

---

# Amazon CloudWatch

## 사용 설명서



## Amazon CloudWatch: 사용 설명서

Copyright © 2019 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
개 경고	7
리소스	8
설정	9
Amazon Web Services(AWS)에 가입	9
Amazon CloudWatch 콘솔에 로그인	9
AWS CLI 설정	9
시작하기	11
모든 AWS 서비스의 주요 지표 살펴보기	13
교차 서비스 대시보드에 나타나지 않도록 서비스 제거	14
단일 서비스에 집중	14
리소스 그룹에 집중	15
대시보드 사용	17
대시보드 생성	17
그래프 추가 또는 제거	18
그래프 이동 또는 크기 조정	19
그래프 편집	20
CloudWatch 대시보드에서 수동으로 지표 그래프 생성	21
그래프 이름 변경	22
텍스트 위젯 추가 또는 제거	23
경보 추가 또는 제거	23
여러 리전의 리소스 모니터링	24
그래프 링크 및 링크 해제	24
즐거찾기 목록에 대시보드 추가	25
기간 재정의 설정 또는 새로 고침 간격 변경	25
시간 범위 또는 시간대 형식 변경	26
지표 사용	27
사용 가능한 지표 보기	27
사용 가능한 지표 검색	29
지표에 대한 통계 얻기	30
특정 리소스에 대한 통계 얻기	30
리소스에서 통계 집계하기	33
Auto Scaling 그룹별 통계 집계	35
AMI의 집계 통계	36
지표 그래프	37
지표 그래프 작성	37
그래프의 시간 범위 또는 시간대 형식 수정	39
그래프의 Y축 수정	40
그래프의 지표에서 경고 생성	41
사용자 지정 지표 게시	42
고분해능 지표	42
차원 사용	42
단일 데이터 요소 게시	43
통계 세트 게시	44
0 값 게시	44

지표 수식 사용 .....	44
CloudWatch 그래프에 수식 표현식 추가 .....	45
지표 수식 구문 및 함수 .....	45
GetMetricData API 작업과 함께 지표 수식 사용 .....	48
경보 사용 .....	49
경보 상태 .....	49
경보 평가 .....	49
경보가 누락 데이터를 처리하는 방법 구성 .....	50
데이터가 누락되었을 때 경보 상태 평가 방법 .....	51
고분해능 경보 .....	52
수식 표현식에 대한 경보 .....	52
백분위수 기반 경보 및 데이터 샘플 부족 .....	53
CloudWatch 경보의 일반적인 기능 .....	53
SNS 주제 설정 .....	53
AWS Management 콘솔을 사용하여 Amazon SNS 주제 설정 .....	54
AWS CLI를 사용하여 SNS 주제 설정 .....	54
단일 지표를 기반으로 경보 생성 .....	56
지표 수식 표현식을 기반으로 경보 생성 .....	57
CloudWatch 경보 편집 .....	58
CPU 사용률 경보 생성 .....	58
AWS Management 콘솔을 사용하여 CPU 사용률 경보를 설정 .....	59
AWS CLI를 사용하여 CPU 사용률 경보를 설정 .....	60
로드 밸런서 지연 경보 생성 .....	61
AWS Management 콘솔을 사용하여 지연 시간 경보 설정 .....	61
AWS CLI를 사용하여 지연 시간 경보 설정 .....	61
스토리지 처리량 경보 생성 .....	62
AWS Management 콘솔을 사용하여 스토리지 처리량 경보 설정 .....	62
AWS CLI를 사용하여 스토리지 처리량 경보 설정 .....	63
인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 .....	63
Amazon CloudWatch 경보에 중지 작업 추가 .....	65
Amazon CloudWatch 경보에 종료 작업 추가 .....	65
Amazon CloudWatch 경보에 재부팅 작업 추가 .....	66
Amazon CloudWatch 경보에 복구 작업 추가 .....	67
트리거된 경보 및 작업 기록 보기 .....	68
결제 경보 만들기 .....	69
결제 경보 활성화 .....	69
결제 경보 만들기 .....	70
경보 상태 확인 .....	71
결제 경보 삭제 .....	71
Amazon EC2 Auto Scaling 경보 숨기기 .....	71
CloudWatch 에이전트로 지표 및 로그 수집 .....	72
CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 .....	73
Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 .....	74
온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성 .....	75
Amazon EC2 인스턴스에 CloudWatch 에이전트 설치 .....	76
시작하기: 첫 번째 인스턴스에 CloudWatch 에이전트 설치 .....	76
에이전트 구성을 사용하여 추가 인스턴스에 CloudWatch 에이전트 설치 .....	83
온프레미스 서버에 CloudWatch 에이전트 설치 .....	89
시작하기: 첫 번째 서버에 CloudWatch 에이전트 설치 .....	89
에이전트 구성을 사용하여 추가 서버에 CloudWatch 에이전트 설치 .....	97
AWS CloudFormation을 사용하여 새 인스턴스에 CloudWatch 에이전트 설치 .....	102
자습서: AWS CloudFormation 인라인 템플릿을 사용하여 설치 .....	103
자습서: AWS CloudFormation 및 Parameter Store를 사용하여 CloudWatch 에이전트 설치 .....	105
AWS CloudFormation에서 CloudWatch 에이전트를 사용하여 문제 해결 .....	106
CloudWatch 에이전트 구성 파일 생성 .....	107
마법사로 CloudWatch 에이전트 구성 파일 만들기 .....	107
CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 .....	111

procstat 플러그인을 사용하여 프로세스 지표 수집 .....	128
procstat를 사용하도록 CloudWatch 에이전트 구성 .....	128
procstat로 수집한 지표 .....	130
StatsD로 시작하는 사용자 지정 지표 검색 .....	136
collectd로 사용자 지정 지표 검색 .....	137
CloudWatch 에이전트를 사용하는 일반적인 시나리오 .....	138
CloudWatch 에이전트에 의해 수집되는 지표에 사용자 지정 차원 추가 .....	139
여러 에이전트 구성 파일 .....	139
CloudWatch 에이전트에 의해 수집되는 지표 집계 또는 롤업 .....	141
CloudWatch 에이전트로 고분해능 지표 수집 .....	141
다른 AWS 계정에 지표 및 로그 전송 .....	142
CloudWatch 에이전트가 수집하는 지표 .....	144
Windows 서버 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 .....	144
Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 .....	144
CloudWatch 에이전트의 문제 해결 .....	151
CloudWatch 에이전트 명령줄 파라미터 .....	151
Run Command를 사용한 CloudWatch 에이전트 설치 실패 .....	151
CloudWatch 에이전트가 시작되지 않음 .....	152
CloudWatch 에이전트가 실행 중인지 확인 .....	152
지표가 저장되는 위치 .....	153
에이전트가 시작되지 않고 오류에 Amazon EC2 리전이 언급되어 있음 .....	153
Windows Server에서 자격 증명을 찾을 수 없음 .....	153
CloudWatch 에이전트 파일 및 위치 .....	153
CloudWatch 에이전트에 의해 생성되는 로그 .....	154
CloudWatch 에이전트 중지 및 다시 시작 .....	154
지표를 게시하는 서비스 .....	156
AWS SDK Metrics를 사용하여 애플리케이션 모니터링 .....	160
SDK Metrics for Enterprise Support에서 수집한 지표와 데이터 .....	160
SDK Metrics에 대해 CloudWatch 에이전트 구성 .....	162
AWS 시스템 관리자를 사용하여 구성 .....	163
수동 구성 .....	163
SDK Metrics에 대한 IAM 권한 설정 .....	164
CloudWatch 자습서 .....	166
시나리오: 예상 요금 모니터링 .....	166
1단계: 결제 경보 활성화 .....	166
2단계: 결제 경보 만들기 .....	167
3단계: 경보 상태 확인 .....	168
4단계: 결제 경보 편집 .....	168
5단계: 결제 경보 삭제 .....	168
시나리오: 지표 게시 .....	168
1단계: 데이터 구성 정의 .....	169
2단계: CloudWatch에 지표 추가 .....	169
3단계: CloudWatch에서 통계 얻기 .....	170
4단계: 콘솔을 사용하여 그래프 보기 .....	170
인터페이스 VPC 엔드포인트와 함께 CloudWatch 사용 .....	171
가용성 .....	171
CloudWatch에 대한 VPC 엔드포인트 생성 .....	171
인증 및 액세스 제어 .....	173
인증 .....	173
액세스 제어 .....	174
CloudWatch 대시보드 권한 업데이트 .....	174
액세스 관리 개요 .....	175
리소스 및 작업 .....	175
리소스 소유권 이해 .....	176
리소스 액세스 관리 .....	176
정책 요소 지정: 작업, 효과, 보안 주체 .....	177
정책에서 조건 지정 .....	178

자격 증명 기반 정책(IAM 정책) 사용 .....	178
CloudWatch 콘솔 사용에 필요한 권한 .....	179
CloudWatch에 대한 AWS 관리형(미리 정의된) 정책 .....	181
고객 관리형 정책 예 .....	181
서비스 연결 역할 사용 .....	183
CloudWatch 경보에 대한 서비스 연결 역할 권한 .....	183
CloudWatch 경보에 대한 서비스 연결 역할 생성 .....	184
CloudWatch 경보에 대한 서비스 연결 역할 편집 .....	184
CloudWatch 경보에 대한 서비스 연결 역할 삭제 .....	185
Amazon CloudWatch 권한 참조 문서 .....	186
API 호출 로깅 .....	193
CloudTrail의 CloudWatch 정보 .....	193
예제: CloudWatch 로그 파일 항목 .....	194
서비스 제한 .....	197
문서 기록 .....	199

# Amazon CloudWatch란 무엇입니까?

Amazon CloudWatch는 Amazon Web Services(AWS) 리소스와 AWS에서 실시간으로 실행 중인 애플리케이션을 모니터링합니다. 를 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 증가된로드를 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 CloudWatch 콘솔 - <https://console.aws.amazon.com/cloudwatch/> —
- AWS CLI - 자세한 정보는 AWS Command Line Interface 사용 설명서의 [AWS 명령줄 인터페이스로 설정](#)을 참조하십시오.
- CloudWatch API - 자세한 정보는 [Amazon CloudWatch API 참조](#)를 참조하십시오. —
- AWS SDKs - 자세한 정보는 [Amazon Web Services용 도구](#)를 참조하십시오. —

## 관련 AWS 서비스

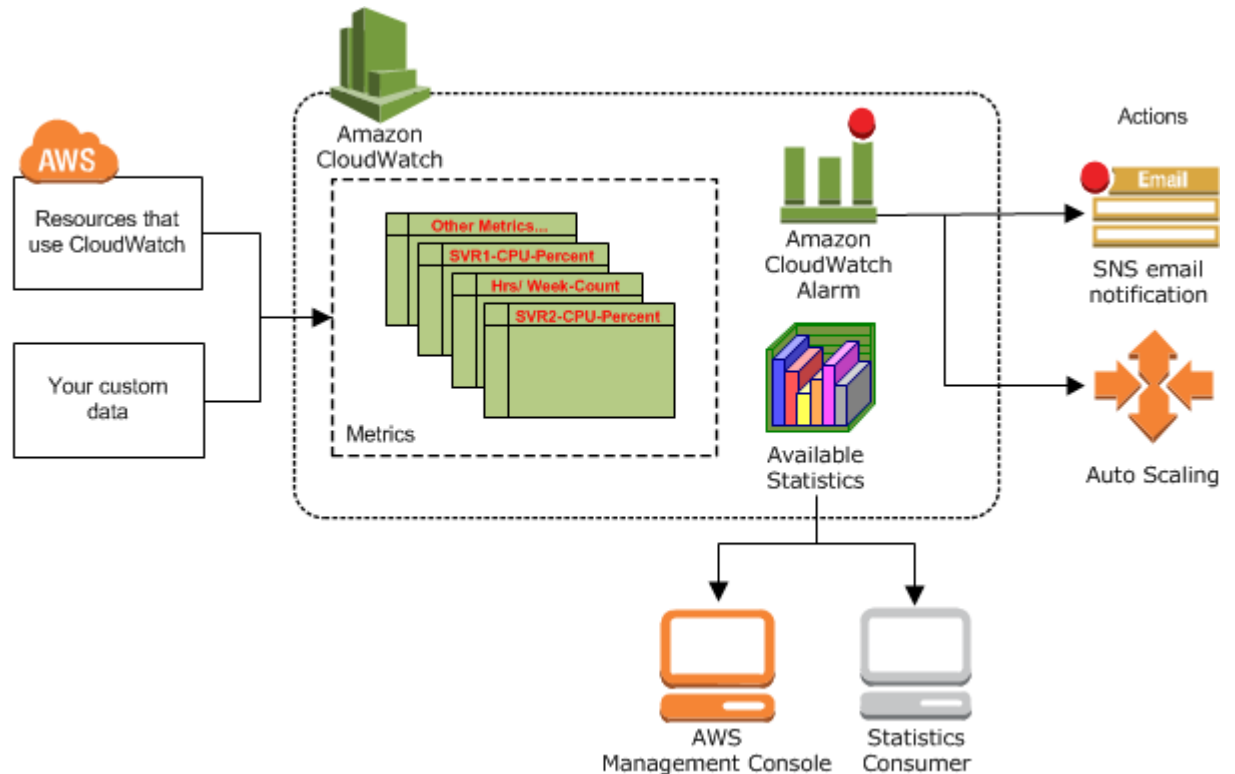
다음 서비스가 Amazon CloudWatch와 함께 사용됩니다.

- 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 정보는 [Amazon SNS 알림 설정 \(p. 53\)](#) 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 정보는 Amazon EC2 Auto Scaling 사용 설명서의 [동적 조정](#)을 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 정보는 [AWS CloudTrail 사용을 통한 Amazon CloudWatch API 호출 로깅 \(p. 193\)](#) 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 정보는 [Amazon CloudWatch에 대한 인증 및 액세스 제어 \(p. 173\)](#)을 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 정보는 [CloudWatch 지표를 게시하는 AWS 서비스 \(p. 156\)](#) 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 정보는 [개경보 \(p. 7\)](#) 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. Amazon CloudWatch에서는 리전 간 데이터는 집계하지 않습니다. 따라서 지표는 리전 간에 완전히 개별적입니다. 자세한 정보는 Amazon Web Services 일반 참조의 [리전 및 엔드포인트](#)를 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 \(p. 7\)](#)
- [개 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 `aws/service`라는 명명 규칙을 사용합니다. 예를 들어 Amazon EC2는 `aws/ec2` 네임스페이스를 사용합니다. AWS 네임스페이스 목록에 대해서는 [CloudWatch 지표를 게시하는 AWS 서비스 \(p. 156\)](#) 단원을 참조하십시오.

## 지표

지표는 CloudWatch의 기본 개념입니다. 지표는 CloudWatch에 게시된 시간 순서별 데이터 요소 세트를 나타냅니다. 지표를 모니터링할 변수로 생각하면 데이터 요소는 시간에 따른 변수의 값을 나타냅니다. 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. 각 데이터 요소에는 타임스탬프와 측정 단위(선택 사항)가 있습니다. 통계를 요청하면 반환된 데이터 스트림은 네임스페이스, 지표 이름, 차원 및 단위(선택 사항)로 식별됩니다.

자세한 정보는 [사용 가능한 지표 보기 \(p. 27\)](#) 및 [사용자 지정 지표 게시 \(p. 42\)](#)을 참조하십시오.

## 타임스탬프

각 지표 데이터 요소에는 타임스탬프가 표시되어 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 정보는 [dateTime](#)을 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 표시되거나 경보가 지연되는 결과로 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 지표를 고유하게 식별하는 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/SampleCount의 값입니다. 이 통계를 Minimum 및 Maximum과 비교하면 지표의 전체 범위와 평균 사용량이 Minimum 및 Maximum에 얼마나 근접했는지 확인할 수 있습니다. 이와 같은 비교를 통해 필요에 따라 리소스를 늘리거나 줄여야 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 정보는 <a href="#">백분위수 (p. 7)</a> 단원을 참조하십시오.

사전 산출된 통계값을 추가할 수 있습니다. 데이터 요소 값 대신 SampleCount, Minimum, Maximum 및 Sum에 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 Percent가 있습니다. CloudWatch에서 지원하는 전체 단위 목록은 Amazon CloudWatch API Reference의 [MetricDatum 데이터 형식](#)을 참조하십시오.

사용자 지정 지표를 만들 때 단위를 지정할 수도 있습니다. 단위를 지정하지 않을 경우 CloudWatch는 None을 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 정보는 [고분해능 지표 \(p. 42\)](#) 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 타임 스탬프를 사용하여 데이터 요소를 원하는 만큼 게시할 수 있습니다. CloudWatch는 기간에 따라 통계를 집계합니다. 집계된 통계는 세부 모니터링을 사용하는 경우에만 사용 가능합니다. 또한 Amazon CloudWatch에서는 리전 간 데이터는 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. 다음 서비스에서 백분위수를 사용할 수 있습니다.

- Amazon EC2
- Amazon RDS
- Kinesis
- Application Load Balancer
- Elastic Load Balancing
- API 게이트웨이

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.

다른 CloudWatch 통계값(평균, 최소값, 최대값, 합계)도 사용되므로 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 대한 원시의 요약되지 않은 데이터 포인트를 게시하는 한 사용자 지정 측정치와 AWS 서비스의 측정치에 대한 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 하나를 충족할 경우에만 이 데이터에 대한 백분위수 통계를 검색할 수 있습니다.

- 통계 세트의 SampleCount는 1.
- 통계 세트의 최소값과 최대값이 동일.

## 개 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 정보는 [사용자 지정 지표 게시 \(p. 42\)](#) 단원을 참조하십시오.

자세한 정보는 [Amazon CloudWatch 경보 사용 \(p. 49\)](#) 및 [그래프의 지표에서 경보 생성 \(p. 41\)](#) 단원을 참조하십시오.

## Amazon CloudWatch 리소스

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

리소스	설명
<a href="#">Amazon CloudWatch FAQ</a>	FAQ는 이 제품에 대해 개발자들이 가장 많이 질문한 상위 개 내용을 소개합니다.
<a href="#">릴리스 정보</a>	릴리스 정보에는 현재 릴리스에 대한 고급 수준의 개요를 제공합니다. 특히 새로운 기능, 상관 관계 및 알려진 문제점에 대해 다룹니다.
<a href="#">AWS Developer Resource Center</a>	설명서, 코드 예제, 릴리스 정보 및 AWS를 사용하여 혁신적인 응용 프로그램을 구현하는 데 도움이 되는 기타 정보를 이 한 곳에서 찾을 수 있습니다.
<a href="#">AWS Management Console</a>	이 콘솔에서는 Amazon CloudWatch 및 다양한 다른 AWS 제품 기능의 대부분을 프로그래밍 작업 없이 수행할 수 있습니다.
<a href="#">Amazon CloudWatch Discussion Forums</a>	Amazon CloudWatch와 관련된 기술적 질문에 대해 토론할 수 있는 개발자를 위한 커뮤니티 기반 포럼입니다.
<a href="#">AWS Support</a>	AWS 지원 사례를 생성 및 관리하는 곳. 또한 포럼, 기술 FAQ, 서비스 상태 및 AWS Trusted Advisor 등의 기타 유용한 자료에 대한 링크가 있습니다.
<a href="#">Amazon CloudWatch 제품 정보</a>	Amazon CloudWatch에 대한 정보를 제공하는 기본 웹 페이지입니다.
<a href="#">문의처</a>	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://aws.amazon.com/>을 열고 Create an AWS Account(AWS 계정 생성)를 선택합니다.

### Note

전에 AWS 계정 루트 사용자 자격 증명을 사용하여 AWS Management 콘솔에 로그인한 적이 있는 경우 Sign in to a different account(다른 계정으로 로그인)를 선택합니다. 전에 IAM 자격 증명을 사용하여 콘솔에 로그인한 적이 있는 경우 Sign-in using root account credentials(루트 계정 자격 증명으로 로그인)를 선택합니다. 그런 다음 Create a new AWS account(새 AWS 계정 생성)를 선택합니다.

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] 섹션에 [Create Alarm] 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 설정](#)을 참조하십시오.

Amazon CloudWatch CLI의 설치 및 구성 방법에 대한 자세한 내용은 Amazon CloudWatch CLI Reference의 [명령줄 인터페이스 설정](#)을 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

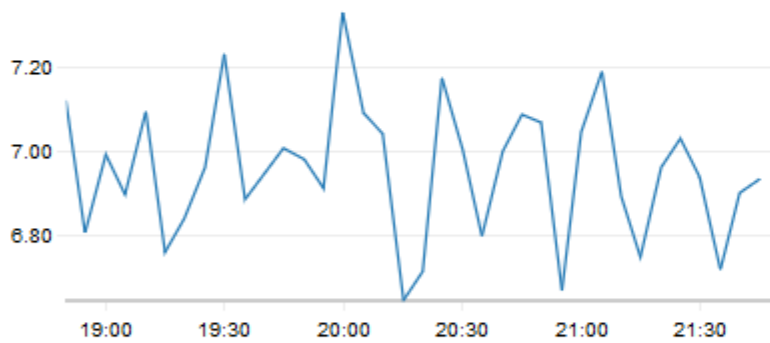
-

🕒 CloudWatch Events

Default dashboard ⓘ [Edit dashboard](#)

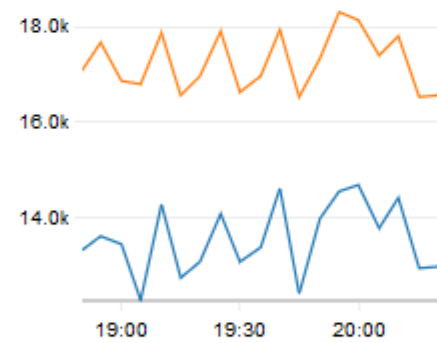
## Custom metric 1

Percent



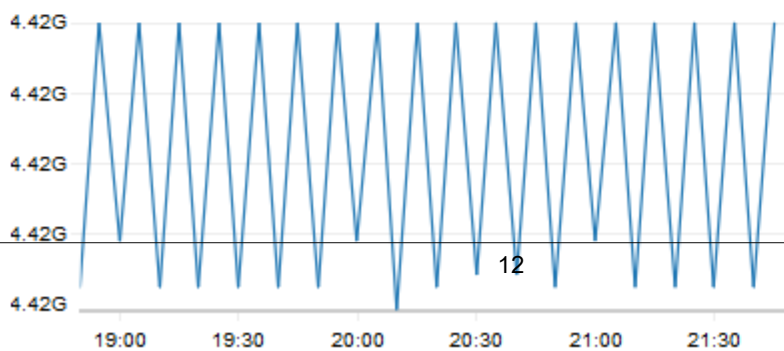
## Custom metric 2

Bytes



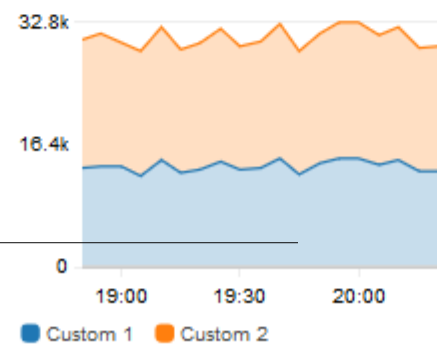
## Custom metrics 5

Bytes



## Custom metrics 2

Bytes



개요에는 다음 항목이 표시되며, 이러한 항목은 자동으로 새로고침됩니다.

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

이러한 상단 영역에서는 모든 서비스의 경고 상태와 최근에 상태가 변경된 경고를 보고 AWS 서비스의 상태를 평가할 수 있습니다. 따라서 모니터링해 문제를 빠르게 진단할 수 있습니다.

- 이러한 영역 아래에는 사용자가 생성해 CloudWatch-Default라고 이름을 지정한 사용자 지정 대시보드(있는 경우)가 있습니다. 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 정보는 [AWS 리소스 그룹이란 무엇입니까?](#)를 참조하십시오.

#### 항목

- [모든 AWS 서비스의 주요 지표 살펴보기](#) (p. 13)
- [단일 AWS 서비스의 지표 및 경고에 집중](#) (p. 14)
- [리소스 그룹의 지표 및 경고에 집중](#) (p. 15)

## 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 경고 수와 ALARM 상태인 경고가 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 경고 기록을 볼 수 있습니다. 이렇게 하려면 경고 그래프 위에 마우스를 올려 놓고 작업 아이콘, 경보에서 보기를 선택합니다.

새 브라우저 탭에 경고 보기가 나타나는데, 여기에는 선택한 경고에 대한 세부 정보와 함께 경고 목록이 표시됩니다. 경고 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 정보는 [리소스 그룹의 지표 및 경보에 집중 \(p. 15\)](#) 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 경고 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 경고 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 정보를 보도록 전환하려면 현재 서비스 대시보드가 표시된 화면 상단에서 경고 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 경고 위에 마우스를 놓고 작업 아이콘 경고에서 보기를 선택합니다.  
새 탭에 경고 보기가 나타나는데, 여기에는 선택한 경고에 대한 세부 정보와 함께 경고 목록이 표시됩니다. 경고 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 경고 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.  
경보는 항상 1분마다 한 번씩 새로 고침됩니다.
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 정보는 [AWS 리소스 그룹이란 무엇입니까?](#)를 참조하십시오.

리소스 그룹을 집중적으로 살펴볼 때 해당 리소스 그룹의 일부로 리소스에 태그를 지정한 서비스만 표시하도록 바뀝니다. 최신 경고 영역에는 해당 리소스 그룹과 관련된 경고만 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 경고 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.  
  
경보는 항상 1분마다 한 번씩 새로 고침됩니다.
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.
- 작동 지침서를 생성하여 운영 이벤트 동안 팀원들에게 특정 사고에 대한 대응 방법에 관해 지침을 제공할 수 있습니다.
- 중요한 리소스 및 애플리케이션 측정값에 대한 공통 뷰를 생성하여 운영 이벤트 동안 신속하고 원활한 의사 소통을 위해 팀원들이 이를 공유하도록 할 수 있습니다.

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

## 목차

- [CloudWatch 대시보드 생성 \(p. 17\)](#)
- [CloudWatch 대시보드에서 그래프를 추가 또는 제거 \(p. 18\)](#)
- [CloudWatch 대시보드에서 그래프를 이동시키거나 크기를 조정합니다. \(p. 19\)](#)
- [CloudWatch 대시보드에서 그래프를 편집하려면 \(p. 20\)](#)
- [CloudWatch 대시보드에서 수동으로 지표 그래프 생성 \(p. 21\)](#)
- [CloudWatch 대시보드에서 그래프 이름 변경 \(p. 22\)](#)
- [CloudWatch 대시보드에서 텍스트 위젯을 추가 또는 제거 \(p. 23\)](#)
- [CloudWatch 대시보드에서 경보를 추가 또는 제거 \(p. 23\)](#)
- [단일 CloudWatch 대시보드를 사용하여 여러 리전의 리소스를 모니터링 \(p. 24\)](#)
- [CloudWatch 대시보드에서 그래프 링크 또는 링크 해제 \(p. 24\)](#)
- [즐거찾기 목록에 대시보드 추가 \(p. 25\)](#)
- [기간 재정의 설정 또는 CloudWatch 대시보드에 대한 새로 고침 간격 변경 \(p. 25\)](#)
- [CloudWatch 대시보드의 시간 범위 또는 시간대 형식 변경 \(p. 26\)](#)

## 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 리전별이 아니라 글로벌 기능입니다.

이 단원에서는 콘솔을 사용하여 대시보드를 생성합니다. PutDashboard API를 사용하여 대시보드를 생성할 수도 있습니다. 그러면 JSON 문자열을 사용하여 대시보드 내용을 정의합니다. PutDashboard를 사용하여 대시보드를 생성하고 기존 대시보드를 기반으로 이 대시보드를 구성하려면 작업, 소스 보기/편집을 선택하여 새 대시보드에 사용할 현재 대시보드의 JSON 문자열을 표시하여 복사합니다.

API를 사용한 대시보드 생성에 대한 자세한 정보는 Amazon CloudWatch API Reference의 [PutDashboard](#)를 참조하십시오.

콘솔을 사용하여 대시보드를 생성하려면

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

2. 탐색 창에서 [Dashboards], [Create dashboard]를 선택합니다.
3. [Create new dashboard] 대화 상자에서 대시보드 이름을 입력하고 [Create dashboard]를 선택합니다.  
  
CloudWatch-Default라는 이름을 사용하는 경우 대시보드가 CloudWatch 홈페이지의 개요에 나타납니다. 자세한 정보는 [Amazon CloudWatch 시작하기 \(p. 11\)](#) 단원을 참조하십시오.  
  
리소스 그룹을 사용하고 대시보드의 이름을 CloudWatch-Default-ResourceGroupName으로 지정하면 해당 리소스 그룹에 마우스를 가져가면 CloudWatch 홈페이지에 대시보드가 나타납니다.
4. [Add to this dashboard] 대화 상자에서 다음 중 하나를 수행합니다.
  - 대시보드에 그래프를 추가하려면 행 또는 누적 면적을 선택하고 구성을 선택합니다. [Add metric graph] 대화 상자에서 그래프 처리할 지표를 선택하고 [Create widget]를 선택합니다. 특정 지표가 14일 이상 데이터를 게시하지 않아 대화 상자에 나타나지 않는 경우 수동으로 추가할 수 있습니다. 자세한 정보는 [CloudWatch 대시보드에서 수동으로 지표 그래프 생성 \(p. 21\)](#) 단원을 참조하십시오.
  - 대시보드에 지표를 표시하는 숫자를 추가하려면 [Number], [Configure]를 선택합니다. [Add metric graph] 대화 상자에서 그래프 처리할 지표를 선택하고 [Create widget]를 선택합니다.
  - 대시보드에 텍스트 블록을 추가하려면 [Text], [Configure]를 선택합니다. 새 텍스트 위젯 대화 상자의 마크다운에서 [마크다운](#)을 사용하여 텍스트를 추가하고 포맷합니다. [Create widget]을 선택합니다.
5. 선택적으로, [Add widget]을 선택하고 4단계를 반복하여 대시보드에 다른 위젯을 추가합니다. 이 단계를 여러 번 반복할 수 있습니다.
6. [Save dashboard]를 선택합니다.

## 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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. [Add widget]을 선택합니다.
4. 행 또는 누적 면적을 선택하고 구성을 선택합니다.
5. [All metrics] 탭에서 그래프로 표시할 지표를 선택합니다. 특정 지표가 14일 이상 데이터를 게시하지 않아 대화 상자에 나타나지 않는 경우 수동으로 추가할 수 있습니다. 자세한 정보는 [CloudWatch 대시보드에서 수동으로 지표 그래프 생성 \(p. 21\)](#) 단원을 참조하십시오.
6. (선택 사항) 그래프 처리할 지표를 선택했으면 그래프에서 색상을 변경할 수 있습니다. 이렇게 하려면 [Graphed metrics]를 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.
7. (선택 사항) 그래프 처리하려는 지표에 대한 자세한 정보를 보려면 범례에 마우스 포인터를 둡니다.
8. (선택 사항) 위젯 유형을 변경하려면 그래프 제목 영역에 마우스 포인터를 놓고 [Widget actions]와 [Widget type]을 선택합니다.
9. (선택 사항) 지표에 사용된 통계를 변경하려면 그래프로 표시된 지표, 통계를 선택한 후 사용하기 원하는 통계를 선택합니다. 자세한 정보는 [통계 \(p. 5\)](#) 단원을 참조하십시오.
10. (선택 사항) 그래프에 표시된 시간 범위를 변경하려면 그래프 맨 위의 사용자 지정이나 사용자 지정 왼쪽의 기간 중 하나를 선택합니다.
11. (선택 사항) 가로 주석을 사용하면 대시보드 사용자가 지표가 특정 수준까지 급상승하는지, 또는 지표가 미리 정의한 범위를 유지하는지 여부를 빠르게 확인할 수 있습니다. 가로 주석을 추가하려면 [Graph options], [Add horizontal annotation]을 선택합니다.



- a. [Label]에 주석의 레이블을 입력합니다.
- b. [Value]에 가로 주석이 표시될 지표 값을 입력합니다.
- c. [Fill]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. [Axis]에서 그래프에 여러 지표가 포함된 경우 Value 값이 왼쪽 Y축 또는 오른쪽 Y축과 연결된 지표를 참조할지 지정합니다.

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

동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

주석을 숨기려면 해당 주석의 왼쪽 옆에서 확인란을 선택 취소합니다.

주석을 삭제하려면 [Actions] 옆에서 [x]를 선택합니다.

12. (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. [Label]에 주석의 레이블을 입력합니다. 주석에 날짜와 시간만 표시하려면 레이블 필드를 비워둡니다.
  - b. 날짜에서, 세로 주석이 표시되는 날짜와 시간을 지정합니다.
  - c. Fill에서, 채우기를 세로 주석 앞에 사용할지 뒤에 사용할지, 또는 2개의 세로 주석 사이에 사용할지 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 옆에서 확인란을 선택 취소합니다.

주석을 삭제하려면 [Actions] 옆에서 [x]를 선택합니다.

13. [Create widget]을 선택합니다.
14. [Save dashboard]를 선택합니다.

대시보드에서 그래프를 제거하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 그래프 제목에 마우스 포인터를 놓고 [Widget actions]와 [Delete]를 선택합니다.
4. [Save dashboard]를 선택합니다. 변경 사항을 저장하기 전에 대시보드에서 다른 곳으로 이동하려고 시도하면 변경 사항을 저장하거나 삭제하라는 메시지가 나타납니다.

## CloudWatch 대시보드에서 그래프를 이동시키거나 크기를 조정합니다.

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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 선택 아이콘이 나타날 때까지 그래프 제목에 마우스 포인터를 둡니다. 원하는 그래프를 선택하고 대시보드의 새로운 위치로 드래그합니다.
4. [Save dashboard]를 선택합니다.

#### 그래프 크기를 조정하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 크기를 늘리거나 줄이려면 그래프에 마우스 포인터를 두고 그래프의 오른쪽 하단 모서리를 드래그합니다.
4. [Save dashboard]를 선택합니다.

#### 그래프를 일시적으로 확대하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 그래프를 선택합니다. 또는 그래프 제목에 마우스 포인터를 놓고 [Widget actions]와 [Enlarge]를 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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 그래프 제목에 마우스 포인터를 놓고 [Widget actions], [Edit]를 선택합니다.
4. 그래프의 제목을 바꾸려면 해당 제목을 선택하고 새로운 제목을 입력한 다음, ENTER를 누릅니다.
5. 그래프에 표시된 시간 범위를 변경하려면 그래프 맨 위의 사용자 지정이나 사용자 지정 왼쪽의 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 [Graphed metrics] 탭에서 제목, 색상, 통계 또는 기간을 변경할 수 있습니다.
  - a. 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.
  - b. 통계를 변경하려면 창 하단에서 [Statistic]을 선택하고 원하는 새 통계를 선택합니다. 자세한 정보는 [통계 \(p. 5\)](#) 단원을 참조하십시오.
  - c. 창 하단의 통계 옆에 있는 기간을 바꾸려면 기간을 선택하고 다른 값을 선택합니다. 대시보드 자체에 대한 기간이 Auto로 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 [Graph options]를 선택합니다.
  - a. 가로 주석을 추가하려면 [Add horizontal annotation]을 선택합니다.
  - b. [Label]에 주석의 레이블을 입력합니다.

- c. [Value]에 가로 주석이 표시될 지표 값을 입력합니다.
- d. [Fill]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 옆에서 색상 정사각형을 선택하여 주석의 채우기 색상을 변경할 수 있습니다.
- e. [Axis]에서 그래프에 여러 지표가 포함된 경우 Value 값이 왼쪽 Y축 또는 오른쪽 Y축과 연결된 지표를 참조할지 지정합니다.

동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

주석을 숨기려면 해당 주석의 왼쪽 옆에서 확인란을 선택 취소합니다.

주석을 삭제하려면 [Actions] 옆에서 [x]를 클릭합니다.

- 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. [Label]에 주석의 레이블을 입력합니다. 주석에 날짜와 시간만 표시하려면 레이블 필드를 비워둡니다.
  - c. 날짜에서, 세로 주석이 표시되는 날짜와 시간을 지정합니다.
  - d. Fill에서, 채우기를 세로 주석 앞에 사용할지 뒤에 사용할지, 또는 2개의 세로 주석 사이에 사용할지 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 옆에서 확인란을 선택 취소합니다.

주석을 삭제하려면 [Actions] 옆에서 [x]를 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 선택하십시오. 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 [Update widget]을 선택합니다.

대시보드의 그래프에서 지표를 일시적으로 숨기려면

- 1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
- 2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
- 3. 그래프의 바닥글에서 범례의 색상 정사각형에 마우스 포인터를 가져갑니다. 정사각형이 X로 바뀌면 클릭합니다.
- 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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 대시보드에 지표를 추가하고자 하는 그래프가 이미 있어야 합니다. 아직 없는 경우 그래프를 생성하고 지표에 추가합니다. 자세한 정보는 [CloudWatch 대시보드에서 그래프를 추가 또는 제거 \(p. 18\)](#) 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 대괄호 뒤에 심표를 추가한 후, 심표 뒤에 비슷한 대괄호로 묶은 섹션을 추가합니다. 이 새 섹션에서 네임스페이스, 지표 이름 및 그래프에 추가하려는 지표의 필요한 모든 차원을 지정합니다. 다음은 그 한 예입니다.

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

JSON으로 지표 형식 지정에 대한 자세한 정보는 [Properties of a Metric Widget Object](#) 단원을 참조하십시오.

6. [Update]를 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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 그래프 제목에 마우스 포인터를 놓고 [Widget actions], [Edit]를 선택합니다.
4. [Edit graph] 화면의 위쪽에서 그래프의 제목을 선택합니다.
5. Title(제목)에 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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. [Add widget]을 선택합니다.
4. [Text], [Configure]를 선택합니다.
5. 마크다운에서 텍스트를 추가하고 **마크다운**을 사용하여 서식을 지정하고 위젯 생성을 선택합니다.
6. [Save dashboard]를 선택합니다.

대시보드에서 텍스트 위젯을 편집하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 텍스트 블록의 오른쪽 상단 모서리에 마우스 포인터를 놓고 위젯 작업, 편집을 선택합니다.
4. 필요에 따라 텍스트를 업데이트하고 위젯 업데이트를 선택합니다.
5. [Save dashboard]를 선택합니다.

대시보드에서 텍스트 위젯을 제거하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 텍스트 블록의 오른쪽 상단 모서리에 마우스 포인터를 놓고 [Widget actions], [Delete]를 선택합니다.
4. [Save dashboard]를 선택합니다.

## CloudWatch 대시보드에서 경보를 추가 또는 제거

경보를 생성해서 대시보드에 추가할 수 있습니다. 대시보드에서 경보가 발행되면 빨간색으로 ALARM 상태라고 표시가 됩니다.

대시보드에 경보를 추가하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms]를 선택하고 추가할 경보를 선택한 다음, [Add to Dashboard]를 선택합니다.
3. 대시보드를 선택하고, 위젯 유형(Line, Stacked area 또는 Number)을 선택한 다음 [Add to dashboard]를 선택합니다.
4. 대시보드에 경보를 표시하려면 탐색 창에서 [Dashboards]를 선택하고 해당 대시보드를 선택합니다.
5. (선택 사항) 경보 그래프를 일시적으로 확대하려면 해당 그래프를 선택합니다.
6. (선택 사항) 위젯 유형을 변경하려면 그래프 제목에 마우스 포인터를 놓고 [Widget actions]와 [Widget type]을 선택합니다.

대시보드에서 경보를 제거하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 그래프 제목에 마우스 포인터를 놓고 [Widget actions]와 [Delete]를 선택합니다.
4. [Save dashboard]를 선택합니다. 변경 사항을 저장하기 전에 대시보드에서 다른 곳으로 이동하려고 시도하면 변경 사항을 저장하거나 삭제하라는 메시지가 나타납니다.

## 단일 CloudWatch 대시보드를 사용하여 여러 리전의 리소스를 모니터링

단일 CloudWatch 대시보드를 사용하여 여러 리전의 AWS 리소스를 모니터링할 수 있습니다. 예를 들어 us-east-1 리전에 위치한 결제 지표와 함께 us-west-2 리전에 위치한 EC2 인스턴스에 대한 CPU 사용률을 보여주는 대시보드를 생성할 수 있습니다.

단일 대시보드에서 여러 리전의 리소스를 모니터링하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. 탐색 모음에서 리전을 선택합니다.
4. 대시보드에 추가하려는 지표를 선택합니다.
5. [Actions]에서 [Add to dashboard]를 선택합니다.
6. [Add to]에 새 대시보드의 이름을 입력하고 [Add to dashboard]를 선택합니다.

아니면 [Existing dashboard]를 선택하고 대시보드를 선택한 다음 [Add to dashboard]를 선택하여 기존 대시보드에 추가할 수도 있습니다.

7. 다른 리전의 지표를 추가하려면 다음 리전을 선택하고 이러한 단계를 반복합니다.
8. [Save dashboard]를 선택합니다.

## 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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. [Actions], [Link graphs]를 선택합니다.

대시보드에서 그래프들의 링크를 해제하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.

3. [Actions]와 [Link graphs]의 선택을 해제합니다.

## 즐거찾기 목록에 대시보드 추가

즐거찾는 대시보드 목록에 CloudWatch 대시보드를 추가하면 빠르게 찾을 수 있습니다. [Favorites] 목록은 탐색 창이 맨 아래에 나타납니다.

[Favorites] 목록에 대시보드를 추가하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택합니다.
3. 추가할 대시보드 옆에 있는 별표 기호를 선택합니다.

## 기간 재정의 설정 또는 CloudWatch 대시보드에 대한 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. [Actions]를 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 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. [Refresh options] 메뉴(오른쪽 상단 모서리)에서 [10 Seconds], [1 Minute], [2 Minutes], [5 Minutes] 또는 [15 Minutes]를 선택합니다.

대시보드를 자동으로 새로 고치려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. [Refresh options], [Auto refresh]를 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 정보는 [고분해능 지표 \(p. 42\)](#) 단원을 참조하십시오.

대시보드 시간 범위를 변경하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. 다음 중 하나를 수행하십시오.
  - 1h, 3h, 12h, 1d, 3d, 1w와 같이 1시간부터 1주까지 사전 정의된 범위 중 하나를 선택합니다.
  - [custom], [Relative]를 선택합니다. 1분부터 15개월까지 사전 정의된 범위 중 하나를 선택합니다.
  - [custom], [Absolute]를 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 [Actions]를 선택하고 [Period:]에 새 값을 선택합니다.

대시보드 시간 형식을 변경하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Dashboards]를 선택하고 대시보드를 선택합니다.
3. [custom]을 선택합니다.
4. 상단 모서리에서 [UTC] 또는 [Local timezone]을 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 처리 및 경보 발행을 위해 계정에 모든 지표(제공된 AWS 리소스 지표 및 애플리케이션 지표)를 로드할 수 있습니다.

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

## 목차

- [사용 가능한 지표 보기 \(p. 27\)](#)
- [사용 가능한 지표 검색 \(p. 29\)](#)
- [지표에 대한 통계 얻기 \(p. 30\)](#)
- [지표 그래프 \(p. 37\)](#)
- [사용자 지정 지표 게시 \(p. 42\)](#)
- [지표 수식 사용 \(p. 44\)](#)

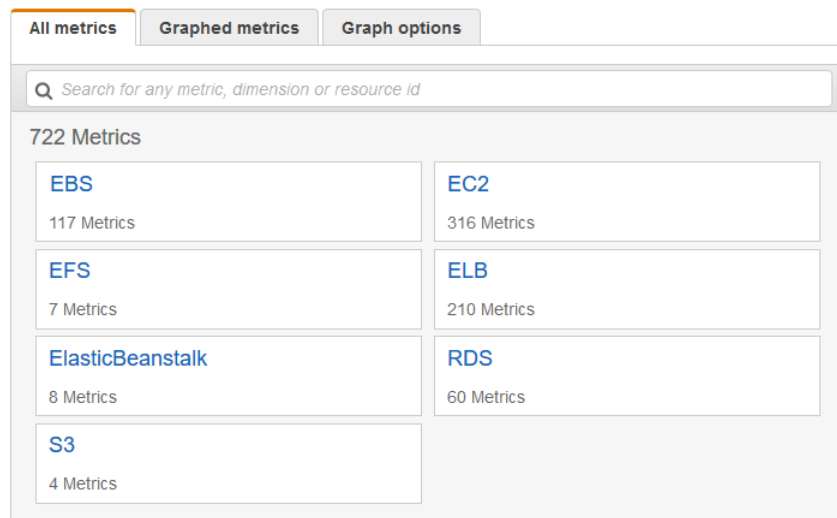
## 사용 가능한 지표 보기

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.

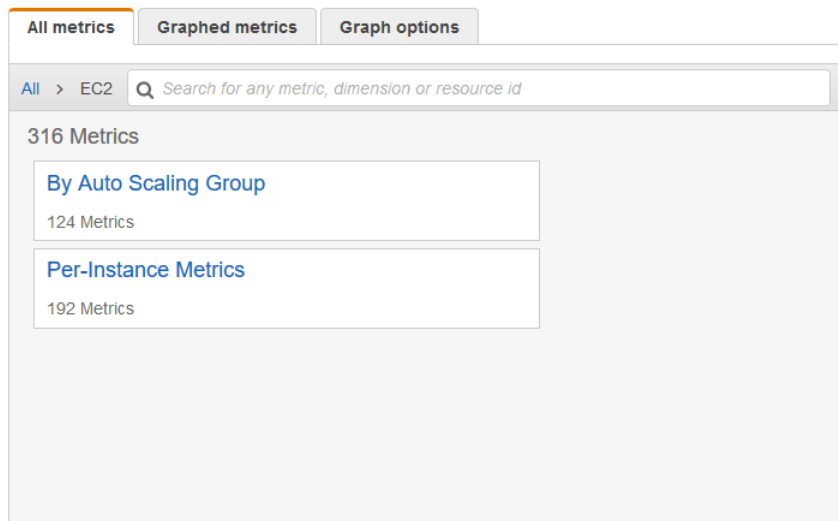
사용 중인 AWS의 서비스에서만 Amazon CloudWatch로 지표를 전송할 수 있습니다.

콘솔을 사용하여 네임스페이스와 차원별로 사용 가능한 지표를 보려면

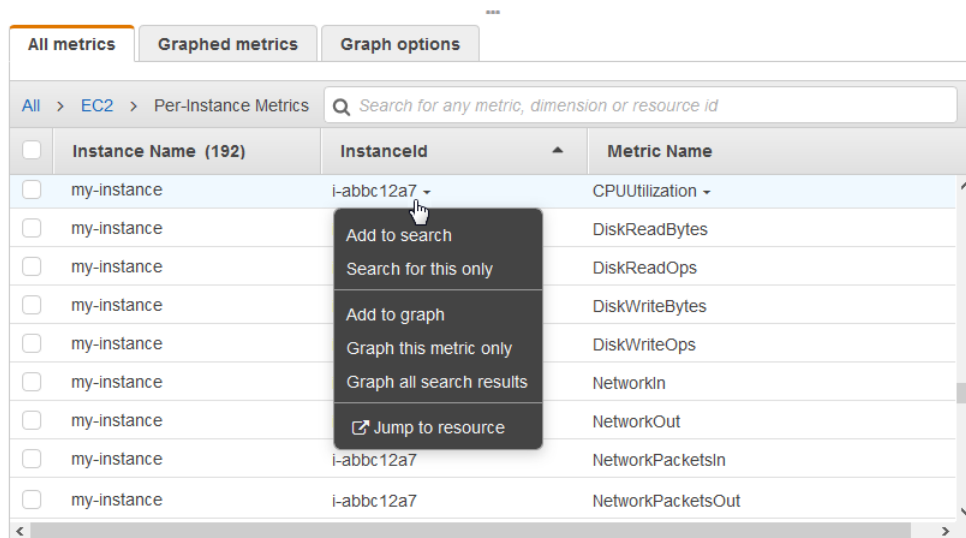
1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. 지표 네임스페이스(예: EC2)를 선택합니다.



4. 지표 차원(예: 인스턴스별 지표)을 선택합니다.



5. [All metrics] 탭에 네임스페이스의 해당 차원에 대한 모든 지표가 표시됩니다. 다음을 수행할 수 있습니다.
- 테이블을 정렬하려면 열 머리글을 사용합니다.
  - 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다. 모든 지표를 선택하려면 테이블의 머리글 행에 있는 확인란을 선택합니다.
  - 리소스로 필터링하려면 리소스 ID를 선택한 후 [Add to search]를 선택합니다.
  - 측정치로 필터링하려면 측정치 이름을 선택한 후 [Add to search]를 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 \(p. 156\)](#) 단원을 참조하십시오.

다음 예제는 에 대한 모든 지표를 볼 수 있도록 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

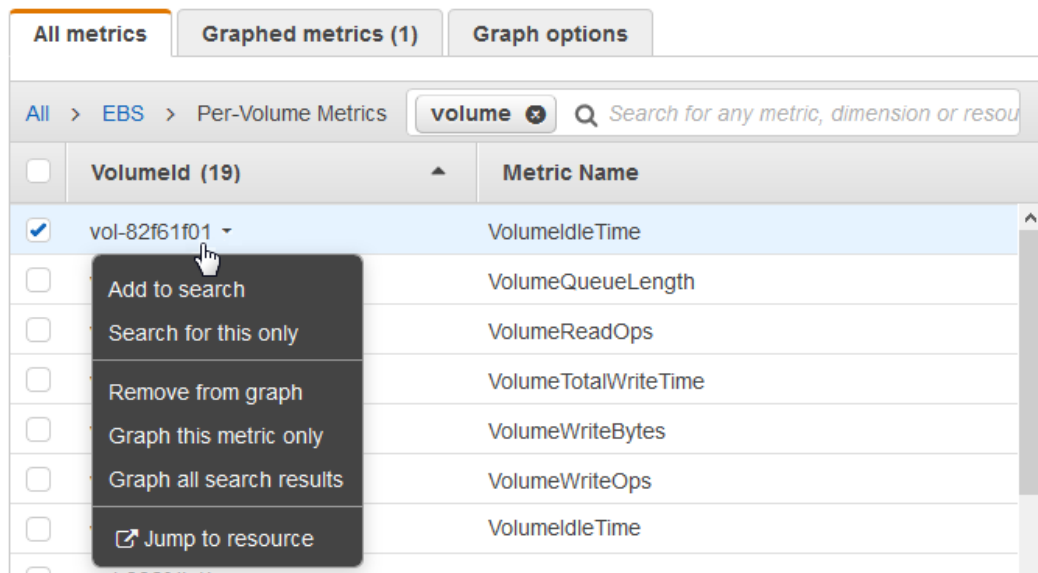
목표하는 검색 단어를 사용하여 계정의 모든 지표를 검색할 수 있습니다. 네임스페이스, 지표 이름 또는 차원 내에서 결과가 일치하는 지표가 반환됩니다.

CloudWatch에서 사용 가능한 지표를 검색하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. [All metrics] 탭의 검색 필드에 검색 단어(예: 지표 이름, 서비스 이름 또는 리소스 이름)를 입력하고 Enter 키를 누릅니다. 이렇게 하면 이 검색 단어를 가진 지표에 모든 네임스페이스가 표시됩니다.

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

4. 지표 검색을 위한 결과와 함께 네임스페이스를 선택합니다. 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. 콘솔에서 리소스 중 하나를 보려면 리소스 ID를 선택하고 [Jump to resource]를 선택합니다.
  - c. 지표에 대한 도움말을 보려면 지표 이름을 선택하고 [What is this?]를 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. 30)
- 리소스에서 통계 집계하기 (p. 33)
- Auto Scaling 그룹별 통계 집계 (p. 35)
- Amazon 머신 이미지(AMI)별로 통계 집계 (p. 36)

## 특정 리소스에 대한 통계 얻기

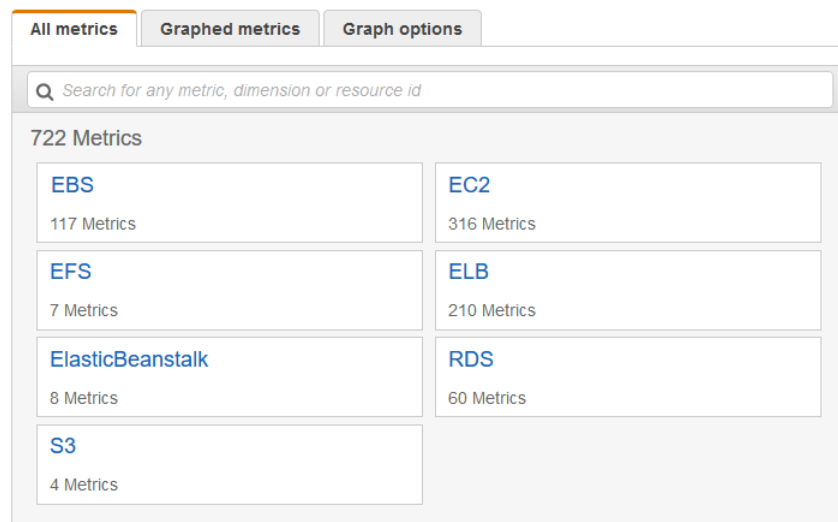
다음 예제는 특정 EC2 인스턴스의 최대 CPU 사용률을 확인하는 방법을 보여 줍니다.

## 요구 사항

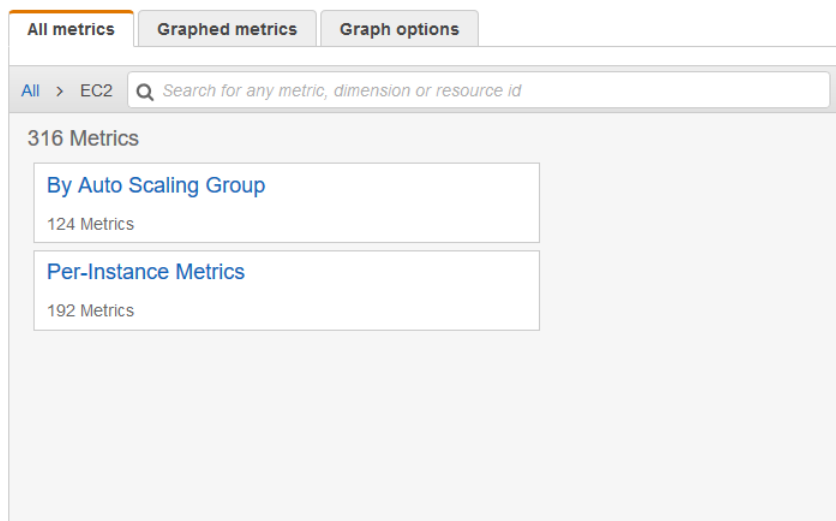
- 인스턴스의 ID가 필요합니다. 인스턴스 ID는 Amazon EC2 콘솔이나 [describe-instances](#) 명령을 사용하여 확인할 수 있습니다.
- 기본적으로 기본 모니터링이 사용되지만 세부 모니터링을 사용하도록 설정할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화](#)를 참조하십시오.

콘솔을 사용하여 특정 인스턴스에 대한 평균 CPU 사용률을 표시하려면

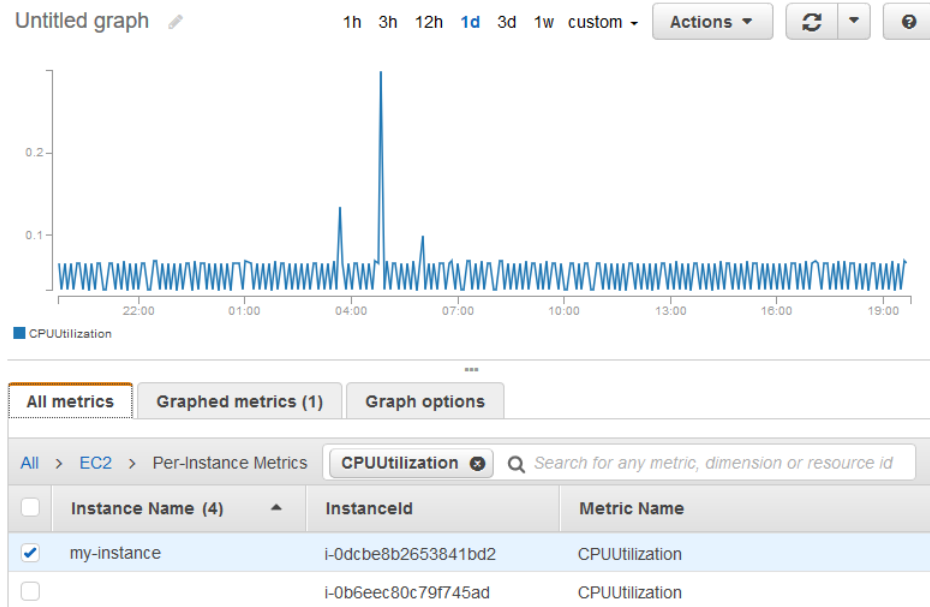
1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. EC2 지표 네임스페이스를 선택합니다.



4. 인스턴스당 지표 차원을 선택합니다.



5. 검색 필드에 **CPUtilization**을 입력하고 Enter를 누릅니다. 특정 인스턴스의 행을 선택합니다. 그러면 해당 인스턴스의 [CPUtilization] 측정치 그래프가 표시됩니다. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 [custom]을 선택합니다.



6. 통계를 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 다음, 통계나 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: p95.45)를 지정합니다.

All metrics Graphed metrics (1) Graph options

Label	Namespace	Dimensions	Metric Name	Statistic	Period
CPUUtilization	AWS/EC2	Dimensions (1)	CPUUtilization	Average	5 Minutes

Average  
 Minimum  
 Maximum  
 Sum  
 Data Samples  
 p99  
 p95  
 p90  
 p50  
 p10  
 Custom percentile...

7. 기간을 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.

AWS CLI를 사용하여 EC2 인스턴스별 CPU 사용률을 얻으려면

다음 `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 부분입니다(전체 출력에는 24시간 동안 6분마다 반환된 데이터 요소가 포함).

```
{
  "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 집계하기

여러 리소스에서 AWS 리소스에 대한 지표를 집계할 수 있습니다. Amazon CloudWatch에서는 리전 간 데이터는 집계하지 않습니다. 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 단위로 데이터를 제공하는 세부 모니터링(추가 요금 부과)을 활성화해야 합니다. 자세한 내용은 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스에 대한 세부 모니터링 활성화 또는 비활성화](#)를 참조하십시오.

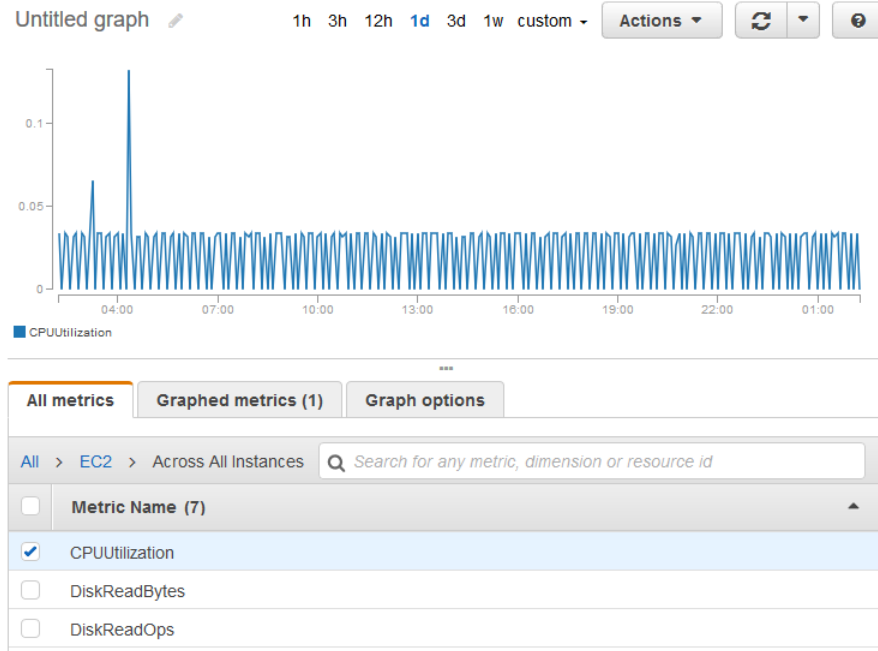
이 예제는 EC2 인스턴스의 평균 CPU 사용률을 확인하는 방법을 보여줍니다. 지정된 차원이 없으므로 CloudWatch에서는 AWS/EC2 네임스페이스의 모든 차원에 대한 통계를 반환합니다. 다른 지표에 대한 통계를 얻으려면 [CloudWatch 지표를 게시하는 AWS 서비스 \(p. 156\)](#) 단원을 참조하십시오.

### Important

AWS 네임스페이스에서 모든 차원을 검색하는 기능은 Amazon 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 [Metrics]를 선택합니다.
3. [EC2] 네임스페이스를 선택한 후 [Across All Instances]를 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 [custom]을 선택합니다.



- 통계를 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 다음, 통계나 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: p95.45)를 지정합니다.
- 기간을 변경하려면 [Graphed metrics] 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 하나의 Auto Scaling 그룹에 집계할 수 있습니다. Amazon CloudWatch에서는 리전 간 데이터는 집계하지 않습니다. 리전마다 지표가 완전히 분리되어 있습니다.

이 예제는 하나의 Auto Scaling 그룹에 대해 디스크에 기록되는 총 바이트 수를 확인하는 방법을 보여줍니다. 이 값은 지정한 Auto Scaling 그룹의 모든 EC2 인스턴스에 대해 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 [Metrics]를 선택합니다.
3. [EC2] 네임스페이스를 선택한 후 [By Auto Scaling Group]을 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 [custom]을 선택합니다.



All metrics			Graphed metrics (1)	Graph options
All > EC2 > By Auto Scaling Group				
Search for any metric, dimension or resource id				
<input type="checkbox"/>	AutoScalingGroupName (28)		Metric Name	
<input type="checkbox"/>	my-asg		DiskReadBytes	
<input type="checkbox"/>	my-asg		DiskReadOps	
<input checked="" type="checkbox"/>	my-asg		DiskWriteBytes	
<input type="checkbox"/>	my-asg		DiskWriteOps	

5. 통계를 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 다음, 통계나 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: p95.45)를 지정합니다.
6. 기간을 변경하려면 [Graphed metrics] 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"
}
```

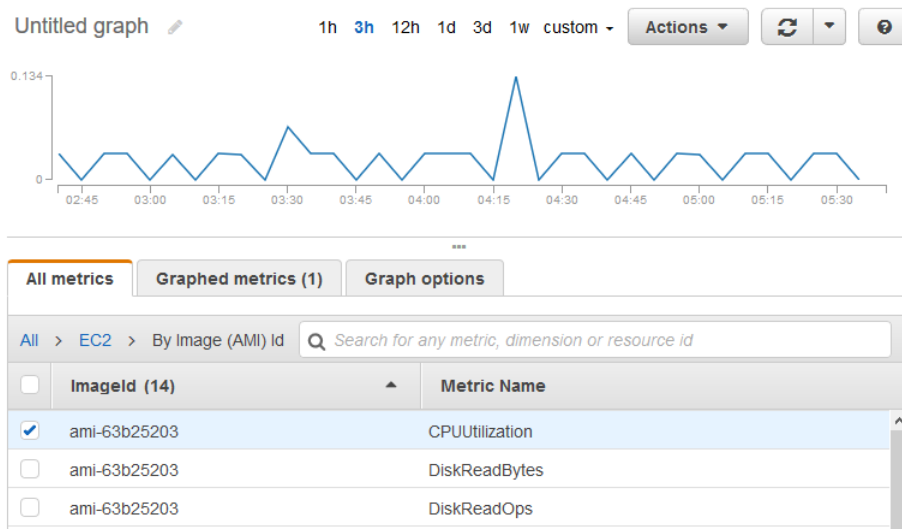
## Amazon 머신 이미지(AMI)별로 통계 집계

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 비활성화](#)를 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 [Metrics]를 선택합니다.
3. [EC2] 네임스페이스를 선택한 후 [By Image (AMI) Id]를 선택합니다.
4. [CPUUtilization] 지표 행과 특정 AMI를 선택합니다. 그러면 지정한 AMI의 그래프가 표시됩니다. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 [custom]을 선택합니다.



5. 통계를 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 다음, 통계나 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: p95.45)를 지정합니다.

6. 기간을 변경하려면 [Graphed metrics] 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 간격에 대한 1분 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": 0.0360000000000000011,
      "Unit": "Percent"
    },
    ...
  ],
  "Label": "CPUUtilization"
}
```

## 지표 그래프

서비스에서 지표 활동을 훨씬 쉽게 파악할 수 있도록 CloudWatch 콘솔을 사용하여 다른 AWS 서비스에서 생성한 지표 데이터를 그래프 처리할 수 있습니다. CloudWatch에서는 다음 절차에 따라 메트릭을 그래프 처리할 수 있습니다.

목차

- [지표 그래프 작성 \(p. 37\)](#)
- [그래프의 시간 범위 또는 시간대 형식 수정 \(p. 39\)](#)
- [그래프의 Y축 수정 \(p. 40\)](#)
- [그래프의 지표에서 경보 생성 \(p. 41\)](#)

## 지표 그래프 작성

CloudWatch 콘솔을 사용하여 지표를 선택하고 데이터의 그래프를 작성할 수 있습니다.

CloudWatch는 지표에 대해 Average, Minimum, Maximum, Sum, SampleCount 등의 통계를 지원합니다. 자세한 내용은 [통계 \(p. 5\)](#)를 참조하십시오.

세부 수준을 달리하여 데이터를 볼 수 있습니다. 예를 들어 세부 뷰(예: 1분)를 선택할 수 있으며, 이는 문제 해결에 유용합니다. 덜 세부적인 뷰(예: 1시간)를 선택할 수도 있으며, 이는 시간 경과에 따른 트렌드를 확인하기 위해 더 넓은 시간 범위(예: 3일)를 확인할 때 유용할 수 있습니다. 자세한 정보는 [기간 \(p. 6\)](#) 단원을 참조하십시오.

## 그래프 작성

지표 그래프를 작성하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. [All metrics] 탭의 검색 필드에 검색 단어(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. 그래프 처리하려는 지표에 대한 자세한 정보를 보려면 범례에 마우스 포인터를 둡니다.
7. 가로 주석을 사용하면 그래프 사용자가 지표가 특정 수준까지 급상승하는 경우 또는 지표가 사전 정의된 범위를 유지하는지 여부를 빠르게 확인할 수 있습니다. 가로 주석을 추가하려면 [Graph options], [Add horizontal annotation]을 선택합니다.
  - a. [Label]에 주석의 레이블을 입력합니다.
  - b. [Value]에 가로 주석이 표시될 지표 값을 입력합니다.
  - c. [Fill]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. [Axis]에서 그래프에 여러 지표가 포함된 경우 Value 값이 왼쪽 Y축 또는 오른쪽 Y축과 연결된 지표를 참조할지 지정합니다.

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

동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

주석을 숨기려면 해당 주석의 왼쪽 옆에서 확인란을 선택 취소합니다.

주석을 삭제하려면 [Actions] 옆에서 [x]를 선택합니다.

8. 그래프에 대한 URL을 얻으려면 [Actions]와 [Share]를 선택합니다. URL을 복사하여 이를 저장 또는 공유합니다.
9. 대시보드에 그래프를 추가하려면 [Actions]와 [Add to dashboard]를 선택합니다.

## 그래프 업데이트

그래프를 업데이트하려면

1. 그래프 이름을 변경하려면 연필 아이콘을 선택합니다.
2. 시간 범위를 변경하려면 제공되는 값 중 하나를 선택하거나 [custom]을 선택합니다. 자세한 내용은 [그래프의 시간 범위 또는 시간대 형식 수정 \(p. 39\)](#) 단원을 참조하십시오.
3. 통계를 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 다음, 통계나 사전 정의된 백분위수 중 하나를 선택하거나 사용자 지정 백분위수(예: p95.45)를 지정합니다.

4. 기간을 변경하려면 [Graphed metrics] 탭을 선택합니다. 열 머리글이나 개별 값을 선택한 후 다른 값을 선택합니다.
5. 가로 주석을 추가하려면 [Graph options], [Add horizontal annotation]을 선택합니다.
  - a. [Label]에 주석의 레이블을 입력합니다.
  - b. [Value]에 가로 주석이 표시될 지표 값을 입력합니다.
  - c. [Fill]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. [Axis]에서 그래프에 여러 지표가 포함된 경우 value 값이 왼쪽 Y축 또는 오른쪽 Y축과 연결된 지표를 참조할지 지정합니다.

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

동일한 그래프에 여러 가로 주석을 추가하려면 이들 단계를 반복합니다.

주석을 숨기려면 해당 주석의 왼쪽 옆에서 확인란을 선택 취소합니다.

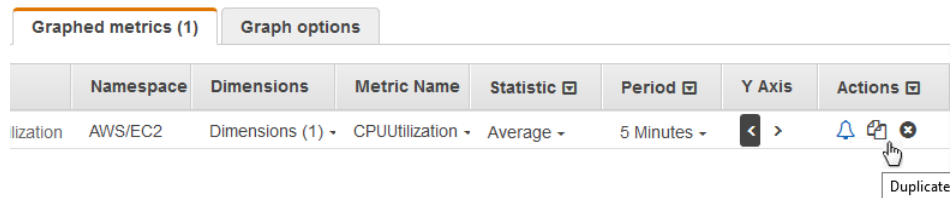
주석을 삭제하려면 [Actions] 옆에서 [x]를 선택합니다.

6. 새로 고침 간격을 변경하려면 [Refresh options]를 선택한 다음 [Auto refresh]를 선택하거나 [1 Minute], [2 Minutes], [5 Minutes] 또는 [15 Minutes]를 선택합니다.

## 지표 복제

지표를 복제하려면

1. [Graphed metrics] 탭을 선택합니다.
2. [Actions]에서 [Duplicate] 아이콘을 선택합니다.



3. 필요에 따라 복제 지표를 업데이트합니다.

## 그래프의 시간 범위 또는 시간대 형식 수정

그래프의 시간 범위 또는 시간대 형식을 변경할 수 있습니다.

### 상대적 시간 범위

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 [Metrics]를 선택합니다.
3. 페이지 상단에 나와 있는 사전 정의된 범위(1시간부터 1주까지) 중 하나를 선택합니다.
4. 더 많은 사전 정의된 범위를 보려면 [custom] 메뉴를 선택하고 [Relative]를 선택합니다. 5분부터 15개월까지 사전 정의된 범위 중 하나를 선택합니다.

## 절대적 시간 범위

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 [Metrics]를 선택합니다.
3. [custom] 메뉴에서 [Absolute]를 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 [Metrics]를 선택합니다.
3. [custom] 메뉴를 선택한 다음 [UTC] 또는 [Local timezone]을 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 [Metrics]를 선택합니다.
3. 그래프 영역을 선택해서 드래그한 다음, 마우스 버튼을 놓습니다.
4. 확대된 그래프를 재설정하려면 [Reset zoom] 아이콘을 선택합니다.

## 그래프의 Y축 수정

데이터 확인이 용이하도록 그래프의 Y축에 대해 사용자 지정 경계를 설정할 수 있습니다. 예를 들어 CPUUtilization 그래프의 경계를 100%로 변경하면 CPU 사용률이 낮은지(그래프 하단 부근에 폴룩되는 선이 표시) 또는 높은지(그래프 상단 부근에 폴룩되는 선이 표시)를 손쉽게 확인할 수 있습니다.

그래프에서 서로 다른 두 Y축 간의 전환이 가능합니다. 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 [Metrics]를 선택합니다.
3. 지표 네임스페이스(예: EC2)를 선택한 다음, 지표 차원(예: Per-Instance Metrics)을 선택합니다.
4. [All metrics] 탭에 네임스페이스의 해당 차원에 대한 모든 지표가 표시됩니다. 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다.
5. [Graph options]에서 [Left Y Axis]에 대한 [Min]과 [Max] 값을 지정합니다. [Min] 값은 [Max] 값보다 클 수 없습니다.

The screenshot shows the 'Graph options' tab with two sections: 'Left Y Axis' and 'Right Y Axis'. Under 'Left Y Axis', there are input fields for 'Limits' with 'Min' set to 0 and 'Max' set to 100. Under 'Right Y Axis', there are input fields for 'Limits' with 'Min' set to 'Auto' and 'Max' set to 'Auto'.

- 두 번째 Y축을 생성할 수 있도록 [Right Y Axis]에 대한 [Min]과 [Max] 값을 지정합니다.
- 두 Y축 간의 전환을 위해서 [Graphed metrics] 탭을 선택합니다. [Y Axis]에서 [Left Y Axis] 또는 [Right Y Axis]를 선택합니다.

The screenshot shows the 'Graphed metrics (1)' tab. Below the tabs is a table with the following columns: Namespace, Dimensions, Metric Name, Statistic, Period, Y Axis, and Actions. The 'Y Axis' column has a dropdown menu with 'Right Y Axis' selected. A tooltip 'Right Y Axis' is visible over the dropdown.

## 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

- <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
- 탐색 창에서 [Metrics]를 선택합니다.
- 지표 네임스페이스(예: EC2)를 선택한 다음, 지표 차원(예: Per-Instance Metrics)을 선택합니다.
- [All metrics] 탭에 네임스페이스의 해당 차원에 대한 모든 지표가 표시됩니다. 지표를 그래프로 표시하려면 지표 옆에 있는 확인란을 선택합니다.
- 지표에 대한 경보를 생성하려면 [Graphed metrics] 탭을 선택합니다. [Actions]에서 경보 아이콘을 선택합니다.

The screenshot shows the 'Graphed metrics (1)' tab. Below the tabs is a table with the following columns: Namespace, Dimensions, Metric Name, Statistic, Period, Y Axis, and Actions. The 'Actions' column has a dropdown menu with 'Create alarm' selected. A tooltip 'Create alarm' is visible over the dropdown.

- [Alarm Threshold]에서 경보의 고유한 이름과 경보에 대한 설명을 입력합니다. [Whenever]에서 임계값과 기간 수를 지정합니다.
- [Actions]에서 경보 트리거 시 수행할 작업의 유형을 선택합니다.
- (선택 사항) [Period]에서 다른 값을 선택합니다. [Statistic]에서 [Standard]를 선택하여 목록의 통계 중 하나를 지정하거나 [Custom]을 선택하여 백분위수(예: p95.45)를 지정합니다.

Period: 5 Minutes

Statistic: ☐ Standard ☒ Custom

p95.45

9. [Create Alarm]을 선택합니다.

## 사용자 지정 지표 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 \(p. 42\)](#)
- [차원 사용 \(p. 42\)](#)
- [단일 데이터 요소 게시 \(p. 43\)](#)
- [통계 세트 게시 \(p. 44\)](#)
- [0 값 게시 \(p. 44\)](#)

## 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 PutMetricData를 호출할 때마다 요금이 부과되며, 따라서 고분해능 지표에 대해 PutMetricData를 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. 하나의 지표에 최대 10개까지 차원을 가질 수 있으며, 각 차원은 이름-값 페어로 정의됩니다.

서로 다른 명령을 사용하면 차원을 지정하는 방법도 달라집니다. put-metric-data에서는 각 차원을 **MyName=MyValue**으로 지정하고, get-metric-statistics 또는 put-metric-alarm에서는 Name=**MyName**, Value=**MyValue** 형식을 사용합니다. 예를 들어 다음 명령은 InstanceId와 InstanceType이라는 두 가지 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 구분합니다. 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-statistics](#)를 사용할 때 정의된 모든 차원에 대해 값을 지정해야 합니다. 예를 들어 Amazon S3 지표의 경우 BucketSizeBytes에 BucketName 및 StorageType 차원이 포함되어 있기 때문에 [get-metric-statistics](#)에서 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
```

[list-metrics](#) 명령을 사용하여 지표에 어떤 차원이 정의되어 있는지 확인할 수 있습니다.

## 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

지표 생성 시 [get-metric-statistics](#) 명령을 사용해 새 지표에 대한 통계를 검색할 수 있기까지 최대 2분이 소요될 수 있습니다. 그러나 [list-metrics](#) 명령을 사용해 검색된 지표 목록에 새 지표가 나타나려면 최대 15분이 걸릴 수 있습니다.

천분의 1초 만큼 세분화된 타임스탬프를 사용하여 데이터 요소를 게시하더라도 CloudWatch에서는 최소 1분으로 세분화되도록 데이터를 집계합니다. CloudWatch는 1분 주기로 수신된 값의 평균(모든 항목의 합을 항목 수로 나눈 값)을 비롯해 샘플 수, 동일 기간에 대한 최대값 및 최소값을 기록합니다. 예를 들어 이전 예의 PageViewCount 지표에는 타임스탬프가 몇 초 간격인 데이터 요소 3개가 들어 있습니다. 데이터 요소 3개의 타임스탬프가 모두 1분 기간 내에 있으므로 CloudWatch에서는 이러한 데이터 요소를 집계합니다.

또한 데이터 요소 집계 시 CloudWatch에서는 1분 경계를 사용합니다. 예를 들어 데이터 요소 3개가 모두 2016-10-20T12:00:00.000Z에서 시작하여 2016-10-20T12:01:00.000Z에서 끝나는 1분 기간 내에 있으므로 CloudWatch에서는 이전 예의 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 SampleCount는 1.
- 통계 세트의 최소값과 최대값이 동일.

## 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. PutMetricData에 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. 결과 시계열을 CloudWatch 콘솔에 표시하고 이를 대시보드에 추가할 수 있습니다. 예를 들어, AWS Lambda 지표를 사용하여 오류 지표를 호출 지표로 나뉜 오류율을 계산하고 결과 시계열을 사용자의 CloudWatch 대시보드 그래프에 추가할 수 있습니다.

GetMetricData API 작업을 사용하여 지표 계산을 프로그래밍 방식으로 실행할 수도 있습니다.

## CloudWatch 그래프에 수학 표현식 추가

CloudWatch 대시보드에서 그래프에 수학 표현식을 추가할 수 있습니다. 각 그래프의 지표와 표현식은 최대 100로 제한되므로, 그래프의 지표가 99개 이하일 경우에만 수학 표현식을 추가할 수 있습니다.

그래프에 수학 표현식을 추가하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 그래프 또는 줄 위젯을 만들거나 편집합니다.
3. [Graphed metrics]를 선택합니다.
4. [Add a math expression]를 선택합니다. 표현식의 새 줄이 나타납니다.
5. [Details] 열에는 수학 표현식만 입력합니다. 다음 단원의 표에 표현식에 사용할 수 있는 함수가 나열됩니다.

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, 밑줄이 포함될 수 있으며, 소문자로 시작해야 합니다. Id의 값을 좀 더 의미 있는 이름으로 변경하면 그래프를 이해하기가 더욱 쉬워집니다. 예를 들어, m1과 m2를 errors와 requests로 변경합니다.

6. 표현식의 [Label] 열에는 표현식으로 계산되는 내용을 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. **Expression-Label(###-###)**을 비워둠으로써 범례에 원래의 지표 레이블만 표시되게 설정할 수 있습니다.

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 수도 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 작성되어야 하며(예: AVG), 모든 지표와 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 생성되어 최종 표현식 결과가 되지 못합니다. 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 목록의 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. 예: 지난 3일 동안 인스턴스 i-1234567890abcdef0에 대한 CPUUtilization 지표.
- TS[]는 시계열 어레이입니다(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. 시계열에서 누락된 값은 0로 처리됩니다. 데이터 요소의 값 때문에 함수에서 0으로 나누려고 시도할 경우 해당 데이터 요소가 누락됩니다.

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

## 지표 수식에 지원되는 함수

다음 표는 수학 표현식에서 사용할 수 있는 함수를 설명합니다. 모든 함수를 대문자로 작성합니다.

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 생성되어 최종 표현식 결과가 되지 못합니다. 그렇지만 이것을 함수 m1-AVG(m1)에서 사용하여 각 개별 데이터 요소와 해당 데이터 요소의 평균 값 차이인 시계열을 표시할 수 있습니다.

아래의 표에서 예제 열의 모든 예제는 단일 시계열이나 시계열 배열을 생성하는 표현식입니다. 이것은 스칼라 수를 반환하는 함수를 단일 시계열을 생성하는 유효한 표현식의 부분으로 사용하는 방법을 보여줍니다.

함수	인수	반환 유형*	설명	예
ABS	TS	TS	각 데이터 요소의 절대값을 반환합니다.	ABS(m1-m2)
	TS[]	TS[]		MIN(ABS([m1, m2]))

함수	인수	반환 유형*	설명	예
				ABS(METRICS())
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 천장값(각 값보다 크거나 같은 최소 정수)을 반환합니다.	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))
FLOOR	TS TS[]	TS TS[]	각 지표의 바닥값(각 값보다 크거나 같은 최소 정수)을 반환합니다.	FLOOR(m1) FLOOR(METRICS())
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 MIN은 단일 시계열을 반환합니다.	m1-MIN(m1) MIN(METRICS())

함수	인수	반환 유형*	설명	예
PERIOD	TS	S	지표의 기간(초)을 반환합니다. 유효한 입력은 지표이지 다른 표현식의 결과가 아닙니다.	m1/PERIOD(m1)
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())
STDDEV	TS TS[]	S TS	단일 시계열의 STDDEV는 지표의 모든 데이터 요소의 표준편차를 나타내는 스칼라를 반환합니다. 시계열 배열의 STDDEV는 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 SUM은 단일 시계열을 반환합니다.	SUM(METRICS())/SUM(m1) SUM([m1,m2]) SUM(METRICS("errors"))/ SUM(METRICS("requests"))*100

\*스칼라 수를 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.

## GetMetricData API 작업과 함께 지표 수식 사용

[GetMetricData]를 사용하여 하나의 API 호출에서 커다란 지표 데이터 배치를 가져올 뿐만 아니라 수학 표현식을 사용하는 계산도 실행할 수 있습니다. 자세한 내용은 [GetMetricData](#) 단원을 참조하십시오.

# Amazon CloudWatch 경보 사용

단일 CloudWatch 지표를 감시하거나 CloudWatch 측정치를 기반으로 하는 수학적 표현식의 결과를 감시하는 CloudWatch 경보를 생성할 수 있습니다. 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.

또한 CloudWatch 대시보드에 경보를 추가해 시각화된 모니터링을 할 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. 작업이 존재하는지 확인하십시오.

## 경보 상태

경보에는 다음과 같은 잠재적인 상태가 있습니다.

- **OK** - 지표 또는 표현식이 정의된 임계값 내에 있습니다.
- **ALARM** - 지표 또는 표현식이 정의된 임계값을 벗어 났습니다.
- **INSUFFICIENT\_DATA** - 경보가 방금 시작되었거나, 측정치를 사용할 수 없거나, 측정치를 통해 경보 상태를 결정하는 데 사용할 충분한 데이터가 없습니다. —

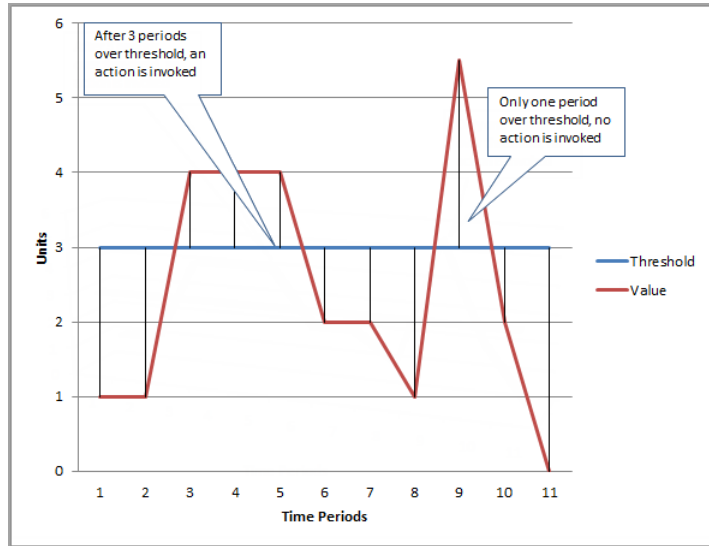
## 경보 평가

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 데이터포인트)는 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.

다음은 경보 임계값을 3개 단위로 설정한 그림입니다. 경보가 ALARM 상태가 되도록 구성하며, Evaluation Period(평가 기간) 및 Datapoints to Alarm(경보에 대한 데이터포인트)는 모두 3입니다. 가장 최근의 연속된 3번의 기간에서 3개 데이터포인트가 모두 임계값 이상일 때 경보가 ALARM 상태가 됩니다. 그림에서는 기간 3

에서 6 사이에 이러한 일이 발생합니다. 기간 6에서는 이 값이 임계값 아래로 떨어져 평가 대상 기간 중 하나가 위반 상태가 아니기 때문에 경보 상태가 OK로 변경됩니다. 9번째 기간에 다시 한 번 임계값이 위반되지만, 오직 하나의 기간 동안에만 그렇습니다. 결과적으로 경보 상태는 OK로 남아 있습니다.



Evaluation Period(평가 기간)와 Datapoints to Alarm(경보에 대한 데이터포인트) 값을 다르게 구성하는 경우 "N 중 M" 경보를 설정합니다. Datapoints to Alarm(경보에 대한 데이터포인트)는 ("M")이며, Evaluation Period(평가 기간)는 ("N")입니다. 평가 간격은 데이터 포인트에 기간을 곱한 값입니다. 예를 들어, 기간을 1분으로 하는 5 중 4 데이터 포인트를 구성하는 경우 평가 간격은 5분입니다. 기간을 10분으로 하는 3 중 3 데이터 포인트를 구성하는 경우에는 평가 간격이 30분이 됩니다.

## 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. 이는 경보를 모니터링 하는 데이터의 유형에 적절한 ALARM 상태로 구성할 수 있도록 도움을 줍니다. 누락된 데이터에 문제가 없는 경우의 오탐을 피할 수 있습니다.

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

- 위반하지 않음(임계값에서)
- 위반(임계값 위반)
- 누락

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

- missing— - 경보가 상태 변경 여부를 평가할 때 누락 데이터 포인트를 고려하지 않습니다.
- notBreaching— - 누락 데이터 포인트를 임계값 내에 있는 데이터 포인트로 처리합니다.
- breaching— - 누락 데이터 포인트를 임계값을 위반한 데이터 포인트로 처리합니다.
- ignore— - 현재 경보 상태를 유지합니다.



최고의 옵션은 지표 유형에 따라 다릅니다. 인스턴스의 CPUUtilization와 같은 데이터를 지속적으로 보고하는 지표의 경우에는 원가 문제가 있음을 나타내기 위해 누락 데이터 포인트를 breaching로 처리하고 싶을 수 있습니다. 그러나 Amazon DynamoDB에서 ThrottledRequests 같은 오류가 발생할 때만 데이터 포인트를 생성하는 지표의 경우에는 누락 데이터를 notBreaching으로 처리하고 싶을 것입니다. 기본값은 missing입니다.

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 경고 상태 평가 방법

CloudWatch는 경고 상태 변경 여부를 평가할 때 누락 데이터를 처리하는 방법에 대해 설정한 값과 상관없이 Evaluation Periods(평가 기간)이 지정한 것보다 더 큰 데이터 포인트 수를 검색하려 시도합니다. 검색을 시도하는 데이터 포인트의 정확한 수는 경고 기간의 길이, 표준 해상도 또는 고 해상도 지표에 토대를 두고 있는 지 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 경고 Evaluation Periods(평가 기간) 이상인 경우, CloudWatch는 성공적으로 검색을 한 가장 최근의 기존 데이터 포인트에 따라 경고 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 경고 평가 기간의 수 이하인 경우, CloudWatch는 사용자가 누락 데이터 처리 방법에 대해 지정한 값으로 누락 데이터 포인트를 채운 다음 경보를 평가합니다. 하지만 보고 시기에 상관 없이 평가 범위 동안의 실제 데이터 포인트는 모두 평가에 포함시킵니다. CloudWatch는 가능한 몇 회만 누락 데이터 포인트를 사용합니다.

이 모든 상황에서 평가 데이터포인트의 수는 Evaluation Periods(평가 기간)의 값과 동일합니다. Datapoints to Alarm(경보에 대한 데이터포인트)의 값보다 작은 값만 위반된 경우 경고 상태는 정상으로 설정됩니다. 나머지 경우는 경보로 설정됩니다.

### Note

이 동작의 특별한 경우에서 지표 흐름이 멈추고 일정 시간 동안 최종 데이터 포인트 세트를 CloudWatch 경고 기능이 계속해서 다시 평가할 수 있습니다. 이 재평가로 인해 지표 스트림 중지 직전에 상태가 변한 경우 경보가 상태를 변경하고 작업을 다시 실행할 수 있습니다. 이 동작을 완화하려면 더 짧은 기간을 사용하십시오.

다음은 경고 평가 동작에 대한 예를 설명한 테이블입니다. 첫 테이블에서 Datapoints to Alarm(경보에 대한 데이터포인트)과 Evaluation Periods(평가 기간)는 모두 3입니다. CloudWatch는 경보를 평가할 때 가장 최근의 데이터 포인트 5개를 검색합니다.

2열은 이 5개 데이터 포인트 중 누락되어, 누락 데이터 처리 방법에 대한 설정을 사용해 채워 넣어야 하는 데이터 포인트의 수를 보여줍니다. 3-6열은 누락 데이터 처리 방법에 대한 설정 각각에 대해 설정하게 되는 경고 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	확인	확인	확인	확인
0 - - - -	2	확인	확인	확인	확인
- - - - -	3	데이터 부족	현재 상태 유지	경보	확인

데이터 포인트	누락 데이터 포인트 가운데 수 (#)	누락	IGNORE	위반	위반하지 않음
0 X X - X	0	경보	경보	경보	경보
---- X	2	데이터 부족	현재 상태 유지	경보	확인

앞 테이블의 2행에서는 누락 데이터를 위반으로 처리하는 경우에도 경보 상태는 정상으로 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 경보 상태가 됩니다. 위반하지 않은 데이터 포인트가 가장 최근 검색한 5개 데이터 포인트에서 빠지기 때문입니다. 4행에서는 경보가 모두 경보 상태입니다. 실제 데이터 포인트가 충분해 누락 데이터 처리 방법에 대한 설정을 고려할 필요가 없기 때문입니다.

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

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

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 정보는 [사용자 지정 지표 게시 \(p. 42\)](#) 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 처리하도록 하는 방법을 지정할 수 있습니다.

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

지표 수학 표현식 및 구문에 대한 자세한 정보는 [지표 수식 사용 \(p. 44\)](#) 단원을 참조하십시오.

## 백분위수 기반 CloudWatch 경고 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 경고 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.

백분위수가 0.5~1.00인 경우, 평가 기간 동안 데이터 요소가  $10/(1-\text{백분위수})$  보다 적을 때 이 설정이 사용됩니다. 예를 들어 p99 백분위수에서 경고 샘플이 1,000개보다 적을 경우 이 설정이 사용됩니다. 백분위수가 0~0.5인 경우, 데이터 요소가  $10/(\text{백분위수})$  보다 적을 때 이 설정이 사용됩니다.

## CloudWatch 경고의 일반적인 기능

아래 기능은 모든 CloudWatch 경고에 적용됩니다.

- AWS 계정마다 리전당 최대 5,000개까지 경보를 만들 수 있습니다. 경보를 만들거나 업데이트하려면 PutMetricAlarm API 작업(mon-put-metric-alarm 명령)을 사용합니다.
- 경고 이름은 ASCII 문자만 포함해야 합니다.
- 현재 구성된 경고의 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 경고 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 경고 정의에 포함시켜야 합니다.
- DescribeAlarmHistory(mon-describe-alarm-history)를 사용하여 경고 기록을 볼 수 있습니다. CloudWatch에서는 경고 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 타임스태프를 사용하여 고유한 상태 변경을 확인할 수 있습니다.
- 경고에 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 경고 세트가 있으면 상태가 [Insufficient Data]로 변경됩니다. 이는 리소스가 비활성 상태를 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 경고 상태 변경 시 이메일 알림을 보낼 수 있습니다. 자세한 정보는 [Amazon Simple Notification Service 시작 안내서](#) 단원을 참조하십시오.

#### Note

또는 AWS Management 콘솔을 사용하여 CloudWatch 경보를 생성할 계획이라면 이 절차를 건너뛸 수 있습니다. 왜냐하면 Create Alarm 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/v2/home>에서 Amazon SNS 콘솔을 엽니다.
2. Amazon SNS 대시보드의 일반 작업에서 주제 생성을 선택합니다.
3. [Create new topic] 대화 상자의 [Topic name]에 주제 이름(예: my-topic)을 입력합니다.
4. [Create topic]을 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/v2/home>에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 [Subscriptions]와 [Create subscription]을 선택합니다.
3. [Create subscription] 대화 상자의 [Topic ARN]에서 이전 작업에서 생성한 주제 ARN을 붙여 넣습니다.
4. 프로토콜에서 이메일을 선택합니다.
5. [Endpoint]에 알림을 받는 데 사용할 수 있는 이메일 주소를 입력한 다음 [Create subscription]을 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/v2/home>에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 [Topics]를 선택합니다.
3. [Topics] 페이지에서 주제를 선택하고 [Publish to topic]을 선택합니다.
4. [Publish a message] 페이지의 [Subject]에 메시지에 대한 제목 줄을 입력하고 [Message]에 간단한 메시지를 입력합니다.
5. [Publish Message]를 선택합니다.
6. 해당 메시지를 받았는지 이메일을 확인합니다.

## AWS CLI를 사용하여 SNS 주제 설정

먼저 SNS 주제를 설정한 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.

# CloudWatch 지표를 기반으로 CloudWatch 경고 생성

감시할 경고에 대한 CloudWatch 지표를 선택하고 해당 지표에 대한 임계값을 선택할 수 있습니다. 지표가 지정된 수의 평가 기간에 대한 임계값을 위반할 경우 경고가 ALARM 상태가 됩니다.

단일 지표를 기반으로 경보를 생성하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.
3. 지표 선택을 선택하고 다음 중 하나를 수행합니다.
  - a. 원하는 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 원하는 지표 옆에 있는 확인란을 선택합니다.
  - 또는
  - b. 검색 상자에 지표 이름, 치수 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 그런 다음 결과 중 하나를 선택하고 지표 목록이 표시될 때까지 계속 진행합니다. 원하는 지표 옆에 있는 확인란을 선택합니다.
4. [Graphed metrics] 탭을 선택합니다.
  - a. 통계에서 통계 또는 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: p95.45)를 지정합니다.
  - b. 기간에서 경고에 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. 경보의 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다.
6. 다음 경우 항상에서 경고 조건을 지정합니다.
  - a. is:에서 지표가 임계값보다 크거나 작아야 하는지 아니면 임계값과 같아야 하는지 지정하고 임계값을 지정합니다.
  - b. for:에서 경보를 트리거하기 위해서는 ALARM 상태인 평가 기간(데이터 포인트)이 몇 개가 있어야 하는지 지정합니다. 처음에는 두 번째 값만 변경할 수 있고 첫 번째 값은 입력한 두 번째 값에 맞춰 바뀝니다. 그러면 다수의 연속 기간이 위반되면 ALARM 상태가 되는 경보가 생성됩니다.

N 중 M 경보를 생성하려면 연필 아이콘을 선택합니다. 그런 다음 M 숫자를 N 숫자와 다르게 변경할 수 있습니다. 자세한 정보는 [경보 평가 \(p. 49\)](#) 단원을 참조하십시오.
7. 추가 설정의 누락 데이터 처리에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 정보는 [CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 \(p. 50\)](#) 단원을 참조하십시오.
8. 경보가 모니터링된 통계값으로 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 경고 상태가 항상 유지됩니다. 자세한 정보는 [백분위수 기반 CloudWatch 경고 및 데이터 샘플 부족 \(p. 53\)](#) 단원을 참조하십시오.
9. [Actions]에서 경고 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. [Create Alarm]을 선택합니다.

대시보드에 경보를 추가할 수도 있습니다. 자세한 정보는 [CloudWatch 대시보드에서 경보를 추가 또는 제거 \(p. 23\)](#) 단원을 참조하십시오.

## 지표 수학 표현식을 기반으로 CloudWatch 경고 생성

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

수학 표현식을 기반으로 경보를 생성하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.
3. 지표 선택을 선택하고 다음 중 하나를 수행합니다.
  - a. 특정 지표가 포함된 서비스 네임스페이스를 선택합니다. 옵션이 나타날 때 계속해서 옵션을 선택하면 선택 범위가 좁아집니다. 지표 목록이 표시되면 원하는 오른쪽 지표 옆에 있는 확인란을 선택합니다.
  - 또는
  - b. 검색 상자에 지표 이름, 치수 또는 리소스 ID를 입력하고 Enter 키를 누릅니다. 그런 다음 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. [Graphed metrics]를 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]를 선택합니다. 표현식을 표시하는 새 행이 나타납니다.
6. 새 행의 세부 정보 옆에 수학 표현식을 입력하고 Enter 키를 누릅니다. 사용 가능한 함수 및 구문에 대한 자세한 정보는 [지표 수학 구문 및 함수 \(p. 45\)](#) 단원을 참조하십시오.

지표를 사용하거나 이 표현식을 위한 공식의 일부로 다른 표현식 결과를 사용하려면 Id 옆에 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. 경보의 이름 및 설명을 입력합니다. 이름은 ASCII 문자만 포함해야 합니다.
10. 다음 경우 항상에서 경보 조건을 지정합니다.
  - a. is:에서 표현식 결과가 임계값보다 크거나 작아야 하는지 아니면 임계값과 같아야 하는지 지정하고 임계값을 지정합니다.



- b. for:에서 경보를 트리거하기 위해서는 ALARM 상태인 평가 기간(데이터 포인트)이 몇 개가 있어야 하는지 지정합니다. 처음에는 두 번째 값만 변경할 수 있고 첫 번째 값은 입력한 두 번째 값에 맞춰 바뀝니다. 그러면 다수의 연속 기간이 위반되면 ALARM 상태가 되는 경보가 생성됩니다.

N 중 M 경보를 생성하려면 연필 아이콘을 선택합니다. 그런 다음 M 숫자를 N 숫자와 다르게 변경할 수 있습니다. 자세한 정보는 [경보 평가 \(p. 49\)](#) 단원을 참조하십시오.

11. 추가 설정의 누락 데이터 처리에서 일부 데이터 포인트가 누락된 경우 경보가 어떻게 동작할지 선택합니다. 자세한 정보는 [CloudWatch 경보가 누락 데이터를 처리하는 방법 구성 \(p. 50\)](#) 단원을 참조하십시오.
12. [Actions]에서 경보 트리거 시 수행할 작업의 유형을 선택합니다. 경보가 여러 작업을 수행하도록 하려면 +알림 또는 +AutoScaling 작업을 선택합니다. 작업을 한 가지 이상 지정합니다.
13. [Create Alarm]을 선택합니다.

대시보드에 경보를 추가할 수도 있습니다. 자세한 정보는 [CloudWatch 대시보드에서 경보를 추가 또는 제거 \(p. 23\)](#) 단원을 참조하십시오.

## CloudWatch 경보 편집

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

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 설명을 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/v2/home>에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 [Topics]를 선택한 다음, 알림 목록(주제)에 대한 ARN을 선택합니다.
3. 다음 중 하나를 수행하십시오.
  - 이메일 주소를 추가하려면 [Create subscription]을 선택합니다. Protocol의 경우 Email을 선택합니다. [Endpoint]에 새 수신자의 이메일 주소를 입력합니다. Create subscription을 선택합니다.
  - 이메일 주소를 제거하려면 [Subscription ID]를 선택합니다. [Other subscription actions]와 [Delete subscriptions]를 선택합니다.
4. [Publish to topic]을 선택합니다.

## CPU 사용률을 기반으로 이메일을 전송하는 경보를 생성

경보 상태가 OK에서 ALARM으로 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 ALARM으로 바뀝니다.

## AWS Management 콘솔을 사용하여 CPU 사용률 경보를 설정

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

CPU 사용률을 기반으로 이메일을 보내는 경보를 만들려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.
3. [EC2 Metrics]에서 지표 범주(예: [Per-Instance Metrics])를 선택합니다.
4. 아래와 같이 지표를 선택합니다.
  - a. 해당 인스턴스와 [CPUUtilization] 지표가 있는 행을 선택합니다.
  - b. 통계의 경우 [Average]를 선택하거나, 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: p95.45)를 지정합니다.
  - c. 기간(예: **5 minutes**)을 선택합니다.
  - d. [Next]를 선택합니다.

The screenshot shows the 'Create Alarm' wizard in the AWS Management Console. It is at the '1. Select Metric' step. The 'EC2' namespace is selected, and 'Per-Instance Metrics' is chosen. A table lists metrics for instance i-0332c3c79f97a3e63, with 'CPUUtilization' selected. Below the table, a graph shows 'CPUUtilization' over time, with 'Average' and '5 Minutes' selected. The 'Time Range' is set to 'Relative' from 3 days ago to 0 hours ago. The 'Left Y-axis' is set to 'Limits' with 'Min' at 0 and 'Max' at Auto. The 'Next' button is highlighted.

5. 다음과 같이 경보를 정의합니다:

- [Alarm Threshold]에 경보의 고유 이름(예: myHighCpuAlarm)과 경보에 대한 설명(예: CPU 사용률이 70%를 초과)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.
- Whenever의 is에서 >를 선택하고 70을 입력합니다. 기간에 2를 입력합니다. 이는 CPU 사용량이 2회의 샘플링 기간 연속으로 70% 이상인 경우 트리거되는 경보를 지정합니다.

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

Description:

Whenever: CPUUtilization

is:

for:  consecutive period(s)

- 누락 데이터 요소가 인스턴스 다운을 나타내면 추가 설정 아래의 누락 데이터 처리에서 bad(breaching threshold)(불량(임계값 위반))를 선택합니다.
- [Actions]의 [Whenever this alarm]에서 [State is ALARM]을 선택합니다. [Send notification to]에 대해 기존 SNS 주제를 선택하거나 새로 만듭니다.

### Actions

Define what actions are taken when your alarm changes state.

Notification Delete

Whenever this alarm:

Send notification to:  New list Enter list ⓘ

Email list:

- SNS 주제를 새로 생성하려면 [New list]를 선택합니다. [Send notification to]에 SNS 주제 이름(예: myHighCpuAlarm)을 입력하고 경보가 ALARM 상태로 변경될 때 알림을 보낼 이메일 주소 목록을 선택표로 구분하여 [Email list]에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 알림을 받으려면 먼저 구독을 설정해야 합니다.
- [Create Alarm]을 선택합니다.

## AWS CLI를 사용하여 CPU 사용률 경보를 설정

다음 단계에 따라 AWS CLI를 사용해 CPU 사용량 경보를 만듭니다.

CPU 사용률을 기반으로 이메일을 보내는 경보를 만들려면

- SNS 주제를 설정합니다. 자세한 정보는 [Amazon SNS 알림 설정 \(p. 53\)](#) 단원을 참조하십시오.
- 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
```

- `set-alarm-state` 명령으로 경보 상태를 강제로 변경하여 경보를 테스트합니다.
  - 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 ALARM로 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 경보를 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 만듭니다.

이메일을 전송하는 로드 밸런서 지연 경보를 생성하려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.
3. [CloudWatch Metrics by Category]에서 [ELB Metrics] 범주를 선택합니다.
4. Classic Load Balancer와 Latency 지표가 있는 행을 선택합니다.
5. 통계의 경우 [Average]를 선택하거나, 사전 정의된 백분위수 중 하나를 선택하거나, 사용자 지정 백분위수(예: p95.45)를 지정합니다.
6. 기간의 경우 [1 Minute]를 선택합니다.
7. [Next]를 선택합니다.
8. 경보 임계값에 경보의 고유 이름(예: **myHighCpuAlarm**)과 경보에 대한 설명(예: 지연 시간이 100s를 초과할 때 경보)을 입력합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.
9. [Whenever]의 [is]에서 [>]를 선택하고 0.1을 입력합니다. [for]에 3을 입력합니다.
10. 누락 데이터 요소가 경보 상태 변경을 트리거하지 않도록 [Additional settings]의 [Treat missing data as]에서 [ignore (maintain alarm state)]를 선택합니다.

경보가 적정 수의 데이터 샘플이 있는 상황만 평가하도록 샘플이 부족한 백분위수에서 무시(경보 상태 유지)를 선택합니다.

11. [Actions]의 [Whenever this alarm]에서 [State is ALARM]을 선택합니다. [Send notification to]에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

SNS 주제를 생성하려면 [New list]를 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. [Create Alarm]을 선택합니다.

## AWS CLI를 사용하여 지연 시간 경보 설정

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

이메일을 전송하는 로드 밸런서 지연 경보를 생성하려면

1. SNS 주제를 설정합니다. 자세한 정보는 [Amazon SNS 알림 설정 \(p. 53\)](#) 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 경고 상태를 강제로 변경하여 경보를 테스트합니다.
  - a. 경고 상태를 `INSUFFICIENT_DATA`에서 `OK`로 변경합니다.

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

- b. 경고 상태를 `OK`에서 `ALARM`로 변경합니다.

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

- c. 경고에 대한 이메일 알림을 받았음을 확인합니다.

## 이메일을 전송하는 스토리지 처리량 경고 생성

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

## AWS Management 콘솔을 사용하여 스토리지 처리량 경고 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 [Alarms], [Create Alarm]을 선택합니다.
3. [EBS Metrics]에서 지표 범주를 선택합니다.
4. 볼륨과 [VolumeWriteBytes] 지표가 있는 행을 선택합니다.
5. 통계의 경우 [Average]를 선택합니다. 기간의 경우 [5 Minutes]를 선택합니다. [Next]를 선택합니다.
6. [Alarm Threshold]에 경보의 고유 이름(예: myHighWriteAlarm)과 경보에 대한 설명(예: VolumeWriteBytes가 100,000KiB/s를 초과)을 입력합니다. 경고 이름은 ASCII 문자만 포함해야 합니다.
7. Whenever의 is에서 >를 선택하고 100000을 입력합니다. 기간에 연속 기간으로 15를 입력합니다.

[Alarm Preview] 아래에 임계값이 그래픽으로 표시됩니다.

8. 누락 데이터 요소가 경고 상태 변경을 트리거하지 않도록 [Additional settings]의 [Treat missing data as]에서 [ignore (maintain alarm state)]를 선택합니다.
9. [Actions]의 [Whenever this alarm]에서 [State is ALARM]을 선택합니다. [Send notification to]에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

SNS 주제를 생성하려면 [New list]를 선택합니다. [Send notification to]에 SNS 주제 이름(예: myHighCpuAlarm)을 입력하고 경보가 ALARM 상태로 변경될 때 알림을 보낼 이메일 주소 목록을 쉼표로 구분하여 [Email list]에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.

10. [Create Alarm]을 선택합니다.

## AWS CLI를 사용하여 스토리지 처리량 경고 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 정보는 [Amazon SNS 알림 설정 \(p. 53\)](#) 단원을 참조하십시오.
2. 경보를 만듭니다.

```
aws cloudwatch put-metric-alarm --alarm-name efs-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 efs-mon --state-reason "initializing" --state-value OK
```

- b. 경보 상태를 OK에서 ALARM로 변경합니다.

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

- c. 경보 상태를 ALARM에서 INSUFFICIENT\_DATA로 변경합니다.

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

- d. 경보에 대한 이메일 알림을 받았음을 확인합니다.

## 인스턴스를 중지, 종료, 재부팅 또는 복구하는 경고 생성

Amazon CloudWatch 경고 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)를 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, AWSServiceRoleForCloudWatchEvents를 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 생성](#)을 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 정보는 IAM 사용 설명서에서 [권한 및 정책](#)을 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 \(p. 65\)](#)
- [Amazon CloudWatch 경보에 종료 작업 추가 \(p. 65\)](#)
- [Amazon CloudWatch 경보에 재부팅 작업 추가 \(p. 66\)](#)
- [Amazon CloudWatch 경보에 복구 작업 추가 \(p. 67\)](#)
- [트리거된 경보 및 작업 기록 보기 \(p. 68\)](#)

## 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 [Alarms], [Create Alarm]을 선택합니다.
3. [Select Metric] 단계에서 다음을 수행합니다.
  - a. [EC2 Metrics]에서 [Per-Instance Metrics]를 선택합니다.
  - b. 해당 인스턴스와 [CPUUtilization] 지표가 있는 행을 선택합니다.
  - c. 통계의 경우 [Average]를 선택합니다.
  - d. 기간(예: **1 Hour**)을 선택합니다.
  - e. [Next]를 선택합니다.
4. [Define Alarm] 단계에서 다음을 수행합니다.
  - a. [Alarm Threshold]에 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.  
  
[Alarm Preview] 아래에 임계값이 그래픽으로 표시됩니다.
  - c. [Notification]의 [Send notification to]에서 기존 SNS 주제를 선택하거나 새로 만듭니다.  
  
SNS 주제를 생성하려면 [New list]를 선택합니다. 알림 보내기에 SNS 주제의 이름을 입력합니다 (예: Stop\_EC2\_Instance). **ALARM** 상태로 경보가 변경되면 알림 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 Action]을 선택합니다.
  - e. [Whenever this alarm]에 [State is ALARM]을 선택합니다. [Take this action]에서 [Stop this instance]를 선택합니다.
  - f. [Create Alarm]을 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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [인스턴스에 대한 종료 보호 활성화](#)를 참조하십시오.

Amazon CloudWatch 콘솔을 사용하여 유휴 인스턴스를 종료하는 경보를 만들려면

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Alarms], [Create Alarm]을 선택합니다.



3. [Select Metric] 단계에서 다음을 수행합니다.
  - a. [EC2 Metrics]에서 [Per-Instance Metrics]를 선택합니다.
  - b. 해당 인스턴스와 [CPUUtilization] 지표가 있는 행을 선택합니다.
  - c. 통계의 경우 [Average]를 선택합니다.
  - d. 기간(예: **1 Hour**)을 선택합니다.
  - e. [Next]를 선택합니다.
4. [Define Alarm] 단계에서 다음을 수행합니다.
  - a. [Alarm Threshold]에 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.  
  
[Alarm Preview] 아래에 임계값이 그래픽으로 표시됩니다.
  - c. [Notification]의 [Send notification to]에서 기존 SNS 주제를 선택하거나 새로 만듭니다.  
  
SNS 주제를 생성하려면 [New list]를 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 Action]을 선택합니다.
  - e. [Whenever this alarm]에 [State is ALARM]을 선택합니다. [Take this action]에서 [Terminate this instance]를 선택합니다.
  - f. [Create Alarm]을 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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서에서 [인스턴스 재부팅](#)을 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 [Alarms], [Create Alarm]을 선택합니다.
3. [Select Metric] 단계에서 다음을 수행합니다.
  - a. [EC2 Metrics]에서 [Per-Instance Metrics]를 선택합니다.
  - b. 해당 인스턴스와 [StatusCheckFailed\_Instance] 지표가 있는 행을 선택합니다.
  - c. 통계의 경우 [Minimum]을 선택합니다.



- d. 기간(예: **1 Minute**)을 선택하고 다음을 선택합니다.
4. [Define Alarm] 단계에서 다음을 수행합니다.
  - a. [Alarm Threshold]에 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 0을 입력합니다. 기간에 연속 기간으로 3를 입력합니다.  
  
[Alarm Preview] 아래에 임계값이 그래픽으로 표시됩니다.
  - c. [Notification]의 [Send notification to]에서 기존 SNS 주제를 선택하거나 새로 만듭니다.  
  
SNS 주제를 생성하려면 [New list]를 선택합니다. 알림 보내기에 SNS 주제의 이름을 입력합니다 (예: Reboot\_EC2\_Instance). ALARM 상태로 경보가 변경되면 알릴 이메일 주소를 심표로 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 Action]을 선택합니다.
  - e. [Whenever this alarm]에 [State is ALARM]을 선택합니다. [Take this action]에서 [Reboot this instance]를 선택합니다.
  - f. [Create Alarm]을 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 구성된 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\_Instance가 아닌 StatusCheckFailed\_System을 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, 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 [Alarms], [Create Alarm]을 선택합니다.
3. [Select Metric] 단계에서 다음을 수행합니다.
  - a. [EC2 Metrics]에서 [Per-Instance Metrics]를 선택합니다.
  - b. 해당 인스턴스와 [StatusCheckFailed\_System] 지표가 있는 행을 선택합니다.
  - c. 통계의 경우 [Minimum]을 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. [Next]를 선택합니다.
4. [Define Alarm] 단계에서 다음을 수행합니다.
    - a. [Alarm Threshold]에 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 0을 입력합니다. 기간에 연속 기간으로 2를 입력합니다.
    - c. [Notification]의 [Send notification to]에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

SNS 주제를 생성하려면 [New list]를 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 Action]을 선택합니다.
    - e. [Whenever this alarm]에 [State is ALARM]을 선택합니다. [Take this action]에서 [Recover this instance]를 선택합니다.
    - f. [Create Alarm]을 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 [Details]를 선택합니다.
4. 최신 기록 항목을 보려면 내역을 선택합니다.

## 예상 AWS 요금을 모니터링하기 위한 결제 경보를 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. 결제 지표 데이터에는 사용한 AWS의 모든 서비스에 대한 예상 요금과 전반적인 총 AWS 예상 요금이 들어 있습니다.

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

사용자의 요금이 임계값을 초과한 시점에 결제 알림을 생성할 경우 경보가 즉시 ALARM 상태가 됩니다.

### 작업

- 결제 경고 활성화 (p. 69)
- 결제 경고 만들기 (p. 70)
- 경고 상태 확인 (p. 71)
- 결제 경고 삭제 (p. 71)

## 결제 경고 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

- AWS 계정 루트 사용자 자격 증명으로 로그인을 해야 합니다. IAM 사용자는 AWS 계정에 대해 결제 알림을 활성화할 수 없습니다.
- 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 비활성화하려면

1. <https://console.aws.amazon.com/billing/home?#>에서 Billing and Cost Management 콘솔을 엽니다.
2. 탐색 창에서 [Preferences]를 선택합니다.
3. [Receive Billing Alerts]를 선택합니다.

Dashboard  
Bills  
Cost Explorer  
Budgets  
Reports  
Cost Allocation Tags  
Payment Methods  
Payment History  
Consolidated Billing  
**Preferences**  
Credits  
Tax Settings  
DevPay

### Preferences

☐ **Receive PDF Invoice By Email**  
Turn on this feature to receive a PDF version of your invoice by email. Invoices are generally available within the first three days of the month.

☒ **Receive Billing Alerts**  
Turn on this feature to monitor your AWS usage charges and recurring fees automatically, making it easier to track and manage your spending on AWS. You can set up billing alerts to receive email notifications when your charges reach a specified threshold. Once enabled, this preference cannot be disabled. [Manage Billing Alerts](#)

☐ **Receive Billing Reports**  
Turn on this feature to receive ongoing reports of your AWS charges once or more daily. AWS delivers these reports to the Amazon S3 bucket that you specify where indicated below. For consolidated billing customers, AWS generates reports only for paying accounts. Linked accounts cannot sign up for billing reports.

Save to S3 Bucket:

4. Save preferences를 선택합니다.

## 결제 경고 만들기

결제 경보를 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 정보는 CloudWatch를 사용하여 예상 요금 모니터링의 [결제 경고 생성 \(p. 167\)](#)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 [Alarms], [Billing], [Create Alarm]를 차례로 선택합니다.
4. [show advanced]를 선택하여 고급 옵션으로 전환합니다.
5. [Alarm Threshold]에서 경보의 기본 이름(예: My Estimated Charges)과 경보에 대한 설명(예: 예상 월별 요금)을 변경합니다. 경보 이름은 ASCII 문자만 포함해야 합니다.
6. [Whenever charges for]의 [is]에서 [>=]를 선택하고 경보를 트리거하고 이메일을 전송하기 위해 초과되어야 할 금액(예: 200)을 입력합니다.

### Note

[Alarm Preview]에는 걱정 금액을 설정하는 데 사용할 수 있는 예상 요금이 나와 있습니다.

7. 누락 데이터 요소가 경보 상태 변경을 트리거하지 않도록 [Additional settings]의 [Treat missing data as]에서 [ignore (maintain alarm state)]를 선택합니다.
8. [Actions]의 [Whenever this alarm]에서 [State is ALARM]을 선택합니다. [Send notification to]에서 기존 SNS 주제를 선택하거나 새로 만듭니다.

SNS 주제를 생성하려면 [New list]를 선택합니다. [Send notification to]에 SNS 주제 이름을 입력하고 이메일 알림을 전송할 이메일 주소 목록을 쉼표로 구분하여 [Email list]에 입력합니다. 각 이메일 주소로 주제 구독 확인 이메일이 전송됩니다. 수신자가 구독을 확인해야만 이 이메일 주소로 알림이 전송될 수 있습니다.

9. [Create Alarm]을 선택합니다.

## 경보 상태 확인

결제 경보의 상태를 확인할 수 있습니다.

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 [Alarms]와 [Billing]을 선택합니다.
4. 경보 옆의 확인란을 선택합니다. 구독이 확인되기 전까지 "확인 보류 중"으로 표시가 됩니다. 구독 확인 후에 콘솔을 새로 고쳐 업데이트된 상태를 보여줍니다.

## 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 [Alarms]와 [Billing]을 선택합니다.
4. 경보 옆의 확인란을 선택한 후 삭제를 선택합니다.
5. 확인 메시지가 나타나면 예, 삭제합니다를 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.

# CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집

통합 CloudWatch 에이전트를 사용하면 다음을 수행할 수 있습니다.

- EC2 인스턴스 지표뿐만 아니라 인게스트 지표를 포함하여 Amazon EC2 인스턴스로부터 더 많은 시스템 수준 지표를 수집할 수 있습니다. 추가 지표는 [CloudWatch 에이전트가 수집하는 지표 \(p. 144\)](#) 섹션에 나열되어 있습니다.
- 온프레미스 서버로부터 시스템 수준 지표를 수집합니다. 여기에는 AWS가 관리하지 않는 서버뿐만 아니라 하이브리드 환경의 서버도 포함될 수 있습니다.
- Linux 또는 Windows Server를 실행하는 Amazon EC2 인스턴스 및 온프레미스 서버로부터 로그를 수집합니다.
- StatsD 및 collectd 프로토콜을 사용하여 애플리케이션 또는 서비스에서 사용자 지정 지표를 검색하십시오. StatsD는 Linux 서버와 Windows Server를 실행하는 서버에서 모두 지원됩니다. collectd는 Linux 서버에서만 지원됩니다.

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 CloudWatch Logs에서 처리되고 저장됩니다. CloudWatch Logs 요금에 대한 자세한 정보는 [Amazon CloudWatch 요금](#)을 참조하십시오.

CloudWatch 에이전트에 의해 수집되는 지표는 사용자 지정 지표로 청구됩니다. CloudWatch 지표 요금에 대한 자세한 정보는 [Amazon CloudWatch 요금](#)을 참조하십시오.

이 단원의 단계에서는 Amazon EC2 인스턴스 및 온프레미스 서버에 통합 CloudWatch 에이전트를 설치하는 방법을 설명합니다. CloudWatch 에이전트에 의해 수집될 수 있는 지표에 대한 자세한 정보는 [CloudWatch 에이전트가 수집하는 지표 \(p. 144\)](#)를 참조하십시오.

지원되는 운영 체제

CloudWatch 에이전트는 다음 운영 체제에서 지원됩니다.

- Amazon Linux 버전 2014.03.02 이상
- Amazon Linux 2
- Ubuntu Server 버전 18.04, 16.04, 14.04
- CentOS 버전 7.0 및 6.5
- Red Hat Enterprise Linux(RHEL) 버전 7.5, 7.4, 7.0 및 6.5
- Debian 8.0
- SUSE Linux Enterprise Server(SLES) 12 이상
- Windows Server 2016, Windows Server 2012 및 Windows Server 2008의 64비트 버전.

## 설치 프로세스 개요

CloudWatch 에이전트를 설치하는 일반적인 흐름은 다음과 같습니다.

1. CloudWatch 에이전트에 필요한 IAM 역할 및 사용자를 생성합니다. 이 역할 및 사용자는 CloudWatch가 서버로부터 지표를 수집하고 AWS 시스템 관리자와 통합하도록 해줍니다.
2. Amazon EC2 인스턴스에 설치하는 경우, IAM 역할을 인스턴스에 연결합니다. 온프레미스 서버에 설치하는 경우, CloudWatch 에이전트가 CloudWatch에 정보를 기록할 수 있도록 IAM 사용자를 생성합니다.
3. AWS 시스템 관리자 Run Command 또는 퍼블릭 Amazon S3 다운로드 링크 중 하나를 사용하여 에이전트 패키지를 다운로드합니다.
4. CloudWatch 에이전트 구성 파일을 수정하고 CloudWatch 에이전트용으로 명명된 프로필을 생성하십시오. Amazon EC2 인스턴스에 에이전트를 설치하는 경우에는 명명된 프로필 생성이 선택 사항입니다.
5. 시스템 관리자 Run Command 또는 명령줄을 사용하여 에이전트를 시작합니다.

## 목차

- [CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 \(p. 73\)](#)
- [Amazon EC2 인스턴스에 CloudWatch 에이전트 설치 \(p. 76\)](#)
- [온프레미스 서버에 CloudWatch 에이전트 설치 \(p. 89\)](#)
- [AWS CloudFormation을 사용하여 새 인스턴스에 CloudWatch 에이전트 설치 \(p. 102\)](#)
- [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#)
- [procstat 플러그인을 사용하여 프로세스 지표 수집 \(p. 128\)](#)
- [StatsD로 시작하는 사용자 지정 지표 검색 \(p. 136\)](#)
- [collectd로 사용자 지정 지표 검색 \(p. 137\)](#)
- [CloudWatch 에이전트를 사용하는 일반적인 시나리오 \(p. 138\)](#)
- [CloudWatch 에이전트가 수집하는 지표 \(p. 144\)](#)
- [CloudWatch 에이전트의 문제 해결 \(p. 151\)](#)

# 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 들어 있는 IAM 역할 및 사용자를 생성할 수 있습니다. Amazon EC2 인스턴스에서 IAM 역할을 사용하고 온프레미스 서버에서 IAM 사용자를 사용하여 에이전트가 CloudWatch에 데이터를 전송하도록 할 수 있습니다.

하나의 역할과 하나의 사용자를 사용하면 CloudWatch 에이전트가 서버에 설치되어 CloudWatch에 지표를 보낼 수 있습니다. CloudWatch 에이전트 구성을 시스템 관리자 Parameter Store에 저장하려면 다른 역할 또는 사용자가 필요합니다. 이는 여러 서버가 하나의 CloudWatch 에이전트 구성을 사용하도록 해줍니다.

Parameter Store에 쓰는 기능은 광범위하고 강력한 권한이며, 필요 시에만 사용해야 하고, 배포 시 여러 인스턴스에 연결해선 안 됩니다. CloudWatch 에이전트 구성을 Parameter Store에 저장하려는 경우, 이 구성을 수행하는 하나의 인스턴스를 설정해야 하며, 이 인스턴스에서만 그리고 CloudWatch 에이전트 구성 파일을 사용하고 저장하는 동안에만 Parameter Store에 대한 쓰기 권한을 갖는 IAM 역할을 사용해야 합니다.

## Note

최근에 당사는 고객들에게 정책을 직접 만들도록 요구하는 대신 Amazon에서 만든 새로운 CloudWatchAgentServerPolicy 및 CloudWatchAgentAdminPolicy 정책을 사용하여 다음과 같은 절차를 수정했습니다. Parameter Store에 파일 쓰기 및 다운로드는 Amazon에서 생성



된 정책에 따라 이름이 "AmazonCloudWatch-"로 시작하는 파일만 지원합니다. 파일 이름이 AmazonCloudWatch-로 시작하지 않는 CloudWatch 에이전트 구성 파일이 있는 경우 Parameter Store에 파일 쓰기 또는 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 에이전트 구성 파일을 시스템 관리자 Parameter Store에 저장하여 다른 서버가 사용할 수 있도록 하려는 경우 CloudWatch 에이전트 구성 파일을 생성하는 데 사용되는 Amazon EC2 인스턴스에 연결하는 데 필요한 IAM 역할을 생성합니다. 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. 왼쪽의 탐색 창에서 [Roles], [Create role]을 선택합니다.
3. [Choose the service that will use this role]의 경우 [EC2 Allows EC2 instances to call AWS services on your behalf]를 선택합니다. 다음: 권한을 선택합니다.
4. 정책 목록에서 [CloudWatchAgentServerPolicy] 옆의 확인란을 선택합니다. 필요하다면 검색 상자를 사용하여 정책을 찾습니다.
5. SSM을 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonEC2RoleforSSM 옆의 확인란을 선택하십시오. 필요하다면 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
6. [Next: Review]를 선택합니다.
7. [CloudWatchAgentServerPolicy] 및 [AmazonEC2RoleforSSM]이 [Policies] 옆에 표시되는지 확인하십시오. [Role name]에 역할 이름(예: CloudWatchAgentServerRole)을 입력합니다. 필요한 경우 설명을 입력하고 [Create role]을 선택합니다.

이제 역할이 생성되었습니다.

다음 절차에서는 Parameter Store에 대한 쓰기 권한도 있는 IAM 역할을 생성합니다. 에이전트 구성 파일을 Parameter Store에 저장하여 다른 서버가 사용할 수 있도록 하려는 경우 이 역할을 사용해야 합니다. 이 역할은 인스턴스의 정보를 읽고 이를 CloudWatch에 쓰는 권한뿐만 아니라 Parameter Store에 쓰는 권한도 제공합니다. Parameter Store에 대한 쓰기 권한은 광범위한 기능을 제공하므로 일부 서버에만 연결해야 하며 관리자만 사용해야 합니다. 에이전트 구성 파일 생성 및 이를 Parameter Store에 복사하는 작업을 마친 이후 이 역할을 인스턴스로부터 분리하고 대신 CloudWatchAgentServerPolicy를 사용해야 합니다.

관리자가 에이전트 구성 파일을 시스템 관리자 Parameter Store에 저장하는 데 필요한 IAM 역할을 만들려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 [Roles], [Create role]을 선택합니다.
3. [Choose the service that will use this role]의 경우 [EC2 Allows EC2 instances to call AWS services on your behalf]를 선택합니다. 다음: 권한을 선택합니다.
4. 정책 목록에서 [CloudWatchAgentAdminPolicy] 옆의 확인란을 선택합니다. 필요하다면 검색 상자를 사용하여 정책을 찾습니다.



5. SSM을 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonEC2RoleforSSM 옆의 확인란을 선택하십시오. 필요하다면 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
6. [Next: Review]를 선택합니다.
7. [CloudWatchAgentAdminPolicy] 및 [AmazonEC2RoleforSSM]이 [Policies] 옆에 표시되는지 확인하십시오. [Role name]에 역할 이름(예: CloudWatchAgentAdminRole)을 입력합니다. 필요한 경우 설명을 입력하고 [Create role]을 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 에이전트 구성 파일을 시스템 관리자 Parameter Store에 저장하여 다른 서버가 사용할 수 있도록 하려는 경우 CloudWatch 에이전트 구성 파일을 생성할 때 사용할 수 있는 IAM 사용자를 생성합니다. 이 사용자는 Parameter Store에 대한 쓰기 권한과 함께 CloudWatch에 대한 데이터 쓰기 권한을 제공합니다.

CloudWatch 에이전트가 CloudWatch에 데이터 쓰기를 수행하는 데 필요한 IAM 사용자를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창 왼쪽에서 [Users]를 선택한 다음 [Add Users]를 선택합니다.
3. 신규 사용자의 사용자 이름을 입력합니다.
4. [Programmatic access]를 선택하고 [Next: Permissions]를 선택합니다.
5. [Attach existing policies directly]를 선택합니다.
6. 정책 목록에서 [CloudWatchAgentServerPolicy] 옆의 확인란을 선택합니다. 필요하다면 검색 상자를 사용하여 정책을 찾습니다.
7. SSM을 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonEC2RoleforSSM 옆의 확인란을 선택하십시오. 필요하다면 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
8. [Next: Review]를 선택합니다.
9. 올바른 정책이 나열되는지 확인하고 [Create user]를 선택합니다.
10. 새 사용자 이름 옆에서 [Show]를 선택합니다. 에이전트를 설치할 때 사용할 수 있도록 액세스 키 및 보안 키를 파일에 복사한 다음, [Close]를 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 사용자를 사용해야 합니다. 이 사용자는 인스턴스의 정보를 읽고 이를 CloudWatch에 쓰는 권한뿐만 아니라 Parameter Store에 쓰는 권한도 제공합니다. 시스템 관리자 Parameter Store에 대한 쓰기 권한은 광범위한 기능을 제공하므로 일부 서버에만 연결해야 하며 관리자만 사용해야 합니다. 에이전트 구성 파일을 에 저장하는 경우에만 이 Parameter Store 사용자를 IAM사용해야 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 창 왼쪽에서 [Users]를 선택한 다음 [Add Users]를 선택합니다.
3. 신규 사용자의 사용자 이름을 입력합니다.
4. [Programmatic access]를 선택하고 [Next: Permissions]를 선택합니다.
5. [Attach existing policies directly]를 선택합니다.
6. 정책 목록에서 [CloudWatchAgentAdminPolicy] 옆의 확인란을 선택합니다. 필요하다면 검색 상자를 사용하여 정책을 찾습니다.
7. SSM을 사용하여 CloudWatch 에이전트를 설치하거나 구성하려면 AmazonEC2RoleforSSM 옆의 확인란을 선택하십시오. 필요하다면 검색 상자를 사용하여 정책을 찾습니다. 명령줄을 통해서만 에이전트를 시작하고 구성하는 경우에는 이 정책이 필요하지 않습니다.
8. [Next: Review]를 선택합니다.
9. 올바른 정책이 나열되는지 확인하고 [Create user]를 선택합니다.
10. 새 사용자 이름 옆에서 [Show]를 선택합니다. 에이전트를 설치할 때 사용할 수 있도록 액세스 키 및 보안 키를 파일에 복사한 다음, [Close]를 선택합니다.

## Amazon EC2 인스턴스에 CloudWatch 에이전트 설치

처음으로 CloudWatch 에이전트 사용을 시작하는 경우, 이를 서버에 다운로드하고 에이전트를 구성해야 합니다. 그러면, 해당 구성이 있는 에이전트를 해당 서버에서 직접 사용할 수 있으며, 구성을 AWS 시스템 관리자 Parameter Store에 저장하는 경우에는, 다른 서버에 CloudWatch 에이전트를 설치할 때도 동일한 구성을 사용할 수 있습니다.

### 항목

- [시작하기: 첫 번째 인스턴스에 CloudWatch 에이전트 설치 \(p. 76\)](#)
- [에이전트 구성을 사용하여 추가 인스턴스에 CloudWatch 에이전트 설치 \(p. 83\)](#)

## 시작하기: 첫 번째 인스턴스에 CloudWatch 에이전트 설치

실행 중인 Amazon EC2 인스턴스에 CloudWatch 에이전트를 다운로드하고 설치하려면 AWS 시스템 관리자 또는 명령줄을 사용하면 됩니다. 어떤 방법을 사용하든, 먼저 IAM 역할을 생성하고 이를 인스턴스에 연결해야 합니다.

### 항목

- [IAM 역할을 인스턴스에 연결 \(p. 76\)](#)
- [Amazon EC2 인스턴스에 CloudWatch 에이전트 패키지 다운로드 \(p. 77\)](#)
- [\(선택 사항\) CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정 \(p. 80\)](#)
- [첫 번째 인스턴스에 에이전트 구성 파일 생성 \(p. 81\)](#)
- [CloudWatch 에이전트 시작 \(p. 81\)](#)

## IAM 역할을 인스턴스에 연결

CloudWatch 에이전트를 Amazon EC2 인스턴스에 설치하는 경우 인스턴스 프로파일용 IAM 역할이 필요합니다. 이 역할은 CloudWatch 에이전트가 인스턴스에 대한 작업을 수행하도록 해줍니다. 앞부분에서 생성한 역할 중 하나를 사용하십시오. 이 역할 생성에 대한 자세한 정보는 [CloudWatch 에이전트와 함께 사용하기 위](#)

[한 IAM 역할 및 사용자 생성 \(p. 73\)](#) 섹션을 참조하십시오. 목록을 스크롤하여 역할을 찾거나 검색 상자를 사용해도 됩니다.

이 인스턴스를 사용하여 CloudWatch 에이전트 구성 파일을 생성하고 이를 시스템 관리자 Parameter Store에 복사하려는 경우, 앞에서 생성한 Parameter Store에 대한 쓰기 권한이 있는 역할을 사용하십시오. 이 역할을 CloudWatchAgentAdminRole이라고 합니다.

다른 모든 인스턴스의 경우, 에이전트를 설치 및 실행하는 데 필요한 권한만 들어 있는 역할을 생성하십시오. 이 역할을 CloudWatchAgentServerRole이라고 합니다.

이 역할을 CloudWatch 에이전트를 설치할 인스턴스에 연결합니다. 자세한 정보는 Windows 인스턴스용 Amazon EC2 사용 설명서의 [IAM 역할을 인스턴스에 연결](#)을 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.

### AWS 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 정보는 AWS 시스템 관리자 사용 설명서의 [Linux 인스턴스에서 SSM Agent 설치 및 구성](#)을 참조하십시오.

SSM Agent를 설치하거나 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [SSM Agent 설치 및 구성](#)을 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 정보는 AWS 시스템 관리자 사용 설명서의 [시스템 관리자 사전 조건](#)을 참조하십시오.

#### 인터넷 액세스 확인

데이터를 CloudWatch 또는 CloudWatch Logs로 전송하려면 Amazon EC2 인스턴스에 아웃바운드 인터넷 액세스 권한이 있어야 합니다. 인터넷 액세스 구성 방법에 대한 자세한 정보는 Amazon VPC 사용 설명서의 [인터넷 게이트웨이](#)를 참조하십시오.

프록시에서 구성할 엔드포인트와 포트는 다음과 같습니다.

- 에이전트를 사용하여 지표를 수집하는 경우 해당 리전의 CloudWatch 엔드포인트 화이트리스트를 지정해야 합니다. 이러한 엔드포인트 목록은 Amazon Web Services 일반 참조의 [Amazon CloudWatch](#)를 참조하십시오.
- 에이전트를 사용하여 로그를 수집하는 경우 해당 리전의 CloudWatch Logs 엔드포인트 화이트리스트를 지정해야 합니다. 이러한 엔드포인트 목록은 Amazon Web Services 일반 참조의 [Amazon CloudWatch Logs](#)를 참조하십시오.

- SSM을 사용하여 에이전트를 설치하는 경우 또는 Parameter Store를 사용하여 구성 파일을 저장하는 경우 해당 리전의 SSM 엔드포인트 화이트리스트를 지정해야 합니다. 이러한 엔드포인트 목록은 Amazon Web Services 일반 참조의 [AWS 시스템 관리자](#)를 참조하십시오.

### CloudWatch 에이전트 패키지를 다운로드합니다.

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

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 command]를 선택합니다.
  4. [Command document] 목록에서 [AWS-ConfigureAWSPackage]를 선택합니다.
  5. 대상 영역에서 CloudWatch 에이전트를 설치할 인스턴스를 선택합니다. 특정 인스턴스가 보이지 않으면 Run Command에 대해 구성되지 않은 것일 수 있습니다. 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [시스템 관리자 사전 조건](#)을 참조하십시오.
  6. [Action] 목록에서 [Install]을 선택합니다.
  7. [Name] 필드에 AmazonCloudWatchAgent를 입력합니다.
  8. [Version]을 [latest]로 설정한 채 그대로 두어 최신 에이전트 버전을 설치합니다.
  9. [Run]을 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.

### S3 다운로드 링크를 사용하여 Amazon EC2 인스턴스에 CloudWatch 에이전트 패키지 다운로드

Amazon S3 다운로드 링크를 사용하여 Amazon EC2 인스턴스 서버에 CloudWatch 에이전트 패키지를 다운로드할 수 있습니다. 아키텍처 및 플랫폼에 따라 이 테이블에서 다운로드 링크를 선택하십시오.

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Amazon Linux and Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Centos	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Redhat	<a href="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</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	SUSE	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Debian	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Windows가 설치된	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig</a>
arm64	Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Redhat	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig</a>

명령줄을 사용하여 Amazon EC2 인스턴스에 CloudWatch 에이전트를 설치하려면

1. CloudWatch 에이전트를 다운로드합니다. Linux 서버의 경우, 다음을 입력합니다. [download-link](#)에 이진 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 GPS 서명 파일을 사용하여 패키지 서명을 확인할 수 있습니다. 자세한 정보는 [CloudWatch 에이전트 패키지의 서명 확인 \(p. 94\)](#) 단원을 참조하십시오.
3. 패키지를 설치합니다. Linux 서버에서 RPM 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Linux 서버에서 DEB 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Windows Server에서 실행 중인 서버에서 MSI 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
msiexec /i amazon-cloudwatch-agent.msi
```

또한 이 명령은 PowerShell 내에서도 작동합니다. MSI 명령 옵션에 대한 자세한 정보는 Microsoft Windows 문서의 [명령줄 옵션](#)을 참조하십시오.

## (선택 사항) CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정

다운로드한 CloudWatch 에이전트 패키지에는 `common-config.toml`이라는 구성 파일이 들어 있습니다. 이 파일을 사용하여 프록시, 자격 증명 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.

Amazon EC2 인스턴스에 CloudWatch 에이전트를 설치할 때는, 프록시 설정을 지정해야 하거나, 인스턴스가 위치한 리전이 아닌 다른 리전에 있는 CloudWatch에 에이전트가 지표를 보내야 하는 경우에만 이 파일을 수정하십시오.

```
# 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 편집할 수도 있고 시스템 관리자의 `RunShellScript` `Run Command`를 사용하여 편집할 수도 있습니다.

- 프록시 설정 서버가 HTTP 또는 HTTPS 프록시를 사용하여 AWS 서비스에 접근하도록 하려면 `http_proxy` 및 `https_proxy` 필드에 해당 프록시를 지정하십시오. 프록시 설정에서 제외해야 하는 URL이 있다면 이를 쉼표로 구분하여 `no_proxy` 필드에 지정하십시오.



- `shared_credential_profile` CloudWatch 에이전트가 인스턴스가 위치한 동일한 리전에 있는 CloudWatch로 지표를 보내도록 하려고 하면 이 줄을 수정하거나 인스턴스에 대한 적절한 권한을 IAM 역할에 연결합니다. IAM 역할을 연결한 경우 해당 에이전트용으로 명명된 프로필을 생성하는데 `aws configure` 명령을 사용하지 않아도 됩니다.

또는 이 줄을 사용하여 CloudWatch 에이전트가 AWS 구성 파일에서 사용할 명명된 프로필을 지정할 수 있습니다. 그렇게 하면, CloudWatch 에이전트가 해당 명명된 프로필에서 리전 설정을 사용하게 됩니다.

- `shared_credential_file` 기본값 경로를 사용하려고 하지 않는 경우 이 줄을 사용하여 사용할 자격 증명을 포함하는 파일에 경로를 지정하십시오.

`common-config.toml`을 수정한 후, CloudWatch 에이전트의 리전 정보를 지정해야 하는 경우에는 AWS 구성 파일에서 CloudWatch 에이전트의 명명된 프로필을 생성하십시오. 이 프로필을 생성할 때는 루트 또는 관리자 권한으로 작업을 수행하십시오.

다음은 구성 파일의 프로필 예입니다.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

CloudWatch 데이터를 다른 리전으로 전송하려면, 이 인스턴스에 연결한 IAM 역할에 해당 리전에서 CloudWatch 데이터를 기록할 권한이 있어야 합니다.

다음은 `aws configure` 명령을 사용하여 CloudWatch 에이전트용으로 명명된 프로필을 생성하는 예입니다. 이 예에서는 `AmazonCloudWatchAgent`의 기본 프로필 이름을 사용하는 것으로 가정합니다.

CloudWatch 에이전트용 `AmazonCloudWatchAgent` 프로필을 생성하려면

- Linux 서버에서 다음 명령을 입력하고 표시되는 메시지에 따릅니다.

```
sudo aws configure --profile AmazonCloudWatchAgent
```

Windows 서버에서는 관리자 권한으로 PowerShell을 열고 다음 명령을 입력한 후 표시되는 메시지에 따릅니다.

```
aws configure --profile AmazonCloudWatchAgent
```

## 첫 번째 인스턴스에 에이전트 구성 파일 생성

CloudWatch 에이전트를 다운로드했으면 서버에서 에이전트를 시작하기 전에 구성 파일을 생성해야 합니다. 자세한 정보는 [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#) 단원을 참조하십시오.

## CloudWatch 에이전트 시작

에이전트 구성 파일을 생성한 서버와 동일한 서버에서 에이전트를 시작하려면 다음 단계를 따르십시오. 다른 서버에서 이 구성 파일을 사용하려면 [에이전트 구성을 사용하여 추가 인스턴스에 CloudWatch 에이전트 설치 \(p. 83\)](#) 단원을 참조하십시오.

### Run Command를 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 command]를 선택합니다.
4. [Command document] 목록에서 [AmazonCloudWatch-ManagedAgent]를 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치한 인스턴스를 선택합니다.
6. [Action] 목록에서 [configure]를 선택합니다.
7. [Optional Configuration Source] 목록에서 [ssm]을 선택합니다.
8. Optional Configuration Location(구성 위치(선택 사항)) 상자에, [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#)에 설명된 대로 생성하고 시스템 관리자 Parameter Store 에 저장한 에이전트 구성 파일의 이름을 입력합니다.
9. [Optional Restart] 목록에서 [yes]를 선택하여 해당 단계를 마친 후 에이전트가 시작되도록 합니다.
10. [Run]을 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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m ec2 -c file:configuration-file-path -s
```



## 에이전트 구성을 사용하여 추가 인스턴스에 CloudWatch 에이전트 설치

CloudWatch 에이전트 구성을 Parameter Store에 저장했으면 다른 서버에 에이전트를 설치할 때 이를 사용할 수 있습니다.

### 항목

- 시스템 관리자용 IAM 역할 및 CloudWatch 에이전트 생성 (p. 83)
- Amazon EC2 인스턴스에 CloudWatch 에이전트 패키지 다운로드 (p. 83)
- (선택 사항) CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정 (p. 86)
- CloudWatch 에이전트 시작 (p. 87)

## 시스템 관리자용 IAM 역할 및 CloudWatch 에이전트 생성

CloudWatch 에이전트를 Amazon EC2 인스턴스에 설치하는 경우 인스턴스 프로파일용 IAM 역할이 필요합니다. 이 역할은 CloudWatch 에이전트가 인스턴스에 대한 작업을 수행하도록 해줍니다. 에이전트를 설치 및 실행하는 데 필요한 권한만 들어 있는 앞에서 생성한 역할을 사용하십시오. 이 역할을 CloudWatchAgentServerPolicy라고 합니다.

이 역할을 CloudWatch 에이전트를 설치할 인스턴스에 연결합니다. 자세한 정보는 Windows 인스턴스용 Amazon EC2 사용 설명서의 [IAM 역할을 인스턴스에 연결](#)을 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.

## 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 정보는 AWS 시스템 관리자 사용 설명서의 [Linux 인스턴스에서 SSM Agent 설치 및 구성](#)을 참조하십시오.

SSM Agent를 설치하거나 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [SSM 에이전트 설치 및 구성](#)을 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 정보는 AWS 시스템 관리자 사용 설명서의 [시스템 관리자 사전 조건](#)을 참조하십시오.

### 인터넷 액세스 확인

데이터를 CloudWatch 또는 CloudWatch Logs 로 전송하려면 Amazon EC2 인스턴스에 아웃바운드 인터넷 액세스 권한이 있어야 합니다. 인터넷 액세스 구성 방법에 대한 자세한 정보는 Amazon VPC 사용 설명서의 [인터넷 게이트웨이](#)를 참조하십시오.

## CloudWatch 에이전트 패키지를 다운로드합니다.

시스템 관리자 Run Command를 사용하면 인스턴스 구성을 관리할 수 있습니다. 시스템 관리자 문서, 파라미터를 지정하고 하나 이상의 인스턴스에 명령을 실행합니다. 인스턴스의 SSM Agent는 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 command]를 선택합니다.
  4. [Command document] 목록에서 [AWS-ConfigureAWSPackage]를 선택합니다.
  5. 대상 영역에서 CloudWatch 에이전트를 설치할 인스턴스를 선택합니다. 특정 인스턴스가 보이지 않으면 Run Command에 대해 구성되지 않은 것일 수 있습니다. 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [시스템 관리자 사전 조건](#)을 참조하십시오.
  6. [Action] 목록에서 [Install]을 선택합니다.
  7. [Name] 상자에 [AmazonCloudWatchAgent]를 입력합니다.
  8. [Version]을 [latest]로 설정한 채 그대로 두어 최신 에이전트 버전을 설치합니다.
  9. [Run]을 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.

## S3 다운로드 링크를 사용하여 Amazon EC2 인스턴스에 CloudWatch 에이전트 패키지 다운로드

Amazon S3 다운로드 링크를 사용하여 Amazon EC2 인스턴스 서버에 CloudWatch 에이전트 패키지를 다운로드할 수 있습니다. 아키텍처 및 플랫폼에 따라 이 테이블에서 다운로드 링크를 선택하십시오.

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Amazon Linux and Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Centos	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Redhat	<a href="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</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	SUSE	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Debian	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Windows가 설치된	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig</a>
arm64	Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Redhat	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig</a>

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 GPS 서명 파일을 사용하여 패키지 서명을 확인할 수 있습니다. 자세한 정보는 [CloudWatch 에이전트 패키지의 서명 확인 \(p. 94\)](#) 단원을 참조하십시오.
3. 패키지를 설치합니다. Linux 서버에서 RPM 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Linux 서버에서 DEB 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Windows Server에서 실행 중인 서버에서 MSI 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
msiexec /i amazon-cloudwatch-agent.msi
```

또한 이 명령은 PowerShell 내에서도 작동합니다. MSI 명령 옵션에 대한 자세한 정보는 Microsoft Windows 문서의 [명령줄 옵션](#)을 참조하십시오.

## (선택 사항) CloudWatch 에이전트용 일반 구성 및 명명된 프로파일 수정

다운로드한 CloudWatch 에이전트 패키지에는 `common-config.toml`이라는 구성 파일이 들어 있습니다. 이 파일을 사용하여 프록시, 자격 증명 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.

Amazon EC2 인스턴스에 CloudWatch 에이전트를 설치할 때는, 프록시 설정을 지정해야 하거나, 인스턴스가 위치한 리전이 아닌 다른 리전에 있는 CloudWatch에 에이전트가 지표를 보내야 하는 경우에만 이 파일을 수정하십시오.

```
# 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 프로파일이나 프록시 설정을 설정하려면 해당 줄에서 `#`을 제거하고 값을 지정하십시오. 이 파일은 수동으로 편집할 수도 있고 시스템 관리자의 `RunShellScript Run Command`를 사용하여 편집할 수도 있습니다.

- `shared_credential_profile` CloudWatch 에이전트가 인스턴스가 위치한 동일한 리전에 있는 CloudWatch로 지표를 보내도록 하려고 하면 이 줄을 수정하거나 인스턴스에 대한 적절한 권한을 IAM 역할에 연결합니다. IAM 역할을 연결한 경우 해당 에이전트용으로 명명된 프로파일을 생성하는데 `aws configure` 명령을 사용하지 않아도 됩니다.

그렇지 않으면, 이 줄을 사용하여 CloudWatch 에이전트가 AWS 자격 증명 및 AWS 구성 파일에 사용하도록 할 명명된 프로필을 지정하면 됩니다. 그렇게 하면, CloudWatch 에이전트가 해당 명명된 프로필의 자격 증명 및 리전 설정을 사용하게 됩니다.

- `shared_credential_file` 기본값 경로를 사용하려고 하지 않는 경우 이 줄을 사용하여 사용할 자격 증명을 포함하는 파일에 경로를 지정하십시오.
- 프록시 설정 서버가 HTTP 또는 HTTPS 프록시를 사용하여 AWS 서비스에 접근하도록 하려면 `http_proxy` 및 `https_proxy` 필드에 해당 프록시를 지정하십시오. 프록시 설정에서 제외해야 하는 URL이 있다면 이를 쉼표로 구분하여 `no_proxy` 필드에 지정하십시오.

`common-config.toml`을 수정한 후, CloudWatch 에이전트에 대해 자격 증명 및 리전 정보를 지정해야 하는 경우, CloudWatch 에이전트에 대한 명명된 프로필을 AWS 자격 증명 및 AWS 구성 파일에 생성하십시오. 이 프로필을 생성할 때는 루트 또는 관리자 권한으로 작업을 수행하십시오. 다음은 자격 증명 파일에 있는 이 프로필의 예입니다.

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

`my_access_key` 및 `my_secret_key`에서, 시스템 관리자 Parameter Store에 대한 쓰기 권한이 없는 IAM 사용자의 키를 사용합니다. CloudWatch 에이전트에 필요한 IAM 사용자에게 대한 자세한 정보는 [온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성 \(p. 75\)](#) 단원을 참조하십시오.

다음은 구성 파일의 프로필 예입니다.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

다음은 `aws configure` 명령을 사용하여 CloudWatch 에이전트용으로 명명된 프로필을 생성하는 예입니다. 이 예에서는 `AmazonCloudWatchAgent`의 기본 프로필 이름을 사용하는 것으로 가정합니다.

CloudWatch 에이전트용 `AmazonCloudWatchAgent` 프로필을 생성하려면

- Linux 서버에서 다음 명령을 입력하고 표시되는 메시지에 따릅니다.

```
sudo aws configure --profile AmazonCloudWatchAgent
```

Windows 서버에서는 관리자 권한으로 PowerShell을 열고 다음 명령을 입력한 후 표시되는 메시지에 따릅니다.

```
aws configure --profile AmazonCloudWatchAgent
```

## 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 command]를 선택합니다.
4. [Command document] 목록에서 [AmazonCloudWatch-ManagedAgent]를 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치한 인스턴스를 선택합니다.
6. [Action] 목록에서 [configure]를 선택합니다.
7. [Optional Configuration Source] 목록에서 [ssm]을 선택합니다.
8. Optional Configuration Location(구성 위치(선택 사항)) 상자에, [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#)에 설명된 대로 생성하고 시스템 관리자 Parameter Store 에 저장한 에이전트 구성 파일의 이름을 입력합니다.
9. [Optional Restart] 목록에서 [yes]를 선택하여 해당 단계를 마친 후 에이전트가 시작되도록 합니다.
10. [Run]을 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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m ec2 -c file:configuration-file-path -s
```

## 온프레미스 서버에 CloudWatch 에이전트 설치

처음으로 CloudWatch 에이전트 사용을 시작하는 경우, 이를 서버에 다운로드하고 에이전트를 구성해야 합니다. 그러면, 해당 구성이 있는 에이전트를 해당 서버에서 직접 사용할 수 있으며, 구성을 AWS 시스템 관리자 Parameter Store에 저장하는 경우에는, 다른 서버에 CloudWatch 에이전트를 설치할 때도 동일한 구성을 사용할 수 있습니다.

### 항목

- 시작하기: 첫 번째 서버에 CloudWatch 에이전트 설치 (p. 89)
- 에이전트 구성을 사용하여 추가 서버에 CloudWatch 에이전트 설치 (p. 97)

## 시작하기: 첫 번째 서버에 CloudWatch 에이전트 설치

온프레미스 서버에 CloudWatch 에이전트를 다운로드하고 설치하려면 AWS 시스템 관리자 또는 명령줄을 사용하면 됩니다.

어떤 방법을 사용하든 먼저 CloudWatch에 대한 쓰기 권한이 있는 IAM 사용자를 생성해야 합니다.

### 항목

- 온프레미스 서버에 CloudWatch 에이전트 다운로드 (p. 89)
- CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정 (p. 91)
- CloudWatch 에이전트 구성 파일 생성 (p. 93)
- CloudWatch 에이전트 시작 (p. 93)
- CloudWatch 에이전트 패키지의 서명 확인 (p. 94)

## 온프레미스 서버에 CloudWatch 에이전트 다운로드

CloudWatch 에이전트를 다운로드하려면 시스템 관리자 또는 명령줄을 사용하면 됩니다.

### 시스템 관리자를 사용하여 다운로드

시스템 관리자 Run Command를 사용하려면 Amazon EC2 Systems Manager을 사용하여 온프레미스 서버를 등록해야 합니다. 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [하이브리드 환경에서 Systems Manager 설정](#)을 참조하십시오.

서버를 이미 등록했다면 SSM Agent를 최신 버전으로 업데이트하십시오.

Linux를 실행하는 서버에서 SSM Agent를 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [Linux 하이브리드 환경의 서버와 VM에 SSM Agent 설치](#)를 참조하십시오.

Windows Server를 실행하는 서버에서 SSM Agent를 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [Windows 하이브리드 환경의 서버와 VM에 SSM Agent 설치](#)를 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 command]를 선택합니다.
4. [Command document] 목록에서 [AWS-ConfigureAWSPackage] 옆의 버튼을 선택합니다.



5. 대상 영역에서 CloudWatch 에이전트를 설치할 서버를 선택합니다. 특정 서버가 보이지 않으면 Run Command에 대해 구성되지 않은 것일 수 있습니다. 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [시스템 관리자 사전 조건](#)을 참조하십시오.
6. [Action] 목록에서 [Install]을 선택합니다.
7. [Name] 상자에 [AmazonCloudWatchAgent]를 입력합니다.
8. [Version]을 비워 두어 최신 에이전트 버전을 설치합니다.
9. [Run]을 선택합니다.

에이전트 패키지가 다운로드되며 다음 단계는 이를 구성하고 시작하는 것입니다.

## S3 다운로드 링크를 사용하여 다운로드

명령줄을 사용하여 에이전트를 다운로드하는 경우 먼저 이 테이블에서 다운로드 링크를 선택하십시오. 아키텍처 및 플랫폼에 따라 링크를 선택하십시오.

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Amazon Linux and Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Centos	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Redhat	<a href="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</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	SUSE	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Debian	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Windows가 설치된	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig</a>
arm64	Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/</a>



Arch	플랫폼	다운로드 링크	서명 파일 링크
			latest/amazon-cloudwatch-agent.rpm.sig
arm64	Redhat	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig</a>

명령줄을 사용하여 CloudWatch 에이전트를 온프레미스 서버에 다운로드하려면

1. 에이전트 패키지를 다운로드할 디렉터리를 만듭니다. 예, tmp/AmazonCloudWatchAgent. 그런 다음 해당 디렉터리로 변경합니다.
2. 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
```

3. 패키지를 다운로드했으면 선택 사항으로 GPS 서명 파일을 사용하여 패키지 서명을 확인할 수 있습니다. 자세한 정보는 [CloudWatch 에이전트 패키지의 서명 확인 \(p. 94\)](#) 단원을 참조하십시오.
4. 패키지를 설치합니다. Linux 서버에서 RPM 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Linux 서버에서 DEB 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Windows Server에서 실행 중인 서버에서 MSI 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
msiexec /i amazon-cloudwatch-agent.msi
```

또한 이 명령은 PowerShell 내에서도 작동합니다. MSI 명령 옵션에 대한 자세한 정보는 Microsoft Windows 문서의 [명령줄 옵션](#)을 참조하십시오.

## CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정

다운로드한 CloudWatch 에이전트 패키지에는 `common-config.toml`이라는 구성 파일이 들어 있습니다. 이 파일을 사용하여 프록시, 자격 증명 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 프로파일이나 프록시 설정을 설정하려면 해당 줄에서 #을 제거하고 값을 지정하십시오. 이 파일은 수동으로 편집할 수도 있고 시스템 관리자의 RunShellScript Run Command를 사용하여 편집할 수도 있습니다.

- shared\_credential\_profile CloudWatch 에이전트가 AWS 자격 증명 및 AWS Config 파일에서 찾아야 하는 명명된 프로파일에 이름을 지정할 수 있습니다. 여기에서 이름을 지정하지 않으면, CloudWatch 에이전트가 기본 프로파일 이름인 AmazonCloudWatchAgent를 찾습니다.
- shared\_credential\_file CloudWatch 에이전트는 다음 위치에서 자격 증명과 리전 정보를 찾습니다.
  - Linux 서버의 /root/.aws
  - Windows Server 2008 및 Windows Server 2012의 %SystemDrive%\Users\Administrator\ .aws
  - Windows Server 2003의 %SystemDrive%\Documents and Settings\Administrator\ .aws%SystemDrive는 일반적으로 C:입니다.

이 기본 경로를 사용하지 않으려면 shared\_credential\_file을 사용하여, 사용할 자격 증명을 포함하는 다른 파일의 경로를 지정하십시오.

- 프록시 설정 서버가 HTTP 또는 HTTPS 프록시를 사용하여 AWS 서비스에 접근하도록 하려면 http\_proxy 및 https\_proxy 필드에 해당 프록시를 지정하십시오. 프록시 설정에서 제외해야 하는 URL이 있다면 이를 쉼표로 구분하여 no\_proxy 필드에 지정하십시오.

common-config.toml을 수정했으면, 지정한 프로파일 이름 또는 AmazonCloudWatchAgent의 기본 프로파일 이름이 루트 사용자의 AWS 자격 증명 및 Config 파일에 있는지 확인해야 합니다. 이 프로파일은 CloudWatch 에이전트 설치 시 자격 증명 및 리전 정보를 제공하는 데 사용됩니다. 다음은 자격 증명 파일에 있는 이 프로파일의 예입니다.

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

my\_access\_key 및 my\_secret\_key에서, 시스템 관리자 Parameter Store 에 대한 쓰기 권한이 없는 IAM 사용자의 키를 사용합니다. CloudWatch 에이전트에 필요한 IAM 사용자에게 대한 자세한 정보는 [온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성 \(p. 75\)](#) 단원을 참조하십시오.

다음은 구성 파일의 프로필 예입니다.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

자격 증명 파일의 명명된 프로필에는 CloudWatch 에이전트에 사용될 자격 증명이 들어 있습니다. 이러한 자격 증명은 CloudWatch 에이전트를 설치하는 동안 지표 데이터를 CloudWatch에 쓰고 시스템 관리자 Parameter Store 로부터 정보를 다운로드하는 권한에 사용됩니다. 이 단원의 앞부분에서 설명한 대로 CloudWatch 에이전트용 IAM 사용자를 생성하면 이 단원에 사용할 자격 증명을 얻을 수 있습니다.

CloudWatch 에이전트가 온프레미스 서버에서 실행되는 경우, 구성 파일에 있는 명명된 프로필은 CloudWatch 지표가 게시된 리전을 나타냅니다. aws configure 명령을 사용하여 프로필을 수정하는 경우, 루트 또는 관리자로 해당 작업을 수행하십시오.

다음은 aws configure 명령을 사용하여 CloudWatch 에이전트용으로 명명된 프로필을 생성하는 예입니다. 이 예에서는 AmazonCloudWatchAgent의 기본 프로필 이름을 사용하는 것으로 가정합니다.

CloudWatch 에이전트용 AmazonCloudWatchAgent 프로필을 생성하려면

- Linux 서버에서 다음 명령을 입력하고 표시되는 메시지에 따릅니다.

```
sudo aws configure --profile AmazonCloudWatchAgent
```

Windows 서버에서는 관리자 권한으로 PowerShell을 열고 다음 명령을 입력한 후 표시되는 메시지에 따릅니다.

```
aws configure --profile AmazonCloudWatchAgent
```

## CloudWatch 에이전트 구성 파일 생성

CloudWatch 에이전트는 구성 파일을 사용하여 수집할 지표 및 다른 에이전트 구성 데이터를 지정합니다. 에이전트를 시작하기 전에 이 파일을 사용자 지정해야 합니다. 자세한 정보는 [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#) 단원을 참조하십시오.

## 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 command]를 선택합니다.
4. [Command document] 목록에서 [AmazonCloudWatch-ManageAgent] 옆의 버튼을 선택합니다.
5. [Targets] 영역에서 에이전트를 설치한 인스턴스를 선택합니다.
6. [Action] 목록에서 [configure]를 선택합니다.
7. [Mode] 목록에서 [onPremise]를 선택합니다.

- Optional Configuration Location(구성 위치(선택 사항)) 상자, 마법사로 생성하고 Parameter Store에 저장한 에이전트 구성 파일의 이름을 입력합니다.
- [Run]을 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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m onPremise -c file:configuration-file-path -s
```

## CloudWatch 에이전트 패키지의 서명 확인

CloudWatch 에이전트 패키지에 대해 GPG 서명 파일이 포함됩니다. 퍼블릭 키를 사용하여 에이전트 다운로드 파일이 원본이며 수정되지 않았는지 확인할 수 있습니다. 먼저 <https://gnupg.org/index.html>를 사용하여 퍼블릭 키를 가져옵니다.

올바른 서명 파일을 찾으려면 다음 표를 참조하십시오.

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Amazon Linux and Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Centos	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Redhat	<a href="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</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	SUSE	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Debian	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Windows가 설치된	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig</a>
arm64	Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Redhat	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig</a>

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 3B789C72입니다.

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. Wget를 사용하여 패키지 서명 파일을 다운로드하십시오. 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 계속 되는 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 않습니다. 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)] 옵션을 포함합니다.

남은 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 적어 둡니다. 이전 예제에서 키 값은 3B789C72입니다.

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. Wget를 사용하여 패키지 서명 파일을 다운로드하십시오. 정확한 서명 파일을 결정하려면 [CloudWatch Agent Download Links \(p. 98\)](#)를 참조하십시오.
6. 서명을 확인하려면 `gpg --verify`를 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 계속 되는 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 않습니다. 단지 해당 사용자가 퍼블릭 키를 확인하지 않은 것입니다.

## 에이전트 구성을 사용하여 추가 서버에 CloudWatch 에이전트 설치

### 항목

- 온프레미스 서버에 CloudWatch 에이전트 다운로드 (p. 97)
- CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정 (p. 100)
- CloudWatch 에이전트 시작 (p. 101)

## 온프레미스 서버에 CloudWatch 에이전트 다운로드

CloudWatch 에이전트를 다운로드하려면 AWS 시스템 관리자 또는 명령줄을 사용하면 됩니다.

### 시스템 관리자를 사용하여 다운로드

시스템 관리자 Run Command를 사용하려면 Amazon EC2 Systems Manager를 사용하여 온프레미스 서버를 등록해야 합니다. 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [하이브리드 환경에서 Systems Manager 설정](#)을 참조하십시오.

서버를 이미 등록했다면 SSM Agent를 최신 버전으로 업데이트하십시오.

Linux를 실행하는 서버에서 SSM Agent를 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [Linux 하이브리드 환경의 서버와 VM에 SSM Agent 설치](#)를 참조하십시오.



Windows Server를 실행하는 서버에서 SSM Agent를 업데이트하는 방법에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [Windows 하이브리드 환경의 서버와 VM에 SSM Agent 설치](#)를 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 command]를 선택합니다.
4. [Command document] 목록에서 [AWS-ConfigureAWSPackage] 옆의 버튼을 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치할 서버를 선택합니다. 특정 서버가 보이지 않으면 Run Command에 대해 구성되지 않은 것일 수 있습니다. 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [시스템 관리자 사전 조건](#)을 참조하십시오.
6. [Action] 목록에서 [Install]을 선택합니다.
7. [Name] 상자에 [AmazonCloudWatchAgent]를 입력합니다.
8. [Version]을 비워 두어 최신 에이전트 버전을 설치합니다.
9. [Run]을 선택합니다.

에이전트 패키지가 다운로드되며 다음 단계는 이를 구성하고 시작하는 것입니다.

## S3 다운로드 링크를 사용하여 다운로드

명령줄을 사용하여 에이전트를 다운로드하려면 먼저 이 테이블에서 다운로드 링크를 선택하십시오. 선택하는 링크는 아키텍처 및 플랫폼에 따라 다릅니다.

Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Amazon Linux and Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Centos	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Redhat	<a href="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</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	SUSE	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig</a>
amd64	Debian	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>



Arch	플랫폼	다운로드 링크	서명 파일 링크
amd64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig</a>
amd64	Windows가 설 치된	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig</a>
arm64	Amazon Linux 2	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Redhat	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig</a>
arm64	Ubuntu	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb</a>	<a href="https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig">https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig</a>

명령줄을 사용하여 CloudWatch 에이전트를 온프레미스 서버에 다운로드하려면

1. 에이전트 패키지를 다운로드할 디렉터리를 만듭니다. 예, `tmp/AmazonCloudWatchAgent`. 그런 다음 해당 디렉터리로 변경합니다.
2. 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
```

3. 패키지를 다운로드했으면 선택 사항으로 GPS 서명 파일을 사용하여 패키지 서명을 확인할 수 있습니다. 자세한 정보는 [CloudWatch 에이전트 패키지의 서명 확인 \(p. 94\)](#) 단원을 참조하십시오.
4. 패키지를 설치합니다. Linux 서버에서 RPM 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Linux 서버에서 DEB 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Windows Server에서 실행 중인 서버에서 MSI 패키지를 다운로드하는 경우 패키지가 들어 있는 디렉터리로 변경하고 다음을 입력합니다.

```
msiexec /i amazon-cloudwatch-agent.msi
```

또한 이 명령은 PowerShell 내에서도 작동합니다. MSI 명령 옵션에 대한 자세한 정보는 Microsoft Windows 문서의 [명령줄 옵션](#)을 참조하십시오.

## CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정

다운로드한 CloudWatch 에이전트 패키지에는 `common-config.toml`이라는 구성 파일이 들어 있습니다. 이 파일을 사용하여 프록시, 자격 증명 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 편집할 수도 있고 시스템 관리자의 `RunShellScript Run Command`를 사용하여 편집할 수도 있습니다.

- `shared_credential_profile` CloudWatch 에이전트가 AWS 자격 증명 및 AWS Config 파일에서 찾아야 하는 명명된 프로필에 이름을 지정할 수 있습니다. 여기에서 이름을 지정하지 않으면, CloudWatch 에이전트가 기본 프로필 이름인 `AmazonCloudWatchAgent`를 찾습니다.
- `shared_credential_file` 기본값 경로를 사용하려고 하지 않는 경우 이 줄을 사용하여 사용할 자격 증명을 포함하는 파일에 경로를 지정하십시오.
- 프록시 설정 서버가 HTTP 또는 HTTPS 프록시를 사용하여 AWS 서비스에 접근하도록 하려면 `http_proxy` 및 `https_proxy` 필드에 해당 프록시를 지정하십시오. 프록시 설정에서 제외해야 하는 URL이 있다면 이를 쉼표로 구분하여 `no_proxy` 필드에 지정하십시오.

`common-config.toml`을 수정했으면, 지정한 프로필 이름 또는 `AmazonCloudWatchAgent`의 기본 프로필 이름이 루트 사용자의 AWS 자격 증명 및 Config 파일에 있는지 확인해야 합니다. 이 프로필은 CloudWatch 에이전트 설치 시 자격 증명 및 리전 정보를 제공하는 데 사용됩니다. 다음은 자격 증명 파일에 있는 이 프로필의 예입니다.

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

my\_access\_key 및 my\_secret\_key에서, 시스템 관리자 Parameter Store 에 대한 쓰기 권한이 없는 IAM 사용자의 키를 사용합니다. CloudWatch 에이전트에 필요한 IAM 사용자에게 대한 자세한 정보는 [온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성 \(p. 75\)](#) 단원을 참조하십시오.

다음은 구성 파일의 프로필 예입니다.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

자격 증명 파일의 명명된 프로필에는 CloudWatch 에이전트에 사용될 자격 증명이 들어 있습니다. 이러한 자격 증명은 CloudWatch 에이전트를 설치하는 동안 지표 데이터를 CloudWatch에 쓰고 시스템 관리자 Parameter Store로부터 정보를 다운로드하는 권한에 사용됩니다. 이 단원의 앞부분에서 설명한 대로 CloudWatch 에이전트용 IAM 사용자를 생성하면 이 단원에 사용할 자격 증명을 얻을 수 있습니다.

CloudWatch 에이전트가 온프레미스 서버에서 실행되는 경우, 구성 파일에 있는 명명된 프로필은 CloudWatch 지표가 게시된 리전을 나타냅니다. aws configure 명령을 사용하여 프로필을 수정하는 경우, 루트 또는 관리자로 해당 작업을 수행하십시오.

다음은 aws configure 명령을 사용하여 CloudWatch 에이전트용으로 명명된 프로필을 생성하는 예입니다. 이 예에서는 AmazonCloudWatchAgent의 기본 프로필 이름을 사용하는 것으로 가정합니다.

CloudWatch 에이전트용 AmazonCloudWatchAgent 프로필을 생성하려면

- Linux 서버에서 다음 명령을 입력하고 표시되는 메시지에 따릅니다.

```
sudo aws configure --profile AmazonCloudWatchAgent
```

Windows 서버에서는 관리자 권한으로 PowerShell을 열고 다음 명령을 입력한 후 표시되는 메시지에 따릅니다.

```
aws configure --profile AmazonCloudWatchAgent
```

## 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 command]를 선택합니다.
4. [Command document] 목록에서 [AmazonCloudWatch-ManageAgent] 옆의 버튼을 선택합니다.
5. [Targets] 영역에서 에이전트를 설치한 인스턴스를 선택합니다.
6. [Action] 목록에서 [configure]를 선택합니다.

7. [Mode] 목록에서 [onPremise]를 선택합니다.
8. Optional Configuration Location(구성 위치(선택 사항)) 상자, 마법사로 생성하고 Parameter Store에 저장한 에이전트 구성 파일의 이름을 입력합니다.
9. [Run]을 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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./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 다음 명령을 사용합니다. PowerShell 콘솔에서 다음을 입력하십시오.

```
./amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m onPremise -c file:configuration-file-path -s
```

## AWS CloudFormation을 사용하여 새 인스턴스에 CloudWatch 에이전트 설치

CloudWatch 에이전트 설치 및 업데이트를 지원하기 위해 Amazon에서는 GitHub에 여러 AWS CloudFormation 템플릿을 업로드했습니다. AWS CloudFormation 사용에 대한 자세한 정보는 [AWS CloudFormation란 무엇입니까?](#)를 참조하십시오.

템플릿 위치는 [Deploy CloudWatch Agent to EC2 instances by AWS CloudFormation](#)입니다. 이 위치에는 inline 및 ssm 디렉터리가 둘 다 포함되어 있습니다. 각 디렉터리에는 Linux 및 Windows 인스턴스 둘 다에 대한 템플릿이 포함되어 있습니다.

- inline 디렉터리의 템플릿에는 AWS CloudFormation 템플릿에 포함된 CloudWatch 에이전트 구성이 들어 있습니다. 기본적으로 Linux 템플릿은 지표 `mem_used_percent` 및 `swap_used_percent`를 수집하고, Windows 템플릿은 Memory % Committed Bytes In Use 및 Paging File % Usage를 수집합니다.

템플릿의 다음 섹션을 수정하여 다른 지표를 수집하도록 이러한 템플릿을 수정할 수 있습니다. 다음은 Linux 서버용 템플릿에서 가져온 예제입니다. 에이전트 구성 파일의 형식 및 구문에 따라 다음과 같이 변경

합니다. 자세한 정보는 [CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 \(p. 111\)](#) 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 위의 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 정보는 [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#) 단원을 참조하십시오.

CloudWatch 에이전트를 설치하고 에이전트 구성을 업데이트하는 데 이러한 두 가지 템플릿 유형을 모두 사용할 수 있습니다.

## 자습서: AWS CloudFormation 인라인 템플릿을 사용하여 CloudWatch 에이전트 설치 및 업데이트

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, 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 Amazon Linux 2용 인라인 템플릿을 다운로드합니다.

```
curl -O https://raw.githubusercontent.com/aws-labs/aws-cloudformation-templates/master/
aws/solutions/AmazonCloudWatchAgent/inline/amazon_linux.template
```

2. <https://console.aws.amazon.com/cloudformation>에서 AWS CloudFormation 콘솔을 엽니다.
3. [Create stack]을 선택합니다.
4. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 다운로드한 템플릿을 선택한 후 다음을 선택합니다.
5. 세부 정보 지정 페이지에서 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 정보는 [Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 \(p. 74\)](#) 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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 키 페어](#)를 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. [Next]를 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. 경우에 따라 그 뒤에 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/cloudformation>에서 AWS CloudFormation 콘솔을 엽니다.
3. AWS CloudFormation 대시보드에서 생성한 스택을 선택하고 스택 업데이트를 선택합니다.
4. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 수정한 템플릿을 선택한 후 다음을 선택합니다.

5. 옵션 페이지에서 다음, 다음을 차례로 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. 에이전트 구성 파일을 생성해 Parameter Store에 저장합니다. 자세한 정보는 [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#) 단원을 참조하십시오.
2. GitHub에서 템플릿을 다운로드합니다.

```
curl -O https://raw.githubusercontent.com/aws-labs/aws-cloudformation-templates/master/aws/solutions/AmazonCloudWatchAgent/ssm/amazon_linux.template
```

3. <https://console.aws.amazon.com/cloudformation>에서 AWS CloudFormation 콘솔을 엽니다.
4. [Create stack]을 선택합니다.
5. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 다운로드한 템플릿을 선택한 후 다음을 선택합니다.
6. Specify Details(세부 정보 지정) 페이지에서 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 정보는 [Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 \(p. 74\)](#) 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 정보는 Linux 인스턴스용 Amazon EC2 사용 설명서의 [Amazon EC2 키 페어](#)를 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.
7. 옵션 페이지에서 스택 리소스에 태그를 지정할 수 있습니다. [Next]를 선택합니다.
8. 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.

9. 인스턴스가 생성되면 Amazon EC2 콘솔에서 해당 인스턴스를 볼 수 있습니다. 경우에 따라 그 뒤에 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” \(p. 105\)](#) 주제에서 다운로드한 AWS CloudFormation 템플릿에서 버전 번호를 변경합니다. 예를 들어, VERSION=1.0을 VERSION=2.0으로 변경할 수 있습니다.
3. <https://console.aws.amazon.com/cloudformation>에서 AWS CloudFormation 콘솔을 엽니다.
4. AWS CloudFormation 대시보드에서 생성한 스택을 선택하고 스택 업데이트를 선택합니다.
5. 템플릿 선택에서 Amazon S3에 템플릿 업로드를 선택하고, 방금 수정한 템플릿을 선택한 후 다음을 선택합니다.
6. 옵션 페이지에서 다음, 다음을 차례로 선택합니다.
7. 검토 페이지에서 정보를 검토하고 업데이트를 선택합니다.

잠시 후 UPDATE\_COMPLETE가 표시됩니다.

## AWS CloudFormation에서 CloudWatch 에이전트를 사용하여 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 업데이트했는데 원하는 지표가 보이지 않는 경우 에이전트가 해당 지표를 수집하도록 구성되어 있는지 확인하십시오. 이렇게 하려면 `amazon-cloudwatch-agent.json` 파일에서 해당 지표가 나열되어 있고, 올바른 지표 네임스페이스를 살펴보고 있는지 확인하십시오. 자세한 정보는 [CloudWatch 에이전트 파일 및 위치 \(p. 153\)](#) 단원을 참조하십시오.



## CloudWatch 에이전트 구성 파일 생성

CloudWatch 에이전트를 Amazon EC2 인스턴스에 설치하든 아니면 온프레미스 서버에 설치하든, 에이전트를 시작하기 전에 CloudWatch 에이전트를 생성해야 합니다.

에이전트 구성 파일은 사용자 지정 지표를 포함하여 에이전트가 수집해야 하는 지표 및 로그가 지정되어 있는 JSON 파일입니다. 이 파일은 마법사를 사용해서 또는 사용자가 처음부터 새로 만들어서 생성할 수 있습니다. 또한 마법사를 사용하여 구성 파일을 처음으로 만든 다음, 수동으로 수정할 수도 있습니다.

구성 파일을 수동으로 생성하거나 수정하는 경우 프로세스가 보다 복잡하지만, 수집되는 지표를 더 많이 제어할 수 있으며 마법사에서 언급하지 않는 지표도 지정할 수 있습니다.

에이전트 구성 파일을 변경할 때마다 에이전트를 다시 시작하여 변경 사항이 적용되도록 해야 합니다.

구성 파일을 생성한 이후 이를 시스템 관리자 Parameter Store에 저장할 수 있습니다. 이렇게 하면 다른 서버에서도 동일한 에이전트 구성을 사용할 수 있습니다.

### 목차

- [마법사로 CloudWatch 에이전트 구성 파일 만들기 \(p. 107\)](#)
- [CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 \(p. 111\)](#)

## 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 실행하니까 아니면 Windows Server를 실행하니까?
- 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 프로토콜을 사용하여 애플리케이션 또는 서비스에서 사용자 지정 지표를 수집하시겠습니까?
- 기존 SSM Agent로부터 마이그레이션하는 중입니까?

마법사를 시작하기 전에 AWS 자격 증명 및 구성 파일이 제자리에 저장되어 있는 경우, 사용할 자격 증명 및 AWS 리전을 자동 검색할 수 있습니다. 이러한 파일에 대한 자세한 정보는 AWS 시스템 관리자 사용 설명서의 [구성 및 자격 증명 파일](#)을 참조하십시오.

마법사가 자격 증명 파일에서 다음과 같은 `AmazonCloudWatchAgent` 섹션을 찾습니다.

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_secret_key
aws_secret_access_key = my_access_key
```

이 섹션이 있는 경우, 마법사가 CloudWatch 에이전트에 대해 이 자격 증명을 사용합니다.

`my_access_key` 및 `my_secret_key`에서, 시스템 관리자 Parameter Store에 대한 쓰기 권한이 있는 IAM 사용자의 키를 사용합니다. CloudWatch 에이전트에 필요한 IAM 사용자에게 대한 자세한 정보는 [온프레미스 서버에서 CloudWatch 에이전트와 함께 사용할 IAM 사용자 생성 \(p. 75\)](#) 단원을 참조하십시오.

구성 파일에서는 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 수준이 다양한 사전 정의된 지표 집합을 사용하여 구성되어 있습니다. 이러한 지표 집합은 다음 표에 표시되어 있습니다. 지표에 대한 자세한 정보는 [CloudWatch 에이전트가 수집하는 지표 \(p. 144\)](#) 섹션을 참조하십시오.

Linux를 실행하는 Amazon EC2 인스턴스

상세 수준	포함된 지표
기본	Mem: mem_used_percent Swap: swap_used_percent
표준	CPU: cpu_usage_idle, cpu_usage_iowait, cpu_usage_user, cpu_usage_system Disk: disk_used_percent, disk_inodes_free, 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

상세 수준	포함된 지표
	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_idle, cpu_usage_guest, cpu_usage_iowait, cpu_usage_steal, 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  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: 사용 중인 메모리 % 커밋된 바이트  Paging: 페이징된 파일 % 사용량
표준	Memory: 사용 중인 메모리 % 커밋된 바이트  Paging: 페이징된 파일 % 사용량  Processor: 프로세서 % 유휴 시간, 프로세서 % 중단 시간, 프로세서 % 사용자 시간, PhysicalDisk: PhysicalDisk % 디스크 시간 LogicalDisk: LogicalDisk % 사용 가능한 공간
Advanced	Memory: 사용 중인 메모리 % 커밋된 바이트  Paging: 페이징된 파일 % 사용량  Processor: 프로세서 % 유휴 시간, 프로세서 % 중단 시간, 프로세서 % 사용자 시간  LogicalDisk: LogicalDisk % 사용 가능한 공간  PhysicalDisk: PhysicalDisk % 디스크 시간, PhysicalDisk 디스크 쓰기 바이트/초, PhysicalDisk 디스크 읽기 바이트/초, PhysicalDisk 디스크 쓰기/초, PhysicalDisk 디스크 읽기/초  TCP: TCPv4 연결 설정됨, TCPv6 연결 설정됨

Windows Server를 실행하는 온프레미스 서버

상세 수준	포함된 지표
기본	<p>Processor: 프로세서 % 프로세서 시간</p> <p>Paging:페이징된 파일 % 사용량</p> <p>LogicalDisk: LogicalDisk % 사용 가능한 공간</p> <p>PhysicalDisk: PhysicalDisk 디스크 쓰기 바이트/초, PhysicalDisk 디스크 읽기 바이트/초, PhysicalDisk 디스크 쓰기/초, PhysicalDisk 디스크 읽기/초</p> <p>Memory: 사용 중인 메모리 % 커밋된 바이트</p> <p>Network Interface: 전송한 네트워크 인터페이스 바이트/초, 수신한 네트워크 인터페이스 바이트/초, 전송한 네트워크 인터페이스 패킷/초, 수신한 네트워크 인터페이스 패킷/초</p>
표준	<p>Paging: 페이징된 파일 % 사용량</p> <p>Processor: 프로세서 % 프로세서 시간, 프로세서 % 유휴 시간 프로세서 % 중단 시간</p> <p>LogicalDisk: LogicalDisk % 사용 가능한 공간</p> <p>PhysicalDisk: PhysicalDisk % 디스크 시간, PhysicalDisk 디스크 쓰기 바이트/초, PhysicalDisk 디스크 읽기 바이트/초, PhysicalDisk 디스크 쓰기/초, PhysicalDisk 디스크 읽기/초</p> <p>Memory: 사용 중인 메모리 % 커밋된 바이트</p> <p>Network Interface: 전송한 네트워크 인터페이스 바이트/초, 수신한 네트워크 인터페이스 바이트/초, 전송한 네트워크 인터페이스 패킷/초, 수신한 네트워크 인터페이스 패킷/초</p>
Advanced	<p>Paging:페이징된 파일 % 사용량</p> <p>Processor: 프로세서 % 프로세서 시간, 프로세서 % 유휴 시간, 프로세서 % 중단 시간, 프로세서 % 사용자 시간</p> <p>LogicalDisk: LogicalDisk % 사용 가능한 공간</p> <p>PhysicalDisk: PhysicalDisk % 디스크 시간, PhysicalDisk 디스크 쓰기 바이트/초, PhysicalDisk 디스크 읽기 바이트/초, PhysicalDisk 디스크 쓰기/초, PhysicalDisk 디스크 읽기/초</p> <p>Memory: 사용 중인 메모리 % 커밋된 바이트</p> <p>Network Interface: 전송한 네트워크 인터페이스 바이트/초, 수신한 네트워크 인터페이스 바이트/초, 전송한 네트워크 인터페이스 패킷/초, 수신한 네트워크 인터페이스 패킷/초</p> <p>TCP: TCPv4 연결 설정됨, TCPv6 연결 설정됨</p>

## 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 Server를 실행하는 서버에서 다음을 입력합니다.

```
cd "C:\Program Files\Amazon\AmazonCloudWatchAgent"  
amazon-cloudwatch-agent-config-wizard.exe
```

2. 질문에 답하여 서버의 구성 파일을 사용자 지정합니다.
3. 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 정보는 [CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 \(p. 73\)](#) 단원을 참조하십시오.

구성 파일을 로컬에 저장하는 경우 config.json 구성 파일이 현재 작업 디렉터리에 저장됩니다. 그런 후 에이전트를 시작할 때 파일 위치를 지정합니다.

## CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집

CloudWatch 에이전트 구성 파일은 에이전트(agent), 지표(metrics) 및 로그(logs) 등 3개의 섹션으로 구성된 JSON 파일입니다.

- 에이전트 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 지표 섹션을 생략할 수 있습니다.
- logs 섹션은 CloudWatch Logs에 게시되는 로그 파일을 지정합니다. 서버가 Windows Server를 실행하는 경우 Windows Event 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.

JSON 파일을 수동으로 생성하거나 편집하면 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.

### CloudWatch 에이전트 구성 파일: 에이전트 섹션

agent 섹션에는 아래 나열된 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 지표의 경우 이 값이 재정의될 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 정보는 [고분해능 지표 \(p. 42\)](#) 단원을 참조하십시오.

기본값은 60입니다.

- `region` – Amazon EC2 인스턴스가 모니터링될 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 `awscloudwatchagent` 프로필로부터 리전을 읽습니다.

- 지표 및 로그를 다른 AWS 계정에 전송할 때는 자격 증명 –을 통해 사용할 IAM 역할을 지정합니다. 지정된 경우 이 필드는 1개의 파라미터 `role_arn`를 포함합니다.
  - 지표 및 로그를 다른 AWS 계정에 전송할 때는 역할 `arn` –을 통해 인증에 사용할 IAM 역할의 ARN을 지정합니다. 자세한 정보는 [다른 AWS 계정에 지표 및 로그 전송 \(p. 142\)](#) 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 `stderr`에 저장됩니다. 이 옵션을 지정하지 않으면 다음과 같이 기본 위치가 사용됩니다.
  - Linux: `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`
  - Windows Server 2003 이상의 Windows Server 버전: `c:\ProgramData\Amazon\CloudWatchAgent\Logs\amazon-cloudwatch-agent.log`

#### Tip

로그가 쌓여 디스크를 채우지 않도록 이 파일에 대해 로그 회전을 설정하는 것이 좋습니다.

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
}
```

## CloudWatch 에이전트 구성 파일: 지표 섹션

Linux 또는 Windows Server를 실행하는 서버의 경우 `metrics` 섹션에 다음 필드가 들어 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. 차원마다, 키 값 쌍을 지정해야 합니다. 여기서 키는 Amazon EC2 차원인 `ImageID: image-id`, `InstanceId: instance-id`, `InstanceType: instance-type` 또는 `AutoScalingGroupName: AutoScaling-group-name`과 일치합니다.

Amazon EC2 메타데이터에 좌우되는 값을 지정하고 프록시를 사용할 경우에는 서버에서 Amazon EC2의 엔드포인트에 액세스할 수 있어야 합니다. 이러한 엔드포인트에 대한 자세한 정보는 Amazon Web Services 일반 참조의 [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)를 참조하십시오.

- `aggregation_dimensions` – 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 전송하는 엔드포인트로 사용할 FIPS 엔드포인트 또는 프라이빗 연결을 지정합니다. 이 옵션을 지정하고 프라이빗 연결을 설정하면 Amazon VPC 엔드포인트로 지표를 전송할 수 있습니다. 자세한 정보는 [Amazon VPC란 무엇입니까?](#)를 참조하십시오.

`endpoint_override`의 값은 URL인 문자열이어야 합니다.

- `metrics_collected` – 필수 사항. StatsD 또는 collectd를 통해 수집된 사용자 지정 지표를 포함하여 수집해야 할 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 상관 없이 버퍼 내 지표의 크기가 40KB에 도달하거나 지표 개수가 20개가 되면 지표가 바로 서버로 전송됩니다.

기본값은 60입니다.

- 지표를 다른 AWS 계정에 전송할 때는 자격 증명 `-`을 통해 사용할 IAM 역할을 지정합니다. 지정된 경우 이 필드는 1개의 파라미터 `role_arn`를 포함합니다.
  - 지표를 다른 AWS 계정에 전송할 때는 역할 `arn` -을 통해 인증에 사용할 IAM 역할의 ARN을 지정합니다. 자세한 정보는 [다른 AWS 계정에 지표 및 로그 전송 \(p. 142\)](#) 단원을 참조하십시오. 여기에 지정한 경우 이를 통해 구성 파일(해당 시)의 `agent` 섹션에 지정되어 있는 `role_arn`을 재정의합니다.

## Linux

Linux를 실행하는 서버에서는 구성 파일의 `metrics_collected` 섹션에 다음 필드가 포함될 수도 있습니다.

- `collectd` – 선택 사항. collectd 프로토콜을 사용하여 사용자 지정 지표를 검색하려는 것을 지정합니다. collectd 소프트웨어를 사용하여 CloudWatch 에이전트에 지표를 전송합니다. 자세한 정보는 [collectd로 사용자 지정 지표 검색 \(p. 137\)](#) 단원을 참조하십시오.
- `cpu` – 선택 사항. CPU 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 3개 필드만큼만 포함할 수 있습니다.
  - `resources` – 선택 사항. CPU별 지표가 수집됨을 나타냅니다. 허용되는 유일한 값은 `*`입니다. 이 필드 및 값을 포함하는 경우 CPU별 지표가 수집됩니다.
  - `totalcpu` – 선택 사항. 모든 CPU 코어에서 집계되는 CPU 지표를 보고할지 여부를 지정합니다. 기본값은 `true`입니다.
  - `measurement` – 수집될 CPU 지표 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 [MetricDatum](#)의 `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 값을 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 정보는 [고분해능 지표 \(p. 42\)](#) 단원을 참조하십시오.

- `append_dimensions` – 선택 사항. CPU 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.
- `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 지표 어레이를 지정합니다. 가능한 값은 `free`, `total`, `used`, `used_percent`, `inodes_free`, `inodes_used`, `inodes_total`입니다. `disk`를 포함하는 경우 이 필드는 필수입니다.

각 `disk` 지표에 대한 기본 단위를 보려면 [Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 \(p. 144\)](#) 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 [MetricDatum](#)의 `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.
- `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 정보는 [고분해능 지표 \(p. 42\)](#) 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. `diskio` 지표가 수집됨을 나타냅니다. 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` 지표 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 [MetricDatum](#)의 `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 정보는 [고분해능 지표 \(p. 42\)](#) 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 필드 하나를 포함할 수 있습니다.
- `measurement` – 수집될 스왑 지표 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 \(p. 144\)](#) 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 [MetricDatum](#)의 `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 정보는 [고분해능 지표 \(p. 42\)](#) 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. 고분해능 지표로 수집됩니다.
- `mem` – 선택 사항. 메모리 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 필드 하나를 포함할 수 있습니다.
- `measurement` – 수집될 스왑 지표 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 \(p. 144\)](#) 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 [MetricDatum](#)의 `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 정보는 [고분해능 지표 \(p. 42\)](#) 단원을 참조하십시오.

- `append_dimensions` – 선택 사항. `mem` 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.
- `net` – 선택 사항. 네트워크 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 2개 필드만큼만 포함할 수 있습니다.

- resources – 선택 사항. 네트워크 인터페이스 어레이를 지정하는 경우, CloudWatch가 해당 인터페이스에서만 지표를 수집합니다. 그렇지 않으면, 모든 디바이스의 지표가 수집됩니다. 값으로 \*를 지정하여 모든 인터페이스에서 지표를 수집할 수도 있습니다.
- measurement – 수집될 네트워킹 지표 어레이를 지정합니다. 가능한 값은 bytes\_sent, bytes\_recv, drop\_in, drop\_out, err\_in, err\_out, packets\_sent, packets\_recv입니다. net를 포함하는 경우 이 필드는 필수입니다.

각 net 지표에 대한 기본 단위를 보려면 [Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 \(p. 144\)](#) 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 [MetricDatum](#)의 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 정보는 [고분해능 지표 \(p. 42\)](#) 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 지정한 차원도 사용됩니다.
- netstat – 선택 사항. TCP 연결 상태 및 UDP 연결 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 필드 하나를 포함할 수 있습니다.
  - measurement – 수집될 netstat 지표 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\_socket입니다. netstat를 포함하는 경우 이 필드는 필수입니다.

각 netstat 지표에 대한 기본 단위를 보려면 [Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표 \(p. 144\)](#) 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 [MetricDatum](#)의 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 정보는 [고분해능 지표 \(p. 42\)](#) 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 지정한 차원도 사용됩니다.
- processes – 선택 사항. 프로세스 지표가 수집됨을 나타냅니다. 이 단원은 리눅스 인스턴스에만 유효합니다. 이 섹션에는 다음의 필드 하나를 포함할 수 있습니다.
  - measurement – 수집될 프로세스 지표 어레이를 지정합니다. 가능한 값은 blocked, dead, idle, paging, running, sleeping, stopped, total, total\_threads, wait, zombies입니다. processes를 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 [MetricDatum](#)의 `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 정보는 [고분해능 지표 \(p. 42\)](#) 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 지정된 차원도 사용됩니다.
- `procstat` – 선택 사항. 개별 프로세스에서 지표를 검색하려 한다고 지정합니다. 자세한 정보는 [procstat 플러그인을 사용하여 프로세스 지표 수집 \(p. 128\)](#) 단원을 참조하십시오.
- `statsd` – 선택 사항. StatsD 프로토콜을 사용하여 사용자 지정 지표를 검색하려 한다는 것을 지정합니다. CloudWatch 에이전트는 프로토콜용 데몬으로 작동합니다. 표준 StatsD 클라이언트를 사용하여 CloudWatch 에이전트에 지표를 전송합니다. 자세한 정보는 [StatsD로 시작하는 사용자 지정 지표 검색 \(p. 136\)](#)를 참조하십시오.

다음은 Linux 서버용 `metrics` 섹션의 예입니다. 이 예에서는 3개의 CPU 지표, 3개의 `netstat` 지표 및 3개의 프로세스 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
    },
    "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 만들 수 있습니다. 사용 가능한 객체 및 카운터에 대한 자세한 정보는 Microsoft Windows 설명서를 참조하십시오.

각 객체의 하위 섹션 내에서 수집할 카운터의 `measurement` 어레이를 지정합니다. `measurement` 어레이는 구성 파일에서 지정하는 객체마다 있어야 합니다. `resources` 필드를 지정하여 지표를 수집해 올 인스턴스의 이름을 지정할 수도 있습니다. 모든 인스턴스에 대해 별도의 지표를 수집하려면 `resources`에 \*를 지정할 수도 있습니다. `resources`를 생략하면 모든 인스턴스의 데이터가 하나의 집합으로 집계됩니다. 인스턴스가 없는 객체의 경우 `resources`를 생략해야 합니다.

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 정보는 [고분해능 지표 \(p. 42\)](#) 단원을 참조하십시오.

- `append_dimensions` – 선택 사항. 이 객체의 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.

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 [MetricDatum](#)의 `unit` 설명에 나와 있는 유효한 CloudWatch 지표 단위여야 합니다.

`metrics_collected`에는 다음 두 가지 다른 섹션을 포함할 수 있습니다.

- `statsd` – StatsD 프로토콜을 사용하여 사용자 지정 지표를 검색할 수 있도록 합니다. CloudWatch 에이전트는 프로토콜용 데몬으로 작동합니다. 표준 StatsD 클라이언트를 사용하여 CloudWatch 에이전트에 지표를 전송합니다. 자세한 정보는 [StatsD로 시작하는 사용자 지정 지표 검색 \(p. 136\)](#) 단원을 참조하십시오.
- `procstat` – 개별 프로세스에서 지표를 검색할 수 있도록 합니다. 자세한 정보는 [procstat 플러그인을 사용하여 프로세스 지표 수집 \(p. 128\)](#) 단원을 참조하십시오.

다음은 Windows Server에서 사용하기 위한 `metrics` 섹션의 예입니다. 이 예에서는 다양한 Windows가 수집되며 컴퓨터는 StatsD 클라이언트로부터 추가 지표를 받도록 설정됩니다.

```
"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 섹션에는 다음 필드가 들어 있습니다.

- logs\_collected – logs 섹션이 포함되어 있는 경우 필수입니다. 서버로부터 수집될 로그 파일 및 Windows 서버 로그를 지정합니다. files 및 windows\_events의 두 필드를 포함할 수 있습니다.
- files – CloudWatch 에이전트에서 어떤 일반 로그 파일을 수집할지 지정합니다. 여기에는 collect\_list라는 하나의 필드만 포함되는데, 이 필드는 이러한 파일을 추가로 정의합니다.
- collect\_list – files가 포함되어 있는 경우 필수입니다. 항목 배열을 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. 슈퍼 별표로 \*\*를 추가한, 규칙과 일치하는 표준 Unix glob는 허용됩니다. 예를 들어 /var/log/\*\*/\*.log를 지정하면 /var/log 디렉터리 트리의 모든 .log 파일이 수집됩니다. 보다 자세한 예는 [Glob Library](#)를 참조하십시오.

표준 별표를 표준 와일드카드로 사용할 수도 있습니다. 예를 들어, /var/log/system.log\*는 /var/log에서 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 파일의 종류에 따라 다른 로그 스트림에 들어가도록 에이전트 구성 파일에 또 다른 로그 스트림 항목을 추가합니다.

- log\_group\_name – 선택 사항. CloudWatch Logs에서 로그 그룹 이름으로 사용할 항목을 지정합니다. 허용되는 문자: a-z, A-Z, 0-9, '\_'(밑줄), '-'(하이픈), '/'(슬래시) 및 '.'(마침표)

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.2017-07-11-14이면 로그 그룹 이름은 /tmp/TestLogFile.log입니다.

- 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}는 EC2 메타데이터에서 호스트 이름을 가져오고, {local\_hostname}는 네트워크 구성 파일에 있는 호스트 이름을 사용합니다.

이 필드를 생략하면 기본값인 {instance\_id}가 사용됩니다. 로그 스트림이 아직 없는 경우 자동으로 생성됩니다.

- timezone – 선택 사항. 로그 이벤트에 타임스탬프를 표시할 때 사용할 시간대를 지정합니다. 유효 값은 UTC와 Local입니다. 기본값은 Local입니다.
- timestamp\_format – 선택 사항. 일반 텍스트와 %로 시작하는 특수 기호를 사용하여 타임스탬프 형식을 지정합니다. 이 필드를 생략한 경우 현재 시간이 사용됩니다. 이 필드를 사용하면 형식의 일부로 다음을 사용할 수 있습니다.

%y

제로 패딩된 10진수 형태의 세기를 제외한 연도

%Y

10진수 형태로 세기가 포함된 연도

%b

로컬 약어 형태의 월

%B

로컬 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)

%Z

시간대(예: PST)

%z

현지 시간대와 UTC 간의 오프셋으로 표시되는 시간대입니다. 예, -0700. 이 형식만 지원됩니다. 예를 들어, -07:00은 유효 형식이 아닙니다.

- `multi_line_start_pattern` – 로그 메시지의 시작을 식별하기 위해 패턴을 지정합니다. 로그 메시지는 패턴과 일치하는 하나의 줄과 패턴과 일치하지 않는 나머지 줄들로 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` – 파일을 정확하게 읽을 수 있도록 로그 파일의 인코딩을 설정합니다. 코딩을 잘못 지정하면 디코딩이 불가능한 문자를 다른 문자들이 대체하면서 데이터 손실이 야기될 수 있습니다.

기본값은 `utf_8`입니다. 아래의 값이 모두 가능합니다.

```
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 Server를 실행하는 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. 이것은 Windows 이벤트 로그 채널 이름과 동일합니다. 예: System, Security, Application 등. 이 필드는 기록할 Windows 이벤트 유형마다 있어야 합니다.
- `event_levels` – 기록할 이벤트 수준을 지정합니다. 기록할 각 수준을 지정해야 합니다. 가능한 값은 INFORMATION, WARNING, ERROR, CRITICAL 및 VERBOSE입니다. 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}`는 EC2 메타데이터에서 호스트 이름을 가져오고, `{local_hostname}`는 네트워크 구성 파일에 있는 호스트 이름을 사용합니다.

이 필드를 생략하면 기본값인 `{instance_id}`가 사용됩니다. 로그 스트림이 아직 없는 경우 자동으로 생성됩니다.

- `event_format` – 선택 사항. CloudWatch Logs에 Windows 이벤트를 저장할 때 사용할 형식을 지정합니다. `xml`은 Windows 이벤트 뷰어에서처럼 XML 형식을 사용합니다. `text`는 기존의 CloudWatch Logs 에이전트 형식을 사용합니다.
- `log_stream_name` – 필수 사항. `collect_list`의 항목에 개별 로그 스트림 이름이 정의되어 있지 않은 Windows 이벤트나 모든 로그에 사용될 기본 로그 스트림 이름을 지정합니다.
- `endpoint_override` – 에이전트가 로그를 전송하는 엔드포인트로 사용할 FIPS 엔드포인트 또는 프라이빗 연결을 지정합니다. 이 옵션을 지정하고 프라이빗 연결을 설정하면 Amazon VPC 엔드포인트로 로그를 전송할 수 있습니다. 자세한 정보는 [Amazon VPC란 무엇입니까?](#)를 참조하십시오.

`endpoint_override`의 값은 URL인 문자열이어야 합니다.



- `force_flush_interval` – 서버로 전송되기 전에 로그가 메모리 버퍼에 남아 있을 최대 시간(초)을 지정합니다. 이 설정과 상관 없이 버퍼 내 로그의 크기가 1MB에 도달하면 로그가 바로 서버로 전송됩니다. 기본값은 5입니다.
- 로그를 다른 AWS 계정에 전송할 때는 자격 증명 –을 통해 사용할 IAM 역할을 지정합니다. 지정된 경우 이 필드는 1개의 파라미터 `role_arn`를 포함합니다.
- 로그를 다른 AWS 계정에 전송할 때는 역할 `arn` –을 통해 인증에 사용할 IAM 역할의 ARN을 지정합니다. 자세한 정보는 [다른 AWS 계정에 지표 및 로그 전송 \(p. 142\)](#) 단원을 참조하십시오. 여기에 지정한 경우 이를 통해 구성 파일(해당 시)의 `agent` 섹션에 지정되어 있는 `role_arn`을 재정의합니다.

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 서버용 전체 에이전트 구성 파일의 예입니다.

```
{
  "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 Server를 실행하는 서버용 전체 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": [
          "customizedInstances"
        ]
      },
      "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"
```

```
}  
}
```

## CloudWatch 에이전트 구성 파일을 시스템 관리자 Parameter Store에 업로드

Amazon EC2 인스턴스에 또는 SSM Agent가 설치되어 있는 온프레미스 서버에 CloudWatch 에이전트를 설치하고 있는 경우, CloudWatch 에이전트 구성 파일을 수동으로 편집한 후에 이를 시스템 관리자 Parameter Store에 업로드해야 합니다. 그러려면 시스템 관리자 `put-parameter` 명령을 사용합니다.

파일을 Parameter Store에 저장할 수 있도록 하려면 충분한 권한이 있는 IAM 역할을 사용해야 합니다. 자세한 정보는 [CloudWatch 에이전트와 함께 사용하기 위한 IAM 역할 및 사용자 생성 \(p. 73\)](#) 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 이상을 실행하는 서버와 Linux 서버에서 지원됩니다.

항목

- [procstat를 사용하도록 CloudWatch 에이전트 구성 \(p. 128\)](#)
- [procstat로 수집한 지표 \(p. 130\)](#)

## 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`: 프로세스가 생성한 프로세스 식별 번호(PID)의 이름으로 프로세스를 선택합니다.
- `exe`: 정규식 일치 규칙을 사용하여 프로세스 이름이 지정한 문자열과 일치하는 프로세스를 선택합니다. 자세한 정보는 [구문](#)을 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 포함했다라도 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 정보는 [LIKE 연산자](#)를 참조하십시오.

어떤 방법을 사용하든 간에 지표 수집 빈도(초)를 지정하는 선택적 `metrics_collection_interval` 파라미터를 포함할 수 있습니다. 이 옵션을 생략하면 기본값인 60초가 사용됩니다.

다음 단원의 예제에서 procstat 섹션은 에이전트 구성 파일의 metrics\_collected 섹션에 포함된 유일한 섹션입니다. 실제 구성 파일에서는 metrics\_collected에 다른 섹션을 포함할 수도 있습니다. 자세한 정보는 [CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 \(p. 111\)](#) 단원을 참조하십시오.

## pid\_file로 구성

다음 예제 procstat 섹션에서는 PID 파일 example1.pid 및 example2.pid를 생성하는 프로세스를 모니터링합니다. 각 프로세스에서는 여러 지표가 수집됩니다. example2.pid를 생성하는 프로세스에서 수집되는 지표는 10초마다 수집되고, example1.pid 프로세스에서 수집되는 지표는 60초마다 수집됩니다(기본 값).

```
{
  "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 plugin과 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 -c와 명령줄이 일치하는 모든 프로세스를 모니터링합니다. 각 프로세스에서 동일한 지표가 수집됩니다.

```
{
  "metrics": {
    "metrics_collected": {
      "procstat": [
        {
          "exe": "config",
          "measurement": [
            "rlimit_memory_data_hard",
            "rlimit_memory_data_soft",
            "rlimit_memory_stack_hard",
            "rlimit_memory_stack_soft"
          ]
        },
        {
          "exe": "-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 지표를 보여줍니다.

CloudWatch 에이전트는 다음 지표 이름의 시작 부분에 procstat를 추가합니다. Linux 서버 또는 Windows Server를 실행하는 서버에서 수집되었는지 여부에 따라 구문이 다릅니다. 예를 들어, cpu\_time 지표는 Linux에서 수집된 경우 procstat\_cpu\_time으로 나타나고, Windows Server에서 수집된 경우에는 procstat cpu\_time으로 나타납니다.

지표 이름	제공 위치	설명
cpu_time	Linux	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



지표 이름	제공 위치	설명
		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
memory_data	Linux	프로세스가 데이터에 사용하는 메모리의 크기입니다.  단위: 바이트
memory_locked	Linux	프로세스가 잠근 메모리의 크기입니다.  단위: 바이트
memory_rss	Linux, Windows Server	프로세스에서 사용 중인 실제 메모리의 크기(실제 상주 메모리)입니다.  단위: 바이트
memory_stack	Linux	프로세스가 사용하는 스택 메모리의 크기입니다.  단위: 바이트
memory_swap	Linux	프로세스가 사용하는 스왑 메모리의 크기입니다.  단위: 바이트

지표 이름	제공 위치	설명
memory_vms	Linux, Windows Server	프로세스가 사용하는 가상 메모리의 크기입니다.  단위: 바이트
read_bytes	Linux, Windows Server	프로세스가 디스크에서 읽어온 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	Linux	프로세스의 컨텍스트가 강제로 전환된 횟수입니다.  단위: 수
voluntary_context_switches	Linux	프로세스의 컨텍스트가 자발적으로 전환된 횟수입니다.  단위: 수
realtime_priority	Linux	프로세스에 대한 실시간 우선 순위의 현재 사용입니다.  단위: 수
nice_priority	Linux	프로세스에 대한 nice priority의 현재 사용입니다.  단위: 수

지표 이름	제공 위치	설명
signals_pending	Linux	프로세스에서 처리하도록 대기 중인 신호 수입니다. 단위: 수
rlimit_cpu_time_hard	Linux	프로세스에 대한 하드 CPU 시간 리소스 제한입니다. 단위: 수
rlimit_cpu_time_soft	Linux	프로세스에 대한 소프트 CPU 시간 리소스 제한입니다. 단위: 수
rlimit_file_locks_hard	Linux	프로세스에 대한 하드 파일 잠금 리소스 제한입니다. 단위: 수
rlimit_file_locks_soft	Linux	프로세스에 대한 소프트 파일 잠금 리소스 제한입니다. 단위: 수
rlimit_memory_data_hard	Linux	데이터에 사용되는 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다. 단위: 바이트
rlimit_memory_data_soft	Linux	데이터에 사용되는 메모리와 관련해 프로세스에 대한 소프트 리소스 제한입니다. 단위: 바이트
rlimit_memory_locked_hard	Linux	잠긴 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다. 단위: 바이트

지표 이름	제공 위치	설명
rlimit_memory_locked_soft	Linux	잠긴 메모리와 관련해 프로세스에 대한 소프트 리소스 제한입니다.  단위: 바이트
rlimit_memory_rss_hard	Linux	물리적 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다.  단위: 바이트
rlimit_memory_rss_soft	Linux	물리적 메모리와 관련해 프로세스에 대한 소프트 리소스 제한입니다.  단위: 바이트
rlimit_memory_stack_hard	Linux	프로세스 스택에 대한 하드 리소스 제한입니다.  단위: 바이트
rlimit_memory_stack_soft	Linux	프로세스 스택에 대한 소프트 리소스 제한입니다.  단위: 바이트
rlimit_memory_vms_hard	Linux	가상 메모리와 관련해 프로세스에 대한 하드 리소스 제한입니다.  단위: 바이트
rlimit_memory_vms_soft	Linux	가상 메모리와 관련해 프로세스에 대한 소프트 리소스 제한입니다.  단위: 바이트
rlimit_nice_priority_hard	Linux	프로세스의 nice priority 값 상한에 대한 하드 리소스 제한입니다.  단위: 수

지표 이름	제공 위치	설명
rlimit_nice_priority_soft	Linux	프로세스의 nice priority 값 상한에 대한 소프트 리소스 제한입니다.  단위: 수
rlimit_num_fds_hard	Linux	프로세스의 파일 설명에 대한 하드 리소스 제한입니다.  단위: 수
rlimit_num_fds_soft	Linux	프로세스의 파일 설명에 대한 소프트 리소스 제한입니다.  단위: 수
rlimit_realtime_priority_hard	Linux	프로세스의 실시간 우선순위 값 상한에 대한 하드 리소스 제한입니다.  단위: 수
rlimit_realtime_priority_soft	Linux	프로세스의 실시간 우선순위 값 상한에 대한 소프트 리소스 제한입니다.  단위: 수
rlimit_signals_pending_hard	Linux	프로세스로 전달 대기 중인 신호 수에 대한 하드 리소스 제한입니다.  단위: 수
rlimit_signals_pending_soft	Linux	프로세스로 전달 대기 중인 신호 수에 대한 소프트 리소스 제한입니다.  단위: 수

지표 이름	제공 위치	설명
num_fds	Linux	프로세스가 사용 중인 파일 서술자 수입니다. 단위: 수
num_threads	Linux, Windows Server	프로세스의 스레드 수입니다. 단위: 수
pid	Linux, Windows Server	프로세스 식별자 (ID)입니다. 단위: 수
pid_count	Linux, Windows Server	프로세스와 관련된 프로세스 ID의 수입니다.  Linux 서버에서 이 지표의 이름은 <code>procstat_lookup_pid_count</code> 이고, Windows Server에서는 <code>procstat_lookup_pid_count</code> 입니다. 단위: 수

## StatsD로 시작하는 사용자 지정 지표 검색

CloudWatch 에이전트와 StatsD 프로토콜을 함께 사용하여 애플리케이션 또는 서비스에서 사용자 지정 지표를 검색할 수 있습니다. StatsD는 Linux 서버와 Windows Server를 실행하는 서버에서 모두 지원됩니다. CloudWatch는 다음의 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 분리된 태그 목록. StatsD 태그는 CloudWatch의 차원과 유사합니다. `env:prod`와 같이 키/값 태그에 콜론을 사용하십시오.

이 형식을 따르는 모든 StatsD 클라이언트를 사용하여 CloudWatch 에이전트에 지표를 전송할 수 있습니다. 사용 가능한 일부 StatsD 클라이언트에 대한 자세한 정보는 [GitHub의 StatsD 클라이언트 페이지](#)를 참조하십시오.

CloudWatch 에이전트를 사용하여 이러한 사용자 지정 지표를 수집하기 위한 기본 구성은 에이전트 구성 파일의 `metrics_collected` 섹션에 `"statsd": {}` 줄을 추가하는 것입니다. 이 줄을 수동으로 추가할 수 있습니다. 마법사를 사용하여 이 구성 파일을 생성하는 경우, 사용자를 위해 이루어집니다. 자세한 정보는 [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#)를 참조하십시오.

StatsD 기본 구성은 대다수 사용자에게 작동합니다. 필요에 따라 에이전트 구성 파일의 `statsd` 섹션에 추가할 수 있는 선택 사항 필드는 3개 있습니다.

- `service_address`: CloudWatch 에이전트가 들어야 하는 서비스 주소. 형식은 `ip:port`입니다. IP 주소를 생략하면 에이전트가 유효한 인터페이스를 모두 수신합니다. UDP 형식만 지원되므로 UDP 접두사를 지정할 필요가 없습니다.

기본값은 `:8125`입니다.

- `metrics_collection_interval`: StatsD 플러그인의 지표 실행 및 수집 간격(초). 기본값은 10초입니다. 범위는 1 ~ 172,000입니다.
- `metrics_aggregation_interval`: CloudWatch에서 지표를 단일 데이터 포인트로 집계하는 간격(초). 기본값은 60초입니다.

예를 들어 `metrics_collection_interval`이 100이고 `metrics_aggregation_interval`이 60이면, CloudWatch에서는 10초마다 데이터를 수집합니다. 매분 후 이 1분의 6개의 데이터 표시값이 단일 데이터 포인트에 집계되어 CloudWatch에 전송됩니다.

범위는 0 ~ 172,000입니다. `metrics_aggregation_interval`를 0에 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 CloudWatch 에이전트에 지표를 전송합니다.

collectd 소프트웨어는 모든 서버에 자동으로 설치되지 않습니다. 자세한 정보는 [collectd 다운로드 페이지](#)를 참조하십시오.

CloudWatch 에이전트를 사용하여 이러한 사용자 지정 지표를 수집하기 위한 기본 구성은 에이전트 구성 파일의 `metrics_collected` 섹션에 `"collectd": {}` 줄을 추가하는 것입니다. 이 줄을 수동으로 추가할 수 있습니다. 마법사를 사용하여 이 구성 파일을 생성하는 경우, 사용자를 위해 이루어집니다. 자세한 정보는 [CloudWatch 에이전트 구성 파일 생성 \(p. 107\)](#)를 참조하십시오.

추가 선택적 파라미터도 사용 가능합니다. collectd를 사용하고 있고 `/etc/collectd/auth_file`를 `collectd_auth_file`로 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 `encrypt`입니다.

`encrypt`는 암호화된 데이터만 수락하도록 지정합니다. `sign`은 서명되고 암호화된 데이터만 수락하도록 지정합니다. `none`은 모든 데이터를 수락하도록 지정합니다. `collectd_auth_file`에 값을 지정하는 경우 가능하면 암호화된 데이터가 복호화됩니다.

자세한 정보는 `collectd` Wiki의 [클라이언트 설정](#) 및 [가능한 상호작용](#)을 참조하십시오.

- `collectd_auth_file` 사용자 이름이 비밀번호에 매핑되는 파일을 설정합니다. 이 비밀번호는 서명을 확인하고 암호화된 네트워크 패킷을 복호화하는 데 사용됩니다. 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_level`이 `none`으로 설정된 경우 이것은 선택 사항입니다. `collectd_security_level`을 `encrypt` 또는 `sign`으로 설정한 경우 `collectd_auth_file`을 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 정보는 <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 에이전트를 사용하는 일반적인 시나리오

다음 단원에서는 CloudWatch 에이전트를 사용할 때 몇 가지 일반적인 구성 및 사용자 지정 작업을 완료하는 방법을 설명합니다.

### 항목

- [CloudWatch 에이전트에 의해 수집되는 지표에 사용자 지정 차원 추가 \(p. 139\)](#)



- 여러 에이전트 구성 파일 (p. 139)
- CloudWatch 에이전트에 의해 수집되는 지표 집계 또는 롤업 (p. 141)
- CloudWatch 에이전트로 고분해능 지표 수집 (p. 141)
- 다른 AWS 계정에 지표 및 로그 전송 (p. 142)

## 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 사항이 적용되도록 해야 함을 잊지 마십시오.

## 여러 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 같이 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 서로 파일 이름이 달라야 하며 초기 구성 파일의 이름과도 달라야 합니다. 에이전트가 이미 실행 중인 구성 파일을 `append-config`에 구성 파일로 사용할 경우, 추가 명령은 첫 번째 구성 파일에 추가되는 것이 아니라 해당 파일의 정보를 겹쳐씹니다. 구성 파일 이름이 같으면 경로가 다르더라도 겹쳐씹니다.

앞의 예는 두 개의 구성 파일을 사용하는 것을 보여주며, 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 "롤업"하려면 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 사항이 적용되도록 해야 함을 잊지 마십시오.

## 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 고분해능 지표이며 10초마다 수집되도록 하려면 `agent` 섹션 아래 `metrics_collection_interval`에서 글로벌 지표 수집 간격 값으로 10을 지정하십시오.

```
"agent": {
  "metrics_collection_interval": 10
}
```

또는 다음의 예에서는 다른 모든 지표는 1분마다 수집되도록 하는 반면 cpu 지표는 1초마다 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 사항이 적용되도록 해야 함을 잊지 마십시오.

## 다른 AWS 계정에 지표 및 로그 전송

CloudWatch 에이전트를 통해 지표, 로그 또는 이 두 가지를 다른 AWS 계정에 전송하려면 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 두 개의 별도의 `role_arn` 문자열을 지정할 수 있는데 하나는 지표를 전송할 때 사용하기 위한 문자열이고 다른 하나는 로그를 전송할 때 사용하기 위한 것입니다.

구성 파일의 `agent` 섹션 부분에 대한 다음의 예에서는 지표와 로그를 다른 AWS 계정에 전송할 때 `CrossAccountAgentRole`를 사용하여 에이전트를 설정합니다.

```
{
  "agent": {
    "credentials": {
      "role_arn": "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 역할은 모두 `CloudWatchAgentServerPolicy`을 포함해야 합니다. 이 정책을 역할에 부여하는 방법에 대한 자세한 정보는 [Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 \(p. 74\)](#) 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 역할은 전송 계정에서 사용하는 IAM 역할을 인식하도록 다음 정책을 포함해야 합니다. 역할을 편집할 때는 IAM 콘솔의 Trust relationships(신뢰 관계) 탭에 이 정책을 추가합니다. 이 정책을 편집하는 대상 계정의 역할은 [Amazon EC2 인스턴스에서 CloudWatch 에이전트와 함께 사용할 IAM 역할 생성 \(p. 74\)](#)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"
    }
  ]
}
```

## 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. 72) 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 Time으로 지표 이름이 지정됩니다. Windows 성능 모니터 카운터에 대한 자세한 정보는 Microsoft Windows 서버 설명서를 참조하십시오.

CloudWatch 에이전트가 수집하는 지표의 기본 네임스페이스는 cwAgent이지만, 에이전트를 구성할 때 다른 네임스페이스를 지정할 수도 있습니다.

### Linux 인스턴스에서 CloudWatch 에이전트가 수집하는 지표

다음 표에 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
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

지표	설명
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
cpu_usage_idle	CPU가 유휴 상태인 시간 비율 단위: 백분율
cpu_usage_iowait	CPU가 I/O 작업이 완료될 때까지 기다리는 시간 비율입니다. 단위: 백분율

지표	설명
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 비율입니다. 도용 시간은 가상화 환경에서 다른 운영 체제에서 사용된 시간입니다. 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
disk_used	디스크의 사용된 공간입니다. 단위: 바이트
disk_used_percent	사용된 총 디스크 공간 비율입니다. 단위: 백분율



지표	설명
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
diskio_reads	디스크 읽기 작업 수입니다. 단위: 수
diskio_read_bytes	디스크에서 읽어온 바이트 수입니다. 단위: 바이트
diskio_read_time	읽기 요청이 디스크에서 대기하고 있는 시간입니다. 동시에 여러 개의 읽기 요청이 대기하게 되면 숫자가 증가합니다. 예를 들어, 5개의 요청 모두가 평균 100밀리초를 대기하는 경우 500이 보고됩니다. 단위: 밀리초
diskio_writes	디스크 쓰기 작업 수입니다. 단위: 수
diskio_write_bytes	디스크에 기록한 바이트 수입니다. 단위: 바이트
diskio_write_time	쓰기 요청이 디스크에서 대기하고 있는 시간입니다. 동시에 여러 개의 쓰기 요청이 대기하게 되면 숫자가 증가합니다. 예를 들어, 8개의 요청 모두가 평균 1000밀리초를 대기하는 경우 8000이 보고됩니다. 단위: 밀리초
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

지표	설명
mem_cached	파일 캐시에 사용되고 있는 메모리 양입니다. 단위: 바이트
mem_free	사용되지 않고 있는 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
net_bytes_recv	네트워크 인터페이스에서 받은 바이트 수입니다. 단위: 바이트
net_bytes_sent	네트워크 인터페이스가 보낸 바이트 수입니다. 단위: 바이트
net_drop_in	이 네트워크 인터페이스에서 받았지만 삭제된 패킷 수입니다. 단위: 수
net_drop_out	이 네트워크 인터페이스가 전송했지만 삭제된 패킷 수입니다. 단위: 수
net_err_in	이 네트워크 인터페이스가 탐지한 수신 오류 수입니다. 단위: 수
net_err_out	이 네트워크 인터페이스가 탐지한 전송 오류 수입니다. 단위: 수
net_packets_sent	이 네트워크 인터페이스가 보낸 패킷 수입니다. 단위: 수
net_packets_recv	이 네트워크 인터페이스에서 받은 패킷 수입니다. 단위: 수

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	Linux에서 X 상태 코드로 나타나는 "데드(Dead)" 상태인 프로세스 수입니다. 단위: 수
processes_idle	유휴 상태(20초 이상 절전 모드)인 프로세스 수입니다. FreeBSD 인스턴스에서만 사용할 수 있습니다. 단위: 수
processes_paging	Linux에서 W 상태 코드로 나타나는 페이징되고 있는 프로세스 수입니다. 단위: 수
processes_running	R 상태 코드로 나타나는 실행 중인 프로세스 수입니다. 단위: 수
processes_sleeping	S 상태 코드로 나타나는 절전 모드의 프로세스 수입니다. 단위: 수
processes_stopped	T 상태 코드로 나타나는 중지된 프로세스 수입니다. 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	FreeBSD 인스턴스에서 W 상태 코드로 나타나는 페이징되고 있는 프로세스 수입니다. 이 지표는 FreeBSD 인스턴스에서만 사용할 수 있습니다. 단위: 수
processes_zombies	Z 상태 코드로 나타나는 좀비 프로세스 수입니다. 단위: 수
swap_free	사용되지 않고 있는 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 에이전트의 문제 해결

다음 정보를 사용하면 CloudWatch 에이전트의 문제를 해결하는 데 도움이 됩니다.

### 항목

- [CloudWatch 에이전트 명령줄 파라미터 \(p. 151\)](#)
- [Run Command를 사용한 CloudWatch 에이전트 설치 실패 \(p. 151\)](#)
- [CloudWatch 에이전트가 시작되지 않음 \(p. 152\)](#)
- [CloudWatch 에이전트가 실행 중인지 확인 \(p. 152\)](#)
- [지표가 저장되는 위치 \(p. 153\)](#)
- [에이전트가 시작되지 않고 오류에 Amazon EC2 리전이 언급되어 있음 \(p. 153\)](#)
- [Windows Server에서 자격 증명을 찾을 수 없음 \(p. 153\)](#)
- [CloudWatch 에이전트 파일 및 위치 \(p. 153\)](#)
- [CloudWatch 에이전트에 의해 생성되는 로그 \(p. 154\)](#)
- [CloudWatch 에이전트 중지 및 다시 시작 \(p. 154\)](#)

## 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 정보는 [SSM Agent 설치 및 구성](#)을 참조하십시오.

## CloudWatch 에이전트가 시작되지 않음

CloudWatch 에이전트가 시작되지 않는 경우 구성에 문제가 있을 수 있습니다. 구성 정보는 configuration-validation.log 파일에 기록되어 있습니다. 이 파일은 Linux 서버의 /opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log 및 Windows Server를 실행하는 서버의 %Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log에 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 Command를 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 command]를 선택합니다.
4. [Command document] 목록에서 [AmazonCloudWatch-ManageAgent]를 선택합니다.
5. [Target] 영역에서, 확인할 인스턴스를 선택합니다.
6. [Action] 목록에서 [status]를 선택합니다.
7. [Optional Configuration Source] 및 [Optional Configuration Location]을 빈 상태로 둡니다.
8. [Run]을 선택합니다.

에이전트가 실행 중인 경우 결과가 다음과 유사하게 표시됩니다.

```
{
  "status": "running",
  "starttime": "2017-12-12T18:41:18",
  "version": "1.73.4"
}
```

에이전트가 중지되어 있는 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 Server를 실행하는 서버의 경우, 관리자로서 PowerShell에 다음을 입력합니다.

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2 -a status
```

## 지표가 저장되는 위치

CloudWatch 에이전트가 실행되어 왔지만, 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 보이지 않는 경우, 구성 파일을 확인하여 사용하고 있는 네임스페이스를 확인하십시오.

CloudWatch 에이전트 패키지를 처음 다운로드하는 경우, 에이전트 구성 파일이 `amazon-cloudwatch-agent.json`입니다. 이 파일은 구성 마법사를 실행한 디렉터리에 있습니다. 아니면 이를 다른 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 정보는 [CloudWatch 에이전트 구성 파일: 지표 섹션 \(p. 112\)](#) 섹션을 참조하십시오.

## 에이전트가 시작되지 않고 오류에 Amazon EC2 리전이 언급되어 있음

에이전트가 시작되지 않고 오류 메시지에 Amazon EC2 리전 엔드포인트가 언급되어 있으면 이 액세스에 권한을 부여하지 않고 Amazon EC2 엔드포인트에 액세스해야 하도록 구성했을 수 있습니다.

예를 들어, 에이전트 구성 파일에 Amazon EC2 메타데이터에 좌우되는 `append_dimensions` 파라미터의 값을 지정하고 프록시를 사용할 경우에는 서버에서 Amazon EC2의 엔드포인트에 액세스할 수 있어야 합니다. 이러한 엔드포인트에 대한 자세한 정보는 Amazon Web Services 일반 참조의 [Amazon Elastic Compute Cloud\(Amazon EC2\)](#)를 참조하십시오.

## Windows Server에서 자격 증명을 찾을 수 없음

Windows Server에서 자격 증명에 `$SystemDrive\Users\Administrator\.aws`에 있지 않거나(Windows Server 2008 또는 Windows Server 2012의 경우), `$SystemDrive\Documents and Settings\Administrator\.aws`에 있지 않으면(Windows Server 2003의 경우) `common.toml`에 `shared_credential_file` 옵션을 사용하여 자격 증명 경로를 직접 지정할 수 있습니다.

자격 증명 파일이 없는 경우 만들어야 합니다. 자세한 정보는 [CloudWatch 에이전트용 일반 구성 및 명명된 프로필 수정 \(p. 91\)](#) 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.	<code>/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl</code> 또는 <code>/usr/bin/amazon-cloudwatch-agent-ctl</code>	<code>\$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1</code>
에이전트가 작성하는 로그 파일입니다. 고객 지원 센터에 문의할 경우 이 파일을 첨부해야 할 수 있습니다.	<code>/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log</code> 또는 <code>/var/log/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.log</code>	<code>\$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log</code>

File	Linux 위치	Windows Server 위치
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
마법사가 생성한 직후에 에이전트를 구성할 때 사용되는 JSON 파일입니다. 자세한 내용은 <a href="#">CloudWatch 에이전트 구성 파일 생성 (p. 107)</a> 단원을 참조하십시오.	/opt/aws/amazon-cloudwatch-agent/bin/config.json	\$Env:ProgramData\Amazon\AmazonCloudWatchAgent\config.json
이 구성 파일을 Parameter Store에서 다운로드한 경우에 에이전트 구성에 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 파일에 있습니다. 이 파일은 Linux 서버의 /opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log 및 Windows Server를 실행하는 서버의 \$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log에 있습니다.

amazon-cloudwatch-agent.log 파일에 추가 세부 정보를 기록하도록 에이전트를 구성할 수 있습니다. 에이전트 구성 파일의 에이전트 섹션에서 디버그 필드를 true로 설정한 다음, CloudWatch 에이전트를 다시 구성하고 다시 시작해야 합니다. 이 추가 정보 기록을 비활성화하려면 [debug] 필드를 [false]로 설정하고 에이전트를 다시 구성하고 다시 시작하십시오. 자세한 정보는 [CloudWatch 에이전트 구성 파일을 수동으로 생성 또는 편집 \(p. 111\)](#) 단원을 참조하십시오.

## CloudWatch 에이전트 중지 및 다시 시작

AWS 시스템 관리자 또는 명령줄을 사용하여 CloudWatch 에이전트를 수동으로 중지할 수 있습니다. 이를 수동으로 중지하면 시스템 재부팅 시 자동으로 시작되지도 않습니다.

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 command]를 선택합니다.
4. [Command document] 목록에서 [AmazonCloudWatch-ManagedAgent]를 선택합니다.
5. 대상 영역에서 CloudWatch 에이전트를 설치한 인스턴스를 선택합니다.
6. [Action] 목록에서 [stop]을 선택합니다.
7. [Optional Configuration Source] 및 [Optional Configuration Location]을 빈 상태로 둡니다.
8. [Run]을 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 Server를 실행하는 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 \(p. 81\)](#) 섹션의 지침을 따르십시오.

# CloudWatch 지표를 게시하는 AWS 서비스

다음 AWS 서비스는 지표를 CloudWatch에 게시합니다. 지표 및 차원에 대한 자세한 정보는 지정된 설명서를 참조하십시오.

서비스	네임스페이스	설명서
Amazon API Gateway	AWS/ApiGateway	<a href="#">Amazon CloudWatch를 사용한 API 실행 모니터링</a>
AppStream 2.0	AWS/AppStream	<a href="#">Amazon AppStream 2.0 리소스 모니터링</a>
AWS Billing and Cost Management	AWS/Billing	<a href="#">알림을 사용한 요금 모니터링</a>
Amazon CloudFront	AWS/CloudFront	<a href="#">CloudWatch를 사용한 CloudFront 활동 모니터링</a>
Amazon CloudSearch	AWS/CloudSearch	<a href="#">Amazon CloudWatch를 사용한 Amazon CloudSearch 도메인 모니터링</a>
Amazon CloudWatch Events	AWS/Events	<a href="#">CloudWatch 지표를 사용한 사용량 모니터링</a>
Amazon CloudWatch Logs	AWS/Logs	<a href="#">CloudWatch 지표를 사용한 사용량 모니터링</a>
AWS CodeBuild	AWS/CodeBuild	<a href="#">AWS CodeBuild 모니터링</a>
Amazon Cognito	AWS/Cognito	<a href="#">고급 보안 측정치 보기</a>
Amazon Connect	AWS/Connect	<a href="#">Amazon CloudWatch 지표의 Amazon Connect 모니터링</a>
AWS Database Migration Service	AWS/DMS	<a href="#">AWS DMS 작업 모니터링</a>
AWS Direct Connect	AWS/DX	<a href="#">Amazon CloudWatch로 모니터링</a>
Amazon DynamoDB	AWS/DynamoDB	<a href="#">DynamoDB 모니터링</a>
Amazon EC2	AWS/EC2	<a href="#">CloudWatch를 사용한 인스턴스 모니터링</a>
Amazon EC2 스팟 집합	AWS/EC2Spot	<a href="#">스팟 집합에 대한 CloudWatch 지표</a>
Amazon EC2 Auto Scaling	AWS/AutoScaling	<a href="#">CloudWatch를 사용한 Auto Scaling 그룹 및 인스턴스 모니터링</a>
AWS Elastic Beanstalk	AWS/ElasticBeanstalk	<a href="#">환경에 대한 Amazon CloudWatch 사용자 지정 지표 게시</a>

서비스	네임스페이스	설명서
Amazon Elastic Block Store	AWS/EBS	<a href="#">볼륨 상태 모니터링</a>
Amazon Elastic Container Service	AWS/ECS	<a href="#">Amazon ECS CloudWatch 지표</a>
Amazon Elastic File System	AWS/EFS	<a href="#">CloudWatch를 사용한 모니터링</a>
Amazon Elastic Inference	AWS/ElasticInference	<a href="#">CloudWatch 지표를 사용하여 Amazon EI 모니터링</a>
Elastic Load Balancing	AWS/ApplicationELB	<a href="#">Application Load Balancer에 대한 CloudWatch 지표</a>
Elastic Load Balancing	AWS/ELB	<a href="#">Classic Load Balancer에 대한 CloudWatch 지표</a>
Elastic Load Balancing	AWS/NetworkELB	<a href="#">Network Load Balancer에 대한 CloudWatch 지표</a>
Amazon Elastic Transcoder	AWS/ElasticTranscoder	<a href="#">Amazon CloudWatch로 모니터링</a>
Amazon ElastiCache for Memcached	AWS/ElastiCache	<a href="#">CloudWatch 지표를 사용한 사용량 모니터링</a>
Redis용 Amazon ElastiCache	AWS/ElastiCache	<a href="#">CloudWatch 지표를 사용한 사용량 모니터링</a>
Amazon Elasticsearch Service	AWS/ES	<a href="#">CloudWatch를 사용한 클러스터 지표 및 통계 모니터링</a>
Amazon EMR	AWS/ElasticMapReduce	<a href="#">CloudWatch를 사용한 지표 모니터링</a>
AWS Elemental MediaConnect	AWS/MediaConnect	<a href="#">Amazon CloudWatch를 사용한 AWS Elemental MediaConnect 모니터링</a>
AWS Elemental MediaConvert	AWS/MediaConvert	<a href="#">CloudWatch 지표</a>
AWS Elemental MediaPackage	AWS/MediaPackage	<a href="#">CloudWatch 지표</a>
AWS Elemental MediaTailor	AWS/MediaTailor	<a href="#">Amazon CloudWatch를 사용한 AWS Elemental MediaTailor 모니터링</a>
Amazon FSx for Lustre	AWS/FSx	<a href="#">Amazon FSx for Lustre 모니터링</a>
Amazon GameLift	AWS/GameLift	<a href="#">CloudWatch를 사용한 Amazon GameLift 모니터링</a>
AWS Glue	AWS/Glue	<a href="#">CloudWatch 지표를 사용한 AWS Glue 모니터링</a>
Amazon Inspector	AWS/Inspector	<a href="#">CloudWatch를 사용한 Amazon Inspector 모니터링</a>

서비스	네임스페이스	설명서
AWS IoT	AWS/IoT	<a href="#">Amazon CloudWatch로 모니터링</a>
AWS IoT Analytics	AWS/IoTAnalytics	<a href="#">네임스페이스, 지표 및 차원</a>
AWS IoT Things Graph	AWS/ThingsGraph	<a href="#">지표</a>
AWS Key Management Service	AWS/KMS	<a href="#">CloudWatch를 사용한 모니터링</a>
Amazon Kinesis Data Analytics	AWS/KinesisAnalytics	Kinesis Data Analytics: <a href="#">CloudWatch를 사용하여 모니터링</a> Java 애플리케이션용 Kinesis Data Analytics: <a href="#">Amazon Kinesis Data Analytics 지표 및 차원 보기</a>
Amazon Kinesis Data Firehose	AWS/Firehose	<a href="#">CloudWatch 지표를 사용한 Kinesis Data Firehose 모니터링</a>
Amazon Kinesis Data Streams	AWS/Kinesis	<a href="#">Amazon CloudWatch를 사용한 Amazon Kinesis Data Streams 모니터링</a>
Amazon Kinesis 비디오 스트림	AWS/KinesisVideo	<a href="#">CloudWatch를 사용한 Kinesis 비디오 스트림 지표 모니터링</a>
AWS Lambda	AWS/Lambda	<a href="#">AWS Lambda 지표</a>
Amazon Lex	AWS/Lex	<a href="#">CloudWatch를 사용한 Amazon Lex 모니터링</a>
Amazon Machine Learning	AWS/ML	<a href="#">CloudWatch 지표를 사용한 Amazon ML 모니터링</a>
Amazon Managed Streaming for Kafka	AWS/Kafka	<a href="#">Amazon CloudWatch를 사용한 Amazon MSK 모니터링</a>
Amazon MQ	AWS/AmazonMQ	<a href="#">Amazon CloudWatch를 사용한 Amazon MQ 브로커 모니터링</a>
Amazon Neptune	AWS/Neptune	<a href="#">CloudWatch를 사용한 Neptune 모니터링</a>
AWS OpsWorks	AWS/OpsWorks	<a href="#">Amazon CloudWatch를 사용한 스택 모니터링</a>
Amazon Polly	AWS/Polly	<a href="#">CloudWatch와 Amazon Polly의 통합</a>
Amazon Redshift	AWS/Redshift	<a href="#">Amazon Redshift 성능 데이터</a>
Amazon Relational Database Service	AWS/RDS	<a href="#">Amazon CloudWatch를 사용한 모니터링</a>
Amazon Route 53	AWS/Route53	<a href="#">Amazon Route 53 모니터링</a>
Amazon SageMaker	AWS/SageMaker	<a href="#">CloudWatch를 사용한 Amazon SageMaker 모니터링</a>
AWS Shield Advanced	AWS/DDoSProtection	<a href="#">CloudWatch를 사용한 모니터링</a>

서비스	네임스페이스	설명서
Amazon Simple Email Service	AWS/SES	<a href="#">CloudWatch에서 Amazon SES 이벤트 데이터 가져오기</a>
Amazon Simple Notification Service	AWS/SNS	<a href="#">CloudWatch를 사용한 Amazon SNS 모니터링</a>
Amazon Simple Queue Service	AWS/SQS	<a href="#">CloudWatch를 사용한 Amazon SQS 대기열 모니터링</a>
Amazon Simple Storage Service	AWS/S3	<a href="#">Amazon CloudWatch를 사용한 지표 모니터링</a>
Amazon Simple Workflow Service	AWS/SWF	<a href="#">CloudWatch에 대한 Amazon SWF 지표</a>
AWS Step Functions	AWS/States	<a href="#">CloudWatch를 사용한 Step Functions 모니터링</a>
AWS Storage Gateway	AWS/StorageGateway	<a href="#">게이트웨이 및 리소스 모니터링</a>
Amazon Textract	AWS/Textract	<a href="#">Amazon Textract용 CloudWatch 지표</a>
Amazon Translate	AWS/Translate	<a href="#">Amazon Translate에 대한 CloudWatch 지표 및 차원</a>
AWS Trusted Advisor	AWS/TrustedAdvisor	<a href="#">CloudWatch를 사용한 Trusted Advisor 경고 생성</a>
Amazon VPC	AWS/NATGateway	<a href="#">CloudWatch를 사용한 NAT 게이트웨이 모니터링</a>
Amazon VPC	AWS/TransitGateway	<a href="#">해당 Transit Gateway용 CloudWatch 지표</a>
Amazon VPC	AWS/VPN	<a href="#">CloudWatch를 사용한 모니터링</a>
AWS WAF	WAF	<a href="#">CloudWatch를 사용한 모니터링</a>
Amazon WorkSpaces	AWS/WorkSpaces	<a href="#">CloudWatch 지표를 사용한 WorkSpaces 모니터링</a>

# AWS SDK Metrics를 사용하여 애플리케이션 모니터링

엔터프라이즈 고객은 CloudWatch 에이전트와 AWS SDK Metrics for Enterprise Support(SDK Metrics)를 함께 사용하여 해당 호스트와 클라이언트에서 AWS SDK에 대한 지표를 수집할 수 있습니다. 이러한 지표는 AWS Enterprise Support와 공유됩니다. SDK Metrics는 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 Metrics는 Enterprise Support 구독 고객만 사용할 수 있습니다. 자세한 정보는 [Amazon CloudWatch 지원 센터](#)를 참조하십시오.

AWS 서비스를 직접 호출하고 AWS SDK 최신 버전을 사용하여 빌드한 애플리케이션에서 SDK Metrics를 사용할 수 있습니다.

SDK Metrics는 AWS SDK를 통해 이루어진 호출을 계속하고, AWS SDK를 사용하는 클라이언트 애플리케이션과 동일한 환경에서 실행하는 CloudWatch 에이전트를 사용합니다. 다음 단원에서는 SDK Metrics 데이터를 생성하도록 CloudWatch 에이전트를 활성화하는 데 필요한 단계를 설명합니다. SDK에서 구성하는 필요한 항목에 대한 설명은 SDK 설명서를 참조하십시오.

## 항목

- [AWS SDK Metrics for Enterprise Support에서 수집한 지표와 데이터 \(p. 160\)](#)
- [SDK Metrics에 대해 CloudWatch 에이전트 구성 \(p. 162\)](#)
- [SDK Metrics에 대한 IAM 권한 설정 \(p. 164\)](#)

## AWS SDK Metrics for Enterprise Support에서 수집한 지표와 데이터

SDK Metrics는 애플리케이션으로부터 데이터를 수집한 후 이를 사용하여 CloudWatch에 지표를 전송합니다. 다음 표에는 SDK Metrics에서 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

테스트	유형
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
대상 IP 주소	문자열

다음 표에는 Enterprise Support 고객이 AWS SDK Metrics for Enterprise Support를 사용하여 수집할 수 있는 지표가 나와 있습니다. 이러한 지표는 AWS/SDKMetrics 네임스페이스에 있습니다.

AWS Support 리소스와 기술 계정 관리자는 귀하의 요청을 해결하기 위해 SDK Metrics 데이터에 액세스할 수 있어야 합니다. 혼란스럽거나 예상치 못한 데이터가 애플리케이션 성능에 부정적인 영향을 미치지 않는 경우, 예정된 비즈니스 검토 중에 기술 계정 관리자와 함께 해당 데이터를 검토하고 기다리십시오.

지표	설명
CallCount	해당 코드에서 AWS 서비스에 대해 수행한 호출 중 성공하거나 실패한 총 API 호출 수. CallCount를 기준으로 사용하여 ServerErrorCount 및 ThrottleCount 등과 같은 다른 지표와의 상관관계를 보여줍니다.  단위: 수
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. 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.  단위: 수
EndToEndLatency	애플리케이션이 AWS SDK를 사용하여 호출(재시도 포함)을 수행한 총 시간.  EndToEndLatency를 사용하여 AWS API 호출이 애플리케이션의 전체 지연 시간에 얼마나 기여했는지를 확인할

지표	설명
	<p>수 있습니다. 네트워크, 방화벽 또는 기타 구성 설정 문제로 인해 예상보다 높은 지연 시간이 발생할 수 있습니다. SDK 재시도로 인해 지연 시간이 발생할 수 있습니다.</p> <p>단위: 밀리초</p>
ConnectionErrorCount	<p>서비스 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.</p> <p>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.</p> <p>단위: 수</p>
ServerErrorCount	<p>AWS 서비스의 서버 오류(5xx HTTP 응답 코드)로 인해 실패한 API 호출 수입니다. 이 실패의 원인은 일반적으로 AWS 서비스입니다.</p> <p>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.</p> <p>단위: 수</p>
ThrottleCount	<p>AWS 서비스의 조정(throttling) 때문에 실패한 API 호출의 수입니다.</p> <p>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.</p> <p>단위: 수</p>

SDK Metrics에서 다음 차원을 사용할 수 있습니다.

차원	설명
DestinationRegion	호출의 대상 AWS 리전입니다.
서비스	애플리케이션에서 호출하는 AWS 서비스입니다.

## SDK Metrics에 대해 CloudWatch 에이전트 구성

시작하기 전에 지표를 받을 애플리케이션을 실행하는 EC2 인스턴스에 CloudWatch 에이전트를 설치합니다. 최선의 결과를 위해 반드시 최신 버전의 CloudWatch 에이전트를 사용하십시오. 자세한 정보는 [Amazon EC2 인스턴스에 CloudWatch 에이전트 설치 \(p. 76\)](#)를 참조하십시오.



CloudWatch 에이전트를 설치한 후 SDK Metrics와 연동하도록 구성해야 합니다. 가장 쉬운 방법은 AWS 시스템 관리자를 사용하는 것이지만 수동으로 할 수도 있습니다.

#### 항목

- [AWS 시스템 관리자를 사용하여 SDK Metrics에 대해 CloudWatch 에이전트 구성 \(p. 163\)](#)
- [SDK Metrics에 대해 수동으로 CloudWatch 에이전트 구성 \(p. 163\)](#)

## AWS 시스템 관리자를 사용하여 SDK Metrics에 대해 CloudWatch 에이전트 구성

이 단원에서는 SSM를 사용하여 SDK Metrics와 연동하도록 CloudWatch를 구성하는 방법을 설명합니다. SSM 에이전트에 대한 자세한 정보는 [SSM Agent 설치 및 구성](#)을 참조하십시오.

SSM를 사용하여 SDK Metrics를 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-UpdateSSMAgent를 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 AmazonCSM으로 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 Metrics에 대해 수동으로 CloudWatch 에이전트 구성

이 단원에서는 SDK Metrics와 연동하도록 CloudWatch 에이전트를 수동으로 구성하는 방법을 설명합니다. 단계는 Linux 서버와 Windows Server 실행 서버별로 다릅니다.

#### Linux

Linux 서버에서 SSM를 사용하지 않고 SDK Metrics를 구성하려면 SSH를 사용하여 호스트에 `sudo` 권한으로 연결해야 합니다. 먼저 Amazon EC2 인스턴스에 로그인합니다. 다음 콘텐츠를 통해 `/temp/csm.json`이라는 파일을 생성합니다.

```
{ "csm": { "memory_limit_in_mb": 20, "port": 31000 } }
```

그리고 다음 명령을 실행합니다.

```
$ cd /tmp
$ mkdir agent
$ cd agent
```

```
$ wget -q https://s3.amazonaws.com/csm-beta-assets/AgentVersion.txt
$ export AGENT_VERSION=$(cat AgentVersion.txt)
$ wget -q https://s3.amazonaws.com/amazon-cloud-watchagent/linux/amd64/${AGENT_VERSION}/
AmazonCloudWatchAgent.zip
$ unzip -q AmazonCloudWatchAgent.zip
$ sudo ./install.sh
$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
ec2 -s -c file:/tmp/csm.json
$ cd ..
$ rm -rf agent
```

구성을 완료하려면 SDK 설명서를 참조하십시오.

#### Windows Server

Windows Server에서 SSM를 사용하지 않고 SDK Metrics를 구성하려면 관리자 권한으로 서버의 PowerShell에 액세스합니다. 서버에 로그인하여 다음 명령을 실행합니다.

```
# rm -r agent
# mkdir agent
# cd agent
# wget https://s3.amazonaws.com/csm-beta-assets/AgentVersion.txt - Outfile
AgentVersion.txt
# $AGENT_VERSION = Get-Content -Raw AgentVersion.txt
# $AGENT_VERSION = $AGENT_VERSION -replace "`n|`r"
# wget https://s3.amazonaws.com/amazon-cloud-watchagent/windows/amd64/${AGENT_VERSION}/
AmazonCloudWatchAgent.zip - Outfile AmazonCloudWatchAgent.zip
# Expand-Archive AmazonCloudWatchAgent.zip
# cd AmazonCloudWatchAgent
# ./install.ps1
# $InstallDir = "${Env:ProgramFiles}\Amazon\AmazonCloudWatchAgent"
# echo '{"csm":{"memory_limit_in_mb":20, "port":31000}}' > ./AmazonCloudWatch-CsmBeta
# powershell -File "${InstallDir}\amazon-cloudwatch-agent-ctl.ps1" -Action fetch-config -
Mode ec2 -Start -ConfigLocation file:AmazonCloudWatch- CsmBeta
# cd ..\..\
# rm -r agent
```

구성을 완료하려면 SDK 설명서를 참조하십시오.

## SDK Metrics에 대한 IAM 권한 설정

SDK Metrics를 사용하기 위해 권한을 구성하려면 SDK Metrics 프로세스를 허용하는 IAM 정책을 생성한 후 EC2 인스턴스를 관리하는 역할이나 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. 모니터링하려는 EC2 인스턴스를 관리하는 IAM 사용자나 역할에 이 정책을 연결합니다.

# 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 \(p. 166\)](#)
- [지표 게시 \(p. 168\)](#)

## 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 \(p. 166\)](#)
- [2단계: 결제 경보 만들기 \(p. 167\)](#)
- [3단계: 경보 상태 확인 \(p. 168\)](#)
- [4단계: 결제 경보 편집 \(p. 168\)](#)
- [5단계: 결제 경보 삭제 \(p. 168\)](#)

## 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.

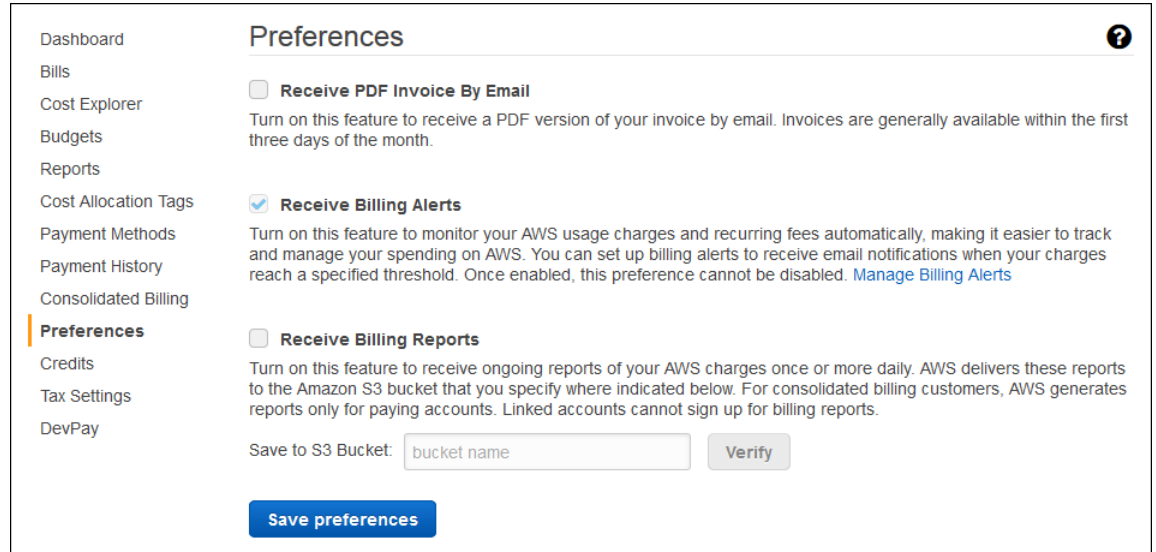
결제 경보를 처음 활성화하고 나서 결제 데이터를 확인하고 결제 경보를 설정할 수 있기까지 약 15분 정도의 시간이 걸립니다.

## 요구 사항

- AWS 계정 루트 사용자 자격 증명으로 로그인을 해야 합니다. IAM 사용자는 AWS 계정에 대해 결제 알림을 활성화할 수 없습니다.
- 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/home?#>에서 Billing and Cost Management 콘솔을 엽니다.
2. 탐색 창에서 [Preferences]를 선택합니다.
3. [Receive Billing Alerts]를 선택합니다.



The screenshot shows the 'Preferences' page in the AWS Billing and Cost Management console. On the left sidebar, 'Preferences' is highlighted. The main content area has three sections: 'Receive PDF Invoice By Email' (unchecked), 'Receive Billing Alerts' (checked), and 'Receive Billing Reports' (unchecked). Below these, there is a 'Save to S3 Bucket' section with a text input field containing 'bucket name' and a 'Verify' button. At the bottom, there is a blue 'Save preferences' button.

4. Save preferences를 선택합니다.

## 2단계: 결제 경고 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 경고 생성의 [결제 경고 만들기 \(p. 70\)](#) 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 [Alarms], [Create Alarm]을 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]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. [send notification to]에서 기존 알림 목록을 선택하거나 새로 만듭니다.

목록을 만들려면 [New list]를 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. [Create Alarm]을 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]에서 경보를 트리거하고 이메일 알림을 전송하기 위해 초과되어야 할 새 금액을 지정합니다.
6. [Save Changes]를 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 설정](#)을 참조하십시오.

#### 작업

- 1단계: 데이터 구성 정의 (p. 169)
- 2단계: CloudWatch에 지표 추가 (p. 169)
- 3단계: CloudWatch에서 통계 얻기 (p. 170)
- 4단계: 콘솔을 사용하여 그래프 보기 (p. 170)

## 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 [Metrics]를 선택합니다.
3. 모든 지표 탭의 검색 상자에 RequestLatency를 입력하고 Enter 키를 누릅니다.
4. RequestLatency 지표에 대한 확인란을 선택합니다. 지표 데이터의 그래프가 위쪽 창에 표시됩니다.

자세한 정보는 [지표 그래프](#) (p. 37) 단원을 참조하십시오.



# 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 Amazon VPC 사용 설명서의 [시작하기](#)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(뭄바이)
- 아시아 태평양(서울)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- EU(프랑크푸르트)
- EU(아일랜드)
- EU(런던)
- EU(파리)
- 남아메리카(상파울루)

## CloudWatch에 대한 VPC 엔드포인트 생성

VPC에서 CloudWatch를 사용하기 시작하려면 CloudWatch에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 엔드포인트의 이름은 `com.amazonaws.Region.monitoring`가 됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트 생성](#)을 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.

# Amazon CloudWatch에 대한 인증 및 액세스 제어

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 \(p. 173\)](#)
- [액세스 제어 \(p. 174\)](#)

## 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 정보는 IAM 사용 설명서의 [IAM 모범 사례 및 관리자 및 그룹 만들기](#)를 참조하십시오.

- **IAM 사용자** – [IAM 사용자](#)는 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 키](#)를 생성할 수 있습니다. [여러 SDK 중 하나](#)를 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 요청을 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 정보는 AWS General Reference의 [서명 버전 4 서명 프로세스](#) 단원을 참조하십시오.

- **IAM 역할** – [IAM 역할](#)은 계정에 만들 수 있는 특정 권한이 있는 또 다른 IAM 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. IAM 역할을 사용하면 AWS 서비스와 리소스에 액세스하는 데 사용할 수 있는 임시 액세스 키를 얻을 수 있습니다. 임시 자격 증명을 가진 IAM 역할은 다음과 같은 상황에서 유용합니다.
- **연합된 사용자 액세스** – IAM 사용자를 만드는 대신 AWS Directory Service, 엔터프라이즈 사용자 디렉터리 또는 웹 자격 증명 공급자(IdP)의 기존 자격 증명을 사용할 수 있습니다. 이러한 사용자를 연합된 사용자라고 합니다. IdP를 통해 액세스를 요청하면 AWS가 연합된 사용자에게 역할을 할당합니다. 자세한 정보는 IAM 사용 설명서의 [연합된 사용자 및 역할](#)을 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 위임](#)을 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 정보는 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 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 정보는 IAM 사용 설명서의 [Amazon EC2에서 실행되는 애플리케이션의 역할 사용하기](#)를 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 \(p. 175\)](#)
- [CloudWatch에 대한 자격 증명 기반 정책\(IAM 정책\) 사용 \(p. 178\)](#)
- [Amazon CloudWatch 권한 참조 문서 \(p. 186\)](#)

## CloudWatch 대시보드 권한 업데이트

2018년 5월 1일에 CloudWatch 대시보드에 액세스하기 위해 필요한 권한이 변경됩니다. 현재, CloudWatch 대시보드를 보기 위해 `cloudwatch:GetMetricData` 권한이 필요하며, 대시보드를 생성 또는 수정하기 위해 `cloudwatch:PutMetricData` 권한이 필요합니다. 5월 1일부터 CloudWatch 콘솔에서 대시보드에 액세스할 때 대시보드 API 작업을 지원하기 위해 2017년에 소개한 다음과 같은 새로운 권한이 대신 필요합니다.

- `cloudwatch:GetDashboard`
- `cloudwatch:ListDashboards`
- `cloudwatch:PutDashboard`
- `cloudwatch:DeleteDashboards`

변경 후 CloudWatch 대시보드에 대한 액세스 권한이 있는지 여부를 확인하려면 CloudWatch 콘솔의 업데이트 메시지에서 권한 확인을 선택합니다. 업데이트 후 확인에서 이러한 권한이 없다고 표시되면 5월 1일 전에 IAM 콘솔을 사용하여 권한을 수정해야 합니다.

CloudWatch 대시보드에 대한 액세스 권한을 보유하려면 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 정보는 [IAM 사용자의 권한 변경](#)을 참조하십시오.

CloudWatch 권한에 대한 자세한 정보는 [Amazon CloudWatch 권한 참조 문서 \(p. 186\)](#)를 참조하십시오.

대시보드 API 작업에 대한 자세한 정보는 Amazon CloudWatch API Reference의 [PutDashboard](#)를 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 사용자)는 관리자 권한이 있는 사용자입니다. 자세한 내용은 IAM 사용 설명서에서 [IAM 모범 사례](#)를 참조하십시오.

권한을 부여하려면 권한을 부여 받을 사용자, 권한 대상이 되는 리소스, 해당 리소스에 허용되는 특정 작업을 결정합니다.

### 항목

- [CloudWatch 리소스 및 작업 \(p. 175\)](#)
- [리소스 소유권 이해 \(p. 176\)](#)
- [리소스 액세스 관리 \(p. 176\)](#)
- [정책 요소 지정: 작업, 효과, 보안 주체 \(p. 177\)](#)
- [정책에서 조건 지정 \(p. 178\)](#)

## CloudWatch 리소스 및 작업

CloudWatch에는 액세스를 제어할 특정 리소스가 없습니다. 따라서 IAM 정책에서 사용할 수 있는 CloudWatch Amazon 리소스 이름(ARN)이 없습니다. 예를 들어 특정 EC2 인스턴스 세트 또는 특정 로드 밸런서에 대해서만 사용자에게 CloudWatch 데이터 액세스 권한을 부여할 수 없습니다. IAM을 사용해 부여된 권한은 CloudWatch에서 사용하거나 모니터링하는 모든 클라우드 리소스에 적용됩니다. 뿐만 아니라 CloudWatch 명령줄 도구에서 IAM 역할을 사용할 수 없습니다.

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 정보는 IAM 사용 설명서의 [ARN](#)을 참조하십시오. CloudWatch Logs ARN에 대한 자세한 정보는 Amazon Web Services 일반 참조의 [Amazon 리소스 이름\(ARN\) 및 AWS 서비스 네임스페이스](#)를 참조하십시오. CloudWatch 작업에 대한 정책의 예는 [CloudWatch에 대한 자격 증명 기반 정책\(IAM 정책\) 사용](#) (p. 178)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: <b>customer-account</b> :action/actions/AWS_EC2.InstanceId.Stop/1.0
Terminate	arn:aws:automate:us-east-1:ec2:terminate	arn:aws:swf:us-east-1: <b>customer-account</b> :action/actions/AWS_EC2.InstanceId.Terminate/1.0
Reboot	해당 사항 없음	arn:aws:swf:us-east-1: <b>customer-account</b> :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 사용할 수 있는 옵션에 대해 설명합니다.

### Note

이 단원에서는 CloudWatch의 맥락에서 IAM을 사용하는 방법에 대해 설명하며, IAM 서비스에 대한 자세한 정보는 다루지 않습니다. 전체 IAM 설명서는 [IAM이란 무엇인가?](#)(출처: IAM 사용 설명서)를 참조하십시오. IAM 정책 구문과 설명에 대한 자세한 내용은 [IAM 사용 설명서](#)에서 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.

### 항목

- [자격 증명 기반 정책\(IAM 정책\)](#) (p. 177)
- [리소스 기반 정책\(IAM 정책\)](#) (p. 177)

## 자격 증명 기반 정책(IAM 정책)

정책을 IAM 자격 증명에 연결할 수 있습니다. 예를 들면,

- 계정에 사용자 또는 그룹에 권한 정책 연결 – metrics와 같은 Amazon CloudWatch 리소스를 생성할 사용자 권한을 부여하려면 권한 정책을 특정 사용자 또는 해당 사용자가 속한 그룹에 연결할 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 주체가 AWS 서비스 보안 주체이기도 합니다.

IAM을 사용하여 권한을 위임하는 방법에 대한 자세한 내용은 [IAM 사용 설명서](#)의 액세스 관리를 참조하십시오.

CloudWatch에서 자격 증명 기반 정책을 사용하는 방법에 대한 자세한 정보는 [CloudWatch에 대한 자격 증명 기반 정책\(IAM 정책\) 사용](#) (p. 178) 단원을 참조하십시오. 사용자, 그룹, 역할 및 권한에 대한 자세한 정보는 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 정보는 [CloudWatch 리소스 및 작업](#) (p. 175) 및 CloudWatch [작업](#)을 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 정보는 [CloudWatch 리소스 및 작업](#) (p. 175)을 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 정보는 IAM 사용 설명서의 [AWS IAM JSON 정책 참조](#)를 참조하십시오.

모든 CloudWatch API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 [Amazon CloudWatch 권한 참조 문서](#) (p. 186) 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 정보는 IAM 사용 설명서의 [조건](#)을 참조하십시오.

조건을 표시하려면 미리 정의된 조건 키를 사용합니다. 각각의 AWS 서비스에서 지원되는 컨텍스트 키 목록과 AWS 전역 정책 키 목록은 IAM 사용 설명서에서 [AWS 서비스 작업과 조건 컨텍스트 키](#) 및 [전역 및 IAM 조건 컨텍스트 키](#)를 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 정보는 [액세스 제어](#) (p. 174) 단원을 참조하십시오.

이 주제의 단원에서는 다음 내용을 학습합니다.

- [CloudWatch 콘솔 사용에 필요한 권한](#) (p. 179)
- [CloudWatch에 대한 AWS 관리형\(미리 정의된\) 정책](#) (p. 181)
- [고객 관리형 정책 예](#) (p. 181)

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 문을 하나 가지고 있지만, 그룹이 요청에서 SSL을 사용하는 경우("aws:SecureTransport": "true")에만 지원됩니다. IAM 정책 문 내의 요소에 대한 자세한 정보는 IAM 사용 설명서의 [정책 요소 지정: 작업, 효과, 보안 주체](#) (p. 177) 및 [IAM 정책 요소 참조](#)를 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

최소 필수 권한보다 더 제한적인 IAM 정책을 만들면 콘솔에서는 해당 IAM 정책에 연결된 사용자에게 의도대로 작동하지 않습니다. 이 사용자가 CloudWatch 콘솔을 사용할 수 있도록 하려면 `CloudWatchReadOnlyAccess` 관리형 정책을 사용자에게 연결합니다([CloudWatch에 대한 AWS 관리형 \(미리 정의된\) 정책 \(p. 181\)](#) 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

## 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 설명서에서 [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 고유의 사용자 지정 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

## 고객 관리형 정책 예

이 단원에서는 다양한 CloudWatch 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 \(p. 182\)](#)
- [예제 2: CloudWatch에 대한 읽기 전용 액세스 허용 \(p. 182\)](#)
- [예제 3: Amazon EC2 인스턴스 중지 또는 종료 \(p. 182\)](#)

## 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": "0000000000000000",
  "Resource": [
    "*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "ec2:DescribeInstanceStatus",
    "ec2:DescribeInstances",
    "ec2:StopInstances",
    "ec2:TerminateInstances"
  ],
  "Sid": "0000000000000000",
  "Resource": [
    "*"
  ],
  "Effect": "Allow"
}
]
```

## CloudWatch 경보에 서비스 연결 역할 사용

Amazon CloudWatch의 경우 AWS Identity and Access Management(IAM) [서비스 연결 역할](#)을 사용합니다. 서비스 연결 역할은 CloudWatch에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 CloudWatch에서 사전 정의하며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다.

CloudWatch에 서비스 연결된 역할을 사용하면 필요한 권한을 수동으로 추가할 필요가 없으므로 Amazon EC2 인스턴스를 보다 쉽게 종료, 중지 또는 재부팅할 수 있는 CloudWatch 경보를 설정할 수 있습니다. CloudWatch에서 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 부주의로 삭제할 수 없기 때문에 CloudWatch 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 정보는 [IAM으로 작업하는 AWS 서비스](#)를 참조하고 서비스 연결 역할 옆에 예가 있는 서비스를 찾으십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 [Yes] 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`

## CloudWatch 경보에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management 콘솔에서 처음으로 경보 생성 시 IAM CLI, 혹은 IAM API, CloudWatch는 사용자를 대신해 서비스 연결 역할을 생성합니다.

### Important

이 서비스 연결 역할은 이 역할이 지원하는 기능을 사용하는 다른 서비스에서 작업을 완료했을 경우 계정에 나타날 수 있습니다. 또한 CloudWatch 서비스가 서비스 연결 역할을 지원하기 시작한 January 1, 2017 이전에 이 서비스를 사용 중이었다면 CloudWatch에서 사용자 계정에 `AWSServiceRoleForCloudWatchEvents` 역할을 이미 생성했습니다. 자세한 정보는 [내 IAM 계정에 표시되는 새 역할](#)을 참조하십시오.

자세한 정보는 IAM 사용 설명서의 [서비스 연결 역할 생성](#)을 참조하십시오.

## CloudWatch 경보에 대한 서비스 연결 역할 편집

CloudWatch는 `AWSServiceRoleForCloudWatchEvents` 역할 편집을 허용하지 않습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다. 단, IAM을 사용하여 `AWSServiceRoleForCloudWatchEvents` 역할 설명을 편집할 수는 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]의 맨 오른쪽에서 [Edit]를 선택합니다.
4. 상자에 새 설명을 입력하고 [Save]를 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`

## CloudWatch 경보에 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 [Summary] 페이지에서 [Access Advisor]를 선택하고 서비스 연결 역할의 최근 활동을 검토합니다.

### Note

CloudWatch에서 AWSServiceRoleForCloudWatchEvents 역할을 사용하는지 잘 모를 경우에는 역할을 삭제해 보십시오. 서비스에서 역할을 사용하는 경우에는 삭제가 안 되어 역할이 사용 중인 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. 이름이나 행 자체가 아닌 AWSServiceRoleForCloudWatchEvents 옆의 확인란을 선택합니다.
3. [Role actions]에 대해 [Delete role]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 [Yes, Delete]를 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 [View details] 또는 [View Resources]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 `DeleteServiceLinkedRole`를 호출합니다. 요청에 `AWSServiceRoleForCloudWatchEvents` 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 `GetServiceLinkedRoleDeletionStatus`를 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.

# Amazon CloudWatch 권한 참조 문서

IAM 자격 증명에 연결할 수 있는 액세스 제어(p. 174) 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 컨텍스트 키](#)를 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)
<a href="#">DeleteAlarms</a>	<code>cloudwatch:DeleteAlarms</code> 경보를 삭제하는 데 필요합니다.
<a href="#">DeleteDashboards</a>	<code>cloudwatch:DeleteDashboards</code> 대시보드를 삭제하는 데 필요합니다.
<a href="#">DescribeAlarmHistory</a>	<code>cloudwatch:DescribeAlarmHistory</code> 경보 기록을 보는 데 필요합니다.
<a href="#">DescribeAlarms</a>	<code>cloudwatch:DescribeAlarms</code> 이름으로 경보 정보를 검색하는 데 필요합니다.
<a href="#">DescribeAlarmsForMetric</a>	<code>cloudwatch:DescribeAlarmsForMetric</code> 지표에 대한 경보를 보는 데 필요합니다.
<a href="#">DisableAlarmActions</a>	<code>cloudwatch:DisableAlarmActions</code> 경보 작업을 비활성화하는 데 필요합니다.
<a href="#">EnableAlarmActions</a>	<code>cloudwatch:EnableAlarmActions</code> 경보 작업을 활성화하는 데 필요합니다.
<a href="#">GetDashboard</a>	<code>cloudwatch:GetDashboard</code> 기존 대시보드에 대한 데이터를 표시하는 데 필요합니다.
<a href="#">GetMetricData</a>	<code>cloudwatch:GetMetricData</code> 커다란 지표 데이터 배치를 가져와 이 데이터의 지표 수식을 계산해야 합니다.
<a href="#">GetMetricStatistics</a>	<code>cloudwatch:GetMetricStatistics</code>

CloudWatchAPI 연산	필요한 권한(API 작업)
	CloudWatch 콘솔의 다른 부분과 대시보드 위젯에서 그래프를 보는 데 필요합니다.
<a href="#">GetMetricWidgetImage</a>	<code>cloudwatch:GetMetricWidgetImage</code>  CloudWatch 지표 1개 이상의 스냅샷 그래프를 비트맵 이미지로 가져오는 데 필요합니다.
<a href="#">ListDashboards</a>	<code>cloudwatch:ListDashboards</code>  계정에서 CloudWatch 대시보드의 목록을 보는 데 필요합니다.
<a href="#">ListMetrics</a>	<code>cloudwatch:ListMetrics</code>  CloudWatch 콘솔 내에서, 그리고 CLI에서 지표 이름을 보거나 검색하는 데 필요합니다. 대시보드 위젯에서 지표를 선택하는 데 필요합니다.
<a href="#">PutDashboard</a>	<code>cloudwatch:PutDashboard</code>  대시보드를 생성하거나 기존 대시보드를 업데이트하는 데 필요합니다.
<a href="#">PutMetricAlarm</a>	<code>cloudwatch:PutMetricAlarm</code>  경보를 생성 또는 업데이트하는 데 필요합니다.
<a href="#">PutMetricData</a>	<code>cloudwatch:PutMetricData</code>  지표를 만드는 데 필요합니다.
<a href="#">SetAlarmState</a>	<code>cloudwatch:SetAlarmState</code>  경보 상태를 수동으로 설정하는 데 필요합니다.

#### CloudWatch 이벤트 API 작업 및 작업에 필요한 권한

CloudWatch 이벤트API 연산	필요한 권한(API 작업)
<a href="#">DeleteRule</a>	<code>events:DeleteRule</code>  규칙을 삭제하는 데 필요합니다.
<a href="#">DescribeRule</a>	<code>events:DescribeRule</code>  규칙에 대한 세부 사항을 나열하는 데 필요합니다.
<a href="#">DisableRule</a>	<code>events:DisableRule</code>  규칙을 비활성화하는 데 필요합니다.
<a href="#">EnableRule</a>	<code>events:EnableRule</code>  규칙을 활성화하는 데 필요합니다.
<a href="#">ListRuleNamesByTarget</a>	<code>events:ListRuleNamesByTarget</code>  대상과 연관된 규칙을 나열하는 데 필요합니다.

CloudWatch 이벤트API 연산	필요한 권한(API 작업)
<a href="#">ListRules</a>	<code>events:ListRules</code> 계정에서 모든 그룹을 나열하는 데 필요합니다.
<a href="#">ListTargetsByRule</a>	<code>events:ListTargetsByRule</code> 규칙과 연관된 모든 대상을 나열하는 데 필요합니다.
<a href="#">PutEvents</a>	<code>events:PutEvents</code> 규칙에 일치시킬 수 있는 사용자 지정 이벤트를 추가하는 데 필요합니다.
<a href="#">PutRule</a>	<code>events:PutRule</code> 규칙을 생성 또는 업데이트하는 데 필요합니다.
<a href="#">PutTargets</a>	<code>events:PutTargets</code> 규칙에 대상을 추가하는 데 필요합니다.
<a href="#">RemoveTargets</a>	<code>events:RemoveTargets</code> 규칙에서 대상을 제거하는 데 필요합니다.
<a href="#">TestEventPattern</a>	<code>events:TestEventPattern</code> 특정 이벤트를 기준으로 이벤트 패턴을 테스트하는 데 필요합니다.

#### CloudWatch Logs API 작업 및 작업에 필요한 권한

CloudWatch LogsAPI 연산	필요한 권한(API 작업)
<a href="#">CancelExportTask</a>	<code>logs:CancelExportTask</code> 보류 또는 실행 중인 내보내기 작업을 취소하는 데 필요합니다.
<a href="#">CreateExportTask</a>	<code>logs:CreateExportTask</code> 로그 그룹에서 Amazon S3 버킷으로 데이터를 내보내는 데 필요합니다.
<a href="#">CreateLogGroup</a>	<code>logs:CreateLogGroup</code> 새 보안 그룹을 생성하는 데 필요합니다.
<a href="#">CreateLogStream</a>	<code>logs:CreateLogStream</code> 로그 그룹에서 로그 스트림을 새로 생성하는 데 필요합니다.
<a href="#">DeleteDestination</a>	<code>logs&gt;DeleteDestination</code> 로그 대상을 삭제하고 이에 대한 모든 구독 필터를 비활성화하는 데 필요합니다.

CloudWatch LogsAPI 연산	필요한 권한(API 작업)
<a href="#">DeleteLogGroup</a>	logs:DeleteLogGroup  로그 그룹과 보관되는 모든 연관 로그 이벤트를 삭제하는 데 필요합니다.
<a href="#">DeleteLogStream</a>	logs:DeleteLogStream  로그 스트림과 보관되는 모든 연관 로그 이벤트를 삭제하는 데 필요합니다.
<a href="#">DeleteMetricFilter</a>	logs:DeleteMetricFilter  로그 그룹과 연관된 지표 필터를 삭제하는 데 필요합니다.
<a href="#">DeleteRetentionPolicy</a>	logs:DeleteRetentionPolicy  로그 그룹의 보존 정책을 삭제하는 데 필요합니다.
<a href="#">DeleteSubscriptionFilter</a>	logs:DeleteSubscriptionFilter  로그 그룹과 연관된 구독 필터를 삭제하는 데 필요합니다.
<a href="#">DescribeDestinations</a>	logs:DescribeDestinations  계정과 연결된 모든 대상들을 보는 데 필요합니다.
<a href="#">DescribeExportTasks</a>	logs:DescribeExportTasks  계정과 연관된 모든 내보내기 작업들을 보는 데 필요합니다.
<a href="#">DescribeLogGroups</a>	logs:DescribeLogGroups  계정과 연관된 모든 로그 그룹들을 보는 데 필요합니다.
<a href="#">DescribeLogStreams</a>	logs:DescribeLogStreams  로그 그룹과 연관된 모든 로그 스트림을 보는 데 필요합니다.
<a href="#">DescribeMetricFilters</a>	logs:DescribeMetricFilters  로그 그룹과 연관된 모든 지표를 보는 데 필요합니다.
<a href="#">DescribeSubscriptionFilters</a>	logs:DescribeSubscriptionFilters  로그 그룹과 연관된 모든 구독 필터를 보는 데 필요합니다.
<a href="#">FilterLogEvents</a>	logs:FilterLogEvents  로그 그룹 필터 패턴에 따라 로그 이벤트를 정렬하는 데 필요합니다.

CloudWatch LogsAPI 연산	필요한 권한(API 작업)
<a href="#">GetLogEvents</a>	<code>logs:GetLogEvents</code> 로그 스트림에서 로그 이벤트를 검색하는 데 필요합니다.
<a href="#">ListTagsLogGroup</a>	<code>logs:ListTagsLogGroup</code> 로그 그룹과 연결된 태그를 나열하는 데 필요합니다.
<a href="#">PutDestination</a>	<code>logs:PutDestination</code> 대상 로그 스트림(예: Kinesis 스트림)을 생성 또는 업데이트하는 데 필요합니다.
<a href="#">PutDestinationPolicy</a>	<code>logs:PutDestinationPolicy</code> 기존 로그 대상과 연관된 액세스 정책을 생성 또는 업데이트하는 데 필요합니다.
<a href="#">PutLogEvents</a>	<code>logs:PutLogEvents</code> 로그 스트림에서 로그 이벤트 배치를 업로드하는 데 필요합니다.
<a href="#">PutMetricFilter</a>	<code>logs:PutMetricFilter</code> 지표 필터를 생성 또는 업데이트하고 이를 로그 그룹과 연관시키는 데 필요합니다.
<a href="#">PutRetentionPolicy</a>	<code>logs:PutRetentionPolicy</code> 로그 그룹에 로그 이벤트를 유지하는 일수(보존 일수)를 설정하는 데 필요합니다.
<a href="#">PutSubscriptionFilter</a>	<code>logs:PutSubscriptionFilter</code> 구독 필터를 생성 또는 업데이트하고 이를 로그 그룹과 연관시키는 데 필요합니다.
<a href="#">TestMetricFilter</a>	<code>logs:TestMetricFilter</code> 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)
<a href="#">DescribeInstanceStatus</a>	<code>ec2:DescribeInstanceStatus</code> EC2 인스턴스 상태에 대한 세부 정보를 보는 데 필요합니다.
<a href="#">DescribeInstances</a>	<code>ec2:DescribeInstances</code> EC2 인스턴스에 대한 세부 정보를 보는 데 필요합니다.

Amazon EC2API 연산	필요한 권한(API 작업)
<a href="#">RebootInstances</a>	<code>ec2:RebootInstances</code> EC2 인스턴스를 재부팅하는 데 필요합니다.
<a href="#">StopInstances</a>	<code>ec2:StopInstances</code> EC2 인스턴스를 중지하는 데 필요합니다.
<a href="#">TerminateInstances</a>	<code>ec2:TerminateInstances</code> 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	<code>autoscaling:Scaling</code> Auto Scaling 그룹을 확장하는 데 필요합니다.
트리거	<code>autoscaling:Trigger</code> Auto Scaling 작업을 트리거하는 데 필요합니다.

# AWS CloudTrail 사용을 통한 Amazon CloudWatch API 호출 로깅

Amazon CloudWatch는 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 Guide](#)를 참조하십시오.

## 항목

- [CloudTrail의 CloudWatch 정보 \(p. 193\)](#)
- [예제: CloudWatch 로그 파일 항목 \(p. 194\)](#)

## 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 내용은 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 Metrics 작업을 로깅할 수 있습니다. 이러한 작업은 SDK Metrics에서만 사용할 수 있으며 해당 코드에서는 사용할 수 없습니다. 자세한 정보는 [AWS SDK Metrics를 사용하여 애플리케이션 모니터링 \(p. 160\)](#) 단원을 참조하십시오.

- `GetPublishingConfiguration` – SDK Metrics에서 지표 게시 방식을 지정하는 데 사용됩니다.
- `GetPublishingSchema` – SDK Metrics에서 지표 게시 방식을 지정하는 데 사용됩니다.
- `PutPublishingMetrics` – SDK Metrics에서 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 사용자에게 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 정보는 [CloudTrail userIdentity 요소](#)를 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": "ctest2",
        "state": "ENABLED",
        "eventPattern": "{\"source\": [\"aws.ec2\"], \"detail-type\": [\"EC2 Instance State-change Notification\"]}",
        "scheduleExpression": ""
    },
    "responseElements": {
        "ruleArn": "arn:aws:events:us-east-1:123456789012:rule/ctest2"
    },
    "requestID": "e9caf887-8d88-11e5-a331-3332aa445952",
    "eventID": "49d14f36-6450-44a5-a501-b0fcdcdfaeb98",
    "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",
    },
    "eventTime": "2015-11-18T00:11:28Z",
    "eventSource": "logs.amazonaws.com",
    "eventName": "CreateExportTask",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS CloudWatch Logs Console",
    "requestParameters": {
        "description": "",
        "name": "ctest2",
        "state": "ENABLED",
        "eventPattern": "{\"source\": [\"aws.ec2\"], \"detail-type\": [\"EC2 Instance State-change Notification\"]}",
        "scheduleExpression": ""
    },
    "responseElements": {
        "ruleArn": "arn:aws:events:us-east-1:123456789012:rule/ctest2"
    },
    "requestID": "e9caf887-8d88-11e5-a331-3332aa445952",
    "eventID": "49d14f36-6450-44a5-a501-b0fcdcdfaeb98",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-10-07",
    "recipientAccountId": "123456789012"
}
```

```
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "someuser"
  },
  "eventTime": "2016-02-08T06:35:14Z",
  "eventSource": "logs.amazonaws.com",
  "eventName": "CreateExportTask",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
  "requestParameters": {
    "destination": "yourdestination",
    "logGroupName": "yourloggroup",
    "to": 123456789012,
    "from": 0,
    "taskName": "yourtask"
  },
  "responseElements": {
    "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
  },
  "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
  "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
  "eventType": "AwsApiCall",
  "apiVersion": "20140328",
  "recipientAccountId": "123456789012"
}
```

# CloudWatch 제한

CloudWatch에는 다음과 같은 제한이 있습니다.

리소스	기본 제한
작업	5/경보. 이 제한값은 변경될 수 없습니다.
개 경보	고객당 월 10회 무료. 계정당 리전당 5000개.
API 요청	고객당 월 1,000,000회 무료.
사용자 지정 지표	제한 없음.
대시보드	계정당 최대 1000개 대시보드 대시보드 위젯당 최대 100개 지표 전체 위젯에서 대시보드당 최대 500개 지표 이 한도는 변경할 수 없습니다.
<a href="#">DescribeAlarms</a>	9건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. <a href="#">한도 증가를 요청</a> 할 수 있습니다.
<a href="#">DeleteAlarms</a> 요청 <a href="#">DescribeAlarmHistory</a> 요청 <a href="#">DescribeAlarmsForMetric</a> 요청 <a href="#">DisableAlarmActions</a> 요청 <a href="#">EnableAlarmActions</a> 요청 <a href="#">SetAlarmState</a> 요청	이러한 각 작업에 대한 3건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. 이 한도는 변경할 수 없습니다.
차원	10/지표. 이 제한값은 변경될 수 없습니다.
<a href="#">GetMetricData</a>	50건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다. <a href="#">한도 증가를 요청</a> 할 수 있습니다.  API 요청에서 사용되는 <code>startTime</code> 이 현재 시간과 같거나 3시간 적을 경우 초당 데이터포인트(DPS)가 180,000입니다. <code>startTime</code> 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 제한값은 변경될 수 없습니다.
<a href="#">GetMetricData</a>	단일 <code>GetMetricData</code> 호출에는 최대 100개의 <code>MetricDataQuery</code> 구조가 포함될 수 있습니다.  이 제한값은 변경될 수 없습니다.

리소스	기본 제한
<a href="#">GetMetricStatistics</a>	400건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.  <a href="#">한도 증가를 요청</a> 할 수 있습니다.
<a href="#">ListMetrics</a>	25건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.  <a href="#">한도 증가를 요청</a> 할 수 있습니다.
지표 데이터	15개월. 이 제한값은 변경될 수 없습니다.
<a href="#">MetricDatum</a> 항목	20/ <a href="#">PutMetricData</a> 요청. <a href="#">MetricDatum</a> 객체는 단일 값을 포함하거나 여러 값을 나타내는 <a href="#">StatisticSet</a> 개체를 포함할 수 있습니다. 이 제한값은 변경될 수 없습니다.
지표	고객당 월 10회 무료.
기간	최대값은 1일(86,400초)입니다. 이 제한값은 변경될 수 없습니다.
<a href="#">PutMetricAlarm</a> 요청	3건의 초당 트랜잭션(TPS). 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.  <a href="#">한도 증가를 요청</a> 할 수 있습니다.
<a href="#">PutMetricData</a> 요청	HTTP POST 요청에 대하여 40KB. <a href="#">PutMetricData</a> 는 150건의 초당 트랜잭션(TPS)을 처리할 수 있으며, 이는 조절 없이 초당 수행할 수 있는 최대 작업 요청 수입니다.  <a href="#">한도 증가를 요청</a> 할 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
<a href="#">AWS SDK Metrics for Enterprise Support (p. 199)</a>	SDK Metrics를 사용하여 AWS 서비스의 상태를 평가하고, 계정 사용 제한에 도달하거나 서비스 중단에 의해 발생하는 지연 시간을 진단할 수 있습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서에서 <a href="#">AWS SDK Metrics를 사용하여 애플리케이션 모니터링</a> 을 참조하십시오.	December 11, 2018
<a href="#">수학 표현식에 대한 경보 (p. 199)</a>	CloudWatch에서는 지표 수학 표현식을 기반으로 경보를 생성할 수 있습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서의 <a href="#">수학 표현식에 대한 경보</a> 를 참조하십시오.	November 20, 2018
<a href="#">새 CloudWatch 콘솔 홈 페이지 (p. 199)</a>	Amazon에서는 CloudWatch 콘솔에 사용 중인 모든 AWS 서비스에 대한 주요 지표 및 경보를 자동으로 표시하는 새로운 홈 페이지를 만들었습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서의 <a href="#">Amazon CloudWatch 시작하기</a> 를 참조하십시오.	November 19, 2018
<a href="#">CloudWatch 에이전트용 AWS CloudFormation 템플릿 (p. 199)</a>	Amazon에서는 CloudWatch 에이전트 설치 및 업데이트에 사용할 수 있는 AWS CloudFormation 템플릿을 업로드했습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서의 <a href="#">AWS CloudFormation을 사용하여 새 인스턴스에 CloudWatch 설치</a> 를 참조하십시오.	November 9, 2018
<a href="#">CloudWatch 에이전트에서 향상된 기능 (p. 199)</a>	CloudWatch 에이전트가 StatsD 및 수집된 프로토콜과 호환되도록 업데이트되었습니다. 또한 교차 계정에 대한 지원도 개선되었습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서에서 <a href="#">StatsD로 시작하는 사용자 지정 지표 검색</a> , <a href="#">collectd로 사용자 지정 지표 검색</a> 및 <a href="#">지표 및 로그를 다른 AWS 계정으로 전송</a> 단원을 참조하십시오.	September 28, 2018

[Amazon VPC 엔드포인트에 대한 지원 \(p. 199\)](#)

이제 VPC와 CloudWatch 간에 프라이빗 연결을 설정할 수 있습니다. 자세한 정보는 [Amazon CloudWatch 사용 설명서의 인터페이스 VPC 엔드포인트와 함께 CloudWatch 사용](#)을 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 정보는 <a href="#">지표 수식 사용 (p. 44)</a> 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 <a href="#">경보 평가 (p. 49)</a> 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. 새로운 에이전트는 Windows Server 및 Linux를 모두 지원하며 CPU당 코어와 같은 하위 리소스 지표를 포함하여 수집할 지표를 사용자 지정할 수 있습니다. 자세한 정보는 <a href="#">CloudWatch 에이전트를 사용하여 Amazon EC2 인스턴스 및 온프레미스 서버로부터 지표 및 로그 수집 (p. 72)</a> 단원을 참조하십시오.	2017년 9월 7일
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 정보는 <a href="#">고분해능 지표 (p. 42)</a> 단원을 참조하십시오.	2017년 7월 26일
대시보드 API	이제 API 및 AWS CLI를 사용하여 대시보드를 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 <a href="#">CloudWatch 대시보드 생성 (p. 17)</a> 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 정보는 <a href="#">CloudWatch 대시보드에서 그래프를 편집하려면 (p. 20)</a> 단원을 참조하십시오.	2017년 2월 27일

변경 사항	설명	릴리스 날짜
대시보드의 경고	경보를 대시보드에 추가할 수 있습니다. 자세한 정보는 <a href="#">CloudWatch 대시보드에서 경보를 추가 또는 제거 (p. 23)</a> 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 정보는 <a href="#">백분위수 (p. 7)</a> 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
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

변경 사항	설명	릴리스 날짜
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 정보는 <a href="#">Amazon CloudWatch 대시보드 사용 (p. 17)</a> 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
새로운 기능: 경보 작업 재부팅	재부팅 경보 작업과 경보 작업에 사용할 새로운 IAM 역할이 추가되었습니다. 자세한 정보는 <a href="#">인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 (p. 63)</a> 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 정보는 <a href="#">인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 (p. 63)</a> 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



변경 사항	설명	릴리스 날짜
AWS CloudTrail에 대한 지원을 추가하도록 설명서 업데이트	Amazon CloudWatch 활동 로깅에 AWS CloudTrail을 사용하는 방법을 설명하기 위해 새 항목이 추가되었습니다. 자세한 정보는 <a href="#">AWS CloudTrail 사용을 통한 Amazon CloudWatch API 호출 로깅 (p. 193)</a> 를 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 <a href="#">AWS 명령줄 인터페이스를 사용한 설정</a> 을 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 정보는 <a href="#">인스턴스를 중지, 종료, 재부팅 또는 복구하는 경보 생성 (p. 63)</a> 단원을 참조하십시오.	2013년 1월 8일
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 정보는 <a href="#">예상 AWS 요금을 모니터링하기 위한 결제 경보를 생성 (p. 69)</a> 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 정보는 <a href="#">사용자 지정 지표 게시 (p. 42)</a> 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

변경 사항	설명	릴리스 날짜
새로운 기능	지표가 임계값을 초과하면 Amazon Simple Notification Service 또는 Auto Scaling 알림을 보내는 기능이 추가되었습니다. 자세한 정보는 <a href="#">개 경보 (p. 7)</a> 를 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