에이전트 설치 및 업데이트에 사용할 수 있는 AWS CloudFormation 템플릿을 업로드했습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 AWS CloudFormation을 사용하여 새 인스턴스에 CloudWatch 설치를 참조하십시오.	November 9, 2018
CloudWatch 에이전트에서 향상된 기능 (p. 395)	CloudWatch 에이전트가 StatsD 및 수집된 프로토콜과 호환되도록 업데이트되었습니다. 또한 교차 계정에 대한 지원도 개선되었습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서에서 StatsD로 시작하는 사용자 지정 지표 검색, collectd로 사용자 지정 지표 검색 및 지표 및 로그를 다른 AWS 계정으로 전송 단원을 참조하십시오.	September 28, 2018
Amazon VPC 엔드포인트에 대한 지원 (p. 395)	이제 VPC와 CloudWatch 간에 프라이빗 연결을 설정할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 인터페이스 VPC 엔드포인트와 함께 CloudWatch 사용을 참조하십시오.	June 28, 2018

아래 표에서는 2018년 6월 이전의 Amazon CloudWatch 사용 설명서에 적용되는 주요 변경 사항을 설명합니다.

변경 사항	설명	릴리스 날짜
지표 수식	이제 CloudWatch 지표에서 수식을 표현하여, 대시보드에서 그래프에 추가할 수 있는 새로운 시계열을 생성할 수 있습니다. 자세한 내용은 지표 수식 사용 (p. 49) 단원을 참조하십시오.	2018년 4월 4일
"N 중 M" 경보	이제는 경보 평가 간격에서 "N 중 M" 데이터 포인트를 기반으로 경보가 트리거되도록 구성할 수 있습니다. 자세한 정보는 경보 평가 (p. 67) 단원을 참조하십시오.	2017년 12월 8일
CloudWatch 에이전트	새로운 통합 CloudWatch 에이전트가 출시되었습니다. 통합된 다중 플랫폼 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버에서 시스템 지표 및 로그 파일을 모두 사용자 지정하여 수집할 수 있습니다. 새	2017년 9월 7일

변경 사항	설명	릴리스 날짜
	로운 에이전트는 Windows Server 및 Linux를 모두 지원하며 CPU당 코어와 같은 하위 리소스 지표를 포함하여 수집할 지표를 사용자 지정할 수 있습니다. 자세한 내용은 CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집 (p. 189) 단원을 참조하십시오.	
NAT 게이트웨이 지표	Amazon VPC NAT 게이트웨이에 대한 지표가 추가되었습니다.	2017년 9월 7일
고분해능 지표	선택적으로, 이제 사용자 지정 지표를 최대 1초 세분화의 고분해능 지표로 설정할 수 있습니다. 자세한 내용은 고분해능 지표 (p. 47) 단원을 참조하십시오.	2017년 7월 26일
대시보드 API	이제 API 및 AWS CLI를 사용하여 대시보드를 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 CloudWatch 대시보드 생성 (p. 17) 단원을 참조하십시오.	2017년 7월 6일
AWS Direct Connect 지표	AWS Direct Connect에 대한 지표가 추가되었습니다.	2017년 6월 29일
Amazon VPC VPN 지표	Amazon VPC VPN에 대한 지표가 추가되었습니다.	2017년 5월 15일
AppStream 2.0 지표	AppStream 2.0에 대한 지표가 추가되었습니다.	2017년 3월 8일
CloudWatch 콘솔 Color Picker	대시보드 위젯에서 각 지표에 대한 색상을 선택할 수 있습니다. 자세한 내용은 CloudWatch 대시보드에서 그래프를 편집하려면 (p. 22) 단원을 참조하십시오.	2017년 2월 27일
대시보드의 경보	경보를 대시보드에 추가할 수 있습니다. 자세한 내용은 CloudWatch 대시보드에서 경보를 추가 또는 제거 (p. 26) 단원을 참조하십시오.	2017년 2월 15일
Amazon Polly에 대한 지표 추가	Amazon Polly에 대한 지표가 추가되었습니다.	2016년 12월 1일
Amazon Kinesis Data Analytics에 대한 지표 추가	Amazon Kinesis Data Analytics에 대한 지표가 추가되었습니다.	2016년 12월 1일
백분위수 통계에 대한 지원 추가	소수점 두 자리까지 사용하여 백분위 수를 지정할 수 있습니다(예: p95.45). 자세한 내용은 백분위수 (p. 7) 단원을 참조하십시오.	2016년 11월 17일
Amazon Simple Email Service에 대한 지표 추가	Amazon Simple Email Service에 대한 지표가 추가되었습니다.	2016년 11월 2일
지표 보존 기간 업데이트	Amazon CloudWatch는 14일이 아닌 15개월 동안 지표 데이터를 보존하고 있습니다.	2016년 11월 1일
지표 콘솔 인터페이스 업데이트	CloudWatch 콘솔은 기존 기능을 개선하고 새 기능을 추가하는 등 업데이트되었습니다.	2016년 11월 1일

변경 사항	설명	릴리스 날짜
Amazon Elastic Transcoder에 대한 지표 추가	Amazon Elastic Transcoder에 대한 지표가 추가되었습니다.	2016년 9월 20일
Amazon API Gateway에 대한 지표 추가	Amazon API Gateway에 대한 지표가 추가되었습니다.	2016년 9월 9일
AWS Key Management Service에 대한 지표 추가	AWS Key Management Service에 대한 지표가 추가되었습니다.	2016년 9월 9일
Elastic Load Balancing에서 지원되는 새 Application Load Balancer에 대한 지표 추가	Application Load Balancer에 대한 지표가 추가되었습니다.	2016년 8월 11일
Amazon EC2에 대한 새로운 NetworkPacketsIn 및 NetworkPacketsOut 측정치를 추가함	Amazon EC2에 대한 새로운 NetworkPacketsIn 및 NetworkPacketsOut 측정치를 추가함.	2016년 3월 23일
Amazon EC2 스팟 집합에 대한 새 지표 추가	Amazon EC2 스팟 집합에 대한 새 지표가 추가되었습니다.	2016년 3월 21일
새 CloudWatch Logs 지표 추가	새 CloudWatch Logs 지표가 추가되었습니다.	2016년 3월 10일
Amazon Elasticsearch Service 및 AWS WAF 지표와 차원 추가	Amazon Elasticsearch Service 및 AWS WAF 지표와 차원이 추가되었습니다.	2015년 10월 14일
CloudWatch 대시보드에 대한 지원 추가	대시보드는 CloudWatch 콘솔에서 사용자 지정이 가능한 홈 페이지로, 다른 리전에 분산되어 있는 리소스들을 단일 뷰에서 모니터링하는 데 사용할 수 있습니다. 자세한 내용은 Amazon CloudWatch 대시보드 사용 (p. 16) 단원을 참조하십시오.	2015년 10월 8일
AWS Lambda 지표 및 차원 추가	AWS Lambda 지표 및 차원이 추가되었습니다.	2015년 9월 4일
Amazon Elastic Container Service 지표 및 차원 추가	Amazon Elastic Container Service 지표 및 차원이 추가되었습니다.	2015년 8월 17일
Amazon Simple Storage Service 지표 및 차원 추가	Amazon Simple Storage Service 지표 및 차원이 추가되었습니다.	2015년 7월 26일

변경 사항	설명	릴리스 날짜
새로운 기능: 경보 작업 재부팅	재부팅 경보 작업과 경보 작업에 사용할 새로운 IAM 역할이 추가되었습니다. 자세한 내용은 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 (p. 89) 단원을 참조하십시오.	2015년 7월 23일
Amazon WorkSpaces 지표 및 차원 추가	Amazon WorkSpaces 지표 및 차원이 추가되었습니다.	2015년 4월 30일
Amazon Machine Learning 지표 및 차원 추가	Amazon Machine Learning 지표 및 차원이 추가되었습니다.	2015년 4월 9일
새로운 기능: Amazon EC2 인스턴스 복구 경보 작업	새로운 EC2 인스턴스 복구 작업이 포함되도록 경보 작업이 업데이트되었습니다. 자세한 내용은 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 (p. 89) 단원을 참조하십시오.	2015년 3월 12일
Amazon CloudFront 및 Amazon CloudSearch 지표와 차원 추가	Amazon CloudFront 및 Amazon CloudSearch 지표와 차원이 추가되었습니다.	2015년 3월 6일
Amazon Simple Workflow Service 지표 및 차원 추가	Amazon Simple Workflow Service 지표 및 차원이 추가되었습니다.	2014년 5월 9일
AWS CloudTrail에 대한 지원을 추가하도록 설명서 업데이트	Amazon CloudWatch 활동 로깅에 AWS CloudTrail을 사용하는 방법을 설명하기 위해 새 항목이 추가되었습니다. 자세한 내용은 AWS CloudTrail 사용을 통한 Amazon CloudWatch API 호출 로깅 (p. 387) 단원을 참조하십시오.	2014년 4월 30일
새 AWS Command Line Interface(AWS CLI)를 사용하도록 설명서 업데이트	AWS CLI는 간단한 설치, 통합 구성, 일관된 명령줄 구문을 특징으로 하는 교차 서비스 CLI입니다. AWS CLI는 Linux/Unix, Windows 및 Mac에서 지원됩니다. 이 설명서의 CLI 예제는 새 AWS CLI를 사용하도록 업데이트되었습니다. 새 AWS CLI의 설치 및 구성 방법에 대한 자세한 정보는 AWS Command Line Interface 사용 설명서의 AWS 명령줄 인터페이스를 사용한 설정을 참조하십시오.	2014년 2월 21일
Amazon Redshift 및 AWS OpsWorks 지표와 차원 추가	Amazon Redshift 및 AWS OpsWorks 지표와 차원이 추가되었습니다.	2013년 7월 16일
Amazon Route 53 지표 및 차원 추가	Amazon Route 53 지표 및 차원이 추가되었습니다.	2013년 6월 26일
새로운 기능: Amazon CloudWatch 경보 작업	Amazon Elastic Compute Cloud 인스턴스를 중지 또는 종료하는 데 사용할 수 있는 Amazon CloudWatch 경보 작업을 문서화하기 위해 새로운 단원이 추가되었습니다. 자세한 내용은 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 (p. 89) 단원을 참조하십시오.	2013년 1월 8일

변경 사항	설명	릴리스 날짜
EBS 지표가 업데이트됨	프로비저닝 IOPS 볼륨에 대해 새 지표 두 개를 포함하도록 EBS 지표가 업데이트되었습니다.	2012년 11월 20일
새로운 결제 알림	이제 Amazon CloudWatch 지표를 사용하여 AWS 요금을 모니터링하고 지정된 임계값을 초과한 경우 경보를 생성할 수 있습니다. 자세한 내용은 예상 AWS 요금을 모니터링하기 위한 결제 경보 생성 (p. 94) 단원을 참조하십시오.	2012년 5월 10일
새로운 지표	이제 다양한 HTTP 응답 코드에 대한 계산을 제공하는 6가지 새로운 Elastic Load Balancing 지표에 액세스할 수 있습니다.	2011년 10월 19일
새로운 기능	이제 Amazon EMR에서 지표에 액세스할 수 있습니다.	2011년 6월 30일
새로운 기능	이제 Amazon Simple Notification Service 및 Amazon Simple Queue Service에서 지표에 액세스할 수 있습니다.	2011년 7월 14일
새로운 기능	`PutMetricData` API를 사용하여 사용자 지정 지표를 게시하는 방법에 대한 정보가 추가되었습니다. 자세한 내용은 사용자 지정 메트릭 게시 (p. 47) 단원을 참조하십시오.	2011년 5월 10일
지표 보존 기간 업데이트	이제 Amazon CloudWatch에서는 경보 기록을 6주가 아니라 2주 동안만 유지합니다. 따라서 경보의 유지 기간이 지표 데이터의 유지 기간과 일치하게 되었습니다.	2011년 4월 7일
새로운 기능	지표가 임계값을 초과하면 Amazon Simple Notification Service 또는 Auto Scaling 알림을 보내는 기능이 추가되었습니다. 자세한 내용은 경보 (p. 7) 단원을 참조하십시오.	2010년 12월 2일
새로운 기능	이제 여러 CloudWatch 작업에 결과 페이지 표시를 제어할 수 있는 MaxRecords 및 NextToken 파라미터가 포함됩니다.	2010년 12월 2일
새로운 기능	이 서비스는 이제 AWS Identity and Access Management(IAM)와 통합됩니다.	2010년 12월 2일