



사용자 가이드

AWS IoT Analytics



AWS IoT Analytics: 사용자 가이드

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께 사용하여 고객에게 혼란을 초래하거나 Amazon을 폄하 또는 브랜드 이미지에 악영향을 끼치는 목적으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

AWS IoT Analytics란 무엇입니까?	1
AWS IoT Analytics 사용 방법	1
주요 기능	2
AWS IoT Analytics 구성 요소 및 개념	3
AWS IoT Analytics 액세스	5
사용 사례	6
시작하기(콘솔)	8
AWS IoT Analytics 콘솔에 로그인	9
채널 생성	9
데이터 스토어 생성	11
파이프라인 생성	12
데이터 세트 생성	13
AWS IoT로 메시지 데이터를 전송합니다.	15
AWS IoT 메시지 진행 상황 확인	17
쿼리 결과 액세스	17
데이터 탐색	18
노트북 템플릿	20
시작하기	21
채널 생성	21
데이터 스토어 생성	23
Amazon S3 정책	23
파일 형식	25
사용자 지정 파티션	28
파이프라인 생성	31
AWS IoT Analytics로 데이터 수집	32
AWS IoT 메시지 브로커 사용	32
BatchPutMessage API 사용	36
수집된 데이터 모니터링	37
데이터 세트 생성	39
데이터 쿼리	40
쿼리된 데이터에 액세스	40
AWS IoT Analytics 데이터 탐색	18
Amazon S3	41
AWS IoT Events	42

Amazon QuickSight	42
Jupyter 노트북	43
다양한 버전의 데이터 세트 유지	43
메시지 페이로드 구문	44
AWS IoT SiteWise 데이터 작업	44
데이터 세트 생성	45
데이터 세트 콘텐츠 액세스	48
튜토리얼: AWS IoT SiteWise 데이터 쿼리	50
파이프라인 활동	58
채널 활동	58
데이터 스토어 활동	58
AWS Lambda 활동	58
Lambda 함수 예시 1	59
Lambda 함수 예시 2	61
AddAttributes 활동	62
RemoveAttributes 활동	63
SelectAttributes 활동	64
Filter 활동	65
DeviceRegistryEnrich 활동	66
DeviceShadowEnrich 활동	68
수학 활동	70
수학 활동 연산자 및 함수	71
RunPipelineActivity	86
채널 메시지 재처리	88
파라미터	88
채널 메시지 재처리(콘솔)	89
채널 메시지 재처리(API)	90
채널 재처리 활동 취소	90
워크플로우 자동화	91
사용 사례	92
Docker 컨테이너 사용	92
사용자 지정 Docker 컨테이너 입력/출력 변수	95
권한	97
CreateDataset(Java 및 AWS CLI)	100
예 1 -- SQL 데이터 세트(java) 생성:	100
예 2 -- delta 기간으로 SQL 데이터 세트(java) 생성:	101

예 3 -- 고유 일정 트리거로 컨테이너 데이터 세트(java) 생성:	102
예 4 -- 트리거인 SQL 데이터 세트로 컨테이너 데이터 세트(java) 생성:	103
예 5 -- SQL 데이터 세트(CLI) 생성:	104
예 6 -- 델타 기간으로 SQL 데이터 세트 생성(CLI)	104
노트북 컨테이너화	106
AWS IoT Analytics 콘솔로 생성하지 않은 노트북 인스턴스 컨테이너화 활성화	106
노트북 컨테이너화 확장 업데이트	109
컨테이너화된 이미지 생성	109
사용자 지정 컨테이너 사용	114
데이터 시각화	123
시각화(콘솔)	123
시각화(QuickSight)	124
태그 지정	128
태그 기본 사항	128
IAM 정책에 태그 사용	129
태그 제한	131
SQL 표현식	132
지원되는 SQL 기능	133
지원되는 데이터 형식	133
지원되는 함수	134
일반적인 문제 해결	135
보안	136
AWS Identity and Access Management	136
고객	136
보안 인증을 통한 인증	137
액세스 관리	140
IAM 작업	141
교차 서비스 혼동된 대리자 예방	145
IAM 정책 예시	151
자격 증명 및 액세스 문제 해결	157
로그 및 모니터링	158
자동 모니터링 도구	158
수동 모니터링 도구	159
CloudWatch Logs를 통한 모니터링	160
CloudWatch 이벤트로 모니터링	164
CloudTrail을 사용하여 API 호출 로깅	173

규정 준수 확인	177
복원력	178
인프라 보안	179
할당량	180
명령	181
AWS IoT Analytics 작업	181
AWS IoT Analytics 데이터	181
문제 해결	182
제 메시지가 AWS IoT Analytics로 들어가고 있는지 어떻게 알 수 있습니까?	182
제 파이프라인에서 메시지가 사라지는 이유는 무엇입니까? 어떻게 해결해야 합니까?	183
제 데이터 스토어에 데이터가 없는 이유는 무엇입니까?	184
데이터 세트에 왜 __dt이 표시되나요?	184
데이터 세트 완료에 따라 실행되는 이벤트를 코딩하려면 어떻게 해야 합니까?	184
AWS IoT Analytics를 사용하도록 노트북 인스턴스를 올바르게 구성하려면 어떻게 해야 합니까?	185
인스턴스에 노트북이 생성되지 않는 이유는 무엇입니까?	185
Amazon QuickSight에서 내 데이터 세트를 볼 수 없는 이유는 무엇입니까?	185
기존 Jupyter Notebook에서 컨테이너화 버튼을 볼 수 없는 이유가 무엇입니까?	186
컨테이너화 플러그인 설치가 실패하는 이유가 무엇입니까?	186
컨테이너화 플러그인에 오류가 발생하는 이유가 무엇입니까?	187
컨테이너화 동안 변수를 볼 수 없는 이유가 무엇입니까?	187
내 컨테이너에 입력으로 추가할 수 있는 변수는 무엇입니까?	187
컨테이너 출력을 이후에 분석을 위해 입력으로 설정하는 방법은 무엇입니까?	187
컨테이너 데이터 세트가 실패하는 이유가 무엇입니까?	188
문서 기록	189
이전 업데이트	190
.....	cxc

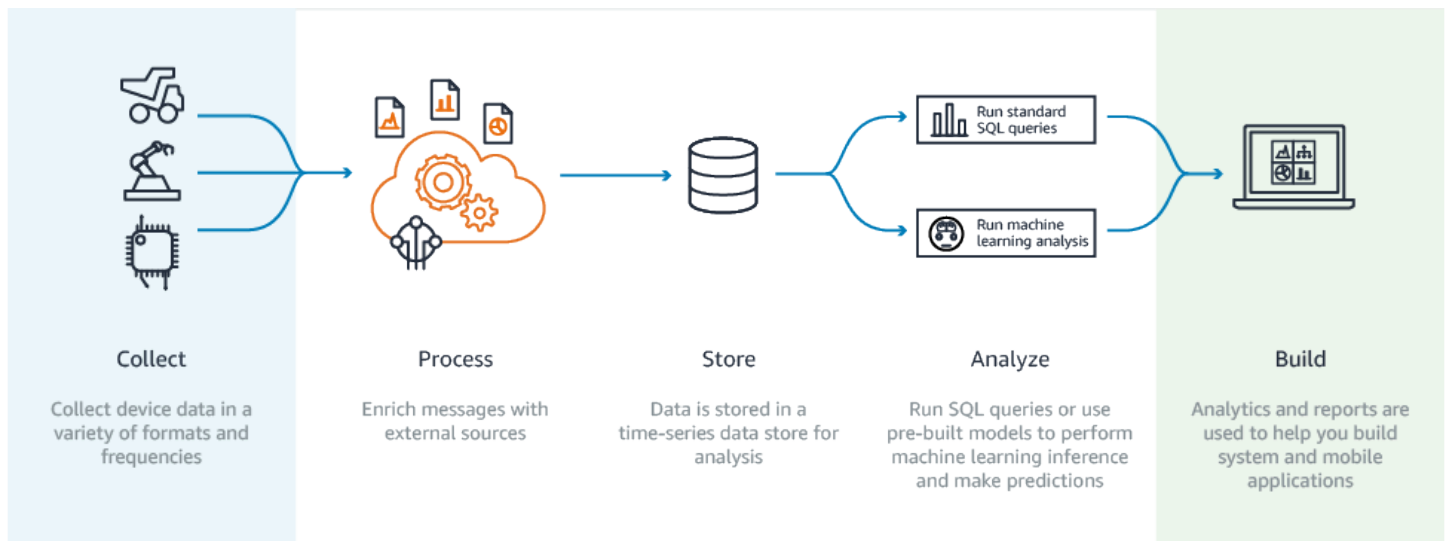
AWS IoT Analytics란 무엇인가요?

AWS IoT Analytics는 IoT 디바이스에서 얻은 데이터를 분석하는 데 필요한 단계를 자동화합니다. AWS IoT Analytics는 IoT 데이터를 필터링, 변환 및 보강한 다음 분석을 위해 시계열 데이터 스토어에 저장합니다. 디바이스에서 필요한 데이터만 수집하도록 서비스를 설정하고, 데이터를 처리하는 데 수학적 변환을 적용하고, 저장하기 전에 디바이스 유형 및 위치와 같은 디바이스별 메타데이터로 데이터를 보강할 수 있습니다. 그런 다음 기본 SQL 쿼리 엔진을 사용하여 쿼리를 실행하여 데이터를 분석하거나 더 복잡한 분석 및 기계 학습 추론을 수행할 수 있습니다. AWS IoT Analytics를 사용하여 [Jupyter Notebook](#)과의 통합을 통해 고급 데이터 탐색을 할 수 있습니다. 또한 AWS IoT Analytics로 [Amazon QuickSight](#)와의 통합을 통해 데이터를 시각화할 수 있습니다. Amazon QuickSight는 다음 [리전](#)에서 사용할 수 있습니다.

기존의 분석 및 비즈니스 인텔리전스 도구는 구조화된 데이터를 처리하도록 설계되어 있습니다. 원시 IoT 데이터는 덜 체계적인 데이터(예: 온도, 동작 또는 소리)를 기록하는 디바이스로부터 오는 경우가 많습니다. 따라서 이러한 디바이스의 데이터는 큰 폭의 차이, 손상된 메시지, 틀린 판독값을 함유할 수 있으며 분석 전에 이를 주기적으로 정리해야 합니다. 또한, IoT 데이터는 외부 소스의 다른 데이터와 관련해서만 의미가 있는 경우가 많습니다. AWS IoT Analytics로 이러한 문제를 해결하고 대량의 디바이스 데이터를 수집하고, 메시지를 처리하고, 저장할 수 있습니다. 그런 다음 데이터를 쿼리하고 분석할 수 있습니다. AWS IoT Analytics에는 일반 IoT 사용 사례를 위한 기본 모델이 포함되어 있습니다. 따라서 어떤 디바이스에서 장애가 발생할지 또는 어떤 고객이 웨어러블 디바이스를 중단할 위험이 있는지와 같은 질문에 답할 수 있습니다.

AWS IoT Analytics 사용 방법

다음 그래픽은 AWS IoT Analytics 사용 방법에 대한 개요를 보여줍니다.



주요 기능

수집

- AWS IoT Core과 통합—AWS IoT Analytics가 AWS IoT Core과 완전히 통합되어 커넥티드 디바이스에서 스트리밍하는 메시지를 수신할 수 있습니다.
- 배치 API를 사용하여 원하는 소스의 데이터 추가 - AWS IoT Analytics는 HTTP를 통해 어떤 소스에서나 데이터를 받을 수 있습니다. 다시 말해, 인터넷에 연결된 모든 디바이스 또는 서비스에서 AWS IoT Analytics로 데이터를 보낼 수 있습니다. 자세한 내용은 AWS IoT Analytics API 참조의 [BatchPutMessage](#)를 참조하세요.
- 저장 및 분석하려는 데이터만 수집 - AWS IoT Analytics 콘솔을 사용하여 다양한 형식 및 빈도의 MQTT 주제 필터를 통해 디바이스로부터 메시지를 수신하도록 AWS IoT Analytics를 구성할 수 있습니다. AWS IoT Analytics는 사용자가 정의한 특정 매개변수 내에 데이터가 있는지 확인하고 채널을 생성합니다. 그런 다음 메시지 처리, 변환 및 보강에 적절한 파이프라인으로 채널을 라우팅합니다.

프로세스

- 정리 및 필터 - AWS IoT Analytics를 사용하면 누락된 데이터가 AWS IoT Analytics에 감지될 때 트리거되는 AWS Lambda 함수를 정의하고, 코드를 실행하여 차이를 추정하고 메울 수 있습니다. 또한 최대 및 최소 필터 및 백분위 임계값을 정의하여 데이터 내 특이값을 제거할 수 있습니다.
- 변환 - AWS IoT Analytics는 사용자가 정의한 수학 또는 조건 논리를 사용하여 메시지를 변환할 수 있습니다. 따라서 섭씨에서 화씨로의 변환과 같은 일반적인 계산을 수행할 수 있습니다.
- 보강—AWS IoT Analytics는 일기 예보와 같은 외부 데이터 원본을 통해 데이터를 보강하고, 이 데이터를 AWS IoT Analytics 데이터 스토어로 라우팅할 수 있습니다.

저장

- 시계열 데이터 스토어 - AWS IoT Analytics는 신속한 검색과 분석을 위해 최적화된 시계열 데이터 스토어에 디바이스 데이터를 저장합니다. 액세스 권한을 관리하고, 데이터 보존 정책을 구현하고, 데이터를 외부 액세스 포인트로 내보낼 수도 있습니다.
- 처리된 데이터 및 원시 데이터 저장 - AWS IoT Analytics는 처리된 데이터를 저장하며, 나중에 처리할 수 있도록 원시 수집 데이터 또한 자동으로 저장합니다.

분석

- 애드혹 SQL 쿼리 실행 - AWS IoT Analytics에는 SQL 쿼리 엔진이 있으므로 애드혹 쿼리를 실행하고 빠르게 결과를 얻을 수 있습니다. 해당 서비스를 통해 표준 SQL 쿼리를 사용하여 데이터 스토어로부터 데이터를 추출하고 전체 커넥티드 차량의 평균 이동 거리 또는 오후 7시 이후 스마트 빌딩에서 잠긴 문의 개수와 같은 질문에 답할 수 있습니다. 이러한 쿼리는 커넥티드 디바이스, 집합 크기 및 분석 요건이 변경되는 경우에도 다시 사용될 수 있습니다.

- 시계열 분석 - AWS IoT Analytics는 시계열 분석을 지원합니다. 따라서 시간에 따른 디바이스의 성능을 분석하고 디바이스가 어떻게 사용되는지 이해할 수 있습니다. 또한 지속적으로 디바이스 데이터를 모니터링하여 유지 관리 문제를 예측하고, 센서를 모니터링하여 환경 조건을 예측하고 대응할 수 있습니다.
- 호스팅된 노트북을 통한 정교한 분석 및 기계 학습 - AWS IoT Analytics에는 통계 분석 및 기계 학습을 위해 Jupyter Notebook의 호스팅 노트북을 지원하는 기능이 있습니다. 이 서비스에는 AWS 작성 기계 학습 모델 및 시각화가 포함된 일련의 노트북 템플릿이 포함됩니다. 템플릿을 이용하여 디바이스 장애 프로파일링, 고객의 제품 종단을 나타낼 수 있는 낮은 사용량과 같은 이벤트 예측 또는 고객 사용량 수준(예: 사용량이 많은 사용자, 주말 사용자)이나 장치 상태를 기준으로 디바이스를 세그먼트화하는 등 IoT 사용 사례를 시작할 수 있습니다. 노트북을 작성한 후 컨테이너화를 해서 지정한 일정에 실행할 수 있습니다. 자세한 내용은 [자동화 워크플로](#)를 참조하세요.
- 예측—로지스틱 회귀라고 하는 방법을 사용하여 통계 분류를 수행할 수 있습니다. 또한 시간 경과에 따라 다른 프로세스의 결과 또는 상태를 예측하는 강력한 신경망 네트워크 기술인 Long Short-Term Memory(LSTM)를 사용할 수 있습니다. 사전 빌드된 노트북 템플릿은 또한 디바이스 세그먼트화에 대해 K-평균 클러스터링 알고리즘을 지원합니다. 이는 디바이스를 유사한 디바이스 집단으로 클러스터링합니다. 이러한 템플릿은 일반적으로 초콜릿 공장의 HVAC 장치 또는 풍력 터빈 블레이드의 마모 같은 디바이스 상태를 프로파일링하는 데 사용됩니다. 이런 노트북 템플릿을 컨테이너화해 일정에 따라 실행할 수 있습니다.

구축 및 시각화

- QuickSight 통합 - AWS IoT Analytics는 Amazon QuickSight 커넥터를 제공하여 QuickSight 대시보드에서 데이터 세트를 시각화할 수 있습니다.
- 콘솔 통합—AWS IoT Analytics' 콘솔에 포함된 Jupyter Notebook의 애드혹 분석이나 결과물을 시각화할 수도 있습니다.

AWS IoT Analytics 구성 요소 및 개념

Channel

채널은 MQTT 주제로부터 데이터를 수집하고, 처리되지 않은 원시 데이터를 보관하여 데이터를 파이프라인으로 게시합니다. 또한 [BatchPutMessage](#) API로 메시지를 직접 채널로 전송할 수 있습니다. 처리되지 않은 메시지는 사용자 또는 AWS IoT Analytics가 관리하는 Amazon Simple Storage Service(S3) 버킷에 저장됩니다.

파이프라인

파이프라인은 채널로부터 메시지를 사용하고 사용자가 데이터 스토어에 저장하기 전에 메시지를 처리할 수 있도록 합니다. 활동([파이프라인 활동](#))이라고 하는 처리 단계는 메시지 변환을 수행합니

다. 그 예로는 메시지 속성 삭제, 이름 변경 또는 추가, 속성 값을 기반으로 한 메시지 필터링, 고급 처리 및 디바이스 데이터 정규화를 위한 수학 변환 수행 시 메시지에서 Lambda 함수 호출이 있습니다.

데이터 스토어

파이프라인은 처리된 메시지를 데이터 스토어에 저장합니다. 데이터 스토어는 데이터베이스가 아니라 확장 및 쿼리가 가능한 메시지 리포지토리입니다. 다양한 디바이스 또는 위치에서 전송되는 메시지 또는 파이프라인 구성 및 요구 사항에 따라 메시지 속성으로 필터링되는 메시지를 저장하기 위해 여러 개의 데이터 스토어를 보유할 수 있습니다. 처리되지 않은 채널 메시지와 마찬가지로, 데이터 스토어의 처리된 메시지는 사용자 또는 AWS IoT Analytics에서 관리하는 [Amazon S3](#) 버킷에 저장됩니다.

데이터 세트

데이터 세트를 생성하여 데이터 스토어에서 데이터를 검색할 수 있습니다. AWS IoT Analytics를 사용하여 SQL 데이터 세트 또는 컨테이너 데이터 세트를 만들 수 있습니다.

데이터 센터를 생성한 뒤에는 [Amazon QuickSight](#)를 사용하여 통합하여 데이터를 탐색하고 분석할 수 있습니다. 아니면 [Jupyter Notebook](#)과 통합하여 더욱 발전된 분석 기능을 수행할 수도 있습니다. Jupyter Notebook은 기계 학습과 각종 통계 분석을 수행할 수 있는 강력한 데이터 과학 도구입니다. 자세한 내용은 [노트북 템플릿](#)을 참조하세요.

데이터 세트 콘텐츠를 [Amazon S3](#) 버킷으로 보내 기존 데이터 레이크와의 통합 또는 사내 애플리케이션 및 시각화 도구에서의 액세스를 활성화할 수 있습니다. 데이터 세트 콘텐츠를 [AWS IoT Events](#)의 입력으로 전송할 수도 있습니다. 디바이스나 프로세스의 장애 또는 작동 변경을 모니터링하고 이러한 이벤트가 발생할 때 추가 작업을 트리거하는 서비스입니다.

SQL 데이터 세트

SQL 데이터 세트는 SQL 데이터베이스의 구체화된 보기와 유사합니다. 실제로 SQL 작업을 적용하여 SQL 데이터 세트를 생성할 수 있습니다. SQL 데이터 세트는 트리거 지정을 통해 반복되는 일정으로 자동 생성될 수 있습니다.

컨테이너 데이터 세트

컨테이너 데이터 세트를 사용하면 자동으로 분석 도구를 실행하고 결과를 생성할 수 있습니다. 자세한 내용은 [자동화 워크플로](#)를 참조하세요. 입력으로서의 SQL 데이터 세트, 분석 도구와 필요한 라이브러리 파일이 포함된 Docker 컨테이너, 입력 및 출력 변수, 선택 사항인 일정 트리거를 결합합니다. 입력 및 출력 변수는 데이터를 가져오고 결과를 저장할 실행 가능한 이미지를 구분합니다. 트리거는 SQL 데이터 세트에서 콘텐츠 생성을 완료할 때 또는 시간 예약 표현식에 따라 분석을 실행

할 수 있습니다. 컨테이너 데이터 세트는 자동으로 실행되어 분석 도구의 결과를 생성한 다음 저장합니다.

트리거

트리거를 지정하여 자동으로 데이터 세트를 생성할 수 있습니다. 트리거는 시간 간격이거나(예: 두 시간마다 이 데이터 세트를 생성), 다른 데이터 세트의 콘텐츠를 생성한 시간(예: myOtherDataset가 콘텐츠를 생성을 완료할 때 이 데이터 세트 생성)일 수 있습니다. 또는 [CreateDatasetContent](#) API를 사용하여 데이터 세트 콘텐츠를 수동으로 생성할 수 있습니다.

Docker 컨테이너

분석 도구들을 패키징화할 Docker 컨테이너를 자체 생성하거나, SageMaker가 제공하는 옵션을 사용할 수 있습니다. 자세한 내용은 [Docker 컨테이너](#)를 참조하세요. 분석 도구들을 패키징화할 Docker 컨테이너를 자체 생성하거나, [SageMaker](#)가 제공하는 옵션을 사용할 수 있습니다. 컨테이너를 지정한 [Amazon ECR](#) 레지스트리에 저장하여 나중에 원하는 플랫폼에 설치할 때 사용할 수 있습니다. Docker 컨테이너는 Matlab, Octave, Wise.io, SPSS, R, Fortran, Python, Scala, Java, C++ 등으로 준비한 사용자 지정 분석 코드를 실행할 수 있습니다. 자세한 내용은 [노트북 컨테이너화](#)를 참조하세요.

Delta 기간

델타 기간이란 일련의 사용자 정의된 비중첩 연속 시간 간격입니다. 델타 기간을 통해 포함되는 데이터 세트 콘텐츠를 생성하고, 마지막 분석 이후 데이터 스토어에 새로 도착한 데이터를 분석할 수 있습니다. 데이터 세트의 queryAction에서 filters의 deltaTime을 설정해 델타 기간을 생성할 수 있습니다. 자세한 정보는 [CreateDataset](#) API를 참조하십시오. 일반적으로 시간 간격 트리거를 설정하여 데이터 세트 콘텐츠를 자동으로 생성합니다(triggers:schedule:expression). 기본적으로 이를 통해 특정 기간 동안 도착한 메시지를 필터링할 수 있어 이전 기간에 온 메시지에 포함된 데이터가 두 번 고려되지 않습니다. 자세한 내용은 [예 6 -- 델타 기간으로 SQL 데이터 세트 생성\(CLI\)](#)을 참조하세요.

AWS IoT Analytics 액세스

AWS IoT의 일부인 AWS IoT Analytics는 디바이스가 데이터를 생성하고 애플리케이션이 생성하는 데이터와 상호 작용할 수 있도록 다음 인터페이스를 제공합니다.

AWS Command Line Interface (AWS CLI)

Windows, OS X 및 Linux에서 AWS IoT Analytics 명령을 실행합니다. 이러한 명령을 사용하여 사물, 인증서, 규칙 및 정책을 생성하고 관리할 수 있습니다. 시작하려면 [AWS Command Line](#)

[Interface 사용 설명서](#)를 참조하세요. AWS IoT 명령에 대한 자세한 내용은 AWS Command Line Interface 참조의 [iot](#)를 참조하세요.

⚠ Important

`aws iotanalytics` 명령을 사용하여 AWS IoT Analytics와 상호 작용합니다. IoT 시스템의 다른 부분과 상호 작용하려면 `aws iot` 명령을 사용합니다.

AWS IoT API

HTTP 또는 HTTPS 요청을 사용하여 IoT 애플리케이션을 빌드합니다. 이러한 API 작업을 사용하여 사물, 인증서, 규칙 및 정책을 생성하고 관리할 수 있습니다. 자세한 내용은 AWS IoTAPI 참조의 [작업](#) 섹션을 참조하세요.

AWS SDK

언어별 API를 사용하여 AWS IoT Analytics 애플리케이션을 빌드합니다. 이러한 SDK는 HTTP와 HTTPS API를 래핑하고 지원되는 언어로 프로그램을 작성할 수 있게 해줍니다. 자세한 내용은 [AWS SDK 및 도구](#) 섹션을 참조하세요.

AWS IoT Device SDK

디바이스에서 실행되면서 AWS IoT Analytics로 메시지를 보내는 애플리케이션을 빌드합니다. 자세한 내용은 [AWS IoT SDK](#)를 참조하세요.

AWS IoT Analytics 콘솔

구성 요소를 구축하여 [AWS IoT Analytics 콘솔](#)에서 결과를 시각화할 수 있습니다.

사용 사례

예측 유지 보수

AWS IoT Analytics는 예측 유지 보수 모델을 빌드하고 이를 디바이스에 적용할 수 있도록 템플릿을 제공합니다. 예를 들어 AWS IoT Analytics를 사용하여 커넥티드 화물 차량의 난방 및 냉각 시스템 장애가 언제 발생할지 예측하고, 이를 통해 차량의 경로를 변경하고 화물 피해를 방지할 수 있습니다. 또는 자동차 제조업체의 경우 어떤 고객의 브레이크 패드가 마모되었는지 감지하고 해당 고객에게 차량 정비를 받으라고 알릴 수 있습니다.

사전 공급 보충

AWS IoT Analytics를 사용하여 재고를 실시간으로 모니터링하는 IoT 애플리케이션을 빌드할 수 있습니다. 예를 들어 식음료 회사에서 자동 판매기의 데이터를 분석하고 공급이 줄어들 때마다 상품을 사전 재주문할 수 있습니다.

프로세스 효율성 평가

AWS IoT Analytics를 사용하여 각기 다른 프로세스의 효율성을 지속적으로 모니터링하고 프로세스 개선을 위한 조치를 취하는 애플리케이션을 빌드할 수 있습니다. 예를 들어 광업 회사에서는 운행 시마다 적재량을 최대화하여 광석 운반 트럭의 효율성을 높일 수 있습니다. 이 회사는 AWS IoT Analytics를 사용하여 시간 경과에 따른 각 위치 또는 트럭의 가장 효율적인 적재량을 파악하고, 목표 적재량과의 편차를 실시간으로 비교하고, 효율성 개선을 위한 적재 지침을 수립할 수 있습니다.

스마트 농업

AWS IoT Analytics는 AWS IoT 레지스트리 데이터 또는 퍼블릭 데이터 원본을 사용하여 컨텍스트 메타데이터로 IoT 장치 데이터를 보강할 수 있습니다. 따라서 분석에 시간, 위치, 온도, 고도 및 기타 환경 조건의 영향을 고려할 수 있습니다. 이러한 분석을 통해 디바이스가 현장에서 취할 권장 작업을 생산하는 모델을 작성할 수 있습니다. 예를 들어 물을 주어야 하는 시기를 결정하기 위해 관개 시스템은 습도 센서 데이터에 강우 데이터를 보강하여 물을 더 효율적으로 사용할 수 있도록 합니다.

AWS IoT Analytics (콘솔) 시작하기

이 튜토리얼을 사용하여 IoT 디바이스 데이터에 대한 유용한 통찰력을 발견하는 데 필요한 AWS IoT Analytics 리소스(구성 요소라고도 함)를 만들 수 있습니다.

주의

- 다음 튜토리얼에서 대문자를 입력하면 AWS IoT Analytics가 자동으로 이를 소문자로 변경합니다.
- AWS IoT Analytics 콘솔에는 빠른 시작 기능이 있어 클릭 한 번으로 채널, 파이프라인, 데이터 스토어 및 데이터 세트를 생성할 수 있습니다. AWS IoT Analytics 콘솔에 로그인할 때 이 기능을 찾을 수 있습니다.
- 이 튜토리얼에서는 AWS IoT Analytics 리소스를 생성하는 각 단계를 안내합니다.

아래 지침에 따라 AWS IoT Analytics 채널, 파이프라인, 데이터 스토어, 데이터 세트를 만드세요. 이 튜토리얼에서는 AWS IoT Core 콘솔을 사용하여 AWS IoT Analytics에 수집될 메시지를 보내는 방법도 보여줍니다.

주제

- [AWS IoT Analytics 콘솔에 로그인](#)
- [채널 생성](#)
- [데이터 스토어 생성](#)
- [파이프라인 생성](#)
- [데이터 세트 생성](#)
- [AWS IoT로 메시지 데이터를 전송합니다.](#)
- [AWS IoT 메시지 진행 상황 확인](#)
- [쿼리 결과 액세스](#)
- [데이터 탐색](#)
- [노트북 템플릿](#)

AWS IoT Analytics 콘솔에 로그인

시작하려면 AWS 계정이 있어야 합니다. 이미 AWS 계정이 있다면 <https://console.aws.amazon.com/iotanalytics/>으로 이동하세요.

AWS 계정이 없다면 다음 단계에 따라 계정을 생성합니다.

AWS 계정을 생성하려면

1. <https://portal.aws.amazon.com/billing/signup>을 엽니다.
2. 온라인 지시 사항을 따릅니다.

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

AWS 계정 루트 사용자에게 가입하면 AWS 계정 루트 사용자가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스하는 권한이 있습니다. 보안 모범 사례는 [관리 사용자에게 관리자 액세스 권한을 할당하고](#), 루트 사용자만 [루트 사용자 액세스 권한이 필요한 태스크](#)를 수행하는 것입니다.

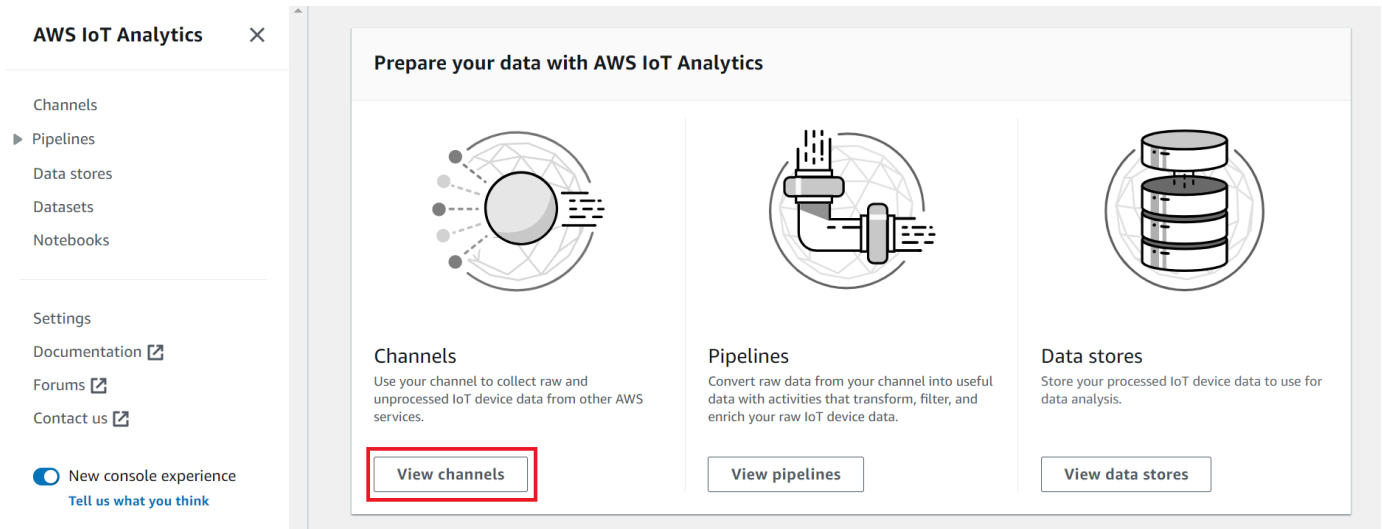
3. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/iotanalytics/>으로 이동합니다.

채널 생성

채널은 원시의 가공되지 않고 구조화되지 않은 IoT 디바이스 데이터를 수집하고 보관합니다. 다음 단계에 따라 채널을 생성합니다.

채널을 생성하려면

1. <https://console.aws.amazon.com/iotanalytics/>의 AWS IoT Analytics로 데이터 준비 섹션에서 채널 보기를 선택합니다.



Tip

탐색 창에서 채널을 선택할 수도 있습니다.

2. 채널 페이지에서 채널 생성을 선택합니다.
3. 채널 세부 정보 지정 페이지에서 채널에 대한 세부정보를 입력합니다.
 - a. 고유하고 쉽게 식별할 수 있는 채널 이름을 입력합니다.
 - b. (선택 사항) 태그에 대해 하나 이상의 사용자 정의 태그(키-값 페어)를 채널에 추가합니다. 태그를 사용하면 AWS IoT Analytics에 대해 생성하는 리소스를 식별하는 데 도움이 됩니다.
 - c. 다음(Next)을 선택합니다.
4. AWS IoT Analytics은 처리되지 않은 원시 IoT 디바이스 데이터를 Amazon Simple Storage Service(S3) 버킷에 저장합니다. 액세스 및 관리할 수 있는 Amazon S3 버킷을 직접 선택하거나 AWS IoT Analytics가 대신 Amazon S3 버킷을 관리할 수 있습니다.
 - a. 이 튜토리얼에서는 스토리지 유형에서 서비스 관리형 스토리지를 선택합니다.
 - b. 원시 데이터를 저장할 기간 선택에서 무기한을 선택합니다.
 - c. 다음(Next)을 선택합니다.
5. 소스 구성 페이지에서 AWS IoT Core에서 메시지 데이터를 수집할 AWS IoT Analytics의 정보를 입력합니다.
 - a. AWS IoT Core 주제 필터를 입력합니다(예: update/environment/dht1). 이 튜토리얼 후 반부에서는 이 주제 필터를 사용하여 채널에 메시지 데이터를 전송해 보겠습니다.

- b. IAM 역할 영역에서 새로 생성을 선택합니다. 새 역할 생성 창에 역할의 이름을 입력한 다음 역할 생성을 선택합니다. 이렇게 하면 적절한 정책이 연결된 역할이 자동 생성됩니다.
 - c. 다음(Next)을 선택합니다.
6. 선택 사항을 검토한 다음 채널 생성을 선택합니다.
 7. 채널 페이지에 새 채널이 나타나는지 확인합니다.

데이터 스토어 생성

데이터 스토어는 메시지 데이터를 수신하고 저장합니다. 데이터 스토어는 데이터베이스가 아닙니다. 대신 데이터 스토어는 Amazon S3 버킷에서 확장 가능하고 쿼리가 가능한 리포지토리입니다. 다른 디바이스 또는 위치에서 보낸 메시지에 대해 여러 데이터 스토어를 사용할 수 있습니다. 또는 파이프라인 구성 및 요구 사항에 따라 메시지 데이터를 필터링할 수 있습니다.

다음 단계에 따라 데이터 스토어를 생성합니다.

데이터 스토어 생성

1. <https://console.aws.amazon.com/iotanalytics/>의 AWS IoT Analytics로 데이터 준비 섹션에서 데이터 스토어 보기를 선택합니다.
2. 데이터 스토어 페이지에서 데이터 스토어 생성을 선택합니다.
3. 데이터 스토어 세부정보 지정 페이지에서 데이터 스토어에 대한 기본 정보를 입력합니다.
 - a. 데이터 스토어 ID에는 고유한 데이터 스토어 ID를 입력합니다. 이를 생성한 후에는 이 ID를 변경할 수 없습니다.
 - b. (선택사항) 태그에서 새 태그 추가를 선택하여 하나 이상의 사용자 지정 태그(키-값 쌍)를 데이터 스토어에 추가합니다. 태그를 사용하면 AWS IoT Analytics에 대해 생성하는 리소스를 식별하는 데 도움이 됩니다.
 - c. 다음(Next)을 선택합니다.
4. 스토리지 유형 구성 페이지에서 데이터 저장 방법을 지정합니다.
 - a. 스토리지 유형에서 서비스 관리 스토리지를 선택합니다.
 - b. 처리된 데이터를 보관할 기간 구성에서 무기한을 선택합니다.
 - c. 다음(Next)을 선택합니다.
5. AWS IoT Analytics 데이터 스토어는 JSON 및 Parquet 파일 형식을 지원합니다. 데이터 스토어 데이터 형식으로 JSON 또는 Parquet을 선택합니다. AWS IoT Analytics 지원 파일 유형에 대한 자세한 내용은 [파일 형식](#)을 참조하세요.

다음(Next)을 선택합니다.

- (선택사항) AWS IoT Analytics는 데이터 스토어의 사용자 지정 파티션을 지원하므로 정리된 데이터를 쿼리하여 지연 시간을 개선할 수 있습니다. 지원되는 사용자 지정 파티션에 대한 자세한 내용은 [사용자 지정 파티션](#) 단원을 참조하세요.

다음(Next)을 선택합니다.

- 선택 사항을 검토한 다음 데이터 스토어 생성을 선택합니다.
- 새 데이터 스토어가 데이터 스토어 페이지에 나타나는지 확인하세요.

파이프라인 생성

채널을 데이터 스토어에 연결하려면 파이프라인을 생성해야 합니다. 기본 파이프라인은 데이터를 수집하는 채널만 지정하고 메시지가 전송되는 데이터 스토어를 식별합니다. 자세한 내용은 [파이프라인 활동](#)을 참조하십시오.

이 튜토리얼에서는 채널을 데이터 스토어에만 연결하는 파이프라인을 생성합니다. 나중에 이 데이터를 처리하는 파이프라인 활동을 추가할 수 있습니다.

이 단계에 따라 파이프라인을 생성합니다.

파이프라인을 생성하려면

- <https://console.aws.amazon.com/iotanalytics/>의 AWS IoT Analytics로 데이터 준비 섹션에서 파이프라인 보기를 선택합니다.

Tip

탐색 창에서 파이프라인을 선택할 수도 있습니다.

- 파이프라인 페이지에서 파이프라인 생성을 선택합니다.
- 파이프라인에 대한 세부 정보를 입력합니다.
 - 파이프라인 ID 및 소스 설정에 파이프라인 이름을 입력합니다.
 - 파이프라인이 메시지를 읽을 AWS IoT Analytics 채널인 파이프라인 소스를 선택합니다.
 - 처리된 메시지 데이터가 저장되는 데이터 스토어인 파이프라인의 출력을 지정합니다.
 - (선택 사항) [태그(Tags)]에서 하나 이상의 사용자 정의 태그(키-값 쌍)를 파이프라인에 추가합니다.

- e. 메시지 속성 추론 페이지에서 속성 이름과 예시 값을 입력하고, 목록에서 데이터 유형을 선택한 다음 속성 추가를 선택합니다.
 - f. 이전 단계를 반복하여 필요한 만큼의 속성을 지정한 후 다음을 선택합니다.
 - g. 지금은 파이프라인 활동을 추가하지 않을 것입니다. 메시지 강화, 변환, 필터링 페이지에서 다음을 선택합니다.
4. 선택 사항을 검토한 다음 파이프라인 생성을 선택합니다.
 5. 파이프라인 페이지에 새 파이프라인이 나타나는지 확인하세요.

Note

AWS IoT Analytics 리소스를 생성하여 다음 작업을 수행하도록 합니다.

- 채널을 통해 처리되지 않은 원시 IoT 디바이스 메시지 데이터를 수집합니다.
- IoT 디바이스 메시지 데이터를 데이터 스토어에 저장합니다.
- 파이프라인을 통해 데이터를 정리, 필터링, 변환 및 보강합니다.

다음으로, AWS IoT Analytics SQL 데이터 세트를 생성하여 IoT 디바이스에 대한 유용한 통찰력을 발견해 보겠습니다.

데이터 세트 생성

Note

데이터 세트는 일반적으로 테이블 형식으로 구성되거나 구성되지 않을 수 있는 데이터 모음입니다. 반대로 AWS IoT Analytics는 데이터 스토어의 데이터에 SQL 쿼리를 적용하여 데이터 세트를 만듭니다.

이제 데이터 스토어에 데이터를 저장하여 쿼리할 수 있는 파이프라인으로 원시 메시지 데이터를 라우팅하는 채널이 생겼습니다. 데이터를 쿼리하려면 데이터 세트를 생성합니다. 데이터 세트에는 데이터 스토어 쿼리에 사용하는 SQL 문과 표현식 및 원하는 날짜 및 시간에 쿼리를 반복하는 일정 옵션이 포함됩니다. 이 일정 옵션은 [Amazon CloudWatch 일정 표현식](#)과 유사한 표현식을 사용하여 생성할 수 있습니다.

데이터 세트를 생성하려면

1. <https://console.aws.amazon.com/iotanalytics/>의 왼쪽 탐색 창에서 데이터 세트를 선택합니다.
2. 데이터 세트 생성 페이지에서 SQL 생성을 선택합니다.
3. 데이터 세트 세부 정보 지정 페이지에서 데이터 세트의 세부 정보를 지정합니다.
 - a. 데이터 세트의 이름을 입력합니다.
 - b. 데이터 스토어 소스의 경우 이전에 만든 데이터 스토어를 식별하는 고유 ID를 선택합니다.
 - c. (선택 사항)태그에서 하나 이상의 사용자 정의 태그(키-값 쌍)를 데이터 세트에 추가합니다.
4. SQL 식을 사용하여 데이터를 쿼리하고 분석 질문에 답변합니다. 쿼리 결과는 이 데이터 세트에 저장됩니다.
 - a. 작성자 쿼리 필드에 와일드카드를 사용하여 최대 5개 행의 데이터를 표시하는 SQL 쿼리를 입력합니다.

```
SELECT * FROM my_data_store LIMIT 5
```

AWS IoT Analytics에서 지원 SQL 함수에 대한 자세한 내용은 [AWS IoT Analytics의 SQL 표현식](#)을 참조하세요.

- b. 쿼리 테스트를 선택하여 입력이 올바른지 확인하고 쿼리 다음에 결과를 테이블로 표시할 수 있습니다.

Note

- 튜토리얼의 이 시점에서는 데이터 스토어가 비어 있을 수 있습니다. 빈 데이터 스토어에서 SQL 쿼리를 실행하면 결과가 반환되지 않으므로 __dt만 표시될 수 있습니다.
- Athena가 [최대 실행 쿼리 최대 수를 제한하므로](#) SQL 쿼리가 장기간 실행되지 않도록 적절한 크기로 제한해야 합니다. 따라서 SQL 쿼리를 적절한 크기로 제한하도록 주의해야 합니다.

테스트 중에는 쿼리에 LIMIT 절을 사용하는 것이 좋습니다. 테스트가 성공하면 이 절을 삭제할 수 있습니다.

5. (선택 사항) 지정된 기간의 데이터를 사용하여 데이터 세트 콘텐츠를 만드는 경우, 일부 데이터가 처리 시간 내에 도착하지 않을 수 있습니다. 지연을 허용하기 위해 오프셋 또는 델타를 지정할 수

있습니다. 자세한 내용은 [Amazon CloudWatch Events를 통해 지연 데이터 알림 받기](#) 섹션을 참조하세요.

지금은 데이터 선택 필터를 구성하지 않을 것입니다. 데이터 선택 필터 구성 페이지에서 다음을 선택합니다.

- (선택 사항) 이 쿼리가 정기적으로 실행되도록 예약하여 데이터 세트를 새로 고침할 수 있습니다. 데이터 세트 일정은 언제든지 만들고 편집할 수 있습니다.

여기서는 쿼리의 반복 실행을 예약하지 않을 것이므로 쿼리 일정 설정 페이지에서 다음을 선택합니다.

- AWS IoT Analytics는 이 데이터 세트 콘텐츠의 버전을 생성하고 지정된 기간 동안 분석 결과를 저장합니다. 90일을 권장하지만, 사용자 지정 보존 정책을 설정하도록 선택할 수 있습니다. 데이터 세트 콘텐츠의 저장된 버전 수를 제한할 수도 있습니다.

기본 데이터 세트 보존 기간을 무기한으로 지정하고 버전 관리를 비활성화 상태로 유지할 수 있습니다. 분석 결과 구성 페이지에서 다음을 선택합니다.

- (선택 사항) AWS IoT Events와 같은 특정 대상에 데이터 세트 결과의 전송 규칙을 구성할 수 있습니다.

이 튜토리얼의 다른 곳으로는 결과를 전달할 수 없으므로, 데이터 세트 콘텐츠 전송 규칙 구성 페이지에서 다음을 선택합니다.

- 선택 사항을 검토한 다음 데이터 세트 생성을 선택합니다.
- 데이터 세트 페이지에 새 데이터 세트가 나타나는지 확인하세요.

AWS IoT로 메시지 데이터를 전송합니다.

쿼리할 수 있는 데이터 스토어에 데이터를 저장하는 파이프라인으로 데이터를 라우팅하는 채널이 있으면 AWS IoT Analytics로 메시지 데이터를 보낼 준비가 된 것입니다. 다음 옵션을 사용하여 데이터를 AWS IoT Analytics에 보낼 수 있습니다.

- AWS IoT 메시지 브로커를 사용합니다.
- AWS IoT Analytics [BatchPutMessage](#) API 작업을 사용합니다.

다음 단계에서는 AWS IoT Core 콘솔의 AWS IoT 메시지 브로커로부터 메시지 데이터를 전송하여 AWS IoT Analytics가 이 데이터를 수집할 수 있도록 합니다.

Note

메시지에 대한 주제 이름을 생성할 때 다음 사항에 유의하세요.

- 주제 이름은 대/소문자를 구분하지 않습니다. 동일한 페이로드에서 example 및 EXAMPLE라는 필드 이름은 중복으로 간주합니다.
- 주제 이름은 \$ 문자로 시작할 수 없습니다. \$로 시작하는 주제 이름은 AWS IoT에서만 사용될 수 있는 예약된 주제입니다.
- 개인 식별 정보가 암호화되지 않은 통신 및 보고서에 나타날 수 있으므로 주제 이름에 개인 식별 정보를 포함하지 마세요.
- AWS IoT Core는 AWS 계정 또는 AWS 리전 간에 메시지를 보낼 수 없습니다.

AWS IoT로 메시지 데이터를 보내려면

1. [AWS IoT 콘솔](#)에 로그인합니다.
2. 탐색 창에서 테스트를 선택하고 MQTT 테스트 클라이언트를 선택합니다.
3. MQTT 테스트 클라이언트에서 주제에 게시를 선택합니다.
4. 주제 이름에는 채널을 만들었을 때 입력한 주제 필터와 일치하는 이름을 입력합니다. 이 예제에서는 update/environment/dht1를 사용합니다.
5. 메시지 페이로드에 다음 JSON 콘텐츠를 입력합니다.

```
{
  "thingid": "dht1",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

6. (선택 사항) 추가 메시지 프로토콜 옵션을 위해 구성 추가를 선택합니다.
7. [Publish]를 선택합니다.

이렇게 하면 채널에서 캡처한 메시지가 게시됩니다. 그러면 파이프라인이 메시지를 데이터 스토어로 라우팅합니다.

AWS IoT 메시지 진행 상황 확인

다음 단계에 따라 채널에서 메시지가 수집되고 있는지 확인할 수 있습니다.

AWS IoT 메시지 진행 상황을 확인하려면

1. <https://console.aws.amazon.com/iotanalytics/>에 로그인합니다.
2. 탐색 창에서 채널을 선택한 후 이전에 생성한 채널 이름을 선택합니다.
3. 채널의 세부 정보 페이지에서 모니터링 섹션까지 아래로 스크롤한 다음, 표시된 시간 프레임(1h 3h 12h 1d 3d 1w)을 조정합니다. 지난 주의 데이터를 보려면 1w와 같은 값을 선택합니다.

유사한 기능을 사용하여 파이프라인의 세부 정보 페이지에서 파이프라인 활동, 런타임 및 오류를 모니터링할 수 있습니다. 이 튜토리얼에서는 활동을 파이프라인의 일부로 지정하지 않았으므로, 런타임 오류가 발생하지 않아야 합니다.

파이프라인 활동을 모니터링하려면

1. 탐색 창에서 파이프라인을 선택한 다음 이전에 생성한 파이프라인의 이름을 선택합니다.
2. 파이프라인의 세부 정보 페이지에서 모니터링 섹션으로 아래로 스크롤한 다음, 기간 표시(1h 3h 12h 1d 3d 1w) 중 하나를 선택하여 표시된 기간을 조정합니다.

쿼리 결과 액세스

데이터 세트 콘텐츠는 쿼리 결과를 포함한 CSV 형식의 파일입니다.

1. <https://console.aws.amazon.com/iotanalytics/>의 왼쪽 탐색 창에서 데이터 세트를 선택합니다.
2. 데이터 세트 페이지에서 이전에 생성한 데이터 세트 이름을 선택합니다.
3. 데이터 세트 정보 페이지의 오른쪽 상단에서 지금 실행을 선택합니다.
4. 데이터 세트가 준비되었는지 확인하려면 데이터 세트에 데이터 집합에 대한 쿼리를 성공적으로 시작했습니다와 유사한 메시지가 있는지 확인합니다. 데이터 세트 콘텐츠 탭에는 쿼리 결과가 포함되며 성공으로 표시됩니다.
5. 쿼리 성공 결과를 미리 보려면 데이터 세트 콘텐츠 탭에서 쿼리 이름을 선택합니다. 쿼리 결과가 포함된 CSV 파일을 보거나 저장하려면 다운로드를 선택합니다.

Note

AWS IoT Analytics는 Jupyter Notebook의 HTML 부분을 데이터 세트 콘텐츠 페이지에 포함시킬 수 있습니다. 자세한 내용은 [콘솔을 사용한 AWS IoT Analytics 데이터 시각화](#) 섹션을 참조하세요.

데이터 탐색

데이터 저장, 분석 및 시각화를 위한 몇 가지 옵션이 있습니다.

Amazon Simple Storage Service

데이터 세트 콘텐츠를 [Amazon Simple Storage Service\(S3\)](#) 버킷으로 보내 기존 데이터 레이크와의 통합 또는 사내 애플리케이션 및 시각화 도구에서의 액세스를 활성화할 수 있습니다. [CreateDataset](#) 작업의 `contentDeliveryRules::destination::s3DestinationConfiguration` 필드를 참조하세요.

AWS IoT Events

데이터 세트 콘텐츠를 AWS IoT Events의 입력으로 전송할 수 있습니다. 디바이스나 프로세스의 장애 또는 작동 변경을 모니터링하고 이러한 이벤트가 발생할 때 추가 작업을 트리거하는 서비스입니다.

이렇게 하려면 [CreateDataset](#) 작업을 사용하여 데이터 세트를 만들고

```
contentDeliveryRules :: destination ::
```

```
iotEventsDestinationConfiguration :: inputName
```

필드에 AWS IoT Events 입력값을 지정해야 합니다. AWS IoT Analytics에 `iotevents:BatchPutMessage` 실행 권한을 부여하는 역할의 `roleArn`도 지정해야 합니다. 데이터 세트 콘텐츠가 생성될 때마다 AWS IoT Analytics는 각 데이터 세트 콘텐츠 항목을 지정된 AWS IoT Events 입력에 메시지로 전송합니다. 예를 들어, 데이터 세트에 다음 콘텐츠가 포함된 경우가 있습니다.

```
"what", "who", "dt"
"overflow", "sensor01", "2019-09-16 09:04:00.000"
"overflow", "sensor02", "2019-09-16 09:07:00.000"
"underflow", "sensor01", "2019-09-16 11:09:00.000"
...
```


그런 다음 AWS IoT Analytics는 다음과 같은 필드가 포함된 메시지를 보냅니다.

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

관심 있는 필드(what, who, dt 중 하나 이상)를 인식하는 AWS IoT Events 입력을 만들고 이벤트에서 이러한 입력 필드를 사용하여 작업을 트리거하거나 내부 변수를 설정하는 AWS IoT Events 감지기 모델을 만드는 것이 좋습니다.

Jupyter 노트북

[Jupyter Notebook](#)은 애드혹 데이터 탐색 및 고급 분석 실행에 사용되는 스크립팅 언어용 오픈 소스 솔루션입니다. IoT 디바이스 데이터에 대해 심층적으로 분석하고 더 복잡한 분석을 적용하고, 예측을 위한 k-means 클러스터링 및 회귀 모델과 같은 기계 학습 방법을 사용할 수 있습니다.

AWS IoT Analytics는 Amazon SageMaker 노트북 인스턴스를 사용하여 Jupyter Notebook을 호스팅합니다. 노트북 인스턴스를 생성하려면 먼저 AWS IoT Analytics와 Amazon SageMaker 사이의 관계를 생성해야 합니다.

1. [SageMaker 콘솔](#)로 이동하여 노트북 인스턴스를 생성합니다.
 - a. 세부 정보를 입력하고 새 역할 생성을 선택합니다. 역할의 ARN을 기록해 둡니다.
 - b. 노트북 인스턴스를 생성합니다.
2. [IAM 콘솔](#)로 이동하여 SageMaker 역할을 수정합니다.
 - a. 역할을 엽니다. 하나의 관리형 정책이 있어야 합니다.
 - b. 인라인 정책 추가를 선택하고 서비스에서 iotAnalytics를 선택합니다. 작업 선택을 선택한 다음 검색 상자에 **GetDatasetContent**을 입력하고 선택합니다. 정책 검토를 선택합니다.
 - c. 정책이 정확한지 검토하고 이름을 입력한 다음 정책 생성을 선택합니다.

이렇게 하면 AWS IoT Analytics의 데이터 세트를 읽을 수 있는 역할 권한이 새로 생성됩니다.

1. <https://console.aws.amazon.com/iotanalytics/>으로 돌아가서 왼쪽 탐색 창에서 노트북을 선택합니다. 노트북 페이지에서 노트북 생성을 선택합니다.
2. 템플릿 선택 페이지에서 IoTA 빈 템플릿을 선택합니다.

3. 노트북 설정 페이지에서 노트북의 이름을 입력합니다. 데이터 세트 소스 선택에서 선택한 다음 이전에 생성한 데이터 세트를 선택합니다. 노트북 인스턴스 선택에서 SageMaker에서 생성한 노트북 인스턴스를 선택합니다.
4. 선택 사항을 검토한 후 노트북 생성을 선택합니다.
5. 노트북 페이지에서 노트북 인스턴스는 [Amazon SageMaker](#) 콘솔에서 열립니다.

노트북 템플릿

AWS IoT Analytics 노트북 템플릿에는 AWS에서 빌드한 기계 학습 모델 및 시각화가 포함되어 AWS IoT Analytics 사용 사례를 시작하는 데 도움이 됩니다. 이러한 노트북 템플릿을 사용하여 자세히 알아보거나 IoT 디바이스 데이터에 맞게 재사용하여 즉각적인 가치를 제공할 수 있습니다.

AWS IoT Analytics 콘솔에서 다음과 같은 노트북 템플릿을 확인할 수 있습니다.

- 상황별 변칙 감지 – Poisson 지수 가중 이동 평균(PEWMA) 모델을 사용하여 측정된 풍속의 상황별 변칙 감지 기능을 적용합니다.
- 솔라 패널 출력 예측 – 솔라 패널의 출력을 예측하는 구분적, 계절적, 선형 시계열 모델 적용.
- 제트 엔진의 예측 유지 보수 – 제트 엔진 고장을 예측하기 위한 장단기 메모리(LSTM) 신경망 및 로지스틱 회귀 분석을 적용합니다.
- 스마트 홈 고객 세분화 – 스마트 홈 사용량 데이터의 고객 세분화를 감지하는 k-means 및 주성분 분석(PCA) 적용.
- 스마트 시티 정체 예측 – 도시 고속도로의 활용도를 예측하는 LSTM 적용.
- 스마트 시티 공기 질 예측 – 도심의 특정 오염을 예측하는 LSTM 적용.

AWS IoT Analytics 시작하기

이 단원은 AWS IoT Analytics를 통해 디바이스 데이터를 수집, 저장, 처리 및 쿼리하는 데 사용할 수 있는 기본 명령에 대해 설명합니다. 여기에 표시된 예제에서는 AWS Command Line Interface(AWS CLI)를 사용합니다. AWS CLI에 대한 자세한 내용은 [AWS Command Line Interface 사용 설명서](#)를 참조하세요. AWS IoT에 사용할 수 있는 CLI 명령에 대한 자세한 내용은 AWS Command Line Interface 참조의 [iot](#)를 참조하세요.

⚠ Important

`aws iotanalytics` 명령을 사용하여 AWS CLI를 사용하는 AWS IoT Analytics와 상호 작용할 수 있습니다. `aws iot` 명령을 사용하여 AWS CLI를 사용하는 IoT 시스템의 다른 부분과 상호 작용할 수 있습니다.

ℹ Note

다음 예에서 AWS IoT Analytics 개체(채널, 데이터 세트, 데이터 스토어, 파이프라인)의 이름을 입력할 때 대문자를 사용할 경우 시스템이 자동으로 소문자로 변경하는 것을 알 수 있습니다. 개체 이름은 소문자로 시작해야 하며 소문자, 밑줄 및 숫자만 포함할 수 있습니다.

채널 생성

채널은 처리되지 않은 원시 메시지 데이터를 파이프라인에 게시하기 전에 이 데이터를 수집 및 보관합니다. 수신 메시지는 채널로 전송되므로, 첫 번째 단계는 데이터를 위한 채널을 생성하는 것입니다.

```
aws iotanalytics create-channel --channel-name mychannel
```

AWS IoT 메시지를 AWS IoT Analytics에서 수집하게 하려면 AWS IoT 규칙 엔진 규칙을 만들어 이 채널에 메시지를 보내십시오. 이 내용은 [AWS IoT Analytics로 데이터 수집](#)의 뒷부분에서 설명합니다. 채널에 데이터를 가져오는 또 다른 방법은 AWS IoT Analytics 명령 `BatchPutMessage`를 사용하는 것입니다.

이미 생성한 채널을 나열하려면:

```
aws iotanalytics list-channels
```

채널에 관한 추가 정보를 가져오려면.

```
aws iotanalytics describe-channel --channel-name mychannel
```

처리되지 않은 채널 메시지는 AWS IoT Analytics에서 관리하는 Amazon S3 버킷 또는 사용자가 관리하는 버킷에 저장됩니다. `channelStorage` 파라미터를 사용하여 저장할 버킷을 지정합니다. 기본값은 서비스 관리형 Amazon S3 버킷입니다. 사용자가 관리하는 Amazon S3 버킷에 채널 메시지를 저장하도록 선택한 경우, AWS IoT Analytics가 Amazon S3 버킷에서 `s3:GetBucketLocation`(버킷 위치 확인) `s3:PutObject`(저장), `s3:GetObject`(읽기), `s3:ListBucket`(재처리) 작업을 대신 수행할 수 있도록 권한을 부여해야 합니다.

Example

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "MyStatementSid",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-iot-analytics-bucket",
        "arn:aws:s3:::my-iot-analytics-bucket/*"
      ]
    }
  ]
}
```

고객 관리형 채널 스토리지의 옵션 또는 권한을 변경하는 경우 이전에 수집된 데이터가 데이터 세트 콘텐츠에 포함되도록 채널 데이터를 재처리해야 할 수 있습니다. [채널 데이터 재처리](#)를 참조하십시오.

데이터 스토어 생성

데이터 스토어는 메시지를 수신하고 저장합니다. 데이터 스토어는 데이터베이스가 아니라 확장 및 쿼리가 가능한 메시지 리포지토리입니다. 여러 데이터 스토어를 생성하여 여러 기기 또는 위치에서 오는 메시지를 저장하거나 단일 데이터 스토어를 사용하여 모든 AWS IoT 메시지를 수신할 수 있습니다.

```
aws iotanalytics create-datastore --datastore-name mydatastore
```

이미 생성한 데이터 스토어를 나열하려면.

```
aws iotanalytics list-datastores
```

데이터 스토어에 관한 추가 정보를 가져오려면.

```
aws iotanalytics describe-datastore --datastore-name mydatastore
```

AWS IoT Analytics 리소스에 대한 Amazon S3 정책

처리된 데이터 스토어 메시지는 AWS IoT Analytics에서 관리하는 Amazon S3 버킷 또는 사용자가 관리하는 버킷에 저장할 수 있습니다. 데이터 스토어를 생성할 때 `datastoreStorage` API 파라미터를 사용하여 원하는 Amazon S3 버킷을 선택합니다. 기본값은 서비스 관리형 Amazon S3 버킷입니다.

사용자가 관리하는 Amazon S3 버킷에 데이터 스토어 메시지를 저장하도록 선택한 경우, AWS IoT Analytics가 Amazon S3 버킷에서 작업을 대신 수행할 수 있도록 권한을 부여해야 합니다.

- `s3:GetBucketLocation`
- `s3:PutObject`
- `s3:DeleteObject`

데이터 스토어를 SQL 쿼리 데이터 세트의 소스로 사용하는 경우 버킷의 콘텐츠에 대해 Amazon Athena 쿼리를 호출할 수 있는 권한을 AWS IoT Analytics에 부여하도록 Amazon S3 버킷 정책을 설정해야 합니다.

Note

혼동된 대리자 보안 문제를 방지하는 데 도움이 되도록 버킷 정책에 `aws:SourceArn`을 지정하는 것이 좋습니다. 이렇게 하면 지정된 계정에서 오는 요청만 허용하여 액세스가 제한됩니다.

다. 혼동된 대리자 문제에 관한 자세한 내용은 [the section called “교차 서비스 혼동된 대리자 예방”](#)를 참조하세요.

다음은 이러한 필수 권한을 부여하는 버킷 정책의 예입니다.

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "MyStatementSid",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:PutObject",
        "s3>DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-EXAMPLE-DATASET",
            "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-EXAMPLE-DATASTORE"
          ]
        }
      }
    }
  ]
}
```

}

자세한 내용은 Amazon Athena 사용 설명서의 [크로스 계정 액세스](#)를 참조하세요.

Note

고객 관리형 데이터 스토어의 옵션 또는 권한을 업데이트하는 경우 이전에 수집된 데이터가 데이터 세트 콘텐츠에 포함되도록 채널 데이터를 재처리해야 할 수 있습니다. 자세한 내용은 [채널 데이터 재처리](#)를 참조하세요.

파일 형식

AWS IoT Analytics 데이터 스토어는 현재 JSON 및 Parquet 파일 형식을 지원합니다. 기본 파일 형식은 JSON입니다.

- [JSON\(JavaScript Object Notation\)](#) - 이름-값 쌍과 순서가 지정된 값 목록을 지원하는 텍스트 형식입니다.
- [Apache Parquet](#) - 대용량 데이터를 효율적으로 저장하고 쿼리하는 데 사용되는 열 방식 저장 형식입니다.

AWS IoT Analytics 데이터 스토어의 파일 형식을 구성하려면 데이터 스토어를 생성할 때 `FileFormatConfiguration` 객체를 사용하면 됩니다.

`fileFormatConfiguration`

파일 형식의 구성 정보가 들어 있습니다. AWS IoT Analytics 데이터 스토어는 JSON 및 Parquet를 지원합니다.

기본 파일 형식은 JSON입니다. 형식은 하나만 지정할 수 있습니다. 데이터 스토어를 생성한 후에는 파일 형식을 변경할 수 없습니다.

`jsonConfiguration`

JSON 형식의 구성 정보가 들어 있습니다.

`parquetConfiguration`

Parquet 형식의 구성 정보가 들어 있습니다.

schemaDefinition

스키마를 정의하는 데 필요한 정보입니다.

columns

데이터를 저장하는 하나 이상의 열을 지정합니다.

각 스키마에는 최대 100개의 열이 포함될 수 있습니다. 각 열에는 최대 100개의 중첩 유형이 포함될 수 있습니다.

name

열의 이름입니다.

길이 제한: 1~255자

type

데이터 유형입니다. 지원되는 데이터 유형에 대한 자세한 내용을 알아보려면 AWS Glue개발자 안내서의 [일반적인 데이터 유형](#)을 참조하세요.

길이 제한: 1~131072자

AWS IoT Analytics는 DECIMAL(*precision*, *scale*) - *precision*를 제외하고 [Amazon Athena의 데이터 유형](#) 페이지에 나열된 모든 데이터 유형을 지원합니다.

데이터 원본(콘솔) 생성

다음 절차는 Parquet 형식으로 데이터를 저장하는 데이터 스토어를 생성하는 방법을 보여줍니다.

데이터 스토어 생성

1. <https://console.aws.amazon.com/iotanalytics/>에 로그인합니다.
2. 탐색 창에서 데이터 스토어를 선택합니다.
3. 데이터 스토어 페이지에서 데이터 스토어 생성을 선택합니다.
4. 데이터 스토어 세부정보 지정 페이지에서 데이터 스토어에 대한 기본 정보를 입력합니다.
 - a. 데이터 스토어 ID에는 고유한 데이터 스토어 ID를 입력합니다. 이를 생성한 후에는 이 ID를 변경할 수 없습니다.
 - b. (선택사항) 태그에서 새 태그 추가를 선택하여 하나 이상의 사용자 지정 태그(키-값 쌍)를 데이터 스토어에 추가합니다. 태그를 사용하면 AWS IoT Analytics에 대해 생성하는 리소스를 식별하는 데 도움이 됩니다.

- c. 다음(Next)을 선택합니다.
- 5. 스토리지 유형 구성 페이지에서 데이터 저장 방법을 지정합니다.
 - a. 스토리지 유형에서 서비스 관리 스토리지를 선택합니다.
 - b. 처리된 데이터를 보관할 기간 구성에서 무기한을 선택합니다.
 - c. 다음(Next)을 선택합니다.
- 6. 데이터 형식 구성 페이지에서 데이터 레코드의 구조 및 형식을 정의합니다.
 - a. 분류에서 Parquet을 선택합니다. 데이터 스토어를 생성한 후에는 이 파일 형식을 변경할 수 없습니다.
 - b. 추론 소스에서 데이터 스토어의 JSON 문자열을 선택합니다.
 - c. 문자열에 다음 예와 같이 JSON 형식으로 스키마를 입력합니다.

```
{
  "device_id": "0001",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

- d. 스키마 추론을 선택합니다.
- e. Parquet 스키마 구성에서 형식이 JSON 예제와 일치하는지 확인합니다. 형식이 일치하지 않는 경우 Parquet 스키마를 수동으로 업데이트합니다.
 - 스키마에 더 많은 열을 표시하려면 새 열 추가를 선택하고 열 이름을 입력한 다음 데이터 유형을 선택합니다.

Note

기본적으로 스키마에는 100개의 열을 사용할 수 있습니다. 자세한 내용은 [AWS IoT Analytics quotas](#)를 참조하세요.

- 기존 열의 데이터 유형을 변경할 수 있습니다. 지원되는 데이터 유형에 대한 자세한 내용을 알아보려면 AWS Glue개발자 안내서의 [일반적인 데이터 유형](#)을 참조하세요.

Note

데이터 스토어를 생성한 후에는 기존 열의 데이터 유형을 변경할 수 없습니다.

- 기존 열을 제거하려면 열 제거를 선택합니다.
- f. 다음(Next)을 선택합니다.
7. (선택사항) AWS IoT Analytics는 데이터 스토어의 사용자 지정 파티션을 지원하므로 정리된 데이터를 쿼리하여 지연 시간을 개선할 수 있습니다. 지원되는 사용자 지정 파티션에 대한 자세한 내용은 [사용자 지정 파티션](#) 단원을 참조하세요.

다음(Next)을 선택합니다.

8. 검토 및 생성 페이지에서 선택 사항을 검토한 다음 데이터 스토어 생성을 선택합니다.

Important

데이터 스토어를 생성한 후에는 데이터 스토어 ID, 파일 형식 또는 열의 데이터 유형을 변경할 수 없습니다.

9. 새 데이터 스토어가 데이터 스토어 페이지에 나타나는지 확인하세요.

사용자 지정 파티션

AWS IoT Analytics는 데이터 파티셔닝을 지원하므로 데이터 스토어의 데이터를 정리할 수 있습니다. 데이터 파티셔닝을 사용하여 데이터를 구성하면 정리된 데이터를 쿼리할 수 있습니다. 이렇게 하면 쿼리당 스캔되는 데이터 양이 줄어들고 지연 시간이 개선됩니다.

메시지 데이터 속성 또는 파이프라인 활동을 통해 추가된 속성에 따라 데이터를 파티셔닝할 수 있습니다.

시작하려면 데이터 스토어의 데이터 파티셔닝을 활성화하세요. 하나 이상의 데이터 파티션 차원을 지정하고 파티셔닝된 데이터 스토어를 AWS IoT Analytics 파이프라인에 연결합니다. 그런 다음 이 WHERE 절을 활용하는 쿼리를 작성하여 성능을 최적화합니다.


데이터 원본(콘솔) 생성

다음 절차에서는 사용자 지정 파티션을 사용하여 데이터 스토어를 생성하는 방법을 보여줍니다.

데이터 스토어 생성

1. [AWS IoT Analytics 콘솔](#)에 로그인합니다.
2. 탐색 창에서 데이터 스토어를 선택합니다.
3. 데이터 스토어 페이지에서 데이터 스토어 생성을 선택합니다.

4. 데이터 스토어 세부정보 지정 페이지에서 데이터 스토어에 대한 기본 정보를 입력합니다.
 - a. 데이터 스토어 ID에는 고유한 데이터 스토어 ID를 입력합니다. 이를 생성한 후에는 이 ID를 변경할 수 없습니다.
 - b. (선택사항) 태그에서 새 태그 추가를 선택하여 하나 이상의 사용자 지정 태그(키-값 쌍)를 데이터 스토어에 추가합니다. 태그를 사용하면 AWS IoT Analytics에 대해 생성하는 리소스를 식별하는 데 도움이 됩니다.
 - c. 다음(Next)을 선택합니다.
5. 스토리지 유형 구성 페이지에서 데이터 저장 방법을 지정합니다.
 - a. 스토리지 유형에서 서비스 관리 스토리지를 선택합니다.
 - b. 처리된 데이터를 보관할 기간 구성에서 무기한을 선택합니다.
 - c. 다음(Next)을 선택합니다.
6. 데이터 형식 구성 페이지에서 데이터 레코드의 구조 및 형식을 정의합니다.
 - a. 데이터 스토어 데이터 형식 분류에서 JSON 또는 Parquet을 선택합니다. AWS IoT Analytics 지원되는 파일 유형에 대한 자세한 내용은 [파일 형식](#)를 참조하세요.

 Note

데이터 스토어를 생성한 후에는 이 파일 형식을 변경할 수 없습니다.

- b. 다음(Next)을 선택합니다.
7. 이 데이터 스토어를 위한 사용자 지정 파티션을 생성합니다.
 - a. 데이터 파티션 추가에서 활성화를 선택합니다.
 - b. 데이터 파티션 소스에서 파티션 소스에 대한 기본 정보를 지정합니다.

샘플 소스를 선택하고 이 데이터 스토어에 대한 메시지를 수집하는 AWS IoT Analytics 채널을 선택합니다.
 - c. 메시지 샘플 속성에서 데이터 스토어를 파티셔닝하는 데 사용할 메시지 속성을 선택합니다. 그런 다음 작업에서 선택 항목을 속성 파티션 차원 또는 타임스탬프 파티션 차원으로 추가합니다.

Note

데이터 스토어에는 타임스탬프 파티션을 하나만 추가할 수 있습니다.

- d. 사용자 지정 데이터 스토어 파티션 차원의 경우 파티션 차원에 대한 기본 정보를 정의합니다. 이전 단계에서 선택한 각 메시지 샘플 속성이 파티션의 차원이 됩니다. 다음 옵션을 사용하여 각 차원을 사용자 지정합니다.
- 파티션 유형 - 이 파티션 차원이 속성 파티션 유형인지 타임스탬프 파티션 유형인지 지정합니다.
 - 속성 이름 및 차원 이름 - 기본적으로 AWS IoT Analytics는 속성 파티션 차원의 식별자로 선택한 메시지 샘플 속성의 이름을 사용합니다. 속성 이름을 편집하여 파티션 차원의 이름을 사용자 지정합니다. WHERE 절의 차원 이름을 사용하여 쿼리 성능을 최적화할 수 있습니다.
 - 모든 파티션 속성 차원의 이름에는 `__partition_` 접두사가 붙습니다.
 - 타임스탬프 파티션 유형의 경우, AWS IoT Analytics가 `__year`, `__month`, `__day`, `__hour`라는 이름으로 다음 네 개의 차원을 생성합니다.
 - 정렬 - 파티션 차원을 재배열하여 쿼리 지연 시간을 개선합니다.

타임스탬프 형식의 경우 메시지 데이터에서 수집된 타임스탬프와 일치시켜 타임스탬프 파티션의 형식을 지정합니다. AWS IoT Analytics 나열된 형식 옵션 중 하나를 선택하거나 데이터 형식과 일치하는 옵션을 지정할 수 있습니다. [날짜 시간 포맷터](#) 지정에 대해 자세히 알아보세요.

메시지 속성이 아닌 새 차원을 추가하려면 새 파티션 추가를 선택합니다.

- e. 다음(Next)을 선택합니다.
8. 검토 및 생성 페이지에서 선택 사항을 검토한 다음 데이터 스토어 생성을 선택합니다.

Important

- 데이터 스토어를 생성한 후에는 데이터 스토어 ID를 변경할 수 없습니다.
- 기존 파티션을 편집하려면 다른 데이터 스토어를 만들고 파이프라인을 통해 데이터를 재처리해야 합니다.

9. 새 데이터 스토어가 데이터 스토어 페이지에 나타나는지 확인하세요.

파이프라인 생성

파이프라인은 채널의 메시지를 사용하고 사용자가 데이터 스토어에 저장하기 전에 메시지를 처리 및 필터링할 수 있습니다. 채널을 데이터 스토어에 연결하려면 파이프라인을 생성합니다. 가장 단순한 파이프라인에는 데이터를 수집할 채널을 지정하고 메시지를 보낼 데이터 스토어를 식별하는 것 외에는 다른 활동이 없습니다. 더 복잡한 파이프라인에 대한 자세한 내용은 [파이프라인 활동](#)을 참조하십시오.

처음에는 채널을 데이터 스토어에 연결하는 것 외에는 아무 작업도 수행하지 않는 파이프라인을 만드는 것이 좋습니다. 그러면 원시 데이터가 어떤 경로로 데이터 스토어까지 이동하는지 확인한 다음 이 데이터를 처리할 파이프라인 활동을 추가할 수 있습니다.

다음 명령을 실행해 파이프라인을 생성합니다.

```
aws iotanalytics create-pipeline --cli-input-json file://mypipeline.json
```

mypipeline.json 파일에는 다음 콘텐츠가 포함되어 있습니다.

```
{
  "pipelineName": "mypipeline",
  "pipelineActivities": [
    {
      "channel": {
        "name": "mychannelactivity",
        "channelName": "mychannel",
        "next": "mystoreactivity"
      }
    },
    {
      "datastore": {
        "name": "mystoreactivity",
        "datastoreName": "mydatastore"
      }
    }
  ]
}
```

다음 명령을 실행하여 기존 파이프라인을 나열합니다.

```
aws iotanalytics list-pipelines
```

다음 명령을 실행하여 개별 파이프라인의 구성을 확인합니다.

```
aws iotanalytics describe-pipeline --pipeline-name mypipeline
```

AWS IoT Analytics로 데이터 수집

쿼리할 수 있는 데이터 스토어에 데이터를 저장하는 파이프라인으로 데이터를 라우팅하는 채널이 있으면 AWS IoT Analytics로 메시지 데이터를 보낼 준비가 된 것입니다. 여기에서는 AWS IoT Analytics로 데이터를 가져오는 두 가지 방법을 보여줍니다. AWS IoT 메시지 브로커를 사용하거나 AWS IoT Analytics BatchPutMessage API를 사용하여 메시지를 보낼 수 있습니다.

주제

- [AWS IoT 메시지 브로커 사용](#)
- [BatchPutMessage API 사용](#)

AWS IoT 메시지 브로커 사용

AWS IoT 메시지 브로커를 사용하려면 AWS IoT 규칙 엔진을 사용하여 규칙을 생성합니다. 이 규칙은 특정 주제의 메시지를 AWS IoT Analytics로 라우팅합니다. 하지만 이 규칙을 사용하려면 먼저 필요한 권한을 부여하는 역할을 생성해야 합니다.

IAM 역할 생성

AWS IoT 메시지를 AWS IoT Analytics 채널로 라우팅하려면 규칙을 설정해야 합니다. 하지만 먼저 메시지 데이터를 AWS IoT Analytics 채널에 전송할 권한을 해당 규칙에 부여하는 IAM 역할을 생성해야 합니다.

다음 명령을 실행해 역할을 생성합니다.

```
aws iam create-role --role-name myAnalyticsRole --assume-role-policy-document file://arpd.json
```

arpd.json 파일 내용은 다음과 같은 형식으로 보입니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

        "Effect": "Allow",
        "Principal": {
            "Service": "iot.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
}

```

그런 다음 정책 문서를 역할에 연결합니다.

```
aws iam put-role-policy --role-name myAnalyticsRole --policy-name myAnalyticsPolicy --policy-document file://pd.json
```

pd.json 파일 내용은 다음과 같은 형식으로 보입니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotanalytics:BatchPutMessage",
      "Resource": [
        "arn:aws:iotanalytics:us-west-2:your-account-number:channel/mychannel"
      ]
    }
  ]
}

```

AWS IoT 규칙 생성

채널에 메시지를 전송하는 AWS IoT 규칙을 생성합니다.

```
aws iot create-topic-rule --rule-name analyticsTestRule --topic-rule-payload file://rule.json
```

rule.json 파일 내용은 다음과 같은 형식으로 보입니다.

```

{
  "sql": "SELECT * FROM 'iot/test'",

```

```

"ruleDisabled": false,
"awsIotSqlVersion": "2016-03-23",
"actions": [ {
  "iotAnalytics": {
    "channelName": "mychannel",
    "roleArn": "arn:aws:iam::your-account-number:role/myAnalyticsRole"
  }
} ]
}

```

iot/test를 라우팅해야 하는 메시지의 MQTT 주제로 바꿉니다. 채널 이름과 역할을 이전 단원에서 생성한 이름과 역할로 바꿉니다.

AWS IoT Analytics에 MQTT 메시지 보내기

규칙을 채널에, 채널을 파이프라인에, 파이프라인을 데이터 스토어에 결합한 후에는 규칙과 일치하는 모든 데이터가 AWS IoT Analytics를 통해 쿼리할 준비가 된 데이터 스토어로 전달됩니다. 이를 테스트 하기 위해 AWS IoT 콘솔을 사용하여 메시지를 보낼 수 있습니다.

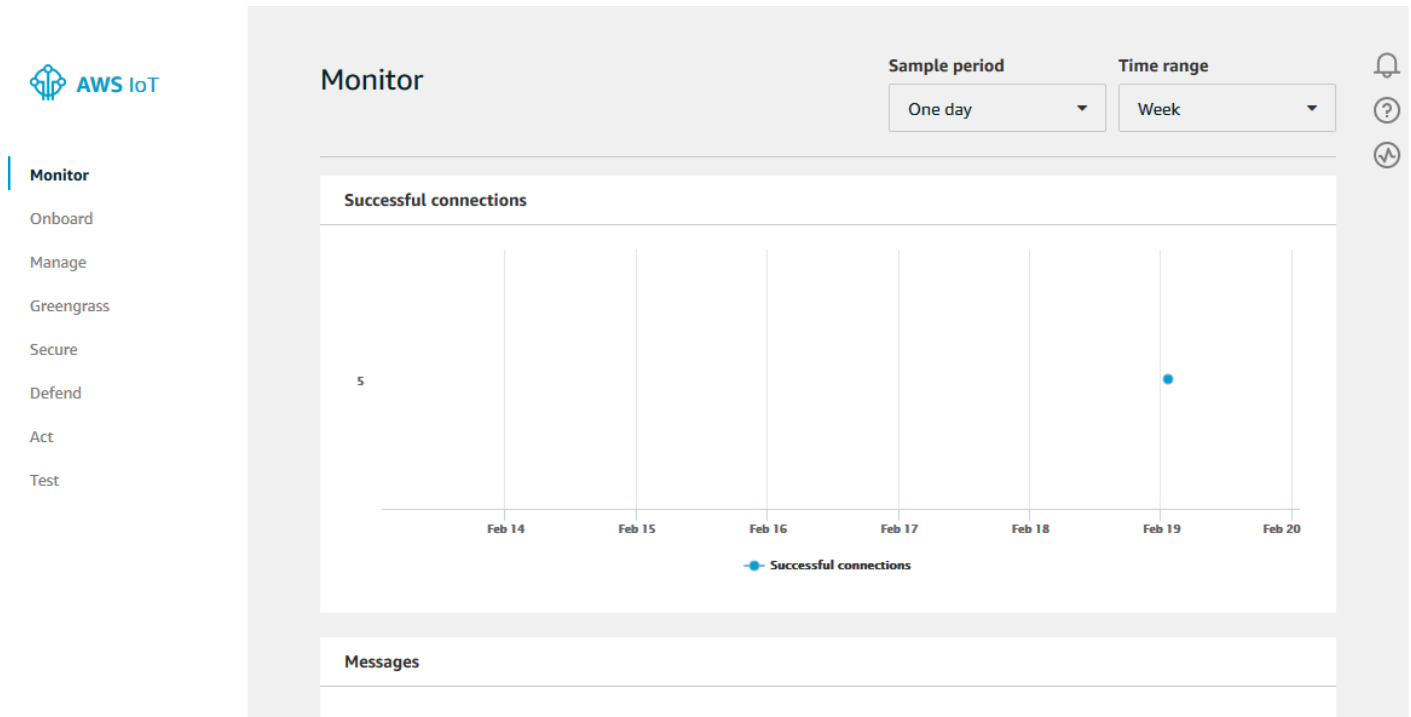
Note

AWS IoT Analytics로 보내는 메시지 페이로드(데이터)의 필드 이름

- 영숫자와 밑줄(_)만 포함해야 하며, 다른 특수 문자는 허용되지 않습니다.
- 알파벳 문자나 1개의 밑줄(_)로 시작해야 합니다.
- 하이픈(-)은 포함할 수 없습니다.
- 정규 표현식 조건: `^[A-Za-z_]([A-Za-z0-9]* | [A-Za-z0-9][A-Za-z0-9_]*)$`.
- 255자를 초과할 수 없습니다.
- 대소문자를 구분하지 않습니다. 동일한 페이로드에서 foo 및 F00라는 필드 이름은 중복으로 간주합니다.

예를 들어, 메시지 페이로드에서 {"temp_01": 29} 또는 {"_temp_01": 29}는 유효하지만, {"temp-01": 29}, {"01_temp": 29} 또는 {"__temp_01": 29}는 잘못된 것입니다.

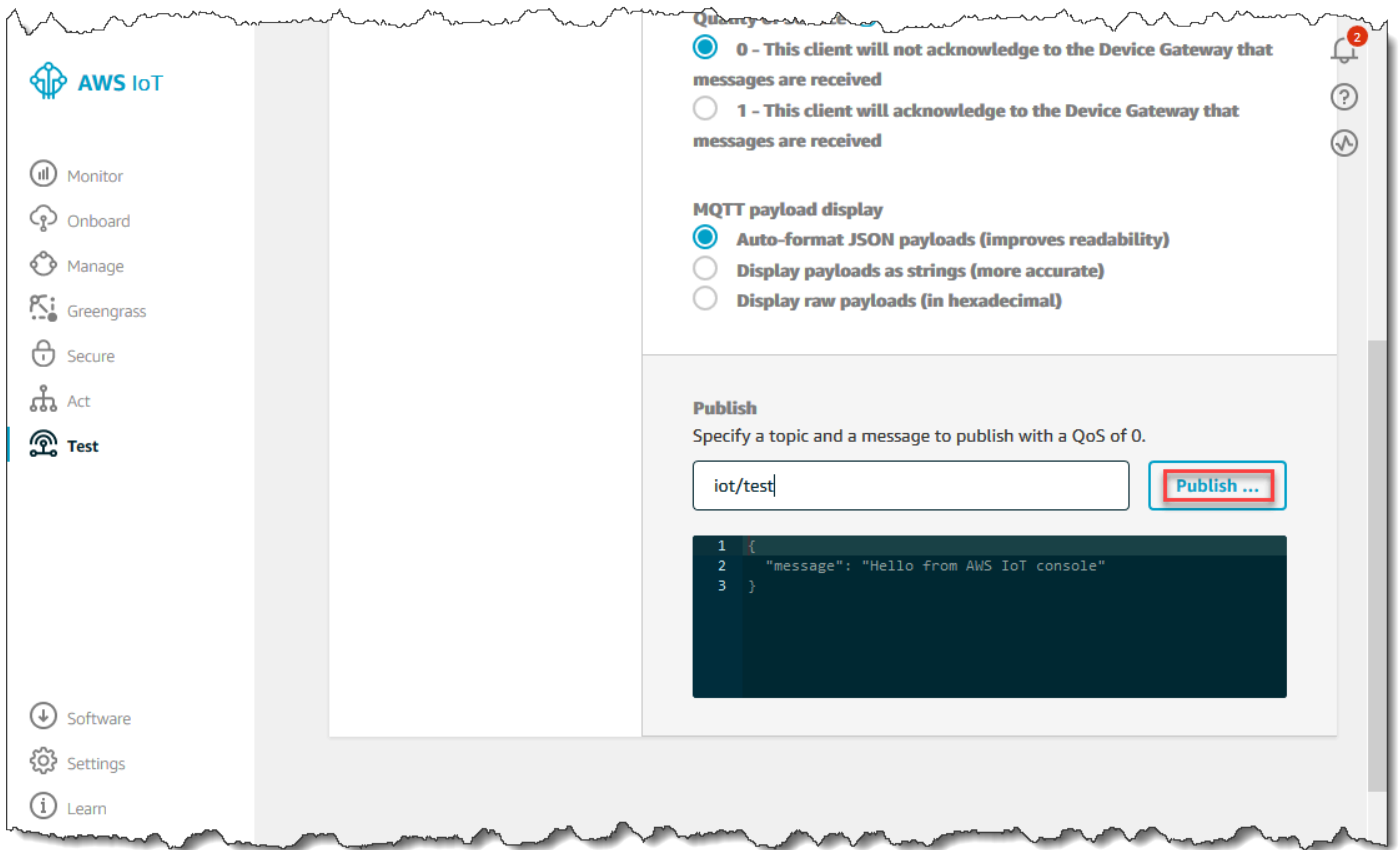
1. [AWS IoT 콘솔](#)의 왼쪽 탐색 창에서 테스트를 선택합니다.



2. [MQTT client] 페이지에서 [Publish] 섹션의 [Specify a topic]에 **iot/test**를 입력합니다. 메시지 페이로드 섹션에서 다음 JSON 내용이 존재하는지 확인하고, 그렇지 않은 경우 내용을 입력합니다.

```
{
  "message": "Hello from the IoT console"
}
```

3. [Publish to topic]을 선택합니다.



이렇게 하면 앞서 생성한 데이터 스토어로 라우팅할 메시지가 게시됩니다.

BatchPutMessage API 사용

AWS IoT Analytics로 메시지 데이터를 가져오는 또 다른 방법은 BatchPutMessage API 명령을 사용하는 것입니다. 이 방법은 특정한 주제의 메시지를 채널로 라우팅하는 AWS IoT 규칙을 설정할 필요가 없습니다. 그러나 채널로 데이터/메시지를 전송하는 디바이스가 AWS SDK로 생성한 소프트웨어를 실행할 수 있어야 하거나, AWS CLI를 사용하여 BatchPutMessage를 호출할 수 있어야 합니다.

1. 전송할 메시지를 포함하는 messages.json 파일을 생성합니다(이 예제에서는 단 하나의 메시지만 전송).

```
[
  { "messageId": "message01", "payload": "{ \"message\": \"Hello from the CLI\n\" }" }
]
```

2. batch-put-message 명령을 실행합니다.

```
aws iotanalytics batch-put-message --channel-name mychannel --messages file://
messages.json --cli-binary-format raw-in-base64-out
```

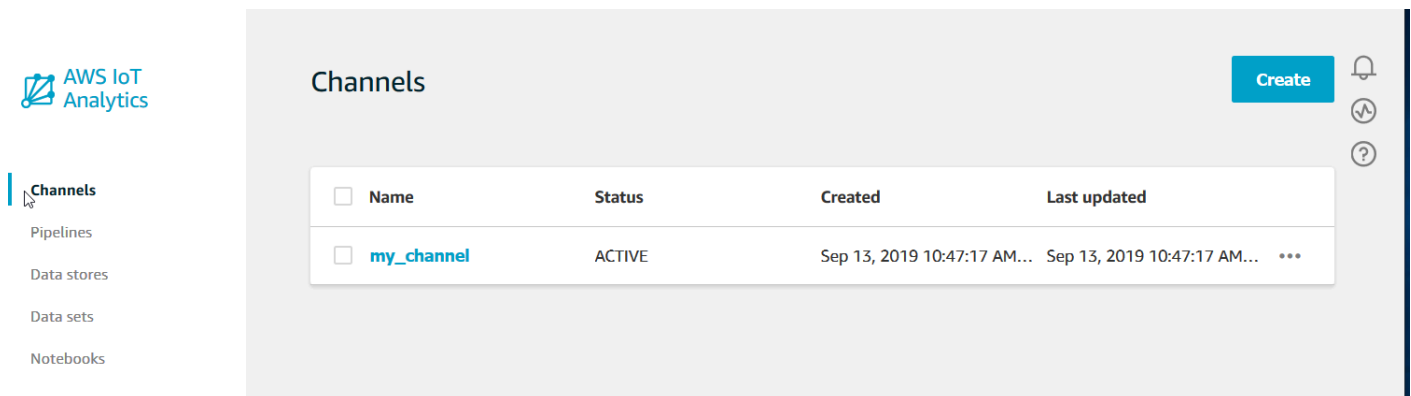
오류가 없다면, 다음과 같은 출력을 확인할 수 있습니다.

```
{
  "batchPutMessageErrorEntries": []
}
```

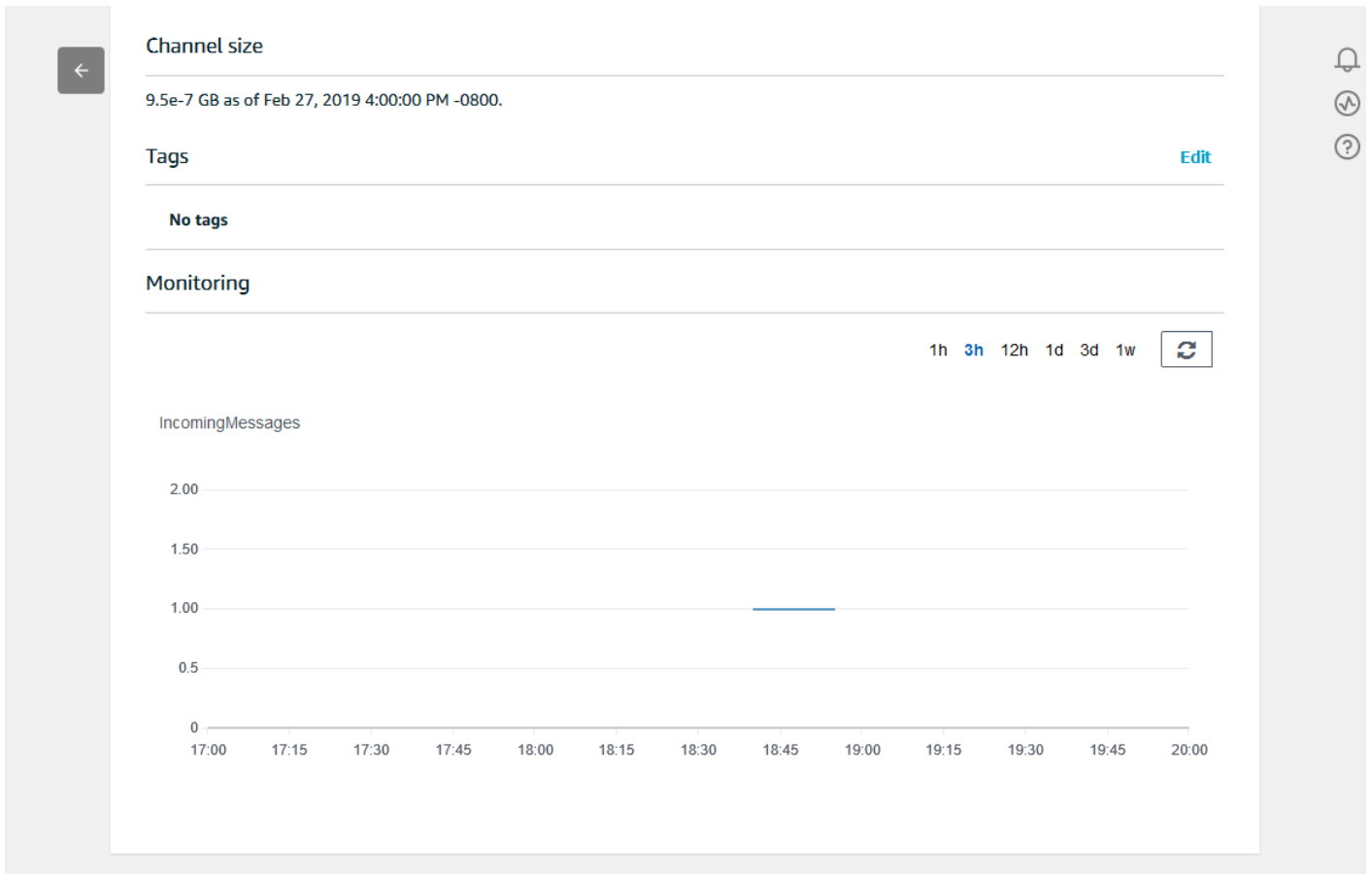
수집된 데이터 모니터링

AWS IoT Analytics 콘솔을 사용하여 전송한 메시지가 채널에서 수집되고 있는지 확인할 수 있습니다.

1. [AWS IoT Analytics 콘솔](#)의 왼쪽 탐색 창에서 준비를 선택하고(필요한 경우) 채널을 선택한 다음 이전에 생성한 채널 이름을 선택합니다.

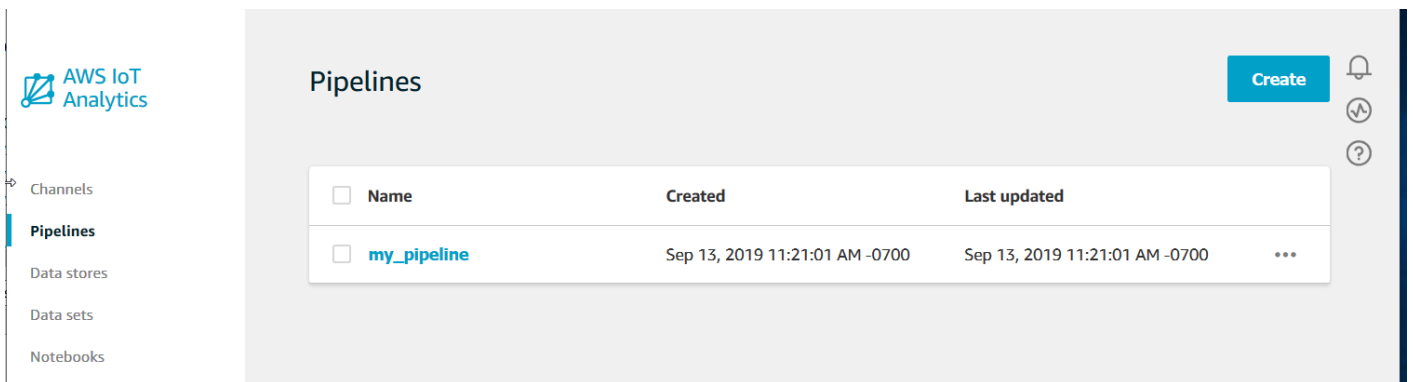


2. 채널 세부 정보 페이지에서 모니터링 섹션까지 아래로 스크롤합니다. 기간 표시기(1시간, 3시간, 12시간, 1일, 3일, 1주) 중 하나를 선택하여 필요에 따라 표시된 기간을 조정합니다. 지정된 기간 동안 이 채널에서 수집된 메시지 수를 나타내는 그래프 선이 표시되어야 합니다.

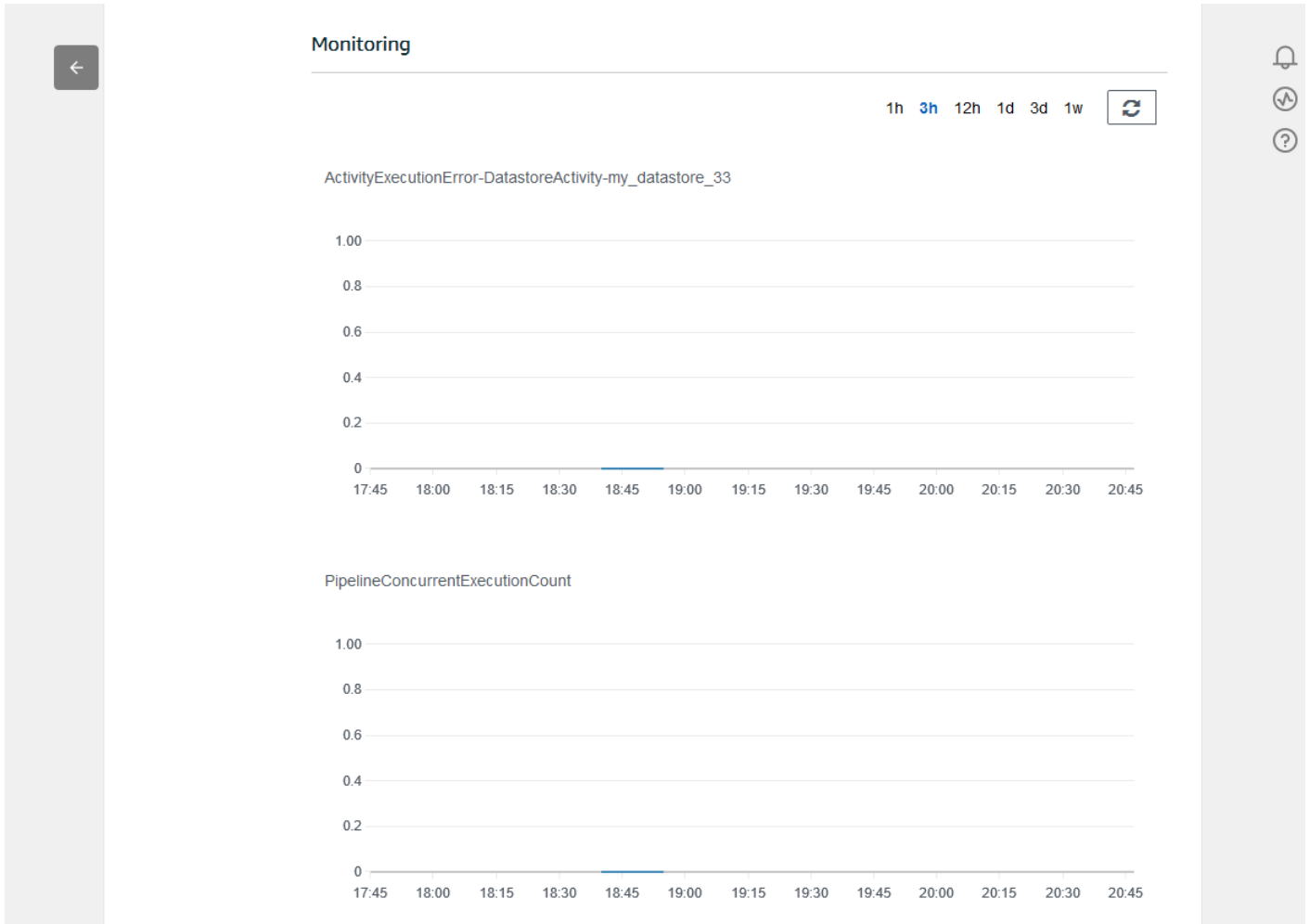


파이프라인 활동 실행을 확인하기 위한 유사한 모니터링 기능이 있습니다. 파이프라인의 세부 정보 페이지에서 활동 실행 오류를 모니터링할 수 있습니다. 파이프라인의 일부로 활동을 지정하지 않은 경우 실행 오류가 0개 표시되어야 합니다.

1. [AWS IoT Analytics 콘솔](#)의 왼쪽 탐색 창에서 준비를 선택하고 파이프라인을 선택한 다음 이전에 생성한 파이프라인 이름을 선택합니다.



2. 파이프라인 세부 정보 페이지에서 모니터링 섹션까지 아래로 스크롤합니다. 기간 표시기(1시간, 3시간, 12시간, 1일, 3일, 1주) 중 하나를 선택하여 필요에 따라 표시된 기간을 조정합니다. 지정된 기간 동안의 파이프라인 활동 실행 오류 수를 나타내는 그래프 선이 표시되어야 합니다.



데이터 세트 생성

SQL 데이터 세트 또는 컨테이너 데이터 세트를 생성하여 데이터 스토어에서 데이터를 검색합니다. AWS IoT Analytics는 데이터를 쿼리하여 분석 질문에 답할 수 있습니다. 데이터 스토어는 데이터베이스가 아니지만, SQL 표현식으로 데이터를 쿼리하고 데이터 세트에 저장된 결과를 얻을 수 있습니다.

주제

- [데이터 쿼리](#)
- [쿼리된 데이터에 액세스](#)

데이터 쿼리

데이터를 쿼리하려면 데이터 세트를 생성합니다. 데이터 세트에는 데이터 스토어 쿼리에 사용하는 SQL과 원하는 날짜 및 시간에 쿼리를 반복하는 일정 옵션이 들어 있습니다. 이 일정 옵션은 [Amazon CloudWatch 일정 표현식](#)과 유사한 표현식으로 생성합니다.

다음 명령을 실행하여 데이터 세트를 생성합니다.

```
aws iotanalytics create-dataset --cli-input-json file://mydataset.json
```

여기 mydataset.json 파일에 다음 콘텐츠가 포함되어 있습니다.

```
{
  "datasetName": "mydataset",
  "actions": [
    {
      "actionName": "myaction",
      "queryAction": {
        "sqlQuery": "select * from mydatastore"
      }
    }
  ]
}
```

다음 명령을 실행하여 쿼리를 실행해 데이터 세트 콘텐츠를 생성합니다.

```
aws iotanalytics create-dataset-content --dataset-name mydataset
```

데이터 세트 콘텐츠가 생성되기까지 몇 분 동안 기다린 후 계속합니다.

쿼리된 데이터에 액세스

쿼리의 결과는 CSV 형식의 파일로 저장된 데이터 세트 콘텐츠입니다. 파일은 Amazon S3을 통해 사용할 수 있습니다. 다음 예에서는 결과가 준비되어 있는지 확인하고 파일을 다운로드하는 방법을 보여줍니다.

다음 get-dataset-content 명령을 실행합니다.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

데이터 세트에 데이터가 포함된 경우, `get-dataset-content`는 아래와 같이 `status` 필드에 `"state": "SUCCEEDED"`를 출력합니다.

```
{
  "timestamp": 1508189965.746,
  "entries": [
    {
      "entryName": "someEntry",
      "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-Amz-"
    }
  ],
  "status": {
    "state": "SUCCEEDED",
    "reason": "A useful comment."
  }
}
```

`dataURI`는 출력 결과에 대한 서명된 URL입니다. 이 URL은 짧은 기간(몇 시간) 동안만 유효합니다. 워크플로우에 따라서는 내용에 액세스하기 전에 항상 `get-dataset-content`를 호출하는 경우도 있습니다. 이 명령을 호출하면 서명된 새 URL이 생성되기 때문입니다.

AWS IoT Analytics 데이터 탐색

AWS IoT Analytics 데이터 저장, 분석 및 시각화를 위한 몇 가지 옵션이 있습니다.

이 페이지의 주제:

- [Amazon S3](#)
- [AWS IoT Events](#)
- [Amazon QuickSight](#)
- [Jupyter 노트북](#)

Amazon S3

데이터 세트 콘텐츠를 [Amazon Simple Storage Service\(S3\)](#) 버킷으로 보내 기존 데이터 레이크와의 통합 또는 사내 애플리케이션 및 시각화 도구에서의 액세스를 활성화할 수 있습니다. [CreateDataset](#)의 `contentDeliveryRules::destination::s3DestinationConfiguration` 필드를 참조하세요.

AWS IoT Events

데이터 세트 콘텐츠를 AWS IoT Events의 입력으로 전송할 수 있습니다. 디바이스나 프로세스의 장애 또는 작동 변경을 모니터링하고 이러한 이벤트가 발생할 때 추가 작업을 트리거하는 서비스입니다.

이렇게 하려면 [CreateDataset](#)을 사용하여 데이터 세트를 만들고 `contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName` 필드에 AWS IoT Events 입력을 지정합니다. 또한 "iotevents:BatchPutMessage"를 실행할 AWS IoT Analytics 권한을 부여하는 역할의 `roleArn`을 지정해야 합니다. 데이터 세트의 콘텐츠가 생성될 때마다 AWS IoT Analytics가 각 데이터 세트 콘텐츠 항목을 지정된 AWS IoT Events 입력에 메시지로 보냅니다. 예를 들어, 데이터 세트에 다음 내용이 포함된 경우,

```
"what", "who", "dt"
"overflow", "sensor01", "2019-09-16 09:04:00.000"
"overflow", "sensor02", "2019-09-16 09:07:00.000"
"underflow", "sensor01", "2019-09-16 11:09:00.000"
...
```

AWS IoT Analytics가 다음과 같이 필드가 포함된 메시지를 보냅니다.

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

관심 있는 필드(what, who, dt 중 하나 이상)를 인식하는 AWS IoT Events 입력을 만들고 이벤트에서 이러한 입력 필드를 사용하여 작업을 트리거하거나 내부 변수를 설정하는 AWS IoT Events 감지기 모델을 만드는 것이 좋습니다.

Amazon QuickSight

AWS IoT Analytics는 [Amazon QuickSight](#)와의 직접 통합을 제공합니다. Amazon QuickSight는 데이터를 사용하여 시각적 객체를 빌드하고, 애드혹 분석을 수행하고, 비즈니스 관련 통찰력을 빠르게 얻을 수 있는 신속한 비즈니스 분석 서비스입니다. Amazon QuickSight는 강력한 인 메모리 엔진(SPICE)을 사용하여 사용자 수 수십만 명 규모로 조직을 확대하고, 즉각적인 성능을 발휘합니다. Amazon QuickSight는 [이러한 리전](#)에서 사용할 수 있습니다.

Jupyter 노트북

또한 Jupyter Notebook에서 AWS IoT Analytics 데이터 세트를 직접 사용하여 고급 분석 및 데이터 탐색을 수행할 수 있습니다. Jupyter Notebook 오픈 소스 솔루션입니다. <http://jupyter.org/install.html>에서 다운로드하여 설치할 수 있습니다. Amazon에서 호스팅하는 노트북 솔루션인 SageMaker와의 추가 통합 또한 가능합니다.

다양한 버전의 데이터 세트 유지

다음과 같은 [CreateDataset](#) 및 [UpdateDataset](#) API를 호출할 때 데이터 세트 retentionPeriod and versioningConfiguration 필드 값을 지정하여 데이터 세트 콘텐츠의 버전 수와 유지 기간을 선택할 수 있습니다.

```
...
"retentionPeriod": {
  "unlimited": "boolean",
  "numberOfDays": "integer"
},
"versioningConfiguration": {
  "unlimited": "boolean",
  "maxVersions": "integer"
},
...
```

이러한 두 파라미터의 설정은 다음과 같은 방식으로 유지할 데이터 세트 콘텐츠의 버전 수와 기간을 결정합니다.

	retentionPeriod	retentionPeriod:	retentionPeriod:
	[지정 안 함]	unlimited = TRUE, numberOfDays = 설정 안 함	unlimited = FALSE, numberOfDays = X
versioningConfiguration: [지정 안 함]	최신 버전과 최근에 성공한 버전(다른 경우)만 90일 간 유지됩니다.	최신 버전과 최근에 성공한 버전(다른 경우)만 무제한 유지됩니다.	최신 버전과 최근에 성공한 버전(다른 경우)만 X일 간 유지됩니다.

<p>versioningConfiguration: unlimited = TRUE, maxVersions 설정 안 함</p>	<p>개수에 관계없이 지난 90일 간의 모든 버전이 유지됩니다.</p>	<p>유지하는 버전 개수에는 제한이 없습니다.</p>	<p>개수에 관계없이 지난 X일 간의 모든 버전이 유지됩니다.</p>
<p>versioningConfiguration: unlimited = FALSE, maxVersions = Y</p>	<p>지난 90일 간의 버전이 Y개 이하로 유지됩니다.</p>	<p>기간에 관계없이 최대 Y개의 버전이 유지됩니다.</p>	<p>지난 X일 간의 버전이 Y개 이하로 유지됩니다.</p>

메시지 페이로드 구문

AWS IoT Analytics로 보내는 메시지 페이로드(데이터)의 필드 이름:

- 영숫자와 밑줄(_)만 포함해야 하며, 다른 특수 문자는 허용되지 않습니다.
- 알파벳 문자나 1개의 밑줄(_)로 시작해야 합니다.
- 하이픈(-)은 포함할 수 없습니다.
- 정규 표현식 조건: `^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$`.
- 255자를 초과할 수 없습니다.
- 대소문자를 구분하지 않습니다. 동일한 페이로드에서 "foo" 및 "FOO"라는 필드 이름은 중복으로 간주합니다.

예를 들어, 메시지 페이로드에서 {"temp_01": 29} 또는 {"_temp_01": 29}는 유효하지만, {"temp-01": 29}, {"01_temp": 29} 또는 {"__temp_01": 29}는 잘못된 것입니다.

AWS IoT SiteWise 데이터 작업

AWS IoT SiteWise는 대규모로 산업 장비 데이터를 수집, 모델링, 구성 및 분석할 수 있는 관리형 서비스입니다. 이 서비스는 산업 디바이스, 프로세스 및 시설 표현을 구축하는 데 사용할 수 있는 자산 모델링 프레임워크를 제공합니다.

AWS IoT SiteWise 자산 모델을 사용하면 사용할 산업 장비 데이터 및 데이터를 복잡한 지표로 처리하는 방식을 정의할 수 있습니다. AWS 클라우드에서 데이터를 수집하고 처리하도록 자산 모델을 구성할 수 있습니다. 자세한 내용은 [AWS IoT SiteWise](#) 사용 설명서를 참조하십시오.

AWS IoT Analytics는 AWS IoT SiteWise와 통합되므로 AWS IoT SiteWise 데이터에 대한 SQL 쿼리를 실행하고 예약할 수 있습니다. AWS IoT SiteWise 데이터 쿼리를 시작하려면 AWS IoT SiteWise 사용 설명서의 [스토리지 설정 구성](#)의 절차에 따라 데이터 스토어를 만듭니다. 그런 다음, [AWS IoT SiteWise 데이터로 데이터 세트 생성\(콘솔\)](#) 또는 [AWS IoT SiteWise 데이터\(AWS CLI\)로 데이터 세트 만들기](#)의 단계에 따라 AWS IoT Analytics 데이터 세트를 만들고 산업 데이터에 대해 SQL 쿼리를 실행합니다.

주제

- [AWS IoT SiteWise 데이터로 AWS IoT Analytics 데이터 세트 생성](#)
- [데이터 세트 콘텐츠 액세스](#)
- [튜토리얼: AWS IoT Analytics에서 AWS IoT SiteWise 데이터 쿼리](#)

AWS IoT SiteWise 데이터로 AWS IoT Analytics 데이터 세트 생성

AWS IoT Analytics 데이터 세트에는 데이터 스토어에서 데이터 쿼리에 사용하는 SQL 문과 표현식 및 원하는 날짜 및 시간에 쿼리를 반복하는 일정 옵션이 포함됩니다. 이 일정 옵션은 [Amazon CloudWatch 일정 표현식](#)과 유사한 표현식을 사용하여 생성할 수 있습니다.

Note

데이터 세트는 일반적으로 테이블 형식으로 구성되거나 구성되지 않을 수 있는 데이터 모음입니다. 반대로 AWS IoT Analytics는 데이터 스토어의 데이터에 SQL 쿼리를 적용하여 데이터 세트를 만듭니다.

다음 단계를 따라 AWS IoT SiteWise 데이터에 대한 데이터 세트 생성 작업을 시작합니다.

주제

- [AWS IoT SiteWise 데이터로 데이터 세트 생성\(콘솔\)](#)
- [AWS IoT SiteWise 데이터\(AWS CLI\)로 데이터 세트 만들기](#)

AWS IoT SiteWise 데이터로 데이터 세트 생성(콘솔)

다음 단계를 사용하여 AWS IoT Analytics 콘솔에서 AWS IoT SiteWise 데이터에 대한 데이터 세트를 만듭니다.

데이터 세트를 생성하려면

1. <https://console.aws.amazon.com/iotanalytics/>에서 왼쪽 탐색 창에서 데이터 세트를 선택합니다.
2. 데이터 세트 생성 페이지에서 SQL 생성을 선택합니다.
3. 데이터 세트 세부 정보 지정 페이지에서 데이터 세트의 세부 정보를 지정합니다.
 - a. 데이터 세트의 이름을 입력합니다.
 - b. 데이터 스토어 소스에서 AWS IoT SiteWise 데이터 스토어를 식별하는 고유 ID를 선택합니다.
 - c. (선택 사항)태그에서 하나 이상의 사용자 정의 태그(키-값 쌍)를 데이터 세트에 추가합니다.
4. SQL 식을 사용하여 데이터를 쿼리하고 분석 질문에 답변합니다.
 - a. 작성자 쿼리 필드에 와일드카드를 사용하여 최대 5행의 데이터를 표시하는 SQL 쿼리를 입력합니다.

```
SELECT * FROM my_iotsitewise_datastore.asset_metadata LIMIT 5
```

AWS IoT Analytics에서 지원 SQL 함수에 대한 자세한 내용은 [AWS IoT Analytics의 SQL 표현식](#)을 참조하세요. 또는 데이터에 대한 통찰력을 제공할 수 있는 통계 쿼리의 예인 [튜토리얼: AWS IoT Analytics에서 AWS IoT SiteWise 데이터 쿼리](#) 항목을 참조하세요.

- b. 쿼리 테스트를 선택하여 입력이 올바른지 확인하고 쿼리 다음에 결과를 테이블로 표시할 수 있습니다.

Note

Amazon Athena가 [최대 실행 쿼리 수를 제한하므로](#) SQL 쿼리가 장기간 실행되지 않도록 적절한 크기로 제한해야 합니다.

5. (선택 사항) 지정된 기간의 데이터를 사용하여 데이터 세트 콘텐츠를 만드는 경우, 일부 데이터가 처리 시간 내에 도착하지 않을 수 있습니다. 지연을 허용하기 위해 오프셋 또는 델타를 지정할 수 있습니다. 자세한 내용은 [Amazon CloudWatch Events를 통해 지연 데이터 알림 받기](#) 섹션을 참조하세요.

데이터 선택 필터 구성 페이지에서 데이터 선택 필터를 구성한 후 다음을 선택합니다.

- (선택 사항) 쿼리 일정 설정 페이지에서 이 쿼리가 정기적으로 실행되도록 예약하여 데이터 세트를 새로 고침할 수 있습니다. 데이터 세트 일정은 언제든지 만들고 편집할 수 있습니다.

Note

AWS IoT SiteWise의 데이터는 6시간마다 AWS IoT Analytics로 수집됩니다. 빈도를 6시간 이상으로 선택하는 것이 좋습니다.

빈도에서 옵션을 선택하고 다음을 선택합니다.

- AWS IoT Analytics는 이 데이터 세트 콘텐츠의 버전을 생성하고 지정된 기간 동안 분석 결과를 저장합니다. 90일을 권장하지만, 사용자 지정 보존 정책을 설정하도록 선택할 수 있습니다. 데이터 세트 콘텐츠의 저장된 버전 수를 제한할 수도 있습니다.

데이터 세트 결과 구성 페이지에서 옵션을 선택한 후 다음을 선택합니다.

- (선택사항) AWS IoT Events와 같은 특정 대상에 데이터 세트 결과의 전송 규칙을 구성할 수 있습니다.

데이터 세트 콘텐츠 전송 규칙 구성 페이지에서 옵션을 선택한 후 다음을 선택합니다.

- 선택 사항을 검토한 다음 데이터 세트 생성을 선택합니다.
- 데이터 세트 페이지에 새 데이터 세트가 나타나는지 확인하세요.

AWS IoT SiteWise 데이터(AWS CLI)로 데이터 세트 만들기

다음 AWS CLI 명령어를 실행하여 AWS IoT SiteWise 데이터 쿼리를 시작합니다.

여기에 표시된 예제에서는 AWS Command Line Interface(AWS CLI)를 사용합니다. AWS CLI에 대한 자세한 내용은 [AWS Command Line Interface 사용 설명서](#)를 참조하세요. AWS IoT Analytics에 사용할 수 있는 CLI 명령에 대한 자세한 내용은 AWS Command Line Interface 참조의 [iotanalytics](#)를 참조하세요.

데이터 세트를 생성하려면

- 다음 create-dataset 명령어를 실행하여 데이터 세트를 생성합니다.

```
aws iotanalytics create-dataset --cli-input-json file://my_dataset.json
```

여기 my_dataset.json 파일에 다음 콘텐츠가 포함되어 있습니다.

```
{
  "datasetName": "my_dataset",
  "actions": [
    {
      "actionName": "my_action",
      "queryAction": {
        "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
LIMIT 5"
      }
    }
  ]
}
```

AWS IoT Analytics에서 지원 SQL 함수에 대한 자세한 내용은 [AWS IoT Analytics의 SQL 표현식](#)을 참조하세요. 또는 데이터에 대한 통찰력을 제공할 수 있는 통계 쿼리의 예인 [튜토리얼: AWS IoT Analytics에서 AWS IoT SiteWise 데이터 쿼리](#) 항목을 참조하세요.

2. 다음 `create-dataset-content` 명령어를 실행하여 쿼리를 실행해 데이터 세트 콘텐츠를 생성합니다.

```
aws iotanalytics create-dataset-content --dataset-name my_dataset
```

데이터 세트 콘텐츠 액세스

SQL 쿼리의 결과는 CSV 형식의 파일로 저장된 데이터 세트 콘텐츠입니다. 파일은 Amazon S3을 통해 사용할 수 있습니다. 다음 단계에서는 결과가 준비되어 있는지 확인하고 파일을 다운로드하는 방법을 보여줍니다.

주제

- [AWS IoT Analytics\(콘솔\)에서 데이터 세트 콘텐츠에 액세스](#)
- [AWS IoT Analytics\(AWS CLI\)에서 데이터 세트 콘텐츠에 액세스](#)

AWS IoT Analytics(콘솔)에서 데이터 세트 콘텐츠에 액세스

데이터 세트에 데이터가 포함된 경우 AWS IoT Analytics 콘솔에서 SQL 쿼리 결과를 미리 보고 다운로드할 수 있습니다.

AWS IoT Analytics 데이터 세트 결과에 액세스하려면

1. 콘솔의 데이터 세트 페이지에서 액세스하려는 데이터 세트의 이름을 선택합니다.
2. 데이터 세트 요약 페이지에서 콘텐츠 탭을 선택합니다.
3. 데이터 세트 콘텐츠 테이블에서 결과를 미리 보려는 쿼리의 이름을 선택하거나 결과의 csv 파일을 다운로드합니다.

AWS IoT Analytics(AWS CLI)에서 데이터 세트 콘텐츠에 액세스

데이터 세트에 데이터가 포함된 경우 SQL 쿼리 결과를 미리 보고 다운로드할 수 있습니다.

여기에 표시된 예제에서는 AWS Command Line Interface(AWS CLI)를 사용합니다. AWS CLI에 대한 자세한 내용은 [AWS Command Line Interface 사용 설명서](#)를 참조하세요. AWS IoT Analytics에 사용할 수 있는 CLI 명령에 대한 자세한 내용은 AWS Command Line Interface 참조의 [iotanalytics](#)를 참조하세요.

AWS IoT Analytics 데이터 세트 결과(AWS CLI)에 액세스하려면

1. 다음 `get-dataset-content` 명령어를 실행하여 쿼리 결과를 확인합니다.

```
aws iotanalytics get-dataset-content --dataset-name my_iotsitewise_dataset
```

2. 데이터 세트에 데이터가 포함된 경우 다음 예와 같이 `get-dataset-content`의 출력이 `status` 필드에 `"state": "SUCCEEDED"`를 포함합니다.

```
{
  "timestamp": 1508189965.746,
  "entries": [
    {
      "entryName": "my_entry_name",
      "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-Amz-"
    }
  ],
  "status": {
    "state": "SUCCEEDED",
    "reason": "A useful comment."
  }
}
```

}

3. `get-dataset-content`의 출력에는 출력 결과에 대한 서명된 URL인 `dataURI`가 포함됩니다. 이 URL은 짧은 기간(몇 시간) 동안만 유효합니다. `dataURI` URL을 방문하여 SQL 쿼리 결과에 액세스할 수 있습니다.

Note

워크플로우에 따라서는 내용에 액세스하기 전에 항상 `get-dataset-content`를 호출하는 경우도 있습니다. 이 명령을 호출하면 서명된 새 URL이 생성되기 때문입니다.

튜토리얼: AWS IoT Analytics에서 AWS IoT SiteWise 데이터 쿼리

이 튜토리얼은 AWS IoT Analytics에서 AWS IoT SiteWise 데이터를 쿼리하는 방법을 설명합니다. 이 튜토리얼은 풍력 발전소의 샘플 데이터 세트를 제공하는 AWS IoT SiteWise 데모에서 데이터를 사용합니다.

Important

이 데모가 생성하고 사용하는 리소스에 대한 요금이 청구됩니다.

주제

- [필수 조건](#)
- [데이터 로드 및 확인](#)
- [데이터 탐색](#)
- [통계 쿼리 실행](#)
- [튜토리얼 리소스 정리](#)

필수 조건

이 튜토리얼을 이해하려면 다음 리소스가 필요합니다.

- AWS IoT SiteWise와 AWS IoT Analytics을 시작하려면 AWS 계정이 있어야 합니다. 계정이 없는 경우에는 [계정 AWS 생성하기](#)에서 절차에 따라 계정을 만드세요.

- Windows, macOS, Linux 또는 Unix를 실행하며 AWS Management Console에 액세스할 수 있는 개발 컴퓨터. 자세한 내용은 [AWS Management Console 시작하기](#)를 참조하십시오.
- AWS IoT SiteWise 모델 및 자산을 정의하고 풍력 발전소 설비의 데이터를 나타내는 스트리밍 데이터를 제공하는 AWS IoT SiteWise 데이터. 데이터를 생성하려면 AWS IoT SiteWise사용 설명서의 [데모 AWS IoT SiteWise 생성하기](#) 단계를 따르세요.
- 관리하는 기존 데이터 저장소의 AWS IoT SiteWise 데모 풍력 발전소 장비 데이터 AWS IoT SiteWise 데이터에 대한 데이터 스토어를 생성하는 방법에 대한 자세한 내용은 AWS IoT SiteWise사용 설명서의 [스토리지 설정 구성](#)을 참조하세요.

Note

AWS IoT SiteWise 메타데이터는 생성 직후 AWS IoT SiteWise 데이터 스토어에 표시되지만, 원시 데이터가 표시되는 데 최대 6시간이 소요될 수 있습니다. 그동안 AWS IoT Analytics 데이터 세트를 생성하고 메타데이터에 대한 쿼리를 실행할 수 있습니다.

다음 단계

[데이터 로드 및 확인](#)

데이터 로드 및 확인

이 튜토리얼에서 쿼리하는 데이터는 풍력 발전소의 풍력 엔진 터빈을 모델링하는 샘플 AWS IoT SiteWise 데이터 세트입니다.

Note

이 튜토리얼에서는 데이터 스토어에 있는 세 개의 테이블을 쿼리합니다.

- raw - 각 자산의 처리되지 않은 원시 데이터를 포함합니다.
- asset_metadata - 각 자산에 대한 일반 정보를 포함합니다.
- asset_hierarchy_metadata - 자산 간의 관계에 대한 정보를 포함합니다.

이 튜토리얼의 SQL 쿼리를 실행하려면

1. [AWS IoT SiteWise 데이터로 데이터 세트 생성\(콘솔\)](#) 또는 [AWS IoT SiteWise 데이터\(AWS CLI\)로 데이터 세트 만들기](#)에 있는 단계를 따라 AWS IoT SiteWise 데이터에 대한 AWS IoT Analytics 데이터 세트를 만듭니다.
2. 이 튜토리얼 전체에서 데이터 세트 쿼리를 업데이트하려면 다음을 수행합니다.
 - a. AWS IoT Analytics 콘솔의 데이터 세트 페이지에서 이전 페이지에서 만든 데이터 세트의 이름을 선택합니다.
 - b. 데이터 세트 요약 페이지에서 편집을 선택하여 SQL 쿼리를 수정합니다.
 - c. 쿼리 다음에 결과를 테이블로 표시하려면 쿼리 테스트를 선택합니다.

또는 다음 `update-dataset` 명령을 실행하여 AWS CLI를 사용하여 SQL 쿼리를 수정할 수 있습니다.

```
aws iotanalytics update-dataset --cli-input-json file://update-query.json
```

`update-query.json`의 콘텐츠:

```
{
  "datasetName": "my_dataset",
  "actions": [
    {
      "actionName": "myDatasetUpdateAction",
      "queryAction": {
        "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
LIMIT 3"
      }
    }
  ]
}
```

3. AWS IoT Analytics 콘솔 또는 AWS CLI를 사용하여 데이터에 대해 다음 쿼리를 실행하여 `asset_metadata` 테이블이 성공적으로 로드되었는지 확인합니다.

```
SELECT COUNT(*) FROM my_iotsitewise_datastore.asset_metadata
```

마찬가지로, `asset_hierarchy_metadata` 및 `raw` 테이블이 비어 있지 않은지 확인할 수 있습니다.

다음 단계

[데이터 탐색](#)

데이터 탐색

AWS IoT SiteWise 데이터를 생성하여 데이터 스토어에 로드한 후에는 AWS IoT Analytics 데이터 세트를 만들고 AWS IoT Analytics에서 SQL 쿼리를 실행하여 자산에 대한 통찰력을 확보할 수 있습니다. 다음 쿼리는 통계 쿼리를 실행하기 전에 데이터를 탐색하는 방법을 보여줍니다.

SQL 쿼리로 데이터를 탐색하려면

1. 원시 테이블과 같은 각 테이블의 열 및 값 샘플을 확인합니다.

```
SELECT * FROM my_iotsitewise_datastore.raw LIMIT 5
```

2. SELECT DISTINCT를 사용하여 asset_metadata 테이블을 쿼리하고 AWS IoT SiteWise 자산의 (고유한) 이름을 나열합니다.

```
SELECT DISTINCT assetname FROM my_iotsitewise_datastore.asset_metadata ORDER BY assetname
```

3. 특정 AWS IoT SiteWise 자산의 속성에 대한 정보를 나열하려면 WHERE 절을 사용하세요.

```
SELECT assetpropertyname,
       assetpropertyunit,
       assetpropertydatatype
FROM my_iotsitewise_datastore.asset_metadata
WHERE assetname = 'Demo Turbine Asset 2'
```

4. AWS IoT Analytics를 사용하면 다음 예제와 같이 데이터 스토어에 있는 둘 이상의 테이블에서 데이터를 조인할 수 있습니다.

```
SELECT * FROM my_iotsitewise_datastore.raw AS raw
JOIN my_iotsitewise_datastore.asset_metadata AS asset_metadata
ON raw.seriesId = asset_metadata.timeseriesId
```

자산 간의 모든 관계를 보려면 다음 쿼리의 JOIN 기능을 사용하세요.

```
SELECT DISTINCT parent.assetName as "Parent name",
       child.assetName AS "Child name"
FROM (
```

```

SELECT sourceAssetId AS parent,
       targetAssetId AS child
FROM my_iotsitewise_datastore.asset_hierarchy_metadata
WHERE associationType = 'CHILD'
)
AS relations
JOIN my_iotsitewise_datastore.asset_metadata AS child
  ON relations.child = child.assetId
JOIN my_iotsitewise_datastore.asset_metadata AS parent
  ON relations.parent = parent.assetId

```

다음 단계

[통계 쿼리 실행](#)

통계 쿼리 실행

이제 AWS IoT SiteWise 데이터를 탐색했으니 산업용 장비에 대한 귀중한 통찰력을 제공하는 통계 쿼리를 실행할 수 있습니다. 다음 쿼리는 검색할 수 있는 일부 정보를 보여줍니다.

AWS IoT SiteWise 데모 풍력 발전소 데이터에 대한 통계 쿼리를 실행하려면

1. 다음 SQL 명령을 실행하여 특정 자산(Demo Turbine Asset 4)에 대한 숫자 값이 있는 모든 속성의 최신 값을 찾습니다.

```

SELECT assetName,
       assetPropertyName,
       assetPropertyUnit,
       max_by(value, timeInSeconds) AS Latest
FROM (
  SELECT *,
         CASE assetPropertyDataType
           WHEN 'DOUBLE' THEN
             cast(doubleValue AS varchar)
           WHEN 'INTEGER' THEN
             cast(integerValue AS varchar)
           WHEN 'STRING' THEN
             stringValue
           WHEN 'BOOLEAN' THEN
             cast(booleanValue AS varchar)
           ELSE NULL
         END AS value

```

```

FROM my_iotsitewise_datastore.asset_metadata AS asset_metadata
JOIN my_iotsitewise_datastore.raw AS raw
    ON raw.seriesId = asset_metadata.timeSeriesId
WHERE startYear=2021
    AND startMonth=7
    AND startDay=8
    AND assetName='Demo Turbine Asset 4'
)
GROUP BY assetName, assetPropertyName, assetPropertyUnit

```

2. 메타데이터 테이블과 원시 테이블을 모두 조인하여 상위 자산 외에도 모든 자산의 최대 풍속 속성을 식별할 수 있습니다.

```

SELECT child_assets_data_set.parentAssetId,
    child_assets_data_set.childAssetId,
    asset_metadata.assetPropertyId,
    asset_metadata.assetPropertyName,
    asset_metadata.timeSeriesId,
    raw_data_set.max_speed
FROM (
    SELECT sourceAssetId AS parentAssetId,
        targetAssetId AS childAssetId
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS child_assets_data_set
JOIN mls_demo.asset_metadata AS asset_metadata
    ON asset_metadata.assetId = child_assets_data_set.childAssetId
JOIN (
    SELECT seriesId, MAX(doubleValue) AS max_speed
    FROM my_iotsitewise_datastore.raw
    GROUP BY seriesId
)
AS raw_data_set
ON raw_data_set.seriesId = asset_metadata.timeseriesid
WHERE assetPropertyName = 'Wind Speed'
ORDER BY max_speed DESC

```

3. 자산(Demo Turbine Asset 2)에 대한 특정 속성(Wind Speed)의 평균 값을 구하려면 다음 SQL 명령을 실행합니다. my_bucket_id를 버킷의 ID로 바꿉니다.

```

SELECT AVG(doubleValue) as "Average wind speed"
FROM my_iotsitewise_datastore.raw

```

```
WHERE seriesId =
  (SELECT timeseriesId
   FROM my_iotsitewise_datastore.asset_metadata as asset_metadata
   WHERE asset_metadata.assetname = 'Demo Turbine Asset 2'
        AND asset_metadata.assetpropertyname = 'Wind Speed')
```

다음 단계

[튜토리얼 리소스 정리](#)

튜토리얼 리소스 정리

튜토리얼을 완료한 후 요금이 발생하지 않도록 리소스를 정리합니다.

AWS IoT SiteWise 데모를 삭제하려면

AWS IoT SiteWise 데모는 일주일 후에 자동으로 삭제됩니다. 데모 리소스의 사용을 마쳤으면 이전 데모를 삭제할 수 있습니다. 데모를 수동으로 삭제하려면 다음 단계를 수행하세요.

1. [AWS CloudFormation 콘솔](#)로 이동합니다.
2. 스택 목록에서 IoTSiteWiseDemoAssets을 선택합니다.
3. 삭제를 선택합니다. 스택을 삭제하면 데모용으로 생성된 모든 리소스가 삭제됩니다.
4. 확인란에 삭제를 입력합니다.

스택을 삭제하는 데 약 15분이 걸립니다. 데모가 삭제되지 않으면 오른쪽 상단에서 삭제를 다시 선택합니다. 또 다시 데모가 삭제되지 않으면 AWS CloudFormation 콘솔의 단계에 따라 삭제에 실패한 리소스를 건너뛰고 다시 시도합니다.

데이터 스토어를 삭제하려면

- 관리형 데이터 스토어를 삭제하려면 다음 예와 같이 CLI 명령 delete-datastore를 실행합니다.

```
aws iotanalytics delete-datastore --datastore-name my_IotSiteWise_datastore
```

AWS IoT Analytics 데이터 세트를 삭제하려면

- 데이터 세트를 삭제하려면 다음 예와 같이 CLI 명령 `delete-dataset`를 실행합니다. 이 작업을 수행하기 전에 데이터 세트의 내용을 삭제할 필요는 없습니다.

```
aws iotanalytics delete-dataset --dataset-name my_dataset
```

Note

이 명령은 출력을 생성하지 않습니다.

파이프라인 활동

가장 단순한 기능 파이프라인은 채널을 데이터 스토어에 연결하는 것으로, 이렇게 하면 channel 활동과 datastore 활동 사이의 파이프라인이 됩니다. 파이프라인에 활동을 더 추가할 경우 더욱 강력한 메시지 처리가 가능합니다.

[RunPipelineActivity](#) 작업을 사용하여 제공한 메시지 페이로드에 파이프라인 활동을 실행한 결과를 시뮬레이션해 볼 수 있습니다. 이 명령은 파이프라인 활동을 개발하고 디버깅할 때 유용합니다. [RunPipelineActivity의 예](#)는 사용 방법을 보여줍니다.

채널 활동

파이프라인의 첫 번째 활동은 처리할 메시지의 소스를 결정하는 channel 활동이어야 합니다.

```
{
  "channel": {
    "name": "MyChannelActivity",
    "channelName": "mychannel",
    "next": "MyLambdaActivity"
  }
}
```

데이터 스토어 활동

처리된 데이터를 어디에 저장할지 지정하는 datastore 활동이 마지막 활동입니다.

```
{
  "datastore": {
    "name": "MyDatastoreActivity",
    "datastoreName": "mydatastore"
  }
}
```

AWS Lambda 활동

lambda 활동은 더욱 복잡한 메시지 처리를 수행하기 위해 사용될 수 있습니다. 예를 들어, 외부 API 작업의 출력 데이터로 메시지를 보강하거나 Amazon DynamoDB의 로직을 기반으로 메시지를 필터링할

수 있습니다. 하지만 데이터 스토어에 들어가기 전에 이 파이프라인 활동을 사용하여 메시지를 추가하거나 기존 메시지를 제거할 수는 없습니다.

lambda 활동에 사용되는 AWS Lambda 함수는 JSON 객체 어레이를 수신 및 반환해야 합니다. 예시는 [the section called “Lambda 함수 예시 1”](#)에서 확인하세요.

Lambda 함수를 호출할 수 있는 AWS IoT Analytics 권한을 부여하려면 정책을 추가해야 합니다. 예를 들어, 다음 CLI 명령을 실행하고 *exampleFunctionName*을 Lambda 함수 이름으로 바꾸고, *123456789012*를 AWS 계정 ID로 바꾸고, 지정된 Lambda 함수를 호출하는 파이프라인의 Amazon 리소스 이름(ARN)을 사용합니다.

```
aws lambda add-permission --function-name exampleFunctionName --
action lambda:InvokeFunction --statement-id iotanalytics --principal
iotanalytics.amazonaws.com --source-account 123456789012 --source-arn
arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline
```

명령은 다음을 반환합니다.

```
{
  "Statement": [{"Sid": "iotanalytica", "Effect": "Allow",
  "Principal": {"Service": "iotanalytics.amazonaws.com"}, "Action":
  "lambda:InvokeFunction", "Resource": "arn:aws:lambda:aws-region:aws-
  account:function:exampleFunctionName", "Condition": {"StringEquals":
  {"AWS:SourceAccount": "123456789012"}, "ArnLike": {"AWS:SourceArn":
  "arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline"}}}]}
}
```

자세한 내용은 AWS Lambda 개발자 안내서에서 [AWS Lambda에 대해 리소스 기반 정책 사용](#)을 참조하세요.

Lambda 함수 예시 1

이 예시에서 Lambda 함수는 원본 메시지의 데이터를 기반으로 정보를 추가합니다. 디바이스는 다음 예시와 유사한 페이로드와 함께 메시지를 게시합니다.

```
{
  "thingid": "00001234abcd",
  "temperature": 26,
  "humidity": 29,
  "location": {
    "lat": 52.4332935,
```

```

    "lon": 13.231694
  },
  "ip": "192.168.178.54",
  "datetime": "2018-02-15T07:06:01"
}

```

그리고 디바이스에는 다음과 같은 파이프라인 정의가 포함됩니다.

```

{
  "pipeline": {
    "activities": [
      {
        "channel": {
          "channelName": "foobar_channel",
          "name": "foobar_channel_activity",
          "next": "lambda_foobar_activity"
        }
      },
      {
        "lambda": {
          "lambdaName": "MyAnalyticsLambdaFunction",
          "batchSize": 5,
          "name": "lambda_foobar_activity",
          "next": "foobar_store_activity"
        }
      },
      {
        "datastore": {
          "datastoreName": "foobar_datastore",
          "name": "foobar_store_activity"
        }
      }
    ],
    "name": "foobar_pipeline",
    "arn": "arn:aws:iotanalytics:eu-west-1:123456789012:pipeline/foobar_pipeline"
  }
}

```

다음 Lambda Python 함수(MyAnalyticsLambdaFunction)는 메시지에 GMaps URL과 화씨 온도를 추가합니다.

```

import logging
import sys

```

```
# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def c_to_f(c):
    return 9.0/5.0 * c + 32

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))
    maps_url = 'N/A'

    for e in event:
        #e['foo'] = 'addedByLambda'
        if 'location' in e:
            lat = e['location']['lat']
            lon = e['location']['lon']
            maps_url = "http://maps.google.com/maps?q={},{}".format(lat,lon)

        if 'temperature' in e:
            e['temperature_f'] = c_to_f(e['temperature'])

        logger.info("maps_url: {}".format(maps_url))
        e['maps_url'] = maps_url

    logger.info("event after processing: {}".format(event))

    return event
```

Lambda 함수 예시 2

유용한 기술은 메시지 페이로드를 압축 및 직렬화하여 전송 및 저장 비용을 줄이는 것입니다. 이 두 번째 예시에서 Lambda 함수는 메시지 페이로드가 압축된 후 문자열로 base64 인코딩(직렬화)된 JSON 원본을 나타내는 것으로 가정합니다. 원본 JSON을 반환합니다.

```
import base64
import gzip
import json
import logging
```

```
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def decode_to_bytes(e):
    return base64.b64decode(e)

def decompress_to_string(binary_data):
    return gzip.decompress(binary_data).decode('utf-8')

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))

    decompressed_data = []

    for e in event:
        binary_data = decode_to_bytes(e)
        decompressed_string = decompress_to_string(binary_data)

        decompressed_data.append(json.loads(decompressed_string))

    logger.info("event after processing: {}".format(decompressed_data))

    return decompressed_data
```

AddAttributes 활동

addAttributes 활동은 메시지 내 기존 속성에 따라 속성을 추가합니다. 이렇게 하면 메시지가 저장되기 전에 메시지의 모양을 변경할 수 있습니다. 예를 들어, addAttributes를 사용하여 서로 다른 세대의 디바이스 펌웨어에서 오는 데이터를 표준화할 수 있습니다.

다음 입력 메시지를 살펴보세요.

```
{
  "device": {
    "id": "device-123",
```

```

    "coord": [ 47.6152543, -122.3354883 ]
  }
}

```

addAttributes 활동은 다음과 같습니다.

```

{
  "addAttributes": {
    "name": "MyAddAttributesActivity",
    "attributes": {
      "device.id": "id",
      "device.coord[0]": "lat",
      "device.coord[1]": "lon"
    },
    "next": "MyRemoveAttributesActivity"
  }
}

```

이 활동은 디바이스 ID를 루트 수준으로 이동하고 coord 배열의 값을 추출하여 이를 lat 및 lon이라는 최상위 속성으로 승격합니다. 이 활동의 결과, 입력 메시지가 다음 예시와 같이 변형됩니다.

```

{
  "device": {
    "id": "device-123",
    "coord": [ 47.6, -122.3 ]
  },
  "id": "device-123",
  "lat": 47.6,
  "lon": -122.3
}

```

원본 디바이스 속성은 아직 남아 있습니다. removeAttributes 활동을 사용하여 이를 제거할 수 있습니다.

RemoveAttributes 활동

removeAttributes 활동은 메시지에서 속성을 제거합니다. 예를 들어 addAttributes 활동의 결과인 메시지의 경우

```

{

```

```

"device": {
  "id": "device-123",
  "coord": [ 47.6, -122.3 ]
},
"id": "device-123",
"lat": 47.6,
"lon": -122.3
}

```

루트 수준에 필요한 데이터만을 포함하도록 메시지를 정규화하려면 다음 `removeAttributes` 활동을 사용합니다.

```

{
  "removeAttributes": {
    "name": "MyRemoveAttributesActivity",
    "attributes": [
      "device"
    ],
    "next": "MyDatastoreActivity"
  }
}

```

그러면 파이프라인을 따라 다음 메시지가 이동합니다.

```

{
  "id": "device-123",
  "lat": 47.6,
  "lon": -122.3
}

```

SelectAttributes 활동

`selectAttributes`는 원본 메시지에서 지정된 속성만 사용하여 새 메시지를 생성합니다. 다른 모든 속성은 버리며, `selectAttributes`는 메시지의 루트 아래에 새 속성을 생성합니다. 이 메시지의 경우:

```

{
  "device": {
    "id": "device-123",
    "coord": [ 47.6152543, -122.3354883 ],

```

```

    "temp": 50,
    "hum": 40
  },
  "light": 90
}

```

이 활동의 경우:

```

{
  "selectAttributes": {
    "name": "MySelectAttributesActivity",
    "attributes": [
      "device.temp",
      "device.hum",
      "light"
    ],
    "next": "MyDatastoreActivity"
  }
}

```

그 결과 다음 메시지가 파이프라인을 따라 이동합니다.

```

{
  "temp": 50,
  "hum": 40,
  "light": 90
}

```

selectAttributes 역시 루트 수준 객체만을 생성할 수 있습니다.

Filter 활동

filter 활동은 속성을 기반으로 메시지를 필터링합니다. 이 활동에 사용되는 표현식은 부울 값을 반환해야 하는 SQL WHERE 절과 유사합니다.

```

{
  "filter": {
    "name": "MyFilterActivity",
    "filter": "temp > 40 AND hum < 20",
    "next": "MyDatastoreActivity"
  }
}

```

```
}
}
```

DeviceRegistryEnrich 활동

deviceRegistryEnrich 활동을 통해 AWS IoT 디바이스 레지스트리의 데이터를 메시지 페이로드에 추가할 수 있습니다. 예를 들어 다음과 같은 메시지가 있다고 가정합니다.

```
{
  "temp": 50,
  "hum": 40,
  "device" {
    "thingName": "my-thing"
  }
}
```

다음과 같은 deviceRegistryEnrich 활동 추가:

```
{
  "deviceRegistryEnrich": {
    "name": "MyDeviceRegistryEnrichActivity",
    "attribute": "metadata",
    "thingName": "device.thingName",
    "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
    "next": "MyDatastoreActivity"
  }
}
```

이제 출력 메시지는 이 예시와 같이 됩니다.

```
{
  "temp" : 50,
  "hum" : 40,
  "device" {
    "thingName" : "my-thing"
  },
  "metadata" : {
    "defaultClientId": "my-thing",
    "thingTypeName": "my-thing",
    "thingArn": "arn:aws:iot:us-east-1:<your-account-number>:thing/my-thing",

```



```

    "version": 1,
    "thingName": "my-thing",
    "attributes": {},
    "thingId": "aaabbbccc-dddeef-gghh-jjkk-llmmnnoopp"
  }
}

```

활동 정의의 `roleArn` 필드에 적절히 권한이 부착된 역할을 지정해야 합니다. 이 역할은 다음 예시와 같은 권한 정책을 갖고 있어야 합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeThing"
      ],
      "Resource": [
        "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
      ]
    }
  ]
}

```

또 신뢰 정책은 다음과 같아야 합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}

```

DeviceShadowEnrich 활동

deviceShadowEnrich 활동은 AWS IoT 디바이스 새도우 서비스의 정보를 메시지에 추가합니다. 예를 들어 메시지가 다음과 같습니다.

```
{
  "temp": 50,
  "hum": 40,
  "device": { "thingName": "my-thing" }
}
```

다음 deviceShadowEnrich 활동:

```
{
  "deviceShadowEnrich": {
    "name": "MyDeviceShadowEnrichActivity",
    "attribute": "shadow",
    "thingName": "device.thingName",
    "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
    "next": "MyDatastoreActivity"
  }
}
```

결과는 다음 예시와 같은 메시지입니다.

```
{
  "temp": 50,
  "hum": 40,
  "device": {
    "thingName": "my-thing"
  },
  "shadow": {
    "state": {
      "desired": {
        "attributeX": valueX, ...
      },
      "reported": {
        "attributeX": valueX, ...
      },
      "delta": {
        "attributeX": valueX, ...
      }
    }
  }
}
```

```

    }
  },
  "metadata": {
    "desired": {
      "attribute1": {
        "timestamp": timestamp
      }, ...
    },
    "reported": ": {
      "attribute1": {
        "timestamp": timestamp
      }, ...
    }
  },
  "timestamp": timestamp,
  "clientToken": "token",
  "version": version
}
}

```

활동 정의의 `roleArn` 필드에 적절히 권한이 부착된 역할을 지정해야 합니다. 이 역할은 다음과 같은 권한 정책을 갖고 있어야 합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:GetThingShadow"
      ],
      "Resource": [
        "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"
      ]
    }
  ]
}

```

또 신뢰 정책은 다음과 같아야 합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [

```

```

    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}

```

수학 활동

math 활동은 메시지의 속성을 사용하여 산수 표현식을 계산합니다. 표현식은 숫자를 반환해야 합니다. 다음 메시지를 예로 들 수 있습니다.

```

{
  "tempF": 50,
}

```

math 활동 처리 이후:

```

{
  "math": {
    "name": "MyMathActivity",
    "math": "(tempF - 32) / 2",
    "attribute": "tempC",
    "next": "MyDatastoreActivity"
  }
}

```

결과 메시지는 다음과 같이 됩니다.

```

{
  "tempF" : 50,
  "tempC": 9
}

```

수학 활동 연산자 및 함수

math 활동에서는 다음 연산자를 사용할 수 있습니다.

+	더하기
-	빼기
*	곱하기
/	나누기
%	모듈로

math 활동에서는 다음 함수를 사용할 수 있습니다.

- [abs\(Decimal\)](#)
- [acos\(Decimal\)](#)
- [asin\(Decimal\)](#)
- [atan\(Decimal\)](#)
- [atan2\(Decimal, Decimal\)](#)
- [ceil\(Decimal\)](#)
- [cos\(Decimal\)](#)
- [cosh\(Decimal\)](#)
- [exp\(Decimal\)](#)
- [ln\(Decimal\)](#)
- [log\(Decimal\)](#)
- [mod\(Decimal, Decimal\)](#)
- [power\(Decimal, Decimal\)](#)
- [round\(Decimal\)](#)
- [sign\(Decimal\)](#)
- [sin\(Decimal\)](#)

- [sinh\(Decimal\)](#)
- [sqrt\(Decimal\)](#)
- [tan\(Decimal\)](#)
- [tanh\(Decimal\)](#)
- [trunc\(Decimal, Integer\)](#)

abs(Decimal)

숫자의 절대값을 반환합니다.

예: $\text{abs}(-5) = 5$.

인수 형식	결과
Int	Int, 인수의 절대값
Decimal	Decimal, 인수의 절대값
Boolean	Undefined .
String	Decimal. 결과는 인수의 절대값입니다. 문자열을 변환할 수 없는 경우 결과는 Undefined 입니다.
배열	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

acos(Decimal)

숫자의 역코사인을 라디안 단위로 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\text{acos}(0) = 1.5707963267948966$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 역코사인. 가상 결과는 Undefined 로 반환됩니다.
Decimal	Decimal(배정밀도), 인수의 역코사인. 가상 결과는 Undefined 로 반환됩니다.
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 역코사인. 문자열을 변환할 수 없는 경우 결과는 Undefined 입니다. 가상 결과는 Undefined 로 반환됩니다.
배열	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

asin(Decimal)

숫자의 역사인을 라디안 단위로 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\text{asin}(0) = 0.0$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 역사인. 가상 결과는 Undefined 로 반환됩니다.
Decimal	Decimal(배정밀도), 인수의 역사인. 가상 결과는 Undefined 로 반환됩니다.
Boolean	Undefined .

인수 형식	결과
String	Decimal(배정밀도), 인수의 역사인. 문자열을 변환할 수 없는 경우 결과는 Undefined 입니다. 가상 결과는 Undefined 로 반환됩니다.
배열	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

atan(Decimal)

숫자의 역탄젠트를 라디안 단위로 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\text{atan}(0) = 0.0$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 역탄젠트. 가상 결과는 Undefined 로 반환됩니다.
Decimal	Decimal(배정밀도), 인수의 역탄젠트. 가상 결과는 Undefined 로 반환됩니다.
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 역탄젠트. 문자열을 변환할 수 없는 경우 결과는 Undefined 입니다. 가상 결과는 Undefined 로 반환됩니다.
배열	Undefined .
객체	Undefined .

인수 형식	결과
Null	Undefined .
정의되지 않음	Undefined .

atan2(Decimal, Decimal)

양의 x축과 두 인수로 정의된 점(x, y) 사이의 각도(라디안)를 반환합니다. 이 각도는 시계 반대 방향 각도(상반면, $y > 0$)의 경우 양수이고, 시계 방향 각도 Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\text{atan}(1, 0) = 1.5707963267948966$

인수 형식	인수 형식	결과
Int / Decimal	Int / Decimal	Decimal(배정밀도), x축과 지정된 점(x,y) 사이의 각도
Int / Decimal / String	Int / Decimal / String	Decimal, 설명된 점의 역탄젠트. 문자열을 변환할 수 없는 경우 결과는 Undefined 입니다.
기타 값	기타 값	Undefined .

ceil(Decimal)

지정된 Decimal을 가장 가까운 Int로 올림합니다.

예제:

$\text{ceil}(1.2) = 2$

$\text{ceil}(11.2) = 12$

인수 형식	결과
Int	Int, 인수 값
Decimal	Int, 문자열은 Decimal로 변환된 후 가장 가까운 Int로 올림됩니다. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
기타 값	Undefined .

cos(Decimal)

숫자의 코사인을 라디안 단위로 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\cos(0) = 1$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 코사인. 가상 결과는 Undefined 로 반환됩니다.
Decimal	Decimal(배정밀도), 인수의 코사인. 가상 결과는 Undefined 로 반환됩니다.
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 코사인. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다. 가상 결과는 Undefined 로 반환됩니다.
배열	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

cosh(Decimal)

숫자의 쌍곡코사인을 라디안 단위로 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\cosh(2.3) = 5.037220649268761$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 쌍곡코사인. 가상 결과는 Undefined 로 반환됩니다.
Decimal	Decimal(배정밀도), 인수의 쌍곡코사인. 가상 결과는 Undefined 로 반환됩니다.
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 쌍곡코사인. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다. 가상 결과는 Undefined 로 반환됩니다.
배열	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

exp(Decimal)

인수로 거듭제곱된 e를 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\exp(1) = 1$

인수 형식	결과
Int	Decimal(배정밀도), $e^{\text{인수}}$.

인수 형식	결과
Decimal	Decimal(배정밀도), e^인수
String	Decimal(배정밀도), e^인수. String을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
기타 값	Undefined .

ln(Decimal)

인수의 자연 로그를 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\ln(e) = 1$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 자연 로그.
Decimal	Decimal(배정밀도), 인수의 자연 로그
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 자연 로그. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
배열	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

log(Decimal)

인수의 기수 10 로그를 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\log(100) = 2.0$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 기수 10 로그.
Decimal	Decimal(배정밀도), 인수의 기수 10 로그.
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 기수 10 로그. String을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
배열	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

mod(Decimal, Decimal)

첫 번째 인수를 두 번째 인수로 나눈 나머지를 반환합니다. 또한 %를 동일한 모듈로 기능의 중위 연산자로 사용할 수도 있습니다.

예: $\text{mod}(8, 3) = 3$

왼쪽 피연산자	오른쪽 피연산자	출력
Int	Int	Int, 두 번째 인수를 법으로 하는 첫 번째 인수.
Int / Decimal	Int / Decimal	Decimal, 두 번째 인수를 법으로 하는 첫 번째 인수.
String / Int / Decimal	String / Int / Decimal	모든 문자열이 Decimals로 변환될 경우 결과는 두 번째 인

왼쪽 피연산자	오른쪽 피연산자	출력
		수를 법으로 하는 첫 번째 인수입니다. 그렇지 않을 경우 Undefined 입니다.
기타 값	기타 값	Undefined .

power(Decimal, Decimal)

두 번째 인수로 거듭제곱된 첫 번째 인수를 반환합니다. Decimal인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: `power(2, 5) = 32.0`

인수 형식 1	인수 형식 2	출력
Int / Decimal	Int / Decimal	Decimal(배정밀도), 두 번째 인수의 승수로 거듭제곱된 첫 번째 인수.
Int / Decimal / String	Int / Decimal / String	Decimal(배정밀도), 두 번째 인수의 승수로 거듭제곱된 첫 번째 인수. 모든 문자열은 Decimals로 변환됩니다. String을(를) Decimal(으)로 변환하지 못한 경우 결과는 Undefined 입니다.
기타 값	기타 값	Undefined .

round(Decimal)

지정된 Decimal을 가장 가까운 Int로 반올림합니다. Decimal이 두 Int 값과 등거리일 경우(예: 0.5) Decimal은 올림됩니다.

예제:

`Round(1.2) = 1`

`Round(1.5) = 2`

`Round(1.7) = 2`

`Round(-1.1) = -1`

`Round(-1.5) = -2`

인수 형식	결과
Int	인수
Decimal	Decimal은 가장 가까운 Int로 내림됩니다.
String	Decimal은 가장 가까운 Int로 내림됩니다. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
기타 값	Undefined .

sign(Decimal)

지정된 숫자의 부호를 반환합니다. 인수의 부호가 플러스일 경우 1이 반환됩니다. 인수의 부호가 마이너스일 경우 -1이 반환됩니다. 인수가 0일 경우 0이 반환됩니다.

예제:

`sign(-7) = -1`

`sign(0) = 0`

`sign(13) = 1`

인수 형식	결과
Int	Int, Int 값의 부호.
Decimal	Int, Decimal 값의 부호.

인수 형식	결과
String	Int, Decimal 값의 부호. 문자열은 Decimal 값으로 변환되고 Decimal 값의 부호가 반환됩니다. String을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
기타 값	Undefined .

sin(Decimal)

숫자의 사인을 라디안 단위로 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\sin(0) = 0.0$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 사인.
Decimal	Decimal(배정밀도), 인수의 사인.
Boolean	Undefined .
String	Decimal, 인수의 사인. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

sinh(Decimal)

숫자의 쌍곡사인을 반환합니다. Decimal 값은 함수 적용 전에 배정밀도로 반올림됩니다. 결과는 배정밀도의 Decimal 값입니다.

예: $\sinh(2.3) = 4.936961805545957$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 쌍곡사인.
Decimal	Decimal(배정밀도), 인수의 쌍곡사인.
Boolean	Undefined .
String	Decimal, 인수의 쌍곡선 사인. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

sqrt(Decimal)

숫자의 제곱근을 반환합니다. Decimal 인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\sqrt{9} = 3.0$

인수 형식	결과
Int	인수의 제곱근.
Decimal	인수의 제곱근.
Boolean	Undefined .
String	인수의 제곱근. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
Array	Undefined .

인수 형식	결과
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

tan(Decimal)

숫자의 탄젠트를 라디안 단위로 반환합니다. Decimal 값은 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\tan(3) = -0.1425465430742778$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 탄젠트.
Decimal	Decimal(배정밀도), 인수의 탄젠트.
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 탄젠트. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

tanh(Decimal)

숫자의 쌍곡탄젠트를 라디안 단위로 반환합니다. Decimal 값은 함수 적용 전에 배정밀도로 반올림됩니다.

예: $\tanh(2.3) = 0.9800963962661914$

인수 형식	결과
Int	Decimal(배정밀도), 인수의 쌍곡탄젠트.
Decimal	Decimal(배정밀도), 인수의 쌍곡탄젠트.
Boolean	Undefined .
String	Decimal(배정밀도), 인수의 쌍곡탄젠트. 문자열을 Decimal로 변환할 수 없는 경우 결과는 Undefined 입니다.
Array	Undefined .
Object	Undefined .
Null	Undefined .
Undefined	Undefined .

trunc(Decimal, Integer)

첫 번째 인수를 두 번째 인수로 지정된 Decimal 자리수로 절사합니다. 두 번째 인수가 0보다 작을 경우 0으로 설정됩니다. 두 번째 인수가 34보다 클 경우 34로 설정됩니다. 끝의 0은 결과에서 제거됩니다.

예제:

$\text{trunc}(2.3, 0) = 2$

$\text{trunc}(2.3123, 2) = 2.31$

$\text{trunc}(2.888, 2) = 2.88$

$\text{trunc}(2.00, 5) = 2$

인수 형식 1	인수 형식 2	결과
Int	Int	소스 값
Int / Decimal / String	Int / Decimal	첫 번째 인수가 두 번째 인수로 지정된 길이로 절사됩니다. 두 번째 인수는 Int가 아닐 경우 가까운 Int로 내림됩니다. 문자열은 Decimal 값으로 변환됩니다. 문자열 변환이 실패할 경우 결과는 Undefined 입니다.
기타 값		Undefined

RunPipelineActivity

다음은 RunPipelineActivity 명령을 사용하여 파이프라인 활동을 테스트하는 방법을 보여주는 예입니다. 이 예의 경우, 수학 연산 활동을 테스트합니다.

1. 테스트하려는 파이프라인 활동에 대한 정의가 포함된 maths.json 파일을 생성합니다.

```
{
  "math": {
    "name": "MyMathActivity",
    "math": "((temp - 32) * 5.0) / 9.0",
    "attribute": "tempC"
  }
}
```

2. 파이프라인 활동 테스트에 사용하는 예가 되는 페이로드가 포함된 payloads.json 파일을 생성합니다.

```
[
  "{\"humidity\": 52, \"temp\": 68 }",
  "{\"humidity\": 52, \"temp\": 32 }"
]
```

3. 명령줄에서 RunPipelineActivities 작업을 호출합니다.

```
aws iotanalytics run-pipeline-activity --pipeline-activity file://maths.json --  
payloads file://payloads.json --cli-binary-format raw-in-base64-out
```

결과는 다음과 같습니다.

```
{  
  "logResult": "",  
  "payloads": [  
    "eyJodW1pZG10eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",  
    "eyJodW1pZG10eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
```

결과에 열거된 페이로드는 Base64 인코딩 문자열입니다. 이 문자열을 디코딩하면 다음 결과가 표시 됩니다.

```
{"humidity":52,"temp":68,"tempC":20}  
{"humidity":52,"temp":32,"tempC":0}
```

채널 메시지 재처리

AWS IoT Analytics는 채널 데이터를 재처리할 수 있습니다. 이는 다음과 같은 경우에 유용할 수 있습니다.

- 처음부터 새로 시작하는 대신, 수집한 기존 데이터를 다시 재생하려는 경우
- 파이프라인을 업데이트하고 기존 데이터에 변경 사항을 적용하여 최신 상태로 만들려는 경우
- 고객 관리형 스토리지 옵션, 채널 권한 또는 데이터 스토어를 변경하기 전에 수집된 데이터를 포함하고자 합니다.

파라미터

AWS IoT Analytics를 통해 파이프라인으로 채널 메시지를 재처리할 때 다음 정보를 지정해야 합니다.

StartPipelineReprocessing

파이프라인을 통한 채널 메시지의 재처리를 시작합니다.

ChannelMessages

재처리하려는 하나 이상의 채널 메시지 세트를 지정합니다.

channelMessages 객체를 사용하는 경우 startTime 및 endTime 값을 지정해서는 안 됩니다.

s3Paths

채널 메시지를 저장하는 Amazon Simple Storage Service(S3) 객체를 식별하는 키를 하나 이상 지정합니다. 키의 전체 경로를 사용해야 합니다.

예시 경로:

00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0.json

유형: 문자열 배열

배열 멤버 제약 조건: 1~100개 항목.

길이 제한: 1~1,024자

endTime

재처리된 채널 데이터의 종료 시간(제외)입니다.

endTime 파라미터 값을 지정하는 경우 channelMessages 객체를 사용해서는 안 됩니다.

유형: Timestamp

startTime

재처리된 원시 메시지 데이터의 시작 시간(포함)입니다.

startTime 파라미터 값을 지정하는 경우 channelMessages 객체를 사용해서는 안 됩니다.

유형: Timestamp

pipelineName

재처리를 시작할 파이프라인의 이름입니다.

유형: 문자열

길이 제한: 1~128자.

채널 메시지 재처리(콘솔)

이 튜토리얼은 AWS IoT Analytics 콘솔의 지정된 Amazon S3 객체에 저장된 채널 데이터를 재처리하는 방법을 보여줍니다.

시작하기 전에 재처리하려는 채널 메시지가 고객 관리형 Amazon S3 버킷에 저장되어 있는지 확인합니다.

1. [AWS IoT Analytics 콘솔](#)에 로그인합니다.
2. 탐색 창에서 파이프라인을 클릭합니다.
3. 대상 파이프라인을 선택합니다.
4. 작업에서 메시지 재처리를 선택합니다.
5. 파이프라인 재처리 페이지에서 메시지 재처리에 대해 S3 객체를 선택합니다.

AWS IoT Analytics 콘솔은 다음 옵션도 제공합니다.

- 사용 가능한 모든 범위 - 채널의 모든 유효한 데이터를 재처리합니다.
- 지난 120일 - 지난 120일 동안 도착한 데이터를 재처리합니다.
- 지난 90일 - 지난 90일 동안 도착한 데이터를 재처리합니다.
- 지난 30일 - 지난 30일 동안 도착한 데이터를 재처리합니다.

- 사용자 지정 범위 - 지정된 시간 범위에 도착한 데이터를 재처리합니다. 원하는 시간 범위를 선택할 수 있습니다.
6. 채널 메시지를 저장하는 Amazon S3 객체의 키를 입력합니다.

키를 찾으려면 다음을 수행합니다.
 - a. [Amazon S3 콘솔](#)을 사용합니다.
 - b. 대상 Amazon S3 객체를 선택합니다.
 - c. 속성의 객체 개요 섹션에서 키를 복사합니다.
 7. 재처리 시작을 선택합니다.

채널 메시지 재처리(API)

StartPipelineReprocessing API를 사용할 때 다음 사항에 유의하세요.

- startTime 및 endTime 파라미터는 원시 데이터의 수집 시점을 지정하지만, 이것은 대강의 추정치에 불과합니다. 가장 가까운 시간 단위로 반올림할 수 있습니다. startTime은 포함되지만 endTime은 제외됩니다.
- 이 명령은 비동기식으로 재처리를 시작하고 즉시 결과를 반환합니다.
- 재처리된 메시지가 원래의 수신 순서에 따라 처리된 것이라는 보장은 없습니다. 거의 동일한 순서지만 정확히 같지는 않습니다.
- 24시간마다 최대 1,000개의 StartPipelineReprocessing API 요청을 생성하여 파이프라인을 통해 동일한 채널 메시지를 재처리할 수 있습니다.
- 원시 데이터를 재처리하는 경우 추가 비용이 발생합니다.

자세한 내용은 AWS IoT Analytics API 참조의 [StartPipelineReprocessing](#) API를 참조하세요.

채널 재처리 활동 취소

파이프라인 재처리 활동을 취소하려면 [CancelPipelineReprocessing](#) API를 사용하거나 AWS IoT Analytics 콘솔의 활동 페이지에서 재처리 취소를 선택합니다. 재처리를 취소하면 나머지 데이터는 재처리되지 않습니다. 재처리 요청을 다시 시작해야 합니다.

[DescribePipeline](#) API를 사용하여 재처리 상태를 확인합니다. 응답의 reprocessingSummaries 필드를 참조하세요.

워크플로우 자동화

AWS IoT Analytics는 AWS IoT에 대한 고급 데이터 분석을 제공합니다. 데이터 분석 및 기계 학습 도구를 사용하여 IoT 데이터를 자동으로 수집, 처리, 저장 및 분석할 수 있습니다. 사용자 지정 분석 코드 또는 Jupyter Notebook을 호스팅하는 컨테이너를 실행하거나 타사 사용자 지정 코드 컨테이너를 사용할 수 있으므로 기존 분석 도구를 다시 생성할 필요가 없습니다. 다음 기능을 사용하여 데이터 스토어에서 입력 데이터를 가져와 자동화된 워크플로우에 제공할 수 있습니다.

반복되는 일정으로 데이터 세트 콘텐츠를 생성합니다.

CreateDataset를 호출할 때 트리거를 지정하여 데이터 세트 콘텐츠 자동 생성을 예약합니다(`triggers:schedule:expression`). 데이터 스토어에 있는 데이터는 데이터 세트 콘텐츠를 생성하는 데 사용됩니다. SQL 쿼리를 사용하여 원하는 필드를 선택합니다(`actions:queryAction:sqlQuery`).

중복되지 않는 연속된 시간 간격을 정의해 새 데이터 세트 콘텐츠에 마지막 시간 이후 도착한 데이터만 포함되도록 만들 수 있습니다. `actions:queryAction:filters:deltaTime` 및 `:offsetSeconds` 필드를 사용해 델타 시간 간격을 지정합니다. 그런 다음 시간 간격이 경과할 때 데이터 세트 콘텐츠를 생성하도록 트리거를 지정합니다. [the section called “예 6 -- 델타 기간으로 SQL 데이터 세트 생성\(CLI\)”](#) 섹션을 참조하세요.

또 다른 데이터 세트가 완료되었을 때 데이터 세트 콘텐츠를 생성합니다.

다른 데이터 세트의 콘텐츠 생성이 완료되면 새 데이터 세트 콘텐츠를 생성을 트리거합니다
`triggers:dataset:name`.

자동으로 분석 애플리케이션을 실행합니다.

자신의 사용자 지정 데이터 분석 애플리케이션을 컨테이너화하고, 다른 데이터 세트 콘텐츠가 생성되었을 때 이를 실행하도록 트리거합니다. 이렇게 하면 반복되는 일정에 따라 생성된 데이터 세트의 콘텐츠 데이터를 애플리케이션에 공급할 수 있습니다. 애플리케이션 내에서 분석 결과에 대한 작업을 자동으로 수행할 수 있습니다. (`actions:containerAction`)

또 다른 데이터 세트가 완료되었을 때 데이터 세트 콘텐츠를 생성합니다.

다른 데이터 세트의 콘텐츠 생성이 완료되면 새 데이터 세트 콘텐츠를 생성을 트리거합니다
`triggers:dataset:name`.

자동으로 분석 애플리케이션을 실행합니다.

자신의 사용자 지정 데이터 분석 애플리케이션을 컨테이너화하고, 다른 데이터 세트 콘텐츠가 생성되었을 때 이를 실행하도록 트리거합니다. 이렇게 하면 반복되는 일정에 따라 생성된 데이터 세트

의 콘텐츠 데이터를 애플리케이션에 공급할 수 있습니다. 애플리케이션 내에서 분석 결과에 대한 작업을 자동으로 수행할 수 있습니다. (actions:containerAction)

사용 사례

제품 품질 측정을 자동화해 운영 비용(OpEx)을 절약

압력, 습도 및 온도를 측정하는 스마트 밸브가 있는 시스템이 있습니다. 이 시스템은 주기적으로, 또한 값 열기 및 닫기와 같은 특정 이벤트가 발생할 때 이벤트를 수집합니다. AWS IoT Analytics에서는 이 정기적인 기간 내에서 중복되지 않은 데이터를 수집하고 최종 제품 품질에 대한 KPI 보고서를 생성하는 분석을 자동화할 수 있습니다. 각 배치를 처리한 후 전체 제품 품질을 측정하고, 실행 볼륨을 극대화해 운영 비용을 낮춥니다.

디바이스 플릿에 대한 분석 자동화

100개의 디바이스에서 생성된 데이터에 대해 15분마다 분석(알고리즘, 데이터 과학 또는 KPI용 ML)을 실행합니다. 각 분석 주기마다 다음 분석 실행을 위한 상태가 생성되고 저장됩니다. 각 분석에서 지정된 기간 내에 받은 데이터만 사용하는 것이 좋습니다. AWS IoT Analytics는 분석을 조율하고, 각 실행에 대한 KPI와 보고서를 생성한 후 향후 분석을 위해 데이터를 저장할 수 있습니다.

자동 이상 탐지

AWS IoT Analytics는 데이터 스토어에 도착한 새로운 데이터에 대해 15분마다 수동으로 실행해야 하는 이상 탐지 워크플로우를 자동화할 수 있습니다. 또한 지정한 기간 내의 디바이스 사용량과 최고 사용자를 표시하도록 대시보드를 자동화할 수 있습니다.

산업 제어 결과 예측

산업 생산 라인이 있습니다. 사용할 수 있는 프로세스에 대한 측정치를 포함하여 AWS IoT Analytics에 전송한 데이터를 사용하면 프로세스 결과를 예측하는 분석 워크플로우를 조작할 수 있습니다. 모델 데이터는 $M \times N$ 매트릭스로 정렬할 수 있으며, 각 행에는 실험실 샘플을 채취하는 다양한 시점의 데이터가 포함됩니다. AWS IoT Analytics는 델타 창을 생성하고 데이터 과학 도구를 사용하여 KPI를 생성하고 측정 장치의 상태를 저장함으로써 분석 워크플로우를 운영할 수 있도록 도와줍니다.

Docker 컨테이너 사용

이 단원에는 고유한 Docker 컨테이너를 빌드하는 방법에 대한 정보가 포함되어 있습니다. 타사에서 빌드한 Docker 컨테이너를 다시 사용하면 보안상 위험할 수 있습니다. 이러한 컨테이너는 사용자 권한으

로 임의의 코드를 실행할 수 있습니다. 타사 컨테이너를 사용하기 전에 타사 컨테이너의 작성자를 신뢰할 수 있는지 확인하십시오.

다음은 최종 분석을 수행한 이후 도착한 데이터에 대해 정기적인 데이터 분석을 실시하도록 설정하는데 필요한 단계들입니다.

1. 데이터 애플리케이션과 모든 필수 라이브러리, 또는 다른 종속성이 포함된 Docker 컨테이너를 생성합니다.

IoTAnalytics Jupyter 확장은 컨테이너화 프로세스를 도와주는 컨테이너화 API를 제공합니다. 직접 만든 이미지를 실행하여 애플리케이션 도구 세트를 만들거나 조합하여 원하는 데이터 분석 또는 계산을 수행할 수도 있습니다. AWS IoT Analytics를 사용하면 변수를 사용하여 컨테이너화된 애플리케이션의 입력 데이터 소스와 Docker 컨테이너의 출력 데이터 대상을 정의할 수 있습니다. ([사용자 지정 Docker 컨테이너 입력/출력 변수](#)에는 사용자 지정 컨테이너의 변수 사용에 대한 추가 정보가 포함되어 있습니다.)

2. 컨테이너를 [Amazon ECR](#) 레지스트리로 업로드합니다.
3. 데이터 스토어를 생성하여 디바이스에서 메시지(데이터)를 받아 저장합니다(`iotanalytics: CreateDatastore`).
4. 메시지가 전송되는 채널을 생성합니다(`iotanalytics: CreateChannel`).
5. 채널을 데이터 스토어에 연결시키는 파이프라인을 생성합니다(`iotanalytics: CreatePipeline`).
6. 메시지 데이터를 AWS IoT Analytics 채널(`iam: CreateRole`.)로 전송하는 권한을 부여하는 IAM 역할을 생성합니다.
7. SQL 쿼리를 사용하여 채널을 메시지 데이터의 소스에 연결하는 IoT 규칙을 생성합니다(`iot: CreateTopicRule` 필드 `topicRulePayload:actions:iotAnalytics`). 디바이스가 올바른 주제의 비자 MQTT로 메시지를 전송할 때 메시지가 채널로 라우팅됩니다. 또는 `iotanalytics: BatchPutMessage`를 사용해 AWS SDK나 AWS CLI를 사용할 수 있는 디바이스에서 직접 채널로 메시지를 전송할 수 있습니다.
8. 시간 일정(`iotanalytics: CreateDataset`, 필드 `actions: queryAction:sqlQuery`)에 따라 생성이 트리거되는 SQL 데이터 세트를 만듭니다.

또한 작업이 마지막으로 실행된 후 도착하는 메시지로만 제한을 하도록 메시지 데이터에 적용할 사전 필터를 지정할 수도 있습니다. (필드 `actions:queryAction:filters:deltaTime:timeExpression`는 메시지 시간이 결정되는 표현식을 제공하고, 필드

`actions:queryAction:filters:deltaTime:offsetSeconds`는 메시지 도착의 지연 시간을 지정합니다.)

사전 필터는 트리거 일정과 함께 델타 기간을 결정합니다. SQL 데이터 세트가 마지막으로 생성된 시간 이후 받은 메시지를 사용해 새로운 SQL 데이터 세트 각각을 생성합니다. (SQL 데이터 세트가 처음으로 생성된 시간은 어떤가요? 일정과 사전 필터를 토대로 데이터 세트가 생성된 마지막 시간을 추정합니다.)

9. 첫 번째 생성에 의해 트리거될 또 다른 데이터 세트를 생성합니다([CreateDataset](#) 필드 `trigger:dataset`). 이 데이터 세트에 첫 번째 단계에서 생성한 Docker 컨테이너 실행에 필요한 정보를 가리키고 제공하는 컨테이너 작업을 지정합니다(필드 `actions:containerAction`). 또 다음을 지정합니다.

- 계정에 저장된 docker 컨테이너의 ARN(image).
- 컨테이너 작업을 실행시키기 위해 필요한 리소스에 액세스할 수 있는 권한을 시스템에 부여하는 역할의 ARN입니다(`executionRoleArn`).
- 컨테이너 작업을 실행하는 리소스의 구성입니다(`resourceConfiguration`).
- 컨테이너 작업을 실행하는 데 사용된 컴퓨팅 리소스의 유형(`computeType`과 가능한 값: `ACU_1 [vCPU=4, memory=16GiB]` or `ACU_2 [vCPU=8, memory=32GiB]`).
- 컨테이너 작업 실행에 사용되는 리소스 인스턴스에 사용할 수 있는 영구 스토리지의 크기(GB) (`volumeSizeInGB`).
- 애플리케이션 실행과 관련된 맥락에서 사용하는 변수 값(기본적으로 애플리케이션으로 전달된 파라미터)입니다(`variables`).

컨테이너가 실행될 때 교체되는 변수입니다. 이렇게 하면, 데이터 세트 콘텐츠를 생성할 때 공급되는 다양한 변수(파라미터)로 동일한 컨테이너를 실행할 수 있습니다. `IoTAnalytics Jupyter` 확장은 자동으로 노트북의 변수를 인식하고, 이를 컨테이너화 프로세스의 일부로 사용할 수 있도록 만들어 이런 프로세스를 단순화시킵니다. 이 인식된 값을 선택하거나, 직접 사용자 지정한 변수를 추가할 수 있습니다. 컨테이너를 실행시키기 전, 시스템이 이런 변수를 실행 시점의 값으로 교체합니다.

- 이런 변수 중 하나는 최신 콘텐츠가 애플리케이션에 대한 입력으로 사용되는 데이터 세트의 이름(이전 단계에서 생성한 데이터 세트의 이름)입니다 (`datasetContentVersionValue:datasetName`).

AWS IoT Analytics는 데이터 세트를 생성하는 델타 기간과 SQL 쿼리, 애플리케이션 컨테이너를 이용해 델타 기간에 데이터에 대해 지정한 간격으로 실행되는 예약 프로덕션 데이터 세트를 생성하며, 원하는 출력을 생산하고 알림을 전송합니다.

선택에 따라, 프로덕션 데이터 세트 애플리케이션을 일시 중지 및 다시 시작할 수 있습니다. 프로덕션 데이터 세트 애플리케이션을 다시 시작할 경우, AWS IoT Analytics는 기본적으로 마지막 실행 이후 도착했지만 아직 분석되지 않은 모든 데이터를 캐치업합니다. 일련의 연속 실행을 수행하여 프로덕션 데이터 세트 작업(기간)을 재개할 방법을 구성할 수도 있습니다. 또는 새로 도착한 데이터 중 지정한 델타 기간의 크기에 부합하는 데이터만 캡처하는 방식으로 프로덕션 데이터 세트 애플리케이션을 다시 시작할 수도 있습니다.

다음은 다른 데이터 세트 생성이 트리거하는 데이터 세트를 생성하거나 정의할 때 적용되는 제한 사항입니다.

- SQL 데이터 세트는 컨테이너 데이터 세트만 트리거 할 수 있습니다.
- SQL 데이터 세트는 최대 10개의 컨테이너 데이터 세트만 트리거할 수 있습니다.

SQL 데이터 세트가 트리거하는 컨테이너 데이터 세트를 생성할 때 다음 오류가 발생할 수도 있습니다.

- "트리거링 데이터 세트는 컨테이너 데이터 세트에만 추가할 수 있습니다."
- "트리거링 데이터 세트는 하나만 허용됩니다."

2개의 서로 다른 SQL 데이터 세트가 트리거하도록 컨테이너 데이터 세트를 정의하려 시도할 때 발생하는 오류입니다.

- "트리거링 데이터 세트 <dataset-name>을 컨테이너 데이터 세트가 트리거할 수 없습니다."

다른 컨테이너 데이터 세트가 트리거하도록 또 다른 컨테이너 데이터 세트를 정의하려 시도할 때 발생하는 오류입니다.

- "<N> 데이터 세트는 이미 <dataset-name> 데이터 세트에 종속되어 있습니다."

이미 10개의 컨테이너 데이터 세트를 트리거한 SQL 데이터 세트가 트리거하도록 다른 컨테이너 데이터 세트를 정의하려 시도할 때 발생하는 오류입니다.

- "정확히 1개의 트리거 유형만 제공되어야 합니다."

일정 트리거와 데이터 세트 트리거 모두를 이용해 트리거하도록 데이터 세트를 정의하려 시도할 때 발생하는 오류입니다.

사용자 지정 Docker 컨테이너 입력/출력 변수

이번 섹션에서는 사용자 지정 도커 이미지가 실행하는 프로그램이 입력 변수를 읽고, 출력을 업로드하는 방법에 대해 설명합니다.

Params 파일

입력 변수와 출력을 업로드하려는 대상은 docker 이미지를 실행시키는 인스턴스의 /opt/ml/input/data/iotanalytics/params에 위치한 JSON 파일에 저장됩니다. 다음은 이런 파일의 콘텐츠에 대한 예입니다.

```
{
  "Context": {
    "OutputUri": {
      "html": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/iotanalytics-xxxxxxx/output.html",
      "ipynb": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/iotanalytics-xxxxxxx/output.ipynb"
    }
  },
  "Variables": {
    "source_dataset_name": "mydataset",
    "source_dataset_version_id": "xxxx",
    "example_var": "hello world!",
    "custom_output": "s3://aws-iot-analytics/dataset-xxxxxxx/notebook/results/iotanalytics-xxxxxxx/output.txt"
  }
}
```

데이터 세트의 이름과 버전 ID에 추가, Variables 섹션에는 iotanalytics:CreateDataset 호출에서 지정한 변수들이 포함되어 있습니다(이 예의 경우, example_var의 값은 hello world!). 또한 사용자 지정 출력 URI가 custom_output 변수에 제공되어 있습니다. OutputUri 필드에는 컨테이너가 출력을 업로드할 수 있는 기본값 위치가 포함됩니다(이 예의 경우, 기본값 출력 URI가 ipynb 및 html 출력 모두에 제공되어 있음).

입력 변수

도커 이미지가 시작하는 프로그램은 params 파일의 변수를 읽을 수 있습니다. 다음은 params 파일을 열고, 파싱하고, example_var 변수 값을 인쇄하는 예제 프로그램입니다.

```
import json

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
    example_var = params["Variables"]["example_var"]
    print(example_var)
```

출력 업로드

도커 이미지가 시작하는 프로그램은 Amazon S3 위치에 출력을 저장합니다. 출력은 "bucket-owner-full-control" [액세스 제어 목록](#)으로 로드해야 합니다. 액세스 목록은 AWS IoT Analytics 서비스에 업로드된 출력을 제어하는 권한을 부여합니다. 이 예에서는 params 파일의 custom_output로 정의한 Amazon S3 위치에 example_var의 내용을 업로드할 수 있도록 이전 예를 확대합니다.

```
import boto3
import json
from urllib.parse import urlparse

ACCESS_CONTROL_LIST = "bucket-owner-full-control"

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
    example_var = params["Variables"]["example_var"]

outputUri = params["Variables"]["custom_output"]
# break the S3 path into a bucket and key
bucket = urlparse(outputUri).netloc
key = urlparse(outputUri).path.lstrip("/")

s3_client = boto3.client("s3")
s3_client.put_object(Bucket=bucket, Key=key, Body=example_var, ACL=ACCESS_CONTROL_LIST)
```

권한

두 가지 역할을 생성해야 합니다. 첫 번째 역할은 SageMaker 인스턴스를 시작해 노트북을 컨테이너화하는 권한을 부여합니다. 컨테이너 실행에 또 다른 역할이 필요합니다.

첫 번째 역할을 자동 또는 수동으로 생성할 수 있습니다. AWS IoT Analytics 콘솔로 새 SageMaker 인스턴스를 생성하는 경우, SageMaker 인스턴스를 실행하고 노트북을 컨테이너화하는 데 필요한 모든 권한을 부여하는 새로운 역할을 자동 생성하는 옵션이 제공됩니다. 또는 수동으로 이런 권한을 가진 역할을 생성할 수도 있습니다. 이렇게 하려면 AmazonSageMakerFullAccess 정책이 연결된 역할을 생성한 후 다음 정책을 추가합니다.

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

```

    "Action": [
      "ecr:BatchDeleteImage",
      "ecr:BatchGetImage",
      "ecr:CompleteLayerUpload",
      "ecr:CreateRepository",
      "ecr:DescribeRepositories",
      "ecr:GetAuthorizationToken",
      "ecr:InitiateLayerUpload",
      "ecr:PutImage",
      "ecr:UploadLayerPart"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject"
    ],
    "Resource": "arn:aws:s3:::iotanalytics-notebook-containers/*"
  }
]
}

```

수동으로 컨테이너를 실행시킬 권한을 부여하는 두 번째 역할을 생성해야 합니다. AWS IoT Analytics 콘솔로 첫 번째 역할을 자동 생성한 경우에도 이렇게 해야 합니다. 다음 정책과 신뢰 정책이 연결된 역할을 생성합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:PutObject",
        "s3:GetObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::aws-*-dataset-*/*"
    },
    {
      "Effect": "Allow",
      "Action": [

```



```

        "iotanalytics:*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ecr:GetAuthorizationToken",
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage",
      "ecr:BatchCheckLayerAvailability",
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:DescribeLogStreams",
      "logs:GetLogEvents",
      "logs:PutLogEvents"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListBucket",
      "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
  }
]
}

```

다음은 신뢰 정책 예시입니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": ["sagemaker.amazonaws.com", "iotanalytics.amazonaws.com"]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

```

    }
  ]
}

```

Java 및 AWS CLI를 통한 CreateDataset API 사용

데이터 세트를 생성합니다. 데이터 세트는 queryAction(SQL 쿼리) 또는 containerAction(컨테이너식 애플리케이션 실행)을 적용하여 데이터 스토어에서 검색된 데이터를 저장합니다. 이 작업은 데이터 세트 스켈레톤을 생성합니다. 데이터 세트는 CreateDatasetContent를 호출해서 수동으로, 또는 지정한 trigger에 따라 자동으로 채울 수 있습니다. 자세한 내용은 [CreateDataset](#) 및 [CreateDatasetContent](#)를 참조하세요.

주제

- [예 1 -- SQL 데이터 세트\(java\) 생성:](#)
- [예 2 -- delta 기간으로 SQL 데이터 세트\(java\) 생성:](#)
- [예 3 -- 고유 일정 트리거로 컨테이너 데이터 세트\(java\) 생성:](#)
- [예 4 -- 트리거인 SQL 데이터 세트로 컨테이너 데이터 세트\(java\) 생성:](#)
- [예 5 -- SQL 데이터 세트\(CLI\) 생성:](#)
- [예 6 -- 델타 기간으로 SQL 데이터 세트 생성\(CLI\)](#)

예 1 -- SQL 데이터 세트(java) 생성:

```

CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("SQLAction1");
action.setQueryAction(new SqlQueryDatasetAction().withSqlQuery("select * from
  DataStoreName"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

```

```
//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

성공에 대한 출력:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>, RetentionPeriod: {unlimited:
true} or {numberOfDays: 10, unlimited: false}}
```

예 2 -- delta 기간으로 SQL 데이터 세트(java) 생성:

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Filter for DeltaTime
QueryFilter deltaTimeFilter = new QueryFilter();
deltaTimeFilter.withDeltaTime(
    new DeltaTime()
        .withOffsetSeconds(-1 * EstimatedDataDelayInSeconds)
        .withTimeExpression("from_unixtime(timestamp)"));

//Create Action
action.setActionName("SQLActionWithDeltaTime");
action.setQueryAction(new SqlQueryDatasetAction()
    .withSqlQuery("SELECT * from DataStoreName")
    .withFilters(deltaTimeFilter));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
```

```
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

성공에 대한 출력:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>, RetentionPeriod: {unlimited:
true} or {numberOfDays: 10, unlimited: false}}
```

예 3 -- 고유 일정 트리거로 컨테이너 데이터 세트(java) 생성:

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
    .withImage(ImageURI)
    .withExecutionRoleArn(ExecutionRoleArn)
    .withResourceConfiguration(
        new ResourceConfiguration()
            .withComputeType(new ComputeType().withAcu(1))
            .withVolumeSizeInGB(1))
    .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue")));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
```

```

DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);

```

성공에 대한 출력:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>, RetentionPeriod: {unlimited: true} or {numberOfDays: 10, unlimited: false}}
```

예 4 -- 트리거인 SQL 데이터 세트로 컨테이너 데이터 세트(java) 생성:

```

CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
    .withImage(ImageURI)
    .withExecutionRoleArn(ExecutionRoleArn)
    .withResourceConfiguration(
        new ResourceConfiguration()
            .withComputeType(new ComputeType().withAcu(1))
            .withVolumeSizeInGB(1))
    .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

```

```
//Create Trigger
DatasetTrigger trigger = new DatasetTrigger()
    .withDataset(new TriggeringDataset()
        .withName(TriggeringSQLDataSetName));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);
final CreateDatasetResult result = iot.createDataset(request);
```

성공에 대한 출력:

```
{DatasetName: <datasetName>, DatasetArn: <datasetARN>}
```

예 5 -- SQL 데이터 세트(CLI) 생성:

```
aws iotanalytics --endpoint <EndPoint> --region <Region> create-dataset --dataset-
name="<dataSetName>" --actions="[{"actionName\":"<ActionName>\", \"queryAction\":
{"sqlQuery\":"<SQLQuery>"}]" --retentionPeriod numberOfDays=10
```

성공에 대한 출력:

```
{
  "datasetName": "<datasetName>",
  "datasetArn": "<datasetARN>",
  "retentionPeriod": {unlimited: true} or {numberOfDays: 10, unlimited: false}
}
```

예 6 -- 델타 기간으로 SQL 데이터 세트 생성(CLI)

델타 기간이란 일련의 사용자 정의된 비중첩 연속 시간 간격입니다. 델타 기간을 통해 포함되는 데이터 세트 콘텐츠를 생성하고, 마지막 분석 이후 데이터 스토어에 새로 도착한 데이터를 분석할 수 있습니다. 데이터 세트의 queryAction에서 filters의 deltaTime을 설정해 델타 기간을 생성할 수 있습니다([CreateDataset](#)). 일반적으로 시간 간격 트리거를 설정하여 데이터 세트 콘텐츠를 자동으로 생성합니다(triggers:schedule:expression). 기본적으로 이를 통해 특정 기간 동안 도착한 메시지를 필터링할 수 있어 이전 기간에 온 메시지에 포함된 데이터가 두 번 고려되지 않습니다.

이 예에서는 마지막 시간 이후 도착한 데이터만 사용하여 15분마다 새 데이터 세트 콘텐츠를 자동으로 생성하는 새 데이터 세트를 만듭니다. 메시지가 지정된 데이터 스토어에 도착하기까지 3분의 지연을 허용하도록 3분(180초)의 `deltaTime` 오프셋을 지정합니다. 따라서 데이터 세트 콘텐츠가 오전 10:30에 생성된 경우 사용된 데이터(데이터 세트 콘텐츠에 포함)의 타임스탬프는 오전 10:12 ~ 오전 10:27(즉, 오전 10:30 - 15분 - 3분 ~ 오전 10:30 - 3분) 사이입니다.

```
aws iotanalytics --endpoint <EndPoint> --region <Region> create-dataset --cli-input-json file://delta-window.json
```

`delta-window.json` 파일에는 다음 코드가 포함되어 있습니다.

```
{
  "datasetName": "delta_window_example",
  "actions": [
    {
      "actionName": "delta_window_action",
      "queryAction": {
        "sqlQuery": "SELECT temperature, humidity, timestamp FROM my_datastore",
        "filters": [
          {
            "deltaTime": {
              "offsetSeconds": -180,
              "timeExpression": "from_unixtime(timestamp)"
            }
          }
        ]
      }
    }
  ],
  "triggers": [
    {
      "schedule": {
        "expression": "cron(0/15 * * * ? *)"
      }
    }
  ]
}
```

성공에 대한 출력:

```
{
```

```

"datasetName": "<datasetName>",
"datasetArn": "<datasetARN>",
}

```

노트북 컨테이너화

이 단원에는 Jupyter Notebook 사용하여 Docker 컨테이너를 빌드하는 방법에 대한 정보가 포함되어 있습니다. 타사에서 빌드한 노트북을 다시 사용하면 보안상 위험할 수 있습니다. 포함된 컨테이너는 사용자 권한으로 임의의 코드를 실행할 수 있습니다. 또한 노트북에서 생성된 HTML을 AWS IoT Analytics 콘솔에 표시할 수 있으므로 HTML을 표시하는 컴퓨터에 잠재적인 공격 벡터를 제공할 수 있습니다. 타사 노트북을 사용하기 전에 해당 노트북의 작성자를 신뢰할 수 있는지 확인하십시오.

고급 분석 기능을 실행하는 한 가지 옵션은 [Jupyter 노트북](#)을 사용하는 것입니다. Jupyter Notebook은 기계 학습과 각종 통계 분석을 수행할 수 있는 강력한 데이터 과학 도구입니다. 자세한 내용은 [노트북 템플릿](#)을 참조하세요. (JupyterLab 내부의 컨테이너화는 현재 지원되지 않습니다.) Jupyter Notebook과 라이브러리를 정의한 델타 기간 동안 AWS IoT Analytics가 수신할 때 새로운 데이터 배치에 대해 정기적으로 실행되는 컨테이너로 패키징할 수 있습니다. 지정 기간 내에 캡처된 새로운 세그먼트화된 데이터와 컨테이너를 사용하는 분석 작업을 예약한 다음 향후 예약된 분석을 위해 작업의 출력을 저장할 수 있습니다.

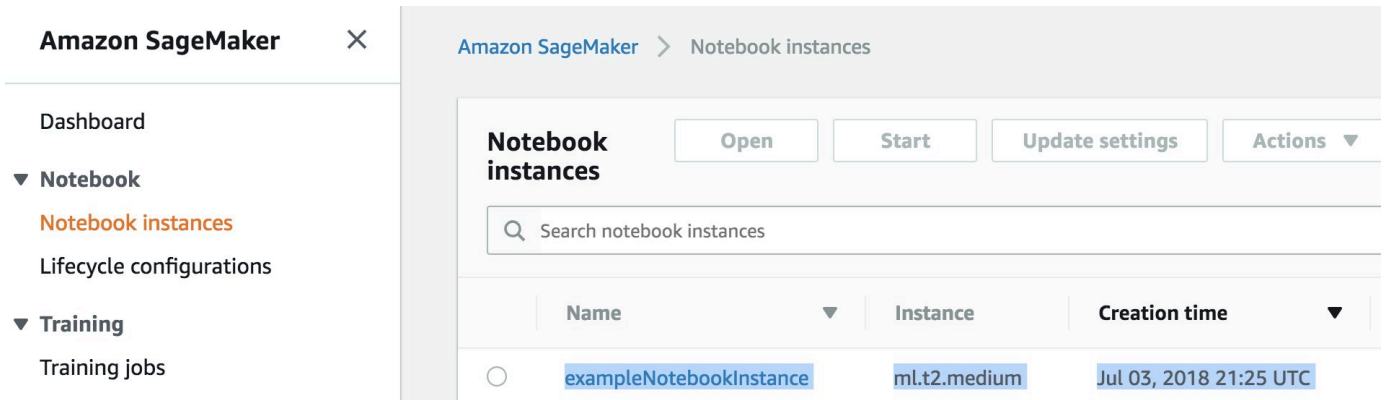
2018년 8월 23일 이후에 AWS IoT Analytics 콘솔을 사용하여 SageMaker 인스턴스를 생성한 경우, 컨테이너화 확장에 대한 설치가 자동으로 완료되고 [컨테이너화된 이미지 생성을 시작할 수 있습니다](#). 그렇지 않은 경우, 이 섹션에 나열된 단계에 따라 SageMaker 인스턴스에서 노트북 컨테이너화를 활성화합니다. 다음은 컨테이너 이미지를 Amazon EC2로 업로드할 수 있도록 SageMaker 실행 역할을 수정하고, 컨테이너화 확장을 설치합니다.

AWS IoT Analytics 콘솔로 생성하지 않은 노트북 인스턴스 컨테이너화 활성화

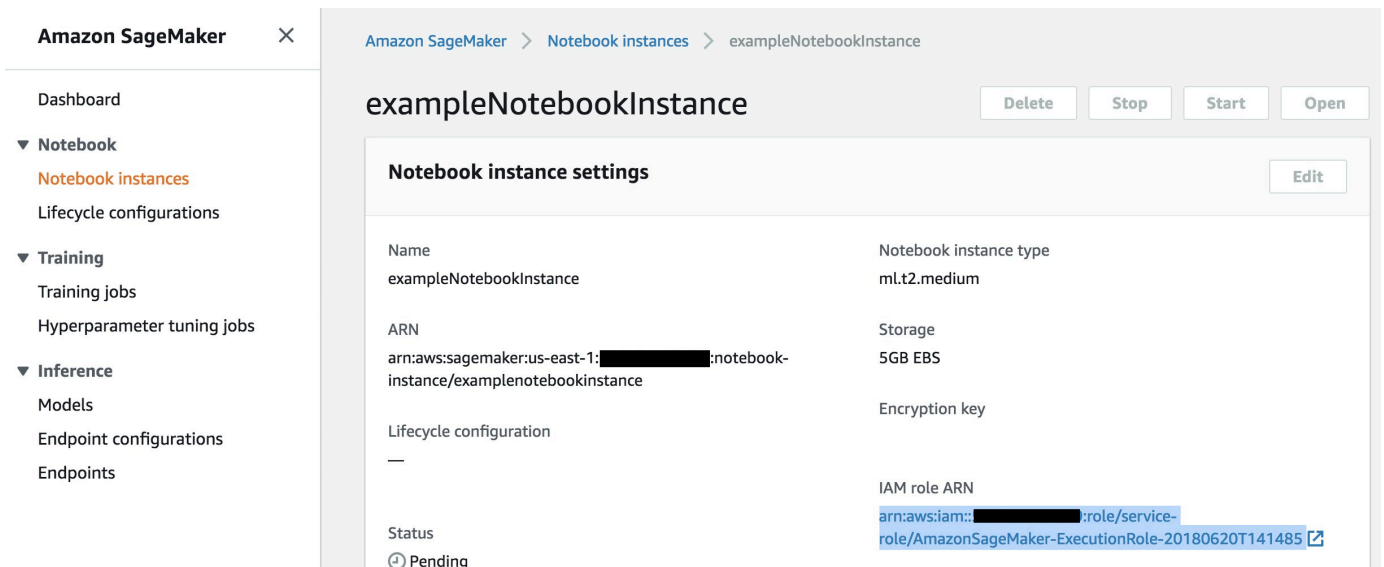
다음 단계 대신 AWS IoT Analytics 콘솔을 통해 새 SageMaker 인스턴스를 생성하는 것이 좋습니다. 새 인스턴스는 자동으로 컨테이너화를 지원합니다.

여기서 보듯, 컨테이너화를 활성화한 후 SageMaker 인스턴스를 다시 시작하면, IAM 역할과 정책을 다시 추가할 필요가 없지만, 마지막 단계에서 보듯 확장을 다시 설치해야 합니다.

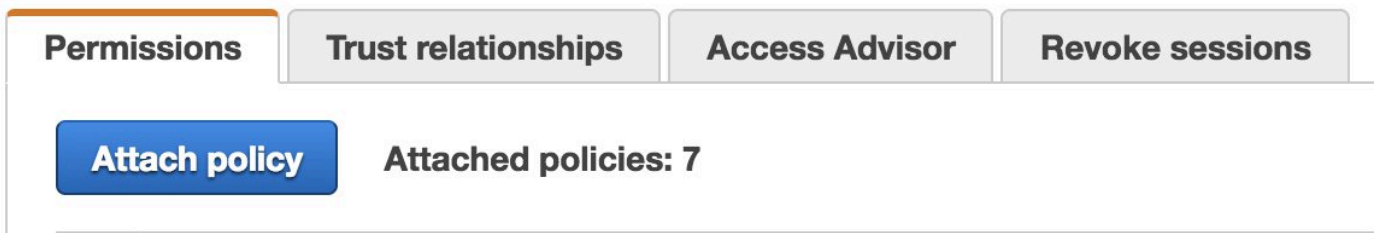
1. 노트북 인스턴스에 Amazon ECS에 대한 액세스 권한을 부여하려면 SageMaker 페이지에서 SageMaker 인스턴스를 선택합니다.



2. IAM 역할 ARN에서 SageMaker 실행 역할을 선택합니다.

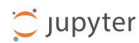


3. Attach Policy(정책 연결)을 선택한 후, 권한에 표시된 것처럼 정책을 정의하고 연결합니다. 아직 연결되어 있지 않은 경우 AmazonSageMakerFullAccess 정책을 연결합니다.



또한 Amazon S3에서 컨테이너화 코드를 다운로드하여 노트북 인스턴스에 설치해야 합니다. 첫 번째 단계는 SageMaker 인스턴스의 터미널에 액세스하는 것입니다.

1. Jupyter에서 새로 만들기를 선택합니다.



Quit

Files Running Clusters SageMaker Examples Conda



Upload New ↕

- 표시되는 메뉴에서 터미널을 선택합니다.



- 터미널에서 다음 명령을 입력해 코드를 다운로드해서 압축을 푼 후 설치합니다. 이 명령은 이 SageMaker 인스턴스의 노트북이 실행하는 모든 프로세스를 종료시킵니다.




```
cd /tmp
aws s3 cp s3://iotanalytics-notebook-containers/iota_notebook_containers.zip /tmp
unzip iota_notebook_containers.zip
cd iota_notebook_containers
chmod u+x install.sh
./install.sh
```

확장이 검증되어 설치되기까지 1~2분을 기다립니다.

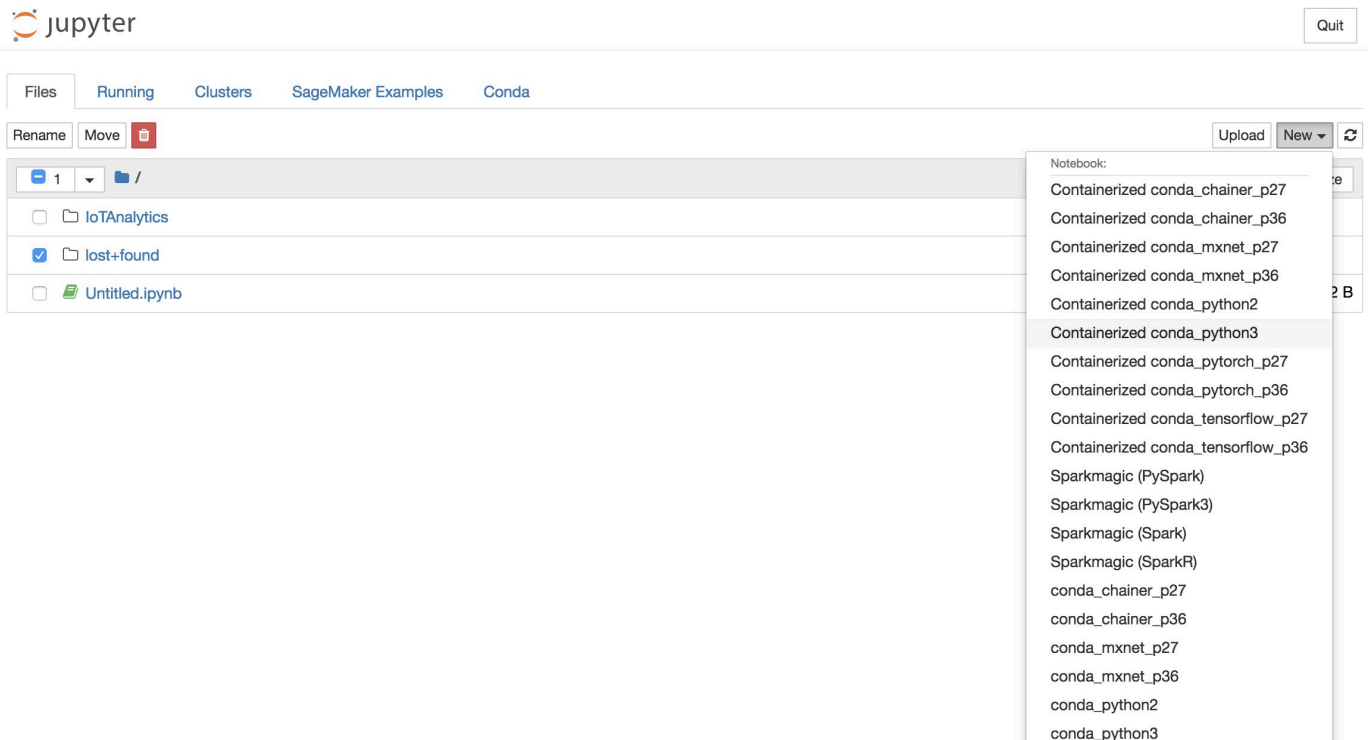
노트북 컨테이너화 확장 업데이트

2018년 8월 23일 이후에 AWS IoT Analytics 콘솔을 통해 SageMaker 인스턴스를 생성한 경우, 컨테이너화 확장이 자동 설치되어 있습니다. SageMaker 콘솔에서 인스턴스를 다시 시작해 이 확장을 업데이트할 수 있습니다. 확장을 수동으로 설치했다면, AWS IoT Analytics 콘솔로 생성하지 않은 노트북 인스턴스 컨테이너화 활성화에 나열된 터미널 명령을 다시 실행해 업데이트할 수 있습니다.

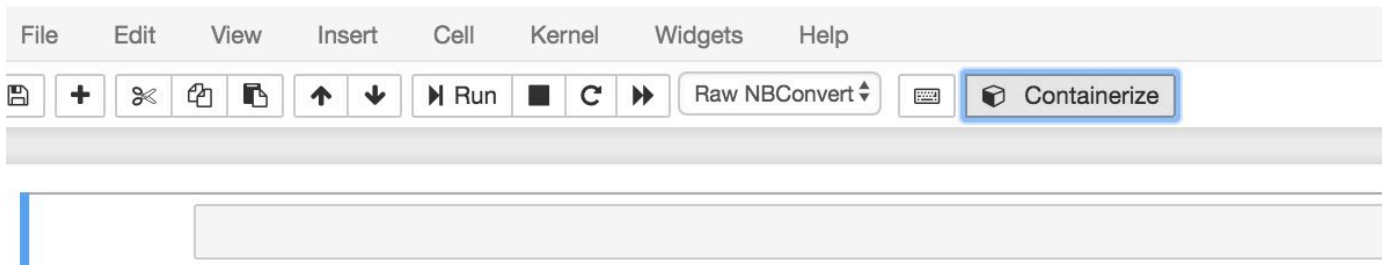
컨테이너화된 이미지 생성

이번 섹션에서는 노트북 컨테이너화에 필요한 단계들을 설명합니다. 먼저 Jupyter 노트북으로 이동해 컨테이너화된 커널을 가진 노트북을 생성합니다.

1. Jupyter 노트북에서 새로 만들기를 선택한 후, 드롭다운 목록에서 원하는 커널 유형을 선택합니다. (커널 유형은 "Containerized"로 시작하고 다른 방법으로는 선택했을 커널로 끝나야 합니다. 예를 들어, "conda_python3"과 같은 평범한 Python 3.0 환경을 원한다면 "Containerized conda_python3"을 선택합니다.)



2. 노트북에서 작업을 완료하고 컨테이너화를 원하면 컨테이너화 버튼을 선택합니다.



3. 컨테이너화된 노트북의 이름을 입력합니다. 또한 설명을 입력할 수 있습니다(선택 사항).



Container Name *

Container Description

Next

Exit

4. 노트북이 호출해야 하는 Input Variables(입력 변수)(파라미터)를 지정합니다. 노트북이 자동으로 감지한 입력 변수를 선택하거나, 사용자 지정 변수를 정의할 수 있습니다. (이전에 노트북을 실행 시킨 경우에만 입력 변수를 감지합니다.) 각 입력 변수에 대한 유형을 선택합니다. 선택에 따라 입력 변수에 대한 설명을 입력할 수도 있습니다.

1. Name

2. Input Variables

3. Select AWS ECR Repository

4. Review

5. Monitor Progress

Name	Type	Description	
<input type="text" value="ounces"/>	<input type="text" value="Double"/>	<input type="text"/>	<input type="button" value="X"/>
<input type="text" value="brand"/>	<input type="text" value="String"/>	<input type="text"/>	<input type="button" value="X"/>

Showing 1 to 2 of 2 variables

Previous Next

5. 노트북에서 생성한 이미지를 업로드할 Amazon ECR 리포지토리를 선택합니다.

1. Name

2. Input Variables

3. Select AWS ECR Repository

4. Review

5. Monitor Progress

Please upload different notebooks to different repositories.

Repository Name Create Search:

Name
my-repo
my-repo2
my-repo3

Showing 1 to 3 of 3 repositories Previous Next

Previous Next

6. 컨테이너화를 선택해 프로세스를 시작합니다.

자신의 입력을 요약한 개요가 제공됩니다. 프로세스를 시작하면 이를 취소할 수 없다는 점에 유의해주세요. 프로세스는 최대 1시간까지 지속될 수 있습니다.

1. Name 2. Input Variables 3. Select AWS ECR Repository **4. Review** 5. Monitor Progress

Container Name: Beer-Tastiness-Calculator
Container Description:
Upload To: my-repo

Variable Name	Type	Description
ounces	Double	
brand	String	

Showing 1 to 2 of 2 variables Previous **1** Next

Previous **Containerize**

Exit

7. 다음 페이지에 진행률이 표시됩니다.

1. Name 2. Input Variables 3. Select AWS ECR Repository 4. Review **5. Monitor Progress**

The containerization process typically completes within 30 minutes.

Creating Image...

Exit

8. 실수로 브라우저를 종료한 경우, AWS IoT Analytics 콘솔의 노트북 섹션에서 컨테이너화 프로세스 상태를 모니터링할 수 있습니다.
9. 프로세스가 완료된 후, 컨테이너화된 이미지가 Amazon ECR에 사용할 수 있는 상태로 저장됩니다.

Containerize Notebook ✕



1. Name

2. Input Variables

3. Select AWS ECR Repository

4. Review

5. Monitor Progress

Creating Image... Uploading Image... 

You can now use this notebook for scheduled analysis of your Data Sets.

[Go To Data Sets](#)[Exit](#)

분석에 사용자 지정 컨테이너 사용

이 단원에는 Jupyter Notebook 사용하여 Docker 컨테이너를 빌드하는 방법에 대한 정보가 포함되어 있습니다. 타사에서 빌드한 노트북을 다시 사용하면 보안상 위험할 수 있습니다. 포함된 컨테이너는 사용자 권한으로 임의의 코드를 실행할 수 있습니다. 또한 노트북에서 생성된 HTML을 AWS IoT Analytics 콘솔에 표시할 수 있으므로 HTML을 표시하는 컴퓨터에 잠재적인 공격 벡터를 제공할 수 있습니다. 타사 노트북을 사용하기 전에 해당 노트북의 작성자를 신뢰할 수 있는지 확인하십시오.

고유의 사용자 지정 컨테이너를 생성하여 AWS IoT Analytics 서비스를 통해 실행할 수 있습니다. 이렇게 하려면 도커 이미지를 설정해 Amazon ECR에 업로드한 다음, 컨테이너 작업을 실행할 데이터 세트를 설정합니다. 이 섹션에서는 Octave를 사용하는 프로세스에 대한 예를 제공합니다.

이 자습서는 다음을 가정합니다.

- 로컬 컴퓨터에 설치된 Octave
- 로컬 컴퓨터에 설정된 Docker 계정
- Amazon ECR 또는 AWS IoT Analytics 액세스 권한이 있는 AWS 계정

1단계: 도커 이미지 설정

이 자습서에 필요한 3가지 주요 파일이 있습니다. 다음은 파일 이름과 내용입니다.

- Dockerfile – Docker 컨테이너화 프로세스의 초기 설정입니다.

```
FROM ubuntu:16.04

# Get required set of software
RUN apt-get update
RUN apt-get install -y software-properties-common
RUN apt-get install -y octave
RUN apt-get install -y python3-pip

# Get boto3 for S3 and other libraries
RUN pip3 install --upgrade pip
RUN pip3 install boto3
RUN pip3 install urllib3

# Move scripts over
ADD moment moment
ADD run-octave.py run-octave.py

# Start python script
ENTRYPOINT ["python3", "run-octave.py"]
```

- run-octave.py - AWS IoT Analytics에서 JSON을 구문 분석하고, Octave 스크립트를 실행하고, 아티팩트를 Amazon S3으로 업로드합니다.

```
import boto3
import json
import os
import sys
from urllib.parse import urlparse

# Parse the JSON from IoT Analytics
with open('/opt/ml/input/data/iotanalytics/params') as params_file:
    params = json.load(params_file)

variables = params['Variables']

order = variables['order']
input_s3_bucket = variables['inputDataS3BucketName']
```

```

input_s3_key = variables['inputDataS3Key']
output_s3_uri = variables['octaveResultS3URI']

local_input_filename = "input.txt"
local_output_filename = "output.mat"

# Pull input data from S3...
s3 = boto3.resource('s3')
s3.Bucket(input_s3_bucket).download_file(input_s3_key, local_input_filename)

# Run Octave Script
os.system("octave moment {} {} {}".format(local_input_filename,
    local_output_filename, order))

# # Upload the artifacts to S3
output_s3_url = urlparse(output_s3_uri)
output_s3_bucket = output_s3_url.netloc
output_s3_key = output_s3_url.path[1:]

s3.Object(output_s3_bucket, output_s3_key).put(Body=open(local_output_filename,
    'rb'), ACL='bucket-owner-full-control')

```

- `moment` – 입력 또는 출력 파일과 지정된 순서를 근거로 모멘트를 계산하는 간단한 Octave 스크립트입니다.

```

#!/usr/bin/octave -qf

arg_list = argv ();
input_filename = arg_list{1};
output_filename = arg_list{2};
order = str2num(arg_list{3});

[D,delimiterOut]=importdata(input_filename)
M = moment(D, order)

save(output_filename, 'M')

```

1. 각 파일의 내용을 다운로드합니다. 새 디렉터리를 생성하고, 그 안에 모든 파일을 넣은 다음 `cd`를 해당 디렉터리에 넣습니다.
2. 다음 명령을 실행합니다.

```
docker build -t octave-moment .
```

3. Docker 리포지터리에 새 이미지가 표시될 것입니다. 다음 명령을 실행하여 인증서를 확인합니다.

```
docker image ls | grep octave-moment
```

2단계: 도커 이미지를 Amazon ECR 리포지토리로 업로드

1. Amazon ECR 리포지토리를 생성합니다.

```
aws ecr create-repository --repository-name octave-moment
```

2. Docker 환경에 로그인을 합니다.

```
aws ecr get-login
```

3. 출력을 복사해 실행시킵니다. 출력은 다음과 같아야 합니다.

```
docker login -u AWS -p password -e none https://your-aws-account-id.dkr.ecr..amazonaws.com
```

4. Amazon ECR 리포지토리 태그로 생성한 이미지를 태그 처리합니다.

```
docker tag your-image-id your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

5. 이미지를 Amazon ECR로 푸시합니다.

```
docker push your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

3단계: 샘플 데이터를 Amazon S3 버킷으로 업로드

1. 다음을 파일 `input.txt`로 다운로드합니다.

```
0.857549 -0.987565 -0.467288 -0.252233 -2.298007
0.030077 -1.243324 -0.692745 0.563276 0.772901
-0.508862 -0.404303 -1.363477 -1.812281 -0.296744
-0.203897 0.746533 0.048276 0.075284 0.125395
```

```
0.829358  1.246402  -1.310275  -2.737117  0.024629
1.206120  0.895101  1.075549  1.897416  1.383577
```

2. `octave-sample-data-your-aws-account-id`로 호출된 Amazon S3 버킷을 생성합니다.
3. 파일 `input.txt`를 방금 생성한 Amazon S3 버킷으로 업로드합니다. `input.txt` 파일을 포함한 `octave-sample-data-your-aws-account-id`이라는 이름의 버킷이 있어야 합니다.

4단계: 컨테이너 실행 역할 생성

1. 다음을 `role1.json` 파일에 복사합니다. `your-aws-account-id`를 AWS 계정 ID로, `aws-region`을 AWS 리소스의 AWS 리전으로 바꿉니다.

Note

이 예에는 혼동된 대리인 보안 문제로부터 보호하기 위한 전역 조건 컨텍스트 키가 포함되어 있습니다. 자세한 내용은 [the section called “교차 서비스 혼동된 대리자 예방”](#) 섹션을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "sagemaker.amazonaws.com",
          "iotanalytics.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "your-aws-account-id"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:your-aws-account-id:dataset/DOC-EXAMPLE-DATASET"
        }
      }
    }
  ]
}
```

```
]
}
```

- 다운로드한 `role1.json` 파일을 사용하여 SageMaker 및 AWS IoT Analytics에 액세스 권한을 부여하는 역할을 생성합니다.

```
aws iam create-role --role-name container-execution-role --assume-role-policy-
document file://role1.json
```

- 이름이 `policy1.json`인 파일에 다음을 다운로드하고, *your-account-id*을 계정 ID로 대체합니다(Statement:Resource의 두 번째 ARN 참조).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:PutObject",
        "s3:GetObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::*-dataset-*/**",
        "arn:aws:s3:::octave-sample-data-your-account-id/**"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
```

```

    "logs:DescribeLogStreams",
    "logs:GetLogEvents",
    "logs:PutLogEvents"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:ListAllMyBuckets"
  ],
  "Resource" : "*"
}
]
}

```

4. 방금 다운로드한 `policy.json` 파일을 사용하여 IAM 정책을 생성합니다.

```
aws iam create-policy --policy-name ContainerExecutionPolicy --policy-document
file://policy1.json
```

5. 정책을 역할에 연결합니다.

```
aws iam attach-role-policy --role-name container-execution-role --policy-arn
arn:aws:iam::your-account-id:policy/ContainerExecutionPolicy
```

5단계: 컨테이너 작업으로 데이터 세트 생성

1. 이름이 `cli-input.json`인 파일에 다음을 다운로드하고, *your-account-id* 및 *region*의 모든 인스턴스를 적절한 값으로 바꿉니다.

```

{
  "datasetName": "octave_dataset",
  "actions": [
    {
      "actionName": "octave",
      "containerAction": {
        "image": "your-account-id.dkr.ecr.region.amazonaws.com/octave-
moment",

```

```

        "executionRoleArn": "arn:aws:iam::your-account-id:role/container-
execution-role",
        "resourceConfiguration": {
            "computeType": "ACU_1",
            "volumeSizeInGB": 1
        },
        "variables": [
            {
                "name": "octaveResultS3URI",
                "outputFileUriValue": {
                    "fileName": "output.mat"
                }
            },
            {
                "name": "inputDataS3BucketName",
                "stringValue": "octave-sample-data-your-account-id"
            },
            {
                "name": "inputDataS3Key",
                "stringValue": "input.txt"
            },
            {
                "name": "order",
                "stringValue": "3"
            }
        ]
    }
}

```

2. 방금 다운로드해 편집한 `cli-input.json` 파일을 사용하여 데이터 세트를 생성합니다.

```
aws iotanalytics create-dataset --cli-input-json file://cli-input.json
```

6단계: 데이터 세트 내용 생성 호출

1. 다음 명령을 실행합니다.

```
aws iotanalytics create-dataset-content --dataset-name octave-dataset
```

7단계: 데이터 세트 내용 가져오기

1. 다음 명령을 실행합니다.

```
aws iotanalytics get-dataset-content --dataset-name octave-dataset --version-id \  
$LATEST
```

2. DatasetContentState이 SUCCEEDED가 되기까지 몇 분 정도 기다려야 할 수도 있습니다.

8단계: 출력을 Octave로 인쇄

1. Octave 셸을 사용해 다음 명령을 실행하고 컨테이너에서 출력을 인쇄합니다.

```
bash> octave  
octave> load output.mat  
octave> disp(M)  
-0.016393 -0.098061 0.380311 -0.564377 -1.318744
```


AWS IoT Analytics 데이터 시각화

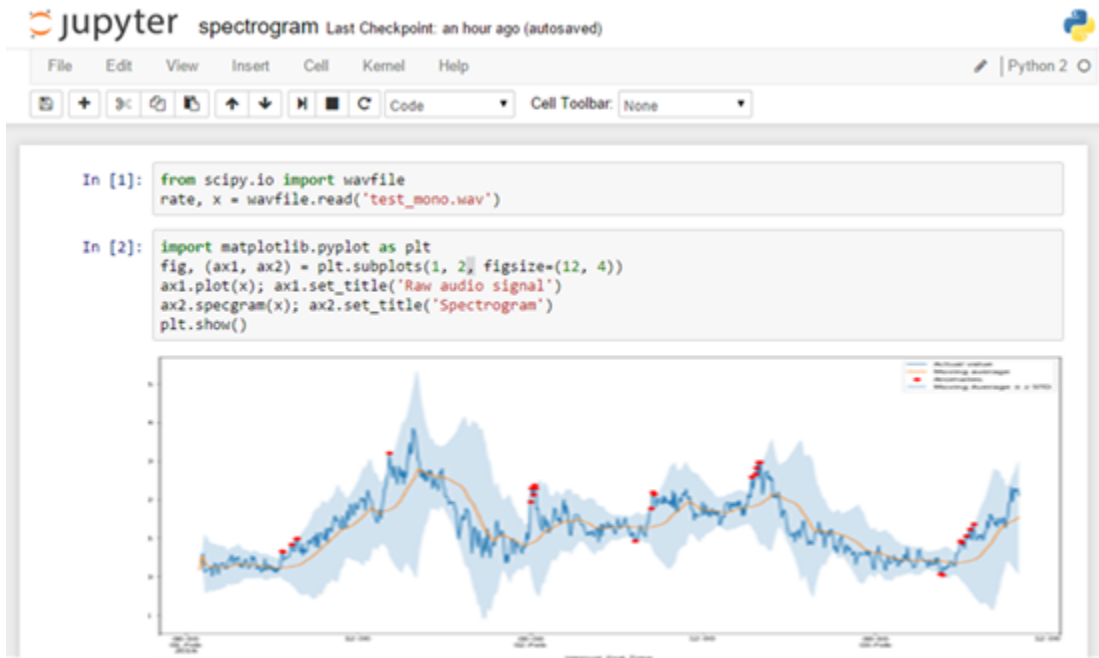
AWS IoT Analytics 데이터를 시각화하기 위해 AWS IoT Analytics 콘솔 또는 Amazon QuickSight를 사용할 수 있습니다.

주제

- [콘솔을 사용한 AWS IoT Analytics 데이터 시각화](#)
- [Amazon QuickSight를 사용한 AWS IoT Analytics 데이터 시각화](#)

콘솔을 사용한 AWS IoT Analytics 데이터 시각화

AWS IoT Analytics는 [AWS IoT Analytics 콘솔](#)의 컨테이너 데이터 세트 콘텐츠 페이지에 컨테이너 데이터 세트의 HTML 출력(output.html 파일에 있음)을 포함할 수 있습니다. 예를 들어, Jupyter Notebook 실행하는 컨테이너 데이터 세트를 정의하고 Jupyter 노트북에서 시각화를 작성하는 경우, 데이터 세트는 다음과 같을 수 있습니다.



컨테이너 데이터 세트 콘텐츠가 작성되면 콘솔의 데이터 세트 콘텐츠 페이지에서 이 시각화를 볼 수 있습니다.



Jupyter Notebook 실행하는 컨테이너 데이터 세트 작성에 대한 자세한 내용은 [내 워크플로우 자동화](#)를 참조하십시오.

Amazon QuickSight를 사용한 AWS IoT Analytics 데이터 시각화

AWS IoT Analytics는 [Amazon QuickSight](#)와의 직접 통합을 제공합니다. Amazon QuickSight는 데이터를 사용하여 시각적 객체를 빌드하고, 애드혹 분석을 수행하고, 비즈니스 관련 통찰력을 빠르게 얻을 수 있는 신속한 비즈니스 분석 서비스입니다. Amazon QuickSight는 강력한 인 메모리 엔진(SPIICE)을 사용하여 사용자 수 수십만 명 규모로 조직을 확대하고, 즉각적인 성능을 발휘합니다. Amazon QuickSight 콘솔에서 AWS IoT Analytics 데이터 세트를 선택하고 대시보드 생성과 시각화를 시작할 수 있습니다. Amazon QuickSight는 [이러한 리전](#)에서 사용할 수 있습니다.

Amazon QuickSight 시각화를 시작하려면 Amazon QuickSight 계정을 생성해야 합니다. 계정을 설정할 때 AWS IoT Analytics 데이터에 대한 액세스 권한을 Amazon QuickSight에 부여해야 합니다. 이미 계정이 있는 경우 Admin, Manage QuickSight, Security & permissions을 선택하여 Amazon QuickSight가 AWS IoT Analytics 데이터에 액세스할 수 있는 권한을 부여합니다. QuickSight access to AWS services에서 추가 또는 제거를 선택한 다음 AWS IoT Analytics 옆에 있는 확인란을 선택하고 업데이트를 선택합니다.

Account name:
Edition: Enterprise

Security & permissions

QuickSight can control access to AWS resources for the entire account in addition to individual users and groups

QuickSight access to AWS services

Amazon Redshift Amazon RDS IAM Amazon S3 AWS IoT Analytics

By configuring access to AWS services, QuickSight can access the data in those services. Access by users and groups can be controlled through the options below.

[Add or remove](#)

Default resource access

① Users and groups have access to all connected resources.

QuickSight can allow or deny access to all users and groups by default, when an individual access control is not in effect for a particular user or group

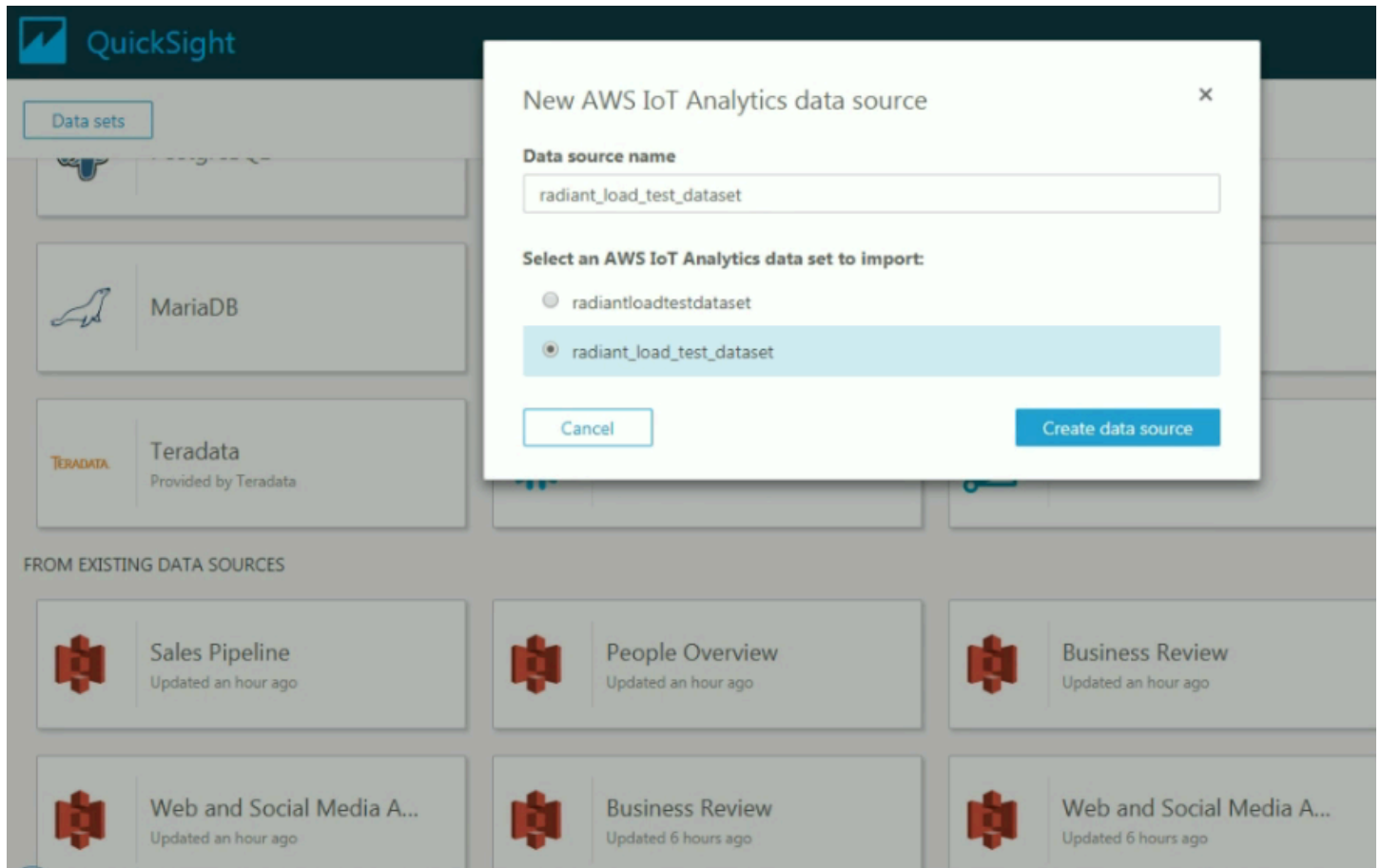
[Change](#)

Resource access for individual users and groups

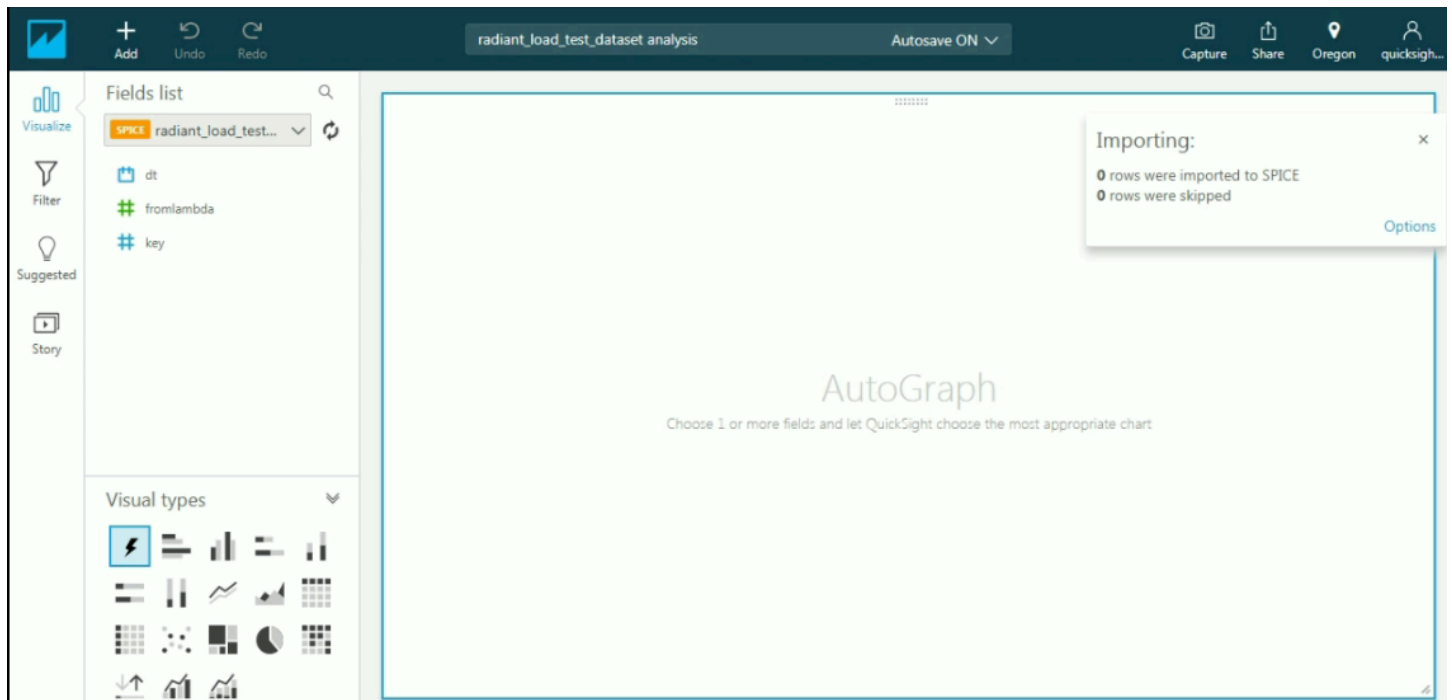
Resource access is controlled by assigning IAM policies.

[IAM policy assignments](#)

계정이 설정된 후 Amazon QuickSight 콘솔 관리 페이지에서 새 분석 및 새 데이터 세트를 선택한 다음 AWS IoT Analytics를 소스로 선택합니다. 데이터 소스의 이름을 입력하고, 가져올 데이터 세트를 선택한 다음 데이터 소스 생성을 선택합니다.



데이터 소스가 생성된 후 Amazon QuickSight에서 시각화를 생성할 수 있습니다.



Amazon QuickSight 대시보드 및 데이터 세트에 대한 자세한 내용은 [Amazon QuickSight 설명서](#)를 참조하세요.

AWS IoT Analytics 리소스에 태그 지정

채널과 데이터 세트, 데이터 스토어, 파이프라인을 쉽게 관리하기 위해 선택적으로 이들 리소스 각각에 태그의 형태로 메타데이터를 할당할 수 있습니다. 이 챕터는 태그에 대해 설명하고, 태그를 생성하는 방법을 보여줍니다.

주제

- [태그 기본 사항](#)
- [IAM 정책에 태그 사용](#)
- [태그 제한](#)

태그 기본 사항

태그를 사용하면 용도, 소유자 또는 환경을 기준으로 하는 등 AWS IoT Analytics 리소스를 다양한 방식으로 분류할 수 있습니다. 이 기능은 지정한 태그에 따라 특정 리소스를 빠르게 식별할 수 있기 때문에 동일한 유형의 리소스가 많을 때 유용합니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다. 예를 들어, 채널에 태그 세트를 정의하면 쉽게 각 채널 메시지 소스를 책임진 디바이스의 유형을 추적할 수 있습니다. 각 리소스 유형에 대한 요건을 충족하는 태그 키 세트를 고안하는 것이 좋습니다. 일관된 태그 키 세트를 사용하면 리소스를 보다 쉽게 관리할 수 있습니다. 추가하는 태그에 따라 리소스를 검색하고 필터링할 수 있습니다.

또한 태그를 사용하여 비용을 분류 및 추적할 수 있습니다. 채널과 데이터 세트, 데이터 스토어에 태그를 적용하면, AWS에서 사용 내역 및 비용이 태그별로 집계된 CSV 파일로 비용 할당 보고서를 만듭니다. 비즈니스 범주를 나타내는 태그(예: 비용 센터, 애플리케이션 이름 또는 소유자)를 적용하여 여러 서비스에 대한 비용을 정리할 수 있습니다. 비용 할당 태그 사용에 대한 자세한 내용은 [AWS Billing 사용자 설명서](#)의 [비용 할당 태그 사용](#)을 참조하세요.

AWS Billing and Cost Management 콘솔의 태그 편집기는 태그를 생성하고 관리할 수 있는 중앙 통합 방식으로, 이 도구를 사용하면 아주 편리합니다. 자세한 내용은 [AWS Management Console 시작하기](#)에서 [태그 편집기로 작업](#)을 참조하세요.

AWS CLI 및 AWS IoT Analytics API를 사용하여 태그 관련 작업을 수행할 수 있습니다. 채널과 데이터 세트, 데이터 스토어, 파이프라인을 생성할 때 다음 명령의 태그 필드를 사용해 태그를 연결할 수 있습니다.

- [CreateChannel](#)

- [CreateDataset](#)
- [CreateDatastore](#)
- [CreatePipeline](#)

태깅을 지원하는 기존 리소스에 대해 태그를 추가, 수정, 삭제할 수 있습니다. 다음 명령을 사용합니다.

- [TagResource](#)
- [ListTagsForResource](#)
- [UntagResource](#)

태그 키와 값을 편집할 수 있으며 언제든지 리소스에서 태그를 제거할 수 있습니다. 태그의 값을 빈 문자열로 설정할 수 있지만 태그의 값을 Null로 설정할 수는 없습니다. 해당 리소스에 대해 키가 기존 태그와 동일한 태그를 추가하는 경우 새 값이 이전 값을 덮어씁니다. 리소스를 삭제하면, 리소스에 대한 연결이 완료된 태그 또한 삭제됩니다.

IAM 정책에 태그 사용

리소스 태그를 기반으로 사용자 액세스(권한)를 제어하기 위해 IAM 정책에서 다음 조건 컨텍스트 키/값과 함께 Condition 요소(Condition 블록이라고도 함)를 사용할 수 있습니다.

- `iotanalytics:ResourceTag/<tag-key>`: `<tag-value>`를 사용하여 특정 태그가 지정된 리소스에 대한 사용자 작업을 허용 또는 거부합니다.
- `aws:RequestTag/<tag-key>`: `<tag-value>`를 사용하여 태그를 허용하는 리소스를 생성하거나 수정하는 API 요청을 작성할 때 특정 태그를 사용하도록(또는 사용하지 않도록) 요구합니다.
- `aws:TagKeys`: [`<tag-key>`, ...]를 사용하여 태깅 가능한 리소스를 생성하거나 수정하는 API 요청을 작성할 때 특정 태그 키 집합을 사용하도록(또는 사용하지 않도록) 요구합니다.

Note

IAM 정책의 조건 컨텍스트 키/값은 태깅 가능한 리소스의 ID가 필수 파라미터인 AWS IoT Analytics 작업에만 적용됩니다. 예를 들어, 이 요청에서 태그를 지정할 수 있는 리소스(채널, 데이터 세트, 데이터 스토어 또는 파이프라인)가 언급되지 않기 때문에 조건 컨텍스트 키/값에 따라 [DescribeLoggingOptions](#)의 사용이 허용/거부되지 않습니다.

자세한 내용은 IAM 사용 설명서의 [태그를 사용한 액세스 제어](#)를 참조하세요. 이 설명서의 [IAM JSON 정책 참조](#) 단원에서는 IAM에서 JSON 정책의 자세한 구문과 설명, 요소의 예, 변수 및 평가 로직을 설명합니다.

다음은 태그 기반 제한을 적용하는 정책 예제입니다. 이 정책으로 제한되는 사용자는 다음과 같습니다.

1. 리소스에 태그 "env=prod"를 지정할 수 없습니다. 이 예제에서는 "aws:RequestTag/env" : "prod" 행을 참조하십시오.
2. 기존 태그 "env=prod"가 지정된 리소스를 수정 또는 액세스할 수 없습니다(이 예제에서는 "iotanalytics:ResourceTag/env" : "prod" 행을 참조하세요).

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "aws:RequestTag/env" : "prod"
        }
      }
    },
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "iotanalytics:ResourceTag/env" : "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    }
  ]
}
```



```

    }
  ]
}

```

또한 다음 예제와 같이 목록에서 태그를 둘러싸 지정된 태그 키에 대해 여러 태그 값을 지정할 수도 있습니다.

```

"StringEquals" : {
  "iotanalytics:ResourceTag/env" : ["dev", "test"]
}

```

Note

태그를 기준으로 리소스에 대한 사용자 액세스를 허용 또는 거부하는 경우 동일한 리소스에서 태그를 추가 또는 제거할 수 있도록 사용자를 명시적으로 거부할 것을 고려해야 합니다. 그렇지 않으면 사용자가 제한을 피해 태그를 수정하여 리소스에 대한 액세스 권한을 얻을 수 있습니다.

태그 제한

태그에 적용되는 기본 제한은 다음과 같습니다.

- 리소스 당 최대 태그 수 - 50개
- 최대 키 길이 - UTF-8의 유니코드 문자 127자
- 최대 값 길이 - UTF8의 유니코드 문자 255자
- 태그 키와 값은 대/소문자를 구분합니다.
- 태그 이름이나 값에서 `aws: prefix`를 사용하지 마세요. 이 단어는 AWS용으로 예약되어 있습니다. 이 접두사가 지정된 태그 이름이나 값은 편집하거나 삭제할 수 없습니다. 이 접두사가 지정된 태그는 소스당 태그 수 제한에 포함되지 않습니다.
- 태깅 스키마를 여러 서비스와 리소스에서 사용하는 경우 다른 서비스 또한 허용되는 문자에 대한 제한이 있을 수 있음을 유의합니다. 일반적으로 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 공백 및 숫자와 특수 문자(+ - = . _ : / @)입니다.

AWS IoT Analytics의 SQL 표현식

데이터 세트는 데이터 스토어의 데이터에 대한 SQL 식을 사용하여 생성됩니다. AWS IoT Analytics는 Amazon Athena와 동일한 SQL 쿼리, 함수 및 연산자를 사용합니다.

AWS IoT Analytics는 ANSI 표준 SQL 구문의 하위 집합을 지원합니다.

```
SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[[ INNER | OUTER ] LEFT | RIGHT | FULL | CROSS JOIN join_item [ ON join_condition ]]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ UNION [ ALL | DISTINCT ] union_query ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ LIMIT [ count | ALL ] ]
```

파라미터에 대한 설명은 Amazon Athena 설명서의 [파라미터](#)를 참조하세요.

AWS IoT Analytics 및 Amazon Athena는 다음을 지원하지 않습니다.

- WITH 절
- CREATE TABLE AS SELECT 문
- INSERT INTO 문
- 준비된 명령문으로는 USING를 사용하여 EXECUTE를 실행할 수 없습니다.
- CREATE TABLE LIKE
- DESCRIBE INPUT 및 DESCRIBE OUTPUT
- EXPLAIN 문
- 사용자 정의 함수(UDF 또는 UDAF)
- 저장 프로시저
- 연동 커넥터

주제

- [AWS IoT Analytics에서 지원되는 SQL 함수](#)
- [AWS IoT Analytics에서 SQL 쿼리와 관련된 일반적인 문제 해결](#)

AWS IoT Analytics에서 지원되는 SQL 함수

데이터 세트는 데이터 스토어의 데이터에 SQL 표현식을 사용하여 생성됩니다. AWS IoT Analytics에서 실행하는 쿼리는 [Presto 0.217](#)을 기반으로 합니다.

지원되는 데이터 형식

AWS IoT Analytics 및 Amazon Athena는 이러한 데이터 유형을 지원합니다.

- primitive_type
 - TINYINT
 - SMALLINT
 - INT
 - BIGINT
 - BOOLEAN
 - DOUBLE
 - FLOAT
 - STRING
 - TIMESTAMP
 - DECIMAL(precision, scale)
 - DATE
 - CHAR(길이 지정된 고정 길이 문자 데이터)
 - VARCHAR(길이 지정된 가변 길이 문자 데이터)
- array_type
 - ARRAY<data_type>
- map_type
 - MAP<primitive_type, data_type>
- struct_type
 - STRUCT<col_name:data_type[COMMENT col_comment][,...]>

Note

AWS IoT Analytics 및 Amazon Athena는 일부 데이터 유형을 지원하지 않습니다.

지원되는 함수

Amazon Athena 및 AWS IoT Analytics SQL 함수는 [Presto 0.217](#)을 기반으로 합니다. 관련된 함수, 연산자, 표현식에 대한 자세한 내용은 Presto 설명서에서 [함수 및 연산자](#) 및 다음의 특정 단원을 참조하세요.

- 논리 연산자
- 비교 함수와 연산자
- 조건식
- 변환 함수
- 수학 함수와 연산자
- 비트 함수
- 소수 함수와 연산자
- 문자열 함수와 연산자
- 이진 함수
- 날짜 및 시간 함수와 연산자
- 정규식 함수
- JSON 함수와 연산자
- URL 함수
- 집계 함수
- 원도 함수
- 색상 함수
- 배열 함수와 연산자
- 맵 함수와 연산자
- 람다 식과 함수
- Teradata 함수

Note

AWS IoT Analytics 및 Amazon Athena에서는 사용자 정의 함수(UDF 또는 UDAF) 또는 저장 프로시저를 지원하지 않습니다.

AWS IoT Analytics에서 SQL 쿼리와 관련된 일반적인 문제 해결

다음 정보를 사용하면 AWS IoT Analytics에서 SQL 쿼리 문제를 해결하는 데 도움이 됩니다.

- 작은따옴표를 이스케이프 처리하려면 작은따옴표 앞에 다른 작은따옴표를 추가합니다. 이를 큰따옴표와 혼동하지 마십시오.

Example 예

```
SELECT '0'Reilly'
```

- 밑줄을 이스케이프 처리하려면 악센트 부호(`)를 사용하여 밑줄로 시작하는 데이터 스토어 열 이름을 묶습니다.

Example 예

```
SELECT ` _myMessageAttribute ` FROM myDataStore
```

- 숫자로 이름을 이스케이프 처리하려면 숫자가 포함된 데이터 스토어 이름을 큰따옴표로 묶어야 합니다.

Example 예

```
SELECT * FROM "myDataStore123"
```

- 예약 키워드를 이스케이프 처리하려면 예약 키워드를 큰따옴표로 묶어야 합니다. 자세한 내용은 SQL SELECT 문의 [예약 키워드 목록](#)을 참조하세요.

보안 내부 AWS IoT Analytics

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 기업 간의 AWS 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 정기적으로 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 테스트하고 검증합니다. 적용되는 규정 준수 프로그램에 대해 알아보려면 규정 [준수 프로그램별 범위 내 AWS 서비스를](#) 참조하십시오. AWS IoT Analytics
- 클라우드에서의 보안 - 귀하의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 데이터의 민감도, 조직의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 공동 책임 모델을 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. AWS IoT Analytics. 다음 항목에서는 보안 및 규정 준수 목표를 AWS IoT Analytics 충족하도록 구성하는 방법을 보여줍니다. 또한 AWS IoT Analytics 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 배우게 됩니다.

AWS Identity and Access Management 에서 AWS IoT Analytics

AWS Identity and Access Management (IAM) 은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 있도록 도와주는 AWS 서비스입니다. AWS IAM 관리자는 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. AWS IoT Analytics IAM은 추가 AWS 비용 없이 사용할 수 있는 서비스입니다.

고객

사용하는 방식 AWS Identity and Access Management (IAM) 은 수행하는 작업에 따라 다릅니다. AWS IoT Analytics

서비스 사용자 - AWS IoT Analytics 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS IoT Analytics 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다.

니다. AWS IoT Analytics의 기능에 액세스할 수 없는 경우 [AWS IoT Analytics ID 및 액세스 문제 해결 단원](#)을 참조하십시오.

서비스 관리자 — 회사에서 AWS IoT Analytics 리소스를 담당하는 경우 전체 액세스 권한이 있을 수 있습니다. 서비스 사용자가 액세스해야 하는 AWS IoT Analytics 기능과 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요. 회사에서 IAM을 어떻게 사용할 수 있는지 자세히 AWS IoT Analytics 알아보려면 [IAM의 AWS IoT Analytics 작동 방식](#).

IAM 관리자 - IAM 관리자라면 AWS IoT Analytics에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AWS IoT Analytics ID 기반 정책의 예를 보려면 [참조하십시오. AWS IoT Analytics ID 기반 정책 예제](#)

보안 인증을 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM 사용자 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 [내 로그인 방법](#)을 참조하십시오. AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK)와 명령줄 인터페이스 (CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS [API 요청 서명](#)을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA)을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [다중 인증](#) 및 IAM 사용 설명서의 [AWS에서 다중 인증\(MFA\) 사용](#)을 참조하세요.

AWS 계정 루트 사용자

계정을 AWS 계정 만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 이를 사용하여 루트 사용자만 수행할 수 있는 작업을 수행하세요. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 [Tasks that require root user credentials](#)를 참조하세요.

IAM 사용자 및 그룹

[IAM 사용자는 단일 사용자](#) 또는 애플리케이션에 대한 특정 권한을 가진 사용자 내의 자격 증명입니다. AWS 계정 가능하다면 암호 및 액세스 키와 같은 장기 보안 인증 정보가 있는 IAM 사용자를 생성하는 대신, 임시 보안 인증 정보를 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증 정보가 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체](#)를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 컬렉션을 지정하는 보안 인증입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증을 가지고 있지만, 역할은 임시 자격 증명만 제공합니다. 자세한 정보는 IAM 사용 설명서의 [IAM 사용자를 만들어야 하는 경우\(역할이 아님\)](#)를 참조하세요.

IAM 역할

[IAM 역할](#)은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console [전환하여](#) 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용 설명서의 [IAM 역할 사용](#)을 참조하세요.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션형 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션형 ID가 인증되면 이 ID는 역할과 연결되며 역할에 의해 정의된 권한이

부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 [타사 자격 증명 공급자의 역할 만들기](#) 부분을 참조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 아이덴티티가 액세스할 수 있는 항목을 통제하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [권한 세트](#)를 참조하세요.

- **임시 IAM 사용자 권한** - IAM 사용자 또는 역할은 IAM 역할을 맡아 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.
- **크로스 계정 액세스** - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스 경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 크로스 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 [IAM 역할과 리소스 기반 정책의 차이](#)를 참조하세요.
- **서비스 간 액세스** — 일부는 다른 AWS 서비스 서비스의 기능을 AWS 서비스 사용합니다. 예를 들어 서비스에서 직접적으로 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이를 수행할 수 있습니다.
- **순방향 액세스 세션 (FAS)** — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스 서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하십시오.
- **서비스 역할** - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [역할을 생성하여 AWS 서비스에게 권한 위임](#)을 참조하세요.
- **서비스 연결 역할** — 서비스 연결 역할은 연결된 서비스 역할의 한 유형입니다. AWS 서비스 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수입할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- **Amazon EC2에서 실행되는 애플리케이션** — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인

증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여](#)를 참조하세요.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 [IAM 사용 설명서](#)의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하세요.

정책을 사용한 액세스 관리

정책을 생성하고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되는지 또는 거부되는지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 [JSON 정책 개요](#)를 참조하세요.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수입할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, iam:GetRole 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

보안 인증 기반 정책

보안 인증 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 보안 인증에 연결할 수 있는 JSON 권한 정책 설명서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하세요.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정 관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책과 인라인 정책의 선택](#)을 참조하십시오.

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 타입에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔티티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 엔티티에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 엔티티의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. Principal 필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 [IAM 엔티티에 대한 권한 경계](#)를 참조하세요.
- 서비스 제어 정책 (SCP) - SCP는 조직 또는 조직 단위 (OU) 에 대한 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations 사업체가 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직에서 모든 기능을 활성화할 경우 서비스 통제 정책 (SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 구성원 계정의 엔티티 (각 엔티티 포함) 에 대한 권한을 제한합니다. AWS 계정 루트 사용자 조직 및 SCP에 대한 자세한 정보는 AWS Organizations 사용 설명서의 [SCP 작동 방식](#)을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 [세션 정책](#)을 참조하세요.

여러 정책 타입

여러 정책 타입이 요청에 적용되는 경우 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련되어 있을 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하십시오.

IAM의 AWS IoT Analytics 작동 방식

IAM을 사용하여 액세스를 AWS IoT Analytics 관리하기 전에 먼저 사용할 수 있는 IAM 기능이 무엇인지 이해해야 합니다. AWS IoT Analytics 기타 AWS 서비스가 AWS IoT Analytics IAM과 연동되는 방식을 자세히 알아보려면 IAM 사용 설명서에서 [IAM과 연동되는 AWS 서비스를](#) 참조하십시오.

이 페이지의 주제:

- [AWS IoT Analytics ID 기반 정책](#)

- [AWS IoT Analytics 리소스 기반 정책](#)
- [태그에 따른 권한 부여 AWS IoT Analytics](#)
- [AWS IoT Analytics IAM 역할](#)

AWS IoT Analytics ID 기반 정책

IAM 자격 증명 기반 정책을 사용하면 허용 또는 거부된 작업 및 리소스와 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. AWS IoT Analytics 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하십시오.

작업

IAM 자격 증명 기반 정책의 Action 요소는 정책에 따라 허용되거나 거부되는 특정 작업에 대해 설명합니다. 정책 액션은 일반적으로 관련 AWS API 오퍼레이션과 이름이 같습니다. 이 작업은 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에서 사용됩니다.

정책 작업은 작업 앞에 다음 접두사를 AWS IoT Analytics 사용합니다. `iotanalytics:` 예를 들어 AWS IoT Analytics `CreateChannel` API 작업으로 AWS IoT Analytics 채널을 만들 수 있는 권한을 누군가에게 부여하려면 해당 작업을 정책에 포함해야 합니다. `iotanalytics:BatchPutMessage` 정책 설명에는 Action OR NotAction 요소가 포함되어야 합니다. AWS IoT Analytics 이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 집합을 정의합니다.

명령문 하나에 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
  "iotanalytics:action1",
  "iotanalytics:action2"
]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `Describe`라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "iotanalytics:Describe*"
```

AWS IoT Analytics 작업 목록을 보려면 IAM 사용 AWS IoT Analytics 설명서에 [정의된 작업을](#) 참조하십시오.

리소스

Resource 요소는 작업이 적용되는 객체를 지정합니다. 명령문에는 Resource 또는 NotResource 요소가 반드시 추가되어야 합니다. ARN을 사용하거나 문이 모든 리소스에 적용됨을 표시하는 와일드카드(*)를 사용하여 리소스를 지정합니다.

AWS IoT Analytics 데이터세트 리소스의 ARN은 다음과 같습니다.

```
arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/${DatasetName}
```

ARN 형식에 대한 자세한 내용은 [Amazon 리소스 이름\(ARN\) 및 AWS 서비스 네임스페이스](#) 단원을 참조하십시오.

예를 들어, 명령문에 Foobar 데이터 세트를 지정하려면 다음 ARN을 사용합니다.

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/Foobar"
```

특정 계정에 속하는 모든 인스턴스를 지정하려면 와일드카드(*)를 사용합니다.

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/*"
```

리소스 생성 작업과 같은 일부 AWS IoT Analytics 작업은 특정 리소스에서 수행할 수 없습니다. 이러한 경우, 와일드카드(*)를 사용해야 합니다.

```
"Resource": "*"
```

일부 AWS IoT Analytics API 작업에는 여러 리소스가 포함됩니다. 예를 들어 CreatePipeline은 채널과 데이터 세트를 참조하므로 사용자에게 해당 채널과 데이터 세트를 사용할 수 있는 권한이 있어야 합니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

```
"Resource": [
  "resource1",
  "resource2"
]
```

AWS IoT Analytics 리소스 유형 및 ARN 목록을 보려면 IAM 사용 AWS IoT Analytics 설명서에 [정의된 리소스](#)를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 [AWS IoT Analytics가 정의한 작업](#)을 참조하십시오.

조건 키

Condition 요소(또는 Condition블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같음, 미만 등 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 빌드할 수 있습니다.

한 문에서 여러 Condition 요소를 지정하거나 단일 Condition 요소에서 여러 키를 지정하는 경우 AWS 은(는) 논리적 AND 작업을 사용하여 평가합니다. 단일 조건 키의 여러 값을 지정하는 경우 AWS 은(는) 논리적 OR 작업을 사용하여 조건을 평가합니다. 명령문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리표시자 변수를 사용할 수도 있습니다. 예를 들어, 사용자에게 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 정책 요소: 변수 및 태그](#) 섹션을 참조하십시오.

AWS IoT Analytics 서비스별 조건 키는 제공하지 않지만 일부 글로벌 조건 키 사용은 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 [AWS 설명서의 글로벌 조건 컨텍스트 키](#)를 참조하십시오.

예

AWS IoT Analytics 자격 증명 기반 정책의 예를 보려면 [AWS IoT Analytics ID 기반 정책 예제](#)를 참조하십시오.

AWS IoT Analytics 리소스 기반 정책

AWS IoT Analytics 리소스 기반 정책을 지원하지 않습니다. 자세한 리소스 기반 정책 페이지의 예를 보려면 AWS Lambda 개발자 안내서에서 [AWS Lambda](#)에 대해 리소스 기반 정책 사용하기를 참조하십시오.

태그에 따른 권한 부여 AWS IoT Analytics

AWS IoT Analytics 리소스에 태그를 첨부하거나 요청에 태그를 전달할 수 있습니다. 태그를 기반으로 액세스를 제어하려면 `iotanalytics:ResourceTag/{key-name}`, `aws:RequestTag/{key-name}` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다. 리소스 태깅에 대한 자세한 내용은 AWS IoT Analytics 리소스 [태그 지정을 참조하십시오](#). [AWS IoT Analytics](#)

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 ID 기반 정책의 예를 보려면 태그 기반 채널 [보기를 AWS IoT Analytics](#) 참조하세요.

AWS IoT Analytics IAM 역할

[IAM 역할](#)은 특정 권한을 가지고 있는 AWS 계정 내 엔터티입니다.

임시 자격 증명 사용 AWS IoT Analytics

임시 보안 인증을 사용하여 페더레이션을 통해 로그인하거나, IAM 역할을 맡거나, 교차 계정 역할을 맡을 수 있습니다. [AssumeRole](#) 또는 [GetFederationToken](#)와 같은 AWS Security Token Service (AWS STS) API 작업을 호출하여 임시 보안 자격 증명을 얻습니다.

AWS IoT Analytics 임시 자격 증명 사용을 지원하지 않습니다.

서비스 링크 역할

[서비스 라인 역할](#)을 사용하면 AWS 서비스가 다른 서비스의 리소스에 액세스하여 사용자를 대신하여 작업을 완료할 수 있습니다. 서비스 연결 역할은 IAM 계정에 나타나며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수 없습니다.

AWS IoT Analytics 서비스 연결 역할을 지원하지 않습니다.

Service roles(서비스 역할)

이 기능을 사용하면 서비스가 사용자를 대신하여 [서비스 역할](#)을 수임할 수 있습니다. 이 역할을 사용하면 서비스가 다른 서비스의 리소스에 액세스해 사용자를 대신해 작업을 완료할 수 있습니다. 서비스 역할은 IAM 계정에 나타나고, 해당 계정이 소유합니다. 즉, IAM 관리자가 이 역할에 대한 권한을 변경할 수 있습니다. 그러나 권한을 변경하면 서비스의 기능이 손상될 수 있습니다.

AWS IoT Analytics 서비스 역할을 지원합니다.

교차 서비스 혼동된 대리자 예방

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. AWS에서는 교차 서비스 가장으로 인해 혼동된 대리자 문제가 발생할 수 있습니다. 교차 서비스 가장은 한 서비스(직접적으로 호출하는 서비스)가 다른 서비스(직접적으로 호출되는 서비스)를 직접적으로 호출할 때 발생할 수 있습니다. 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

리소스 정책에는 [aws:SourceArn](#) 및 [aws:SourceAccount](#) 전역 조건 컨텍스트 키를 사용하는 것이 좋습니다. 이렇게 하면 AWS IoT Analytics가 리소스에 대해 다른 서비스에 부여하는 권한을 제한합니

다. 두 글로벌 조건 컨텍스트 키를 모두 사용하는 경우 `aws:SourceAccount` 값과 `aws:SourceArn` 값의 계정은 동일한 정책 문에서 사용할 경우 동일한 계정 ID를 사용해야 합니다.

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 Amazon 리소스 이름(ARN)이 포함된 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용하는 것입니다. 리소스의 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드(*)를 포함한 `aws:SourceArn` 글로벌 조건 컨텍스트 키를 사용합니다. 예: `arn:aws:iotanalytics::123456789012:*`.

주제

- [Amazon S3 버킷에 대한 예방](#)
- [Amazon CloudWatch Logs를 통한 예방](#)
- [고객 관리형 AWS IoT Analytics 리소스에 대한 혼동된 대리자 예방 조치](#)

Amazon S3 버킷에 대한 예방

AWS IoT Analytics 데이터 스토어에 고객 관리형 Amazon S3 스토리지를 사용하는 경우, 데이터를 저장하는 Amazon S3 버킷은 혼동된 대리자 문제에 노출될 수 있습니다.

`## ## Nikki Wolf# DOC-EXAMPLE-BUCKET`이라는 고객 소유의 Amazon S3 버킷을 사용합니다. 버킷에는 `us-east-1` 리전에서 생성된 AWS IoT Analytics 데이터 스토어에 대한 정보가 저장됩니다. 그녀는 AWS IoT Analytics 서비스 주체가 자신을 대신하여 `DOC-EXAMPLE-BUCKET`을 쿼리할 수 있도록 하는 정책을 지정합니다. Nikki의 동료인 Li Juan은 자신의 계정에서 `DOC-EXAMPLE-BUCKET`을 쿼리하고 결과가 포함된 데이터 세트를 만듭니다. 그 결과 Li가 자신의 계정에서 쿼리를 실행했음에도 불구하고 AWS IoT Analytics 서비스 주체가 Li를 대신하여 Nikki의 Amazon S3 버킷을 쿼리했습니다.

이를 방지하기 위해 Nikki는 `DOC-EXAMPLE-BUCKET` 정책에 `aws:SourceAccount` 조건 또는 `aws:SourceArn` 조건을 지정할 수 있습니다.

`aws:SourceAccount` 조건 지정 - `## ## ## ### Nikki ##(123456789012)# AWS IoT Analytics ##### DOC-EXAMPLE-BUCKET`에 액세스할 수 있도록 지정합니다.

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
```



```

    "Principal": {
      "Service": "iotanalytics.amazonaws.com"
    },
    "Action": [
      "s3:GetBucketLocation",
      "s3:GetObject",
      "s3:ListBucket",
      "s3:ListBucketMultipartUploads",
      "s3:ListMultipartUploadParts",
      "s3:AbortMultipartUpload",
      "s3:PutObject",
      "s3>DeleteObject"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
]
}

```

aws:SourceArn 조건 지정 - 또는 Nikki가 `aws:SourceArn` 조건을 사용할 수도 있습니다.

```

{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",

```

```

        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:PutObject",
        "s3>DeleteObject"
    ],
    "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ],
    "Condition": {
        "ArnLike": {
            "aws:SourceArn": [
                "arn:aws:iotanalytics:us-east-1:123456789012:dataset/DOC-EXAMPLE-DATASET",
                "arn:aws:iotanalytics:us-east-1:123456789012:datastore/DOC-EXAMPLE-DATASTORE"
            ]
        }
    }
}

```

Amazon CloudWatch Logs를 통한 예방

Amazon CloudWatch Logs를 사용하여 모니터링하는 동안 혼동된 대리자 문제를 방지할 수 있습니다. 다음과 같은 리소스 정책은 다음 항목으로 혼동된 대리자 문제를 방지하는 방법을 보여줍니다.

- 전역 조건 컨텍스트 키, `aws:SourceArn`
- AWS 계정 ID가 있는 `aws:SourceAccount`
- AWS IoT Analytics의 `sts:AssumeRole` 요청과 관련된 고객 리소스

다음 예제에서는 `123456789012`를 AWS 계정 ID로 바꾸고, `us-east-1`을 AWS IoT Analytics 계정의 리전으로 바꿉니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {

```

```

        "Service": "iotanalytics.amazonaws.com"
    },
    "Action": "logs:PutLogEvents",
    "Resource": "*",
    "Condition":{
        "ArnLike":{
            "aws:SourceArn":"arn:aws:iotanalytics:us-east-1:123456789012:*/*"
        },
        "StringEquals":{
            "aws:SourceAccount":"123456789012"
        }
    }
}
]
}

```

Amazon CloudWatch Logs 활성화 및 구성에 대한 자세한 내용은 [the section called “로깅 및 모니터링”](#)을 참조하세요.

고객 관리형 AWS IoT Analytics 리소스에 대한 혼동된 대리자 예방 조치

AWS IoT Analytics 리소스에 대한 작업을 수행할 수 있는 AWS IoT Analytics 권한을 부여하면, 리소스가 혼동된 대리자 문제에 노출될 수 있습니다. 혼동된 대리자 문제를 방지하기 위해 다음 예제 리소스 정책을 사용하여 AWS IoT Analytics에 부여되는 권한을 제한할 수 있습니다.

주제

- [AWS IoT Analytics 채널 및 데이터 스토어 방지](#)
- [AWS IoT Analytics 데이터 세트 콘텐츠 전송 규칙에 대한 크로스 서비스 혼동된 대리자 예방](#)

AWS IoT Analytics 채널 및 데이터 스토어 방지

IAM 역할을 사용하여 AWS IoT Analytics가 사용자를 대신하여 액세스할 수 있는 AWS 리소스를 제어합니다. 혼동된 대리자 문제에 역할이 노출되는 것을 방지하려면 `aws:SourceAccount` 요소에 AWS 계정을, 역할에 연결하는 신뢰 정책의 `aws:SourceArn` 요소에 AWS IoT Analytics 리소스의 ARN을 지정할 수 있습니다.

다음 예제에서는 `123456789012`를 AWS 계정 ID로 바꾸고 `arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-EXAMPLE-CHANNEL`을 AWS IoT Analytics 채널 또는 데이터 스토어의 ARN로 바꿉니다.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotanalytics.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-EXAMPLE-CHANNEL"
      }
    }
  }
]
}

```

채널 및 데이터 스토어의 고객 관리형 S3 스토리지 옵션에 대한 자세한 내용은 AWS IoT Analytics API 참조의 [CustomerManagedChannelS3Storage](#) 및 [CustomerManagedDatastoreS3Storage](#) 항목을 참조하세요.

AWS IoT Analytics 데이터 세트 콘텐츠 전송 규칙에 대한 크로스 서비스 혼동된 대리자 예방

AWS IoT Analytics가 Amazon S3 또는 AWS IoT Events에 데이터 세트 쿼리 결과를 전달하는 IAM 역할은 혼동된 대리자 문제에 노출될 수 있습니다. 혼동된 대리자 문제를 방지하기 위해 `aws:SourceAccount` 요소에 AWS 계정을, 역할에 연결하는 신뢰 정책의 `aws:SourceArn` 요소에 AWS IoT Analytics 리소스의 ARN을 지정합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExampleTrustPolicyDocument",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": "sts:AssumeRole",

```

```

    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:dataset/DOC-EXAMPLE-DATASET"
      }
    }
  }
]
}

```

데이터 세트 콘텐츠 전송 규칙을 구성하는 방법에 대한 자세한 내용은 AWS IoT Analytics API 참조의 [contentDeliveryRules](#) 항목을 참조하세요.

AWS IoT Analytics ID 기반 정책 예제

기본적으로 사용자 및 역할에는 AWS IoT Analytics 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console AWS CLI, 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 사용자와 역할에게 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 사용자 또는 그룹에 이러한 정책을 연결해야 합니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용자 설명서의 [JSON 탭에서 정책 생성](#)을 참조하십시오.

이 페이지의 주제:

- [정책 모범 사례](#)
- [콘솔 사용 AWS IoT Analytics](#)
- [사용자가 자신의 고유한 권한을 볼 수 있도록 허용](#)
- [단일 AWS IoT Analytics 입력에 액세스](#)
- [태그를 기반으로 AWS IoT Analytics 채널 보기](#)

정책 모범 사례

자격 증명 기반 정책은 매우 강력합니다. 계정에서 다른 사람이 AWS IoT Analytics 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. 자격 증명 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책 사용 시작하기 - 관리형 정책을 AWS IoT Analytics 빠르게 사용하려면 AWS 관리형 정책을 사용하여 직원에게 필요한 권한을 부여하세요. 이 정책은 이미 계정에서 사용할 수 있으며 AWS에 의해 유지 관리 및 업데이트됩니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책으로 권한 사용 시작하기](#)를 참조하십시오.
- 최소 권한 부여 - 사용자 지정 정책을 생성할 때 오직 작업을 수행하는 데 필요한 권한만을 부여합니다. 최소한의 권한 조합으로 시작하여 필요에 따라 추가 권한을 부여합니다. 처음부터 권한을 많이 부여한 후 나중에 줄이는 방법보다 이 방법이 안전합니다. 자세한 정보는 IAM 사용 설명서의 [최소 권한 부여](#)를 참조하십시오.
- 중요한 작업에 대해 MFA 활성화 - 보안을 강화하기 위해 사용자가 중요한 리소스 또는 API 작업에 액세스하려면 다중 인증(MFA)을 사용해야 합니다. 자세한 정보는 [IAM 사용 설명서](#)의 AWS에서 다중 인증(MFA) 사용을 참조하십시오.
- 보안 강화를 위해 정책 조건 사용 - 실제로 가능한 경우 자격 증명 기반 정책이 리소스에 대한 액세스를 허용하는 조건을 정의합니다. 예를 들어 요청을 할 수 있는 IP 주소의 범위를 지정하도록 조건을 작성할 수 있습니다. 지정된 날짜 또는 시간 범위 내에서만 요청을 허용하거나, SSL 또는 MFA를 사용해야 하는 조건을 작성할 수도 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하십시오.

콘솔 사용 AWS IoT Analytics

AWS IoT Analytics 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한을 통해 내 AWS IoT Analytics 리소스의 세부 정보를 나열하고 볼 수 있어야 AWS 계정입니다. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

해당 엔티티가 AWS IoT Analytics 콘솔을 계속 사용할 수 있도록 하려면 다음 AWS 관리형 정책도 엔티티에 연결하십시오. 자세한 내용은 IAM 사용 설명서의 [사용자에게 권한 추가](#)를 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:BatchPutMessage",
        "iotanalytics:CancelPipelineReprocessing",
        "iotanalytics:CreateChannel",
        "iotanalytics:CreateDataset",
        "iotanalytics:CreateDatasetContent",
```

```

        "iotanalytics:CreateDatastore",
        "iotanalytics:CreatePipeline",
        "iotanalytics>DeleteChannel",
        "iotanalytics>DeleteDataset",
        "iotanalytics>DeleteDatasetContent",
        "iotanalytics>DeleteDatastore",
        "iotanalytics>DeletePipeline",
        "iotanalytics:DescribeChannel",
        "iotanalytics:DescribeDataset",
        "iotanalytics:DescribeDatastore",
        "iotanalytics:DescribeLoggingOptions",
        "iotanalytics:DescribePipeline",
        "iotanalytics:GetDatasetContent",
        "iotanalytics:ListChannels",
        "iotanalytics:ListDatasetContents",
        "iotanalytics:ListDatasets",
        "iotanalytics:ListDatastores",
        "iotanalytics:ListPipelines",
        "iotanalytics:ListTagsForResource",
        "iotanalytics:PutLoggingOptions",
        "iotanalytics:RunPipelineActivity",
        "iotanalytics:SampleChannelData",
        "iotanalytics:StartPipelineReprocessing",
        "iotanalytics:TagResource",
        "iotanalytics:UntagResource",
        "iotanalytics:UpdateChannel",
        "iotanalytics:UpdateDataset",
        "iotanalytics:UpdateDatastore",
        "iotanalytics:UpdatePipeline"
    ],
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:channel/
${channelName}",
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/
${datasetName}",
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:datastore/
${datastoreName}",
    "Resource": "arn:${Partition}:iotanalytics:${Region}:${Account}:pipeline/
${pipelineName}"
    }
]
}

```

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여 줍니다. 이 정책에는 콘솔에서 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": [
        "arn:aws:iam::*:user/${aws:username}"
      ]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```


단일 AWS IoT Analytics 입력에 액세스

이 예시에서는 사용자에게 AWS IoT Analytics 채널 중 하나에 AWS 계정 대한 액세스 권한을 부여하려고 `exampleChannel` 합니다. 또한 사용자가 채널을 추가, 업데이트 및 삭제하도록 허용하려고 합니다.

정책은 사용자에게 `iotanalytics:ListChannels`, `iotanalytics:DescribeChannel`, `iotanalytics:CreateChannel`, `iotanalytics>DeleteChannel`, and `iotanalytics:UpdateChannel` 권한을 부여합니다. 사용자에게 권한을 부여하고 콘솔을 사용하여 테스트하는 Amazon S3 서비스의 안내 예제를 보려면 [안내 예제: 사용자 정책을 통해 버킷에 대한 액세스 제어](#)를 참조하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListChannelsInConsole",
      "Effect": "Allow",
      "Action": [
        "iotanalytics:ListChannels"
      ],
      "Resource": "arn:aws:iotanalytics:::*"
    },
    {
      "Sid": "ViewSpecificChannelInfo",
      "Effect": "Allow",
      "Action": [
        "iotanalytics:DescribeChannel"
      ],
      "Resource": "arn:aws:iotanalytics:::exampleChannel"
    },
    {
      "Sid": "ManageChannels",
      "Effect": "Allow",
      "Action": [
        "iotanalytics:CreateChannel",
        "iotanalytics>DeleteChannel",
        "iotanalytics:DescribeChannel",
        "iotanalytics:ListChannels",
        "iotanalytics:UpdateChannel"
      ],
      "Resource": "arn:aws:iotanalytics:::exampleChannel/*"
    }
  ]
}
```

```

    }
  ]
}

```

태그를 기반으로 AWS IoT Analytics 채널 보기

ID 기반 정책의 조건을 사용하여 태그를 기반으로 AWS IoT Analytics 리소스에 대한 액세스를 제어할 수 있습니다. 이 예제에서는 channel 보기를 허용하는 정책을 생성할 수 있는 방법을 보여 줍니다. 하지만 channel 태그 Owner가 해당 사용자의 사용자 이름 값을 가지고 있는 경우에만 권한이 부여됩니다. 이 정책은 콘솔에서 이 작업을 완료하는 데 필요한 권한도 부여합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListChannelsInConsole",
      "Effect": "Allow",
      "Action": "iotanalytics:ListChannels",
      "Resource": "*"
    },
    {
      "Sid": "ViewChannelsIfOwner",
      "Effect": "Allow",
      "Action": "iotanalytics:ListChannels",
      "Resource": "arn:aws:iotanalytics:*:*:channel/*",
      "Condition": {
        "StringEquals": {"iotanalytics:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}

```

이 정책을 계정의 사용자에게 연결할 수 있습니다. 라는 사용자가 보기를 richard-roe 시도하는 경우 태그를 AWS IoT Analytics channel channel 지정해야 합니다. Owner=richard-roe or owner=richard-roe 그렇지 않으면 액세스가 거부됩니다. 조건 키 이름은 대소문자를 구분하지 않기 때문에 조건 태그 키 Owner는 Owner 및 owner 모두와 일치합니다. 자세한 정보는 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하십시오.

AWS IoT Analytics ID 및 액세스 문제 해결

다음 정보를 사용하면 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 해결하는 데 도움이 AWS IoT Analytics 됩니다.

주제

- [에서 작업을 수행할 권한이 없습니다. AWS IoT Analytics](#)
- [iam:PassRole을 수행할 권한이 없음](#)
- [외부 사용자가 내 AWS IoT Analytics 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.](#)

에서 작업을 수행할 권한이 없습니다. AWS IoT Analytics

작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 도움을 요청해야 합니다. 관리자는 사용자 이름과 암호를 제공한 사람입니다.

다음 예제 오류는 mateojackson 용자가 콘솔을 사용하여 channel에 대한 세부 정보를 보려고 하지만 iotanalytics:ListChannels 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotanalytics:``ListChannels`` on resource: ``my-example-channel``
```

이 경우 Mateo는 iotanalytics:ListChannel 작업을 사용하여 my-example-channel 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

iam:PassRole을 수행할 권한이 없음

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS IoT Analytics에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 만드는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 AWS IoT Analytics에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우 Mary가 iam:PassRole 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 보안 인증을 제공한 사용자입니다.

외부 사용자가 내 AWS IoT Analytics 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 ACL(액세스 제어 목록)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 이러한 기능의 AWS IoT Analytics 지원 여부를 알아보려면 [IAM의 AWS IoT Analytics 작동 방식을](#) 참조하십시오.
- 소유한 리소스에 대한 액세스를 제공하는 방법을 알아보려면 IAM 사용 설명서의 [다른 AWS 계정 IAM 사용자에게 액세스 권한 제공](#)을 참조하십시오. AWS 계정
- [제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 AWS 계정 소유에 대한 액세스 제공](#)을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [외부에서 인증된 사용자에게 액세스 권한 제공\(ID 페더레이션\)](#)을 참조하세요.
- 크로스 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 [IAM 역할과 리소스 기반 정책의 차이](#)를 참조하세요.

AWS IoT Analytics의 로깅 및 모니터링

AWS는 AWS IoT Analytics를 모니터링하는 데 사용할 수 있는 도구를 제공합니다. 이러한 도구 중 일부를 구성하여 모니터링을 수행할 수 있습니다. 일부 도구는 수동 개입이 필요합니다. 모니터링 작업은 최대한 자동화하는 것이 좋습니다.

자동 모니터링 도구

다음과 같은 자동 모니터링 도구를 사용하여 AWS IoT를 관찰하고 문제 발생 시 보고할 수 있습니다.

- Amazon CloudWatch Logs – AWS CloudTrail 또는 기타 소스의 로그 파일을 모니터링, 저장 및 액세스합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 [AWS CloudTrail 로그 파일 모니터링](#)은 무엇입니까를 참조하세요.

- AWS CloudTrail 로그 모니터링 – 계정 간에 로그 파일을 공유하고, CloudTrail 로그 파일을 CloudWatch Logs에 전송하여 실시간으로 모니터링하며, Java에서 로그 처리 애플리케이션을 작성하고, CloudTrail에서 전송한 후 로그 파일이 변경되지 않았는지 확인합니다. 자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail 로그 파일 작업](#)을 참조하세요.

수동 모니터링 도구

AWS IoT 모니터링의 또 한 가지 중요한 부분은 CloudWatch 경보에 포함되지 않는 항목을 수동으로 모니터링해야 한다는 점입니다. AWS IoT, CloudWatch, 및 기타 AWS 서비스 콘솔 대시보드에서는 AWS 환경의 상태를 한 눈에 볼 수 있습니다. 또한 AWS IoT Analytics에서 로그 파일을 확인하는 것이 좋습니다.

- AWS IoT Analytics 콘솔에는 다음이 표시됩니다.
 - 채널
 - 파이프라인
 - 데이터 스토어
 - 데이터 세트
 - 노트북
 - 설정
 - 자세히 알아보기
- CloudWatch 홈 페이지에는 다음 내용이 표시됩니다.
 - 현재 경보 및 상태
 - 경보 및 리소스 그래프
 - 서비스 상태

또한 CloudWatch를 사용하여 다음을 수행할 수 있습니다.

- [사용자 지정 대시보드](#)를 생성하여 관심 있는 서비스 모니터링
- 지표 데이터를 그래프로 작성하여 문제를 해결하고 추세 파악
- 모든 AWS 리소스 지표 검색 및 찾아보기
- 문제에 대해 알려주는 경보 생성 및 편집

Amazon CloudWatch Logs를 사용한 모니터링

AWS IoT Analytics는 Amazon CloudWatch를 사용한 로깅을 지원합니다. [PutLoggingOptions API 작업을 사용하여 AWS IoT Analytics에 대한 Amazon CloudWatch 로깅을 활성화하고 구성할 수 있습니다.](#) 이 섹션에서는 AWS Identity and Access Management(IAM)으로 PutLoggingOptions을 사용하여 AWS IoT Analytics에 대한 Amazon CloudWatch 로깅을 구성하고 활성화하는 방법을 설명합니다.

CloudWatch Logs에 대한 자세한 내용은 [Amazon CloudWatch Logs User Guide](#)를 참조하세요. AWS IAM에 대한 자세한 내용은 [AWS Identity and Access Management 사용 설명서](#)를 참조하세요.

Note

AWS IoT Analytics 로깅을 활성화하기 전에 CloudWatch Logs 액세스 권한을 이해해야 합니다. CloudWatch 로그 액세스 권한이 있는 사용자는 디버깅 정보를 볼 수 있습니다. 자세한 내용은 [Amazon CloudWatch Logs에 대한 인증 및 액세스 제어](#) 단원을 참조하세요.

IAM 역할을 생성하여 로깅을 활성화합니다.

Amazon CloudWatch에 대한 로깅을 활성화하기 위한 IAM 역할을 만들려면

1. [AWS IAM 콘솔](#) 또는 다음 AWS IAM CLI 명령인 [CreateRole](#)을 사용하여 신뢰 관계 정책(신뢰 정책)으로 새 IAM 역할을 생성합니다. 신뢰 정책은 Amazon CloudWatch와 같은 엔터티에 해당 역할을 수임할 권한을 부여합니다.

```
aws iam create-role --role-name exampleRoleName --assume-role-policy-document
exampleTrustPolicy.json
```

exampleTrustPolicy.json 파일에는 다음 콘텐츠가 포함되어 있습니다.

Note

이 예에는 혼동된 대리인 보안 문제로부터 보호하기 위한 전역 조건 컨텍스트 키가 포함되어 있습니다. **123456789012**를 AWS 계정 ID로 바꾸고, **aws-region**은 AWS 리소스의 AWS 리전으로 바꿉니다. 자세한 내용은 [the section called “교차 서비스 혼동된 대리자 예 방”](#) 섹션을 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:aws-region:123456789012:*"
        }
      }
    }
  ]
}
```

나중에 AWS IoT Analytics PutLoggingOptions 명령을 호출할 때 이 역할의 ARN을 사용합니다.

2. AWS IAM [PutRolePolicy](#)를 사용하여 1단계에서 생성한 역할에 권한 정책(role policy)을 연결합니다.

```
aws iam put-role-policy --role-name exampleRoleName --policy-name
examplePolicyName --policy-document exampleRolePolicy.json
```

exampleRolePolicy.json 파일에는 다음과 같은 내용이 포함되어 있습니다.

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

```

    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
}

```

3. Amazon CloudWatch에 로깅 이벤트를 추가할 수 있는 AWS IoT Analytics 권한을 부여하려면 Amazon CloudWatch 명령 [PutResourcePolicy](#)를 사용합니다.

Note

혼동된 대리자 보안 문제를 방지하려면 리소스 정책에 `aws:SourceArn`을 지정하는 것이 좋습니다. 이렇게 하면 지정된 계정에서 오는 요청만 허용하도록 액세스가 제한됩니다. 혼동된 대리자 문제에 관한 자세한 내용은 [the section called “교차 서비스 혼동된 대리자 예방”](#)를 참조하세요.

```
aws logs put-resource-policy --policy-in-json
exampleResourcePolicy.json
```

`exampleResourcePolicy.json` 파일에는 다음 리소스 정책이 포함됩니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotanalytics.amazonaws.com"
      },
      "Action": "logs:PutLogEvents",
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iotanalytics:us-east-1:123456789012:*/*"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}

```



```

    ]
  }
}

```

로깅 구성 및 활성화

PutLoggingOptions 명령을 사용하여 AWS IoT Analytics에 대한 Amazon CloudWatch 로깅을 구성 및 활성화합니다. loggingOptions 필드의 roleArn은 이전 섹션에서 생성이 완료된 역할의 ARN이어야 합니다. 또한 DescribeLoggingOptions 명령을 사용하여 로깅 옵션 설정을 점검할 수 있습니다.

PutLoggingOptions

AWS IoT Analytics 로깅 옵션을 설정하거나 업데이트합니다. loggingOptions 필드 값을 업데이트한 경우 변경 사항이 적용되기까지 최대 1분이 소요될 수 있습니다. 또한 roleArn 필드에서 지정한 역할에 연결된 정책을 변경하는 경우(예: 잘못된 정책 수정), 변경 사항이 적용되기까지 최대 5분이 소요될 수 있습니다. 자세한 내용은 [PutLoggingOptions](#) 섹션을 참조하세요.

DescribeLoggingOptions

AWS IoT Analytics 로깅 옵션의 현재 설정을 가져옵니다. 자세한 정보는 [DescribeLoggingOptions](#) 섹션을 참조하세요.

네임스페이스, 지표 및 측정기준

AWS IoT Analytics는 다음 지표들을 Amazon CloudWatch 리포지토리에 제공합니다.

네임스페이스

AWS/IoTAnalytics

지표	설명
ActionExecution	실행된 작업의 수입입니다.
ActionExecutionThrottled	제한된 작업 수.
ActivityExecutionError	파이프라인 활동을 실행하는 동안 생성된 오류의 수입입니다.

지표	설명
IncomingMessages	채널로 들어오는 메시지의 수입니다.
PipelineConcurrentExecutionCount	동시에 실행된 파이프라인 활동 수.

차원	설명
ActionType	모니터링하는 작업의 유형입니다.
ChannelName	모니터링하는 채널의 이름입니다.
DatasetName	모니터링하는 데이터 세트의 이름입니다.
DatastoreName	모니터링하는 데이터 스토어의 이름입니다.
PipelineActivityName	모니터링하는 파이프라인 활동의 이름입니다.
PipelineActivityType	모니터링하는 파이프라인 활동의 유형입니다.
PipelineName	모니터링하는 파이프라인의 이름입니다.

Amazon CloudWatch Events를 사용하여 모니터링

AWS IoT Analytics는 AWS Lambda 활동 중에 런타임 오류가 발생하면 Amazon CloudWatch Events에 이벤트를 자동으로 게시합니다. 이 이벤트에는 자세한 오류 메시지와 처리되지 않은 채널 메시지를 저장하는 Amazon Simple Storage Service(S3) 객체의 키가 포함되어 있습니다. Amazon S3 키를 사용하여 처리되지 않은 채널 메시지를 재처리할 수 있습니다. 자세한 내용은 [채널 메시지 재처리](#), AWS IoT Analytics API 참조의 [StartPipelineReprocessing](#) API, Amazon CloudWatch Events 사용 설명서의 [Amazon CloudWatch Events는 무엇입니까?](#)를 참조하세요.

또한 Amazon CloudWatch Events가 알림을 보내거나 추가 조치를 취할 수 있도록 대상을 구성할 수 있습니다. 예를 들어 Amazon Simple Queue Service(Amazon SQS) 대기열에 알림을 보낸 다음 StartReprocessingMessage API를 호출하여 Amazon S3 객체에 저장된 채널 메시지를 처리할 수 있습니다. Amazon CloudWatch Events는 다음과 같은 다양한 유형의 대상을 지원합니다.

- Amazon Kinesis Streams
- AWS Lambda 함수

- Amazon Simple Notification Service(Amazon SNS) 주제
- Amazon Simple Queue Service(Amazon SQS) 대기열

지원되는 대상에 대한 자세한 정보는 Amazon EventBridge 사용 설명서의 [Amazon EventBridge 대상](#)을 참조하세요.

CloudWatch Events 리소스와 관련 대상은 AWS IoT Analytics 리소스를 생성한 AWS 리전에 있어야 합니다. 자세한 내용은 AWS 일반 참조의 [서비스 엔드포인트 및 할당량](#)을 참조하세요.

AWS Lambda 활동의 런타임 오류에 대해 Amazon CloudWatch Events로 전송되는 알림은 다음 형식을 사용합니다.

```
{
  "version": "version-id",
  "id": "event-id",
  "detail-type": "IoT Analytics Pipeline Failure Notification",
  "source": "aws.iotanalytics",
  "account": "aws-account",
  "time": "timestamp",
  "region": "aws-region",
  "resources": [
    "pipeline-arn"
  ],
  "detail": {
    "event-detail-version": "1.0",
    "pipeline-name": "pipeline-name",
    "error-code": "LAMBDA_FAILURE",
    "message": "error-message",
    "channel-messages": {
      "s3paths": [
        "s3-keys"
      ]
    },
    "activity-name": "lambda-activity-name",
    "lambda-function-arn": "lambda-function-arn"
  }
}
```

알림 예제:

```
{
  "version": "0",
```

```

    "id": "204e672e-ef12-09af-4cfd-de3b53673ec6",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "123456789012",
    "time": "2020-10-15T23:47:02Z",
    "region": "ap-southeast-2",
    "resources": [
      "arn:aws:iotanalytics:ap-southeast-2:123456789012:pipeline/
test_pipeline_failure"
    ],
    "detail": {
      "event-detail-version": "1.0",
      "pipeline-name": "test_pipeline_failure",
      "error-code": "LAMBDA_FAILURE",
      "message": "Temp unavaliable",
      "channel-messages": {
        "s3paths": [
          "test_pipeline_failure/channel/cmr_channel/__dt=2020-10-15
00:00:00/1602805530000_1602805560000_123456789012_cmr_channel_0_257.0.json.gz"
        ]
      },
      "activity-name": "LambdaActivity_33",
      "lambda-function-arn": "arn:aws:lambda:ap-
southeast-2:123456789012:function:lambda_activity"
    }
  }
}

```

Amazon CloudWatch Events를 통해 지연 데이터 알림 받기

지정 기간 내의 데이터를 사용해 데이터 세트 콘텐츠를 생성할 때, 일부 데이터가 처리할 시간 내에 도착하지 않을 수도 있습니다. 지연을 허용하려면 `queryAction`(SQL 쿼리)를 적용하여 [데이터 세트를 생성할 때](#) `QueryFilter`에 `deltaTime` 오프셋을 지정할 수 있습니다. AWS IoT Analytics는 여전히 델타 시간 내에 도착하는 데이터를 처리하며, 데이터 세트 콘텐츠에는 시간 지연이 있습니다. 지연 데이터 알림 기능을 사용하면 델타 시간 이후에 데이터가 도착할 때 AWS IoT Analytics가 [Amazon CloudWatch Events](#)를 통해 알림을 보낼 수 있습니다.

AWS IoT Analytics 콘솔, [API](#), [AWS Command Line Interface\(AWS CLI\)](#) 또는 [AWSSDK](#)를 사용하여 데이터 세트에 대한 지연 데이터 규칙을 지정할 수 있습니다.

AWS IoT Analytics API에서 `LateDataRuleConfiguration` 객체는 데이터 세트의 지연 데이터 규칙 설정을 나타냅니다. 이 객체는 `CreateDataset` 및 `UpdateDataset` API 작업과 관련된 `Dataset` 객체에 포함되어 있습니다.

파라미터

AWS IoT Analytics을 사용하여 데이터 세트에 대해 지연 데이터 규칙을 생성할 때 다음 정보를 지정해야 합니다.

ruleConfiguration (LateDataRuleConfiguration)

지연 데이터 규칙의 구성 정보를 포함하는 구조입니다.

deltaTimeSessionWindowConfiguration

델타 시간 세션 기간의 구성 정보를 포함하는 구조입니다.

[DeltaTime](#)은 시간 간격을 지정합니다. DeltaTime을 사용하면 마지막 실행 이후 데이터 스토어에 도착한 데이터로 데이터 세트 콘텐츠를 생성할 수 있습니다. DeltaTime의 예제를 알아보려면 [델타 기간으로 SQL 데이터 세트 생성\(CLI\)](#)을 참조하세요.

timeoutInMinutes

시간 간격입니다. AWS IoT Analytics가 마지막 실행 이후 생성된 지연 데이터 알림을 일괄 처리할 수 있도록 timeoutInMinutes를 사용할 수 있습니다. AWS IoT Analytics는 한 번에 하나의 알림 배치를 CloudWatch Events로 전송합니다.

유형: 정수

유효 범위: 1~60

ruleName

지연 데이터 규칙의 이름입니다.

유형: 문자열

Important

lateDataRules를 지정하려면 데이터 세트에 DeltaTime 필터를 사용해야 합니다.

지연 데이터 규칙 구성(콘솔)

다음 절차는 AWS IoT Analytics 콘솔에서 데이터 세트의 지연 데이터 규칙을 구성하는 방법을 보여줍니다.

최신 데이터 규칙을 구성하려면

1. [AWS IoT Analytics 콘솔](#)에 로그인합니다.
2. 탐색 창에서 데이터 세트를 선택합니다.
3. 데이터 세트에서 대상 데이터 세트를 선택합니다.
4. 탐색 창에서 세부사항을 선택합니다.
5. 델타 윈도우 섹션에서 편집을 선택합니다.
6. 데이터 선택 필터 구성에서 다음을 수행합니다.
 - a. 데이터 선택 창에서 델타 시간을 선택합니다.
 - b. 오프셋에 시간을 입력한 다음 단위를 선택합니다.
 - c. 타임스탬프 표현식에는 표현식을 입력합니다. 타임스탬프 필드의 이름이나 `from_unixtime(time)`과 같이 시간을 도출할 수 있는 SQL 표현식이 될 수 있습니다.

타임스탬프 표현식을 작성하는 방법에 대한 자세한 내용을 알아보려면 Presto 0.172 설명서의 [날짜 및 시간 함수와 연산자](#)를 참조하세요.

- d. 지연 데이터 알림의 경우 활성을 선택합니다.
- e. 델타 시간에는 정수를 입력합니다. 값의 범위는 1~60입니다.
- f. 저장을 선택합니다.

UPDATE DATA SET

Configure data selection filter

When creating a SQL data set, you can specify a `deltaTime` pre-filter to be applied to the message data to help limit the messages to those which have arrived since the last time the SQL data set content was created. [Learn more](#)

Data selection window

Offset

Specifies possible latency in the arrival of a message

Timestamp expression

Late data notification

Enable late data notification to receive CloudWatch events if late data is detected.

Delta time

IoT Analytics will emit a notification if late data is received within the value below

 Minutes

지연 데이터 규칙(CLI) 구성

AWS IoT Analytics API에서 `LateDataRuleConfiguration` 객체는 데이터 세트의 지연 데이터 규칙 설정을 나타냅니다. 이 객체는 `CreateDataset` 및 `UpdateDataset`와 연결된 `Dataset` 객체에 포함되어 있습니다. [API](#), [AWS CLI](#) 또는 [AWSSDK](#)를 사용하여 데이터 세트에 대한 지연 데이터 규칙을 지정할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

지정된 지연 데이터 규칙으로 데이터 세트를 생성하려면 다음 명령어를 실행합니다. 명령은 `dataset.json` 파일이 현재 디렉터리에 있다고 가정합니다.

Note

[UpdateDataset](#) API를 사용하여 기존 데이터 세트를 업데이트할 수 있습니다.

```
aws iotanalytics create-dataset --cli-input-json file://dataset.json
```

dataset.json 파일에 다음을 포함해야 합니다.

- *demo_dataset*를 대상 데이터 세트 이름으로 바꿉니다.
- *demo_datastore*를 대상 데이터 스토어 이름으로 바꿉니다.
- *from_unixtime(time)*을 타임스탬프 필드의 이름 또는 시간을 도출할 수 있는 SQL 표현식으로 바꿉니다.

타임스탬프 표현식을 작성하는 방법에 대한 자세한 내용을 알아보려면 Presto 0.172 설명서의 [날짜 및 시간 함수와 연산자](#)를 참조하세요.

- *timeout*을 1~60 사이의 정수로 바꿉니다.
- *demo_rule*을 원하는 이름으로 바꿉니다.

```
{
  "datasetName": "demo_dataset",
  "actions": [
    {
      "actionName": "myDatasetAction",
      "queryAction": {
        "filters": [
          {
            "deltaTime": {
              "offsetSeconds": -180,
              "timeExpression": "from_unixtime(time)"
            }
          }
        ],
        "sqlQuery": "SELECT * FROM demo_datastore"
      }
    }
  ],
  "retentionPeriod": {
```



```

    "unlimited": false,
    "numberOfDays": 90
  },
  "lateDataRules": [
    {
      "ruleConfiguration": {
        "deltaTimeSessionWindowConfiguration": {
          "timeoutInMinutes": timeout
        }
      },
      "ruleName": "demo_rule"
    }
  ]
}

```

지연 데이터 알림 수신 구독

AWS IoT Analytics에서 전송된 지연 데이터 알림을 처리하는 방법을 정의하는 규칙을 CloudWatch Events에서 만들 수 있습니다. CloudWatch Events는 알림을 수신하면 규칙에 정의된 지정된 대상 작업을 호출합니다.

CloudWatch Events 규칙 생성을 위한 사전 조건

AWS IoT Analytics에 대한 CloudWatch Events 규칙을 생성하기 전에 다음을 수행해야 합니다.

- CloudWatch Events의 이벤트, 규칙 및 대상을 익힙니다.
- CloudWatch Events 규칙에 의해 호출되는 [대상](#)을 생성하고 구성해야 합니다. 규칙은 다음과 같은 다양한 유형의 대상을 호출할 수 있습니다.
 - Amazon Kinesis Streams
 - AWS Lambda 함수
 - Amazon Simple Notification Service(Amazon SNS) 주제
 - Amazon Simple Queue Service(Amazon SQS) 대기열

CloudWatch Events 규칙 및 관련 대상은 AWS IoT Analytics 리소스를 생성한 AWS 리전에 있어야 합니다. 자세한 내용은 AWS 일반 참조의 [서비스 엔드포인트 및 할당량](#)을 참조하세요.

자세한 정보는 Amazon CloudWatch Events 사용 설명서의 [Amazon CloudWatch Events란 무엇인가요?](#) 및 [Amazon CloudWatch Events 시작하기](#)를 참조하세요.

지연 데이터 알림 이벤트

지연 데이터 알림 이벤트는 다음 형식을 사용합니다.

```
{
  "version": "0",
  "id": "7f51dfa7-ffef-97a5-c625-abddbac5eadd",
  "detail-type": "IoT Analytics Dataset Lifecycle Notification",
  "source": "aws.iotanalytics",
  "account": "123456789012",
  "time": "2020-05-14T02:38:46Z",
  "region": "us-east-2",
  "resources": ["arn:aws:iotanalytics:us-east-2:123456789012:dataset/demo_dataset"],
  "detail": {
    "event-detail-version": "1.0",
    "dataset-name": "demo_dataset",
    "late-data-rule-name": "demo_rule",
    "version-ids": ["78244852-8737-4650-aa4d-3071a01338fa"],
    "message": null
  }
}
```

지연 데이터 알림을 수신할 CloudWatch Events 규칙 생성

다음 절차에서는 Amazon SQS 대기열에 AWS IoT Analytics 지연 데이터 알림을 보내는 규칙을 생성하는 방법을 보여줍니다.

CloudWatch Events 규칙을 생성하려면

1. [Amazon CloudWatch 콘솔](#)에 로그인합니다.
2. 탐색 창의 이벤트 아래에서 규칙을 선택합니다.
3. 규칙 페이지에서 규칙 생성을 선택합니다.
4. 이벤트 소스 아래 이벤트 패턴을 선택합니다.
5. 서비스별 이벤트와 일치시킬 이벤트 패턴을 작성 섹션에서 다음을 수행합니다.
 - a. 서비스 이름에서 IoT Analytics를 선택합니다.
 - b. 이벤트 유형에서 IoT Analytics 데이터 세트 수명 주기 알림을 선택합니다.
 - c. 특정 데이터 세트 이름을 선택한 다음 대상 데이터 세트의 이름을 입력합니다.
6. 대상에서 대상 추가를 선택합니다.
7. SQS 대기열을 선택하고 다음을 수행합니다.

- 대기열*에서 대상 대기열을 선택합니다.
8. 세부 정보 구성을 선택합니다.
 9. 2단계: 규칙 세부 정보 구성 페이지에 이름 및 설명을 입력합니다.
 10. 규칙 생성을 선택합니다.

AWS CloudTrail을 사용하여 AWS IoT Analytics API 호출 로깅

AWS IoT Analytics는 AWS IoT Analytics에서 사용자, 역할, 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 AWS IoT Analytics 콘솔의 호출 및 AWS IoT Analytics API에 대한 코드 호출을 포함하여 AWS IoT Analytics에 대한 API 호출 하위 집합을 이벤트로 캡처합니다. 추적을 생성하면 AWS IoT Analytics 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 Event history(이벤트 기록)에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS IoT Analytics에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용 설명서](#)를 참조하세요.

AWS CloudTrail의 AWS IoT Analytics 정보

CloudTrail은 계정 생성 시 AWS 계정에서 사용되도록 설정됩니다. AWS IoT Analytics에서 활동이 발생하면 해당 활동이 [이벤트 기록(Event history)]의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

AWS IoT Analytics에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 정보는 다음을 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 지역에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)

AWS IoT Analytics는 CloudTrail 로그 파일에 다음 작업을 이벤트로 로깅합니다.

- [CancelPipelineReprocessing](#)
- [CreateChannel](#)
- [CreateDataset](#)
- [CreateDatasetContent](#)
- [CreateDatastore](#)
- [CreatePipeline](#)
- [DeleteChannel](#)
- [DeleteDataset](#)
- [DeleteDatasetContent](#)
- [DeleteDatastore](#)
- [DeletePipeline](#)
- [DescribeChannel](#)
- [DescribeDataset](#)
- [DescribeDatastore](#)
- [DescribeLoggingOptions](#)
- [DescribePipeline](#)
- [GetDatasetContent](#)
- [ListChannels](#)
- [ListDatasets](#)
- [ListDatastores](#)
- [ListPipelines](#)
- [PutLoggingOptions](#)
- [RunPipelineActivity](#)
- [SampleChannelData](#)
- [StartPipelineReprocessing](#)
- [UpdateChannel](#)
- [UpdateDataset](#)
- [UpdateDatastore](#)
- [UpdatePipeline](#)

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 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하세요.

AWS IoT Analytics 로그 파일 항목 이해

추적이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있도록 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 퍼블릭 API 호출에 대한 순서 지정된 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

다음은 CreateChannel 작업을 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ABCDE12345FGHIJ67890B:AnalyticsChannelTestFunction",
    "arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/AnalyticsChannelTestFunction",
    "accountId": "123456789012",
    "accessKeyId": "ABCDE12345FGHIJ67890B",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-02-14T23:43:12Z"
      }
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "ABCDE12345FGHIJ67890B",
      "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
      "accountId": "123456789012",
      "userName": "AnalyticsRole"
    }
  }
}
```

```

},
"eventTime": "2018-02-14T23:55:14Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateChannel",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"channelName": "channel_channeltest"
},
"responseElements": {
"retentionPeriod": {
"unlimited": true
},
"channelName": "channel_channeltest",
"channelArn": "arn:aws:iotanalytics:us-east-1:123456789012:channel/channel_channeltest"
},
"requestID": "7f871429-11e2-11e8-9eee-0781b5c0ac59",
"eventID": "17885899-6977-41be-a6a0-74bb95a78294",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}

```

다음은 CreateDataset 작업을 보여주는 CloudTrail 로그 항목이 나타낸 예제입니다.

```

{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsDatasetTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/AnalyticsDatasetTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
"mfaAuthenticated": "false",
"creationDate": "2018-02-14T23:41:36Z"
}
},
"sessionIssuer": {
"type": "Role",
"principalId": "ABCDE12345FGHIJ67890B",
"arn": "arn:aws:iam::123456789012:role/AnalyticsRole",

```

```

"accountId": "123456789012",
"userName": "AnalyticsRole"
}
},
"eventTime": "2018-02-14T23:53:39Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateDataset",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"datasetName": "dataset_datasettest"
},
"responseElements": {
"datasetArn": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/
dataset_datasettest",
"datasetName": "dataset_datasettest"
},
"requestID": "46ee8dd9-11e2-11e8-979a-6198b668c3f0",
"eventID": "5abe21f6-ee1a-48ef-afc5-c77211235303",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}

```

규정 준수 검증: AWS IoT Analytics


특정 규정 준수 프로그램의 범위 내에 AWS 서비스 있는지 알아보려면 AWS 서비스 규정 준수 [프로그램의 AWS 서비스 범위별, 규정](#) 참조하여 관심 있는 규정 준수 프로그램을 선택하십시오. 일반 정보는 [AWS 규정 준수 프로그램 AWS 보증 프로그램 규정 AWS](#) 참조하십시오.

를 사용하여 AWS Artifact 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 의 보고서 <https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html> 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS 서비스 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- [보안 및 규정 준수 킷스타트 가이드](#) - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 AWS 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.

- [Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계](#) — 이 백서에서는 기업이 HIPAA 적격 애플리케이션을 만드는 AWS 데 사용할 수 있는 방법을 설명합니다.

 Note

모든 AWS 서비스 사람이 HIPAA 자격을 갖춘 것은 아닙니다. 자세한 내용은 [HIPAA 적격 서비스 참조](#)를 참조하세요.

- [AWS 규정 준수 리소스 AWS](#) — 이 워크북 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- [AWS 고객 규정 준수 가이드](#) — 규정 준수의 관점에서 공동 책임 모델을 이해하십시오. 이 가이드에서는 보안을 유지하기 위한 모범 사례를 AWS 서비스 요약하고 여러 프레임워크 (미국 표준 기술 연구소 (NIST), 결제 카드 산업 보안 표준 위원회 (PCI), 국제 표준화기구 (ISO) 등) 에서 보안 제어에 대한 지침을 매핑합니다.
- AWS Config 개발자 안내서의 [규칙을 사용하여 리소스 평가](#) — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- [AWS Security Hub](#) — 이를 AWS 서비스 통해 조직 내 AWS 보안 상태를 포괄적으로 파악할 수 있습니다. Security Hub는 보안 통제를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 [Security Hub 제어 참조](#)를 참조하세요.
- [AWS Audit Manager](#) — 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 단순화할 수 있습니다.

의 레질리언스 AWS IoT Analytics

AWS 글로벌 인프라는 AWS 지역 및 가용 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 지역 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하십시오.

의 인프라 보안 AWS IoT Analytics

관리형 서비스로서 AWS 글로벌 네트워크 보안의 보호를 받습니다. AWS IoT Analytics AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 [AWS 클라우드 보안을](#) 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 [인프라 보호를](#) 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 액세스할 수 있습니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service\(AWS STS\)](#)를 사용하여 임시 보안 보안 인증을 생성하여 요청에 서명할 수 있습니다.

AWS IoT Analytics 할당량

AWS 일반 참조 안내서는 AWS 계정에 대해 AWS IoT Analytics에 대한 기본 할당량을 제공합니다. 달리 지정되지 않는 한 각 할당량은 AWS 리전별로 적용됩니다. 자세한 내용은 AWS 일반 참조안내서의 [AWS IoT Analytics엔드포인트 및 할당량](#) 및 [AWS서비스 할당량](#)을 참조하세요.

서비스 할당량 증가를 요청하려면 [지원 센터](#) 콘솔에서 지원 사례를 제출하세요. 자세한 내용은 Service Quotas 사용 설명서의 [할당량 증가 요청](#)을 참조하세요.

AWS IoT Analytics 명령

지원되는 웹 서비스 프로토콜에 대한 샘플 요청, 응답, 오류 등 AWS IoT Analytics에 대한 API 작업에 대해 알아보려면 이 주제를 읽어보세요.

AWS IoT Analytics 작업

AWS IoT Analytics API 명령을 사용하여 IoT 데이터를 수집, 처리, 저장 및 분석할 수 있습니다. 자세한 내용은 AWS IoT Analytics API 참조의 AWS IoT Analytics에서 지원하는 [작업](#)을 참조하세요.

AWS CLI Command Reference의 [AWS IoT Analytics 섹션](#)에는 AWS IoT Analytics를 관리 및 조작하는데 사용할 수 있는 AWS CLI 명령이 포함되어 있습니다.

AWS IoT Analytics 데이터

AWS IoT Analytics 데이터 API 명령을 사용하여 AWS IoT Analytics channel, pipeline, datastore, dataset로 고급 작업을 수행할 수 있습니다. 자세한 내용은 AWS IoT Analytics API 참조의 AWS IoT Analytics 데이터로 지원되는 [데이터 유형](#)을 참조하세요.

AWS IoT Analytics 문제 해결

오류를 해결하고 AWS IoT Analytics 문제를 해결할 수 있는 가능한 해결 방법을 찾으려면 다음 섹션을 참조하세요.

주제

- [제 메시지가 AWS IoT Analytics로 들어가고 있는지 어떻게 알 수 있습니까?](#)
- [제 파이프라인에서 메시지가 사라지는 이유는 무엇입니까? 어떻게 해결해야 합니까?](#)
- [제 데이터 스토어에 데이터가 없는 이유는 무엇입니까?](#)
- [데이터 세트에 왜 __dt이 표시되나요?](#)
- [데이터 세트 완료에 따라 실행되는 이벤트를 코딩하려면 어떻게 해야 합니까?](#)
- [AWS IoT Analytics를 사용하도록 노트북 인스턴스를 올바르게 구성하려면 어떻게 해야 합니까?](#)
- [인스턴스에 노트북이 생성되지 않는 이유는 무엇입니까?](#)
- [Amazon QuickSight에서 내 데이터 세트를 볼 수 없는 이유는 무엇입니까?](#)
- [기존 Jupyter Notebook에서 컨테이너화 버튼을 볼 수 없는 이유가 무엇입니까?](#)
- [컨테이너화 플러그인 설치가 실패하는 이유가 무엇입니까?](#)
- [컨테이너화 플러그인에 오류가 발생하는 이유가 무엇입니까?](#)
- [컨테이너화 동안 변수를 볼 수 없는 이유가 무엇입니까?](#)
- [내 컨테이너에 입력으로 추가할 수 있는 변수는 무엇입니까?](#)
- [컨테이너 출력을 이후에 분석을 위해 입력으로 설정하는 방법은 무엇입니까?](#)
- [컨테이너 데이터 세트가 실패하는 이유가 무엇입니까?](#)

제 메시지가 AWS IoT Analytics로 들어가고 있는지 어떻게 알 수 있습니까?

규칙 엔진을 통해 채널에 데이터를 주입하게 되어 있는 규칙이 제대로 구성되었는지 확인하십시오.

```
aws iot get-topic-rule --rule-name your-rule-name
```

응답은 다음과 같아야 합니다.

```
{
  "ruleArn": "arn:aws:iot:us-west-2:your-account-id:rule/your-rule-name",
```

```

"rule": {
  "awsIotSqlVersion": "2016-03-23",
  "sql": "SELECT * FROM 'iot/your-rule-name'",
  "ruleDisabled": false,
  "actions": [
    {
      "iotAnalytics": {
        "channelArn":
"arn:aws:iotanalytics:region:your_account_id:channel/your-channel-name"
      }
    }
  ],
  "ruleName": "your-rule-name"
}
}

```

규칙에 사용된 리전과 채널 이름이 올바른지 확인합니다. 데이터가 규칙 엔진으로 전송되고 규칙이 제대로 실행되는지 확인하기 위해, 새로운 대상을 추가하여 들어오는 메시지가 일시적으로 Amazon S3 버킷에 저장되도록 할 수 있습니다.

제 파이프라인에서 메시지가 사라지는 이유는 무엇입니까? 어떻게 해결해야 합니까?

- 잘못된 JSON 입력이 활동에 수신되었습니다.

Lambda 활동을 제외한 모든 활동에는 특히 유효한 JSON 문자열이 입력값으로 필요합니다. 활동에 수신된 JSON이 유효하지 않다면 해당 메시지는 삭제되고 데이터 스토어로 전달되지 않습니다. 올바른 JSON을 받아서 서비스에 입력해야 합니다. 이진수 입력인 경우, 이진수 데이터를 유효한 JSON으로 변환하고 이를 다음 활동으로 보내거나 데이터 스토어에 저장하는 Lambda 활동이 파이프라인의 첫 번째 활동이 되어야 합니다. 자세한 내용은 [Lambda 함수 예제 2](#)를 참조하세요.

- Lambda 활동으로 호출된 Lambda 함수에 권한이 부족합니다.

Lambda 활동의 각 Lambda 함수에 AWS IoT Analytics 서비스에서 호출하기에 충분한 권한이 있어야 합니다. 다음 AWS CLI 명령을 사용하여 권한을 부여할 수 있습니다.

```
aws lambda add-permission --function-name <name> --region <region> --statement-id <id> --principal iotanalytics.amazonaws.com --action lambda:InvokeFunction
```

- 필터 또는 removeAttribute 활동이 잘못 정의되었습니다.

`filter` 또는 `removeAttribute` 활동의 정의가 올바른지 확인하십시오. 메시지를 필터링하거나 메시지의 속성을 모두 제거하면 그 메시지는 데이터 스토어에 추가되지 않습니다.

제 데이터 스토어에 데이터가 없는 이유는 무엇입니까?

- 데이터 수집 시점과 데이터 가용 시점 사이에 지연 시간이 있습니다.

데이터가 채널에 수집된 뒤 데이터 스토어에서 그 데이터를 사용할 수 있게 되기까지 몇 분이 걸릴 수 있습니다. 이 지연 시간은 파이프라인 활동의 수와 파이프라인에서 이루어지는 사용자 지정 Lambda 활동의 정의에 따라 달라집니다.

- 파이프라인에서 메시지가 필터링되고 있습니다.

파이프라인에서 메시지가 누락되지 않도록 하십시오. (이전의 질문과 답변을 참조하십시오.)

- 데이터 세트 쿼리가 잘못되었습니다.

데이터 스토어에서 데이터 세트를 생성하는 쿼리가 올바른지 확인하십시오. 불필요한 필터는 쿼리에서 제거하여 데이터가 데이터 스토어에 전달되도록 합니다.

데이터 세트에 왜 `__dt`이 표시되나요?

- 이 열은 서비스에서 자동으로 추가한 것이며, 데이터의 수집 시간이 근사값으로 나와 있습니다. 이를 통해 쿼리를 최적화할 수 있습니다. 데이터 세트에 이것밖에 들어 있지 않다면 이전의 질문과 답변을 참조하십시오.

데이터 세트 완료에 따라 실행되는 이벤트를 코딩하려면 어떻게 해야 합니까?

- `describe-dataset` 명령을 토대로 폴링을 설정하여 특정한 타임스탬프의 데이터 세트 상태가 성공인지 확인해야 합니다.

AWS IoT Analytics를 사용하도록 노트북 인스턴스를 올바르게 구성하려면 어떻게 해야 하나요?

노트북 인스턴스를 만들 때 사용한 IAM 역할에 필요한 권한이 있는지 다음 단계에 따라 확인하십시오.

1. SageMaker 콘솔로 이동하여 노트북 인스턴스를 생성합니다.
2. 세부 정보를 입력하고 새 역할 생성을 선택합니다. 역할 ARN을 기록해 둡니다.
3. 노트북 인스턴스를 생성합니다. 이렇게 하면 SageMaker가 사용하는 역할도 생성됩니다.
4. IAM 콘솔로 이동하여 새로 생성된 SageMaker 역할을 수정합니다. 이 역할을 열면 관리형 정책이 있어야 합니다.
5. 인라인 정책 추가를 클릭하고 서비스로 IoTAnalytics를 선택한 다음, 읽기 권한에서 GetDatasetContent를 선택합니다.
6. 정책을 검토하고 정책 이름을 추가한 다음 생성합니다. 이렇게 하면 새로 생성된 역할이 AWS IoT Analytics에서 데이터 세트를 읽을 수 있는 정책 권한을 가지게 됩니다.
7. AWS IoT Analytics 콘솔로 이동하여 노트북 인스턴스에 노트북을 생성합니다.
8. 노트북 인스턴스가 "In Service" 상태가 되기를 기다립니다.
9. 노트북 생성을 선택하고 생성된 노트북 인스턴스를 선택합니다. 이렇게 하면 데이터 세트에 액세스할 수 있는 Jupyter Notebook이 선택한 템플릿으로 생성됩니다.

인스턴스에 노트북이 생성되지 않는 이유는 무엇입니까?

- 올바른 IAM 정책으로 노트북 인스턴스를 만들어야 합니다. (이전 질문의 단계에 따르십시오.)
- 노트북 인스턴스가 "In Service" 상태인지 확인합니다. 인스턴스를 생성할 때는 "Pending" 상태로 시작합니다. "In Service" 상태가 되기까지는 보통 5분 정도 걸립니다. 약 5분 후 노트북 인스턴스가 "Failed" 상태가 되면 권한을 다시 확인하세요.

Amazon QuickSight에서 내 데이터 세트를 볼 수 없는 이유는 무엇입니까?

Amazon QuickSight에는 AWS IoT Analytics 데이터 세트 콘텐츠를 읽을 수 있는 권한이 필요할 수 있습니다. 권한을 부여하려면 다음 단계를 수행합니다.

1. Amazon QuickSight의 오른쪽 상단에서 계정 이름을 선택하고 QuickSight 관리를 선택합니다.

2. 왼쪽 탐색 창에서 보안 & 권한을 선택합니다. QuickSight access to AWS services에서 AWS IoT Analytics에 액세스 권한이 부여되었는지 확인합니다.
 - a. AWS IoT Analytics에 액세스 권한이 없는 경우 추가 또는 제거를 선택합니다.
 - b. AWS IoT Analytics 옆에 있는 상자를 선택한 다음 업데이트를 선택합니다. 이렇게 하면 Amazon QuickSight가 데이터 세트 콘텐츠 읽기 권한을 가지게 됩니다.
3. 데이터를 다시 시각화합니다.

AWS와 Amazon QuickSight와 둘 모두에 대해 동일한 AWS IoT Analytics 리전을 선택했는지 확인하세요. 그렇지 않으면 AWS 리소스에 액세스하는 데 문제가 있을 수 있습니다. 지원되는 리전 목록은 Amazon Web Services 일반 참조의 [AWS IoT Analytics 엔드포인트 및 할당량](#), [Amazon QuickSight 엔드포인트 및 할당량](#)을 참조하세요.

기존 Jupyter Notebook에서 컨테이너화 버튼을 볼 수 없는 이유가 무엇입니까?

- AWS IoT Analytics 컨테이너화 플러그인이 누락되어 있어 발생하는 문제입니다. 2018년 8월 23일 이전에 SageMaker 노트북 인스턴스를 생성한 경우, [노트북 컨테이너화](#)에 있는 지침에 따라 수동으로 플러그인을 설치해야 합니다.
- AWS IoT Analytics 콘솔에서 SageMaker 노트북 인스턴스를 생성한 후, 또는 수동으로 설치한 후에도 컨테이너화 버튼이 표시되지 않는 경우에는 AWS IoT Analytics 기술 지원에 문의하십시오.

컨테이너화 플러그인 설치가 실패하는 이유가 무엇입니까?

- 일반적으로 SageMaker 노트북 인스턴스에 권한이 누락되어 있어 플러그인 설치가 실패합니다. 노트북 인스턴스의 필수 권한은 [권한](#)을 참조하고, 노트북 인스턴스 역할의 필수 권한을 추가하십시오. 문제가 계속되는 경우 AWS IoT Analytics 콘솔에서 새 노트북 인스턴스를 생성하십시오.
- 플러그인 설치 동안 로그에 "노트북(또는 다른 앱)을 로드할 때마다 브라우저에서 이 확장을 초기화"라는 메시지가 표시되는 경우 이를 무시할 수 있습니다.

컨테이너화 플러그인에 오류가 발생하는 이유가 무엇입니까?

- 여러 이유에서 컨테이너화가 실패해 오류가 발생할 수 있습니다. 노트북을 컨테이너화하기 전에 올바른 커널을 사용 중인지 확인하십시오. 컨테이너화된 커널은 "Containerized"라는 접두사로 시작됩니다.
- 플러그인이 ECR 리포지토리에 도커 이미지를 생성해 저장한 후, 노트북 인스턴스 역할에 ECR 리포지토리를 읽고, 열거하고, 생성할 충분한 권한이 있는지 확인합니다. 노트북 인스턴스의 필수 권한은 [권한](#)을 참조하고, 노트북 인스턴스 역할의 필수 권한을 추가하십시오.
- 또 리포지토리 이름이 ECR 요구 사항을 준수하는지 확인하십시오. ECR 리포지토리 이름은 문자로 시작해야 하고, 소문자와 숫자, 하이픈, 밑줄, 슬래시(/)만 포함할 수 있습니다.
- 컨테이너화 프로세스가 "이 인스턴스에는 컨테이너화를 실행시킬 여유 공간이 부족합니다."라는 오류와 함께 실패할 경우, 이 문제를 해결하기 위해 더 큰 인스턴스 사용을 시도합니다.
- 연결 오류나 이미지 생성 오류가 발생할 경우 다시 시도하십시오. 문제가 계속될 경우 인스턴스를 다시 시작해서 최신 플러그인 버전을 설치합니다.

컨테이너화 동안 변수를 볼 수 없는 이유가 무엇입니까?

- AWS IoT Analytics 컨테이너화 플러그인은 "컨테이너화된" 커널로 노트북을 실행시킨 후 자동으로 노트북의 모든 변수를 인식합니다. 컨테이너화된 커널 중 하나를 사용해 노트북을 실행한 후 컨테이너화를 수행하십시오.

내 컨테이너에 입력으로 추가할 수 있는 변수는 무엇입니까?

- 실행 시간 동안 수정하려는 값을 가진 모든 변수를 컨테이너에 입력으로 추가할 수 있습니다. 이렇게 하면, 데이터 세트 생성 시 제공해야 하는 여러 파라미터로 동일한 컨테이너를 실행할 수 있습니다. AWS IoT Analytics 컨테이너화 Jupyter 플러그인은 자동으로 노트북의 변수를 인식하고, 이를 컨테이너화 프로세스의 일부로 사용할 수 있도록 만들어 이런 프로세스를 단순화시킵니다.

컨테이너 출력을 이후에 분석을 위해 입력으로 설정하는 방법은 무엇입니까?

- 컨테이너 데이터 세트를 실행할 때마다 실행된 아티팩트를 저장할 수 있는 특정 S3 위치가 생성됩니다. 이 출력 위치에 액세스하려면 컨테이너 데이터 세트에 `outputFileUriValue`를 입력해 변수를

생성하십시오. 이 변수 값은 추가 출력 파일을 저장할 때 사용할 S3 경로여야 합니다. 이후 실행에서 저장한 아티팩트에 액세스하려면 `getDatasetContent` API 코드를 사용하여 이후 실행에 필요한 올바른 출력 파일을 선택합니다.

컨테이너 데이터 세트가 실패하는 이유가 무엇입니까?

- 컨테이너 데이터 세트에 올바른 `executionRole`을 전달했는지 확인하십시오. `executionRole`의 신뢰 정책에는 `iotanalytics.amazonaws.com` 및 `sagemaker.amazonaws.com`가 모두 포함되어야 합니다.
- 실패 이유가 `AlgorithmError`인 경우, 컨테이너 코드를 수동으로 디버깅하십시오. 컨테이너 코드에 버그가 있거나, 실행 역할에 컨테이너를 실행할 권한이 없을 때 발생하는 문제입니다. AWS IoT Analytics Jupyter 플러그인을 사용해 컨테이너화한 경우, `containerDataset`의 `executionRole`과 동일한 역할로 새 SageMaker 노트북 인스턴스를 생성한 후 수동으로 노트북을 실행합니다. Jupyter 플러그인 밖에서 컨테이너를 생성한 경우, 수동으로 코드를 실행하고 권한을 `executionRole`로 제한합니다.

문서 기록

아래 표에 2020년 11월 3월 이후 AWS IoT Analytics 사용 설명서의 주요 변경 사항이 설명되어 있습니다. 이 설명서에 대한 자세한 정보를 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
리전 출시	이제 아시아 태평양(뭄바이) 리전에서 AWS IoT Analytics을 사용할 수 있습니다.	2021년 8월 18일
JOIN로 쿼리하기	이 업데이트를 사용하면 JOIN을 사용하여 AWS IoT Analytics 데이터 세트를 쿼리할 수 있습니다.	2021년 7월 27일
AWS IoT SiteWise와 통합	이제 AWS IoT Analytics를 사용하여 AWS IoT SiteWise 데이터를 쿼리할 수 있습니다.	2021년 7월 27일
사용자 지정 파티션	이제 AWS IoT Analytics는 일반적으로 메시지 속성 또는 파이프라인 활동을 통해 추가된 속성에 따라 데이터를 파티셔닝할 수 있습니다.	2021년 6월 14일
채널 메시지 재처리	이 업데이트를 통해 지정된 Amazon S3 객체의 채널 데이터를 재처리할 수 있습니다.	2020년 12월 15일
Parquet 스키마	이제 AWS IoT Analytics 데이터 스토어가 Parquet 파일 형식을 지원합니다.	2020년 12월 15일
CloudWatch 이벤트로 모니터링	AWS IoT Analytics는 AWS Lambda 활동 중에 런타임 오류가 발생하면 Amazon	2020년 12월 15일

CloudWatch Events에 이벤트를 자동으로 게시합니다.

[지연 데이터 알림](#)

이 기능을 사용하여 지연 데이터가 도착할 때 Amazon CloudWatch Events를 통해 알림을 수신할 수 있습니다.

2020년 11월 9일

[리전 출시](#)

중국(베이징)에 AWS IoT Analytics가 출시되었습니다.

2020년 11월 4일

이전 업데이트

아래 표에서는 2020년 11월 4일 이전의 AWS IoT Analytics 사용 설명서에 적용된 주요 변경 사항을 설명합니다.

변경 사항	설명	날짜
리전 출시	아시아 태평양(시드니) 리전에 AWS IoT Analytics가 출시되었습니다.	2020년 7월 16일
업데이트	재구성된 설명서	2020년 5월 7일

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