사용자 가이드

AWS IoT 분석

AWS IoT 분석: 사용자 가이드

Table of Contents

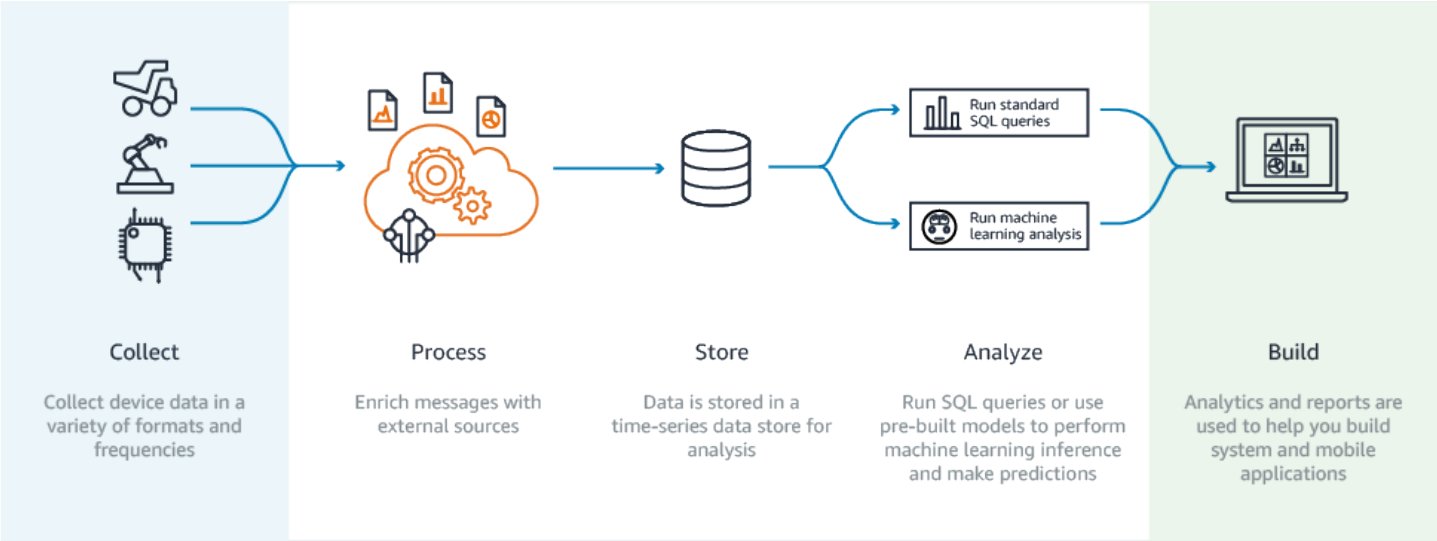
AWS IoT 분석란 무엇인가요?

AWS IoT 분석 는 IoT 디바이스의 데이터를 분석하는 데 필요한 단계를 자동화합니다.는 분석을 위해 시계열 데이터 스토어에 저장하기 전에 IoT 데이터를 AWS IoT 분석 필터링, 변환 및 보강합니다. 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 내장 SQL 쿼리 엔진을 사용하여 쿼리를 실행하여 데이터를 분석하거나 보다 복잡한 분석 및 기계 학습 추론을 수행할 수 있습니다. AWS IoT 분석 는 Jupyter Notebook과의 통합을 통해 고급 데이터 탐색을 가능하게 합니다. AWS IoT 분석 또한는 Quick Suite와의 통합을 통해 데이터 시각화를 가능하게 합니다. Quick Suite는 다음 리전에서 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 데이터 컨텍스트에서만 의미가 있는 경우가 많습니다. AWS IoT 분석 에서는 이러한 문제를 해결하고 대량의 디바이스 데이터를 수집하고 메시지를 처리하고 저장할 수 있습니다. 그런 다음 데이터를 쿼리하고 분석할 수 있습니다. 에는 장애가 발생할 예정이거나 웨어러블 디바이스를 중단할 위험이 있는 고객과 같은 질문에 답할 수 있도록 일반적인 IoT 사용 사례에 대해 사전 구축된 모델이 AWS IoT 분석 포함되어 있습니다.

사용 방법 AWS IoT 분석

다음 그래픽은 AWS IoT 분석사용 방법에 대한 개요를 보여줍니다.



주요 기능

수집

- 와 통합 AWS IoT Core-AWS IoT 분석 는와 완전히 통합되어 스트리밍할 때 연결된 디바이스에서 메시지를 수신할 AWS IoT Core 수 있습니다.
- 배치 API를 사용하여 모든 소스의 데이터를 추가합니다.는 HTTP를 통해 모든 소스의 데이터를 수신할AWS IoT 분석 수 있습니다. 다시 말해, 인터넷에 연결된 모든 디바이스 또는 서비스에서 AWS IoT 분석로 데이터를 보낼 수 있습니다. 자세한 내용은 AWS IoT Analytics API 참조의 BatchPutMessage를 참조하십시오.
- 저장 및 분석하려는 데이터만 수집 - 콘솔을 사용하여 AWS IoT 분석 다양한 형식 및 빈도의 MQTT 주제 필터를 통해 디바이스에서 메시지를 수신 AWS IoT 분석 하도록를 구성할 수 있습니다. AWS IoT 분석 는 데이터가 사용자가 정의한 특정 파라미터 내에 있는지 확인하고 채널을 생성합니다. 그런 다음 메시지 처리, 변환 및 보강에 적절한 파이프라인으로 채널을 라우팅합니다.

프로세스

- 정리 및 필터링 -AWS IoT 분석 가 누락된 데이터를 AWS IoT 분석 감지할 때 트리거되는 AWS Lambda 함수를 정의하여 코드를 실행하여 격차를 추정하고 채울 수 있습니다. 또한 최대 및 최소 필터 및 백분위 임계값을 정의하여 데이터 내 특이값을 제거할 수 있습니다.
- 변환 - 사용자가 정의한 수학 또는 조건부 로직을 사용하여 메시지를AWS IoT 분석 변환할 수 있으므로 섭씨에서 화씨로의 변환과 같은 일반적인 계산을 수행할 수 있습니다.
- 보강 - 일기 예보와 같은 외부 데이터 소스로 데이터를 보강한 다음 데이터를 AWS IoT 분석 데이터 스토어로 라우팅할AWS IoT 분석 수 있습니다.

저장

- 시계열 데이터 스토어 - 더 빠른 검색 및 분석을 위해 디바이스 데이터를 최적화된 시계열 데이터 스토어에AWS IoT 분석 저장합니다. 액세스 권한을 관리하고, 데이터 보존 정책을 구현하고, 데이터를 외부 액세스 포인트로 내보낼 수도 있습니다.
- 처리된 데이터 및 원시 데이터 저장 -는 처리된 데이터를AWS IoT 분석 저장하고 원시 수집된 데이터도 자동으로 저장하므로 나중에 처리할 수 있습니다.

분석

- 임시 SQL 쿼리 실행 -는 임시 쿼리를 실행하고 결과를 빠르게 가져올 수 있도록 SQL 쿼리 엔진을AWS IoT 분석 제공합니다. 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 - 시계열 분석을AWS IoT 분석 지원하므로 시간 경과에 따른 디바이스의 성능을 분석하고 디바이스가 사용되는 방식과 위치를 이해하고, 디바이스 데이터를 지속적으로 모니터링하여 유지 관리 문제를 예측하고, 센서를 모니터링하여 환경 조건을 예측하고 대응할 수 있습니다.
- 호스팅된 노트북을 통한 정교한 분석 및 기계 학습 -AWS IoT 분석 에는 통계 분석 및 기계 학습을 위해 Jupyter Notebook의 호스팅 노트북을 지원하는 기능이 있습니다. 이 서비스에는 AWS작성 기계 학습 모델 및 시각화가 포함된 노트북 템플릿 세트가 포함되어 있습니다. 템플릿을 이용하여 디바이스 장애 프로파일링, 고객의 제품 중단을 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 워크플로를 참조하십시오.
- 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

- Quick Suite 통합 - QuickSight 대시보드에서 데이터 세트를 시각화할 수 있도록 QuickSight Suite에 커넥터를AWS IoT 분석 제공합니다.
- 콘솔 통합 - AWS IoT 분석콘솔의 임베디드 Jupyter Notebook에서 결과 또는 임시 분석을 시각화할 수도 있습니다.

AWS IoT 분석 구성 요소 및 개념

채널

채널은 MQTT 주제로부터 데이터를 수집하고, 데이터를 파이프라인에 게시하기 전에 처리되지 않은 원시 메시지를 보관합니다. 또한 BatchPutMessage API로 메시지를 직접 채널로 전송할 수 있습니다. 처리되지 않은 메시지는 사용자가 관리하거나 AWS IoT 분석 관리하는 Amazon Simple Storage Service(Amazon 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 분석 관리하는 Amazon S3 버킷에 저장됩니다.

데이터 세트

데이터 세트를 생성하여 데이터 스토어에서 데이터를 검색합니다. SQL 데이터 세트 또는 컨테이너 데이터 세트를 생성할 수 AWS IoT 분석 있습니다.

데이터 세트가 있으면 Quick Suite를 사용한 통합을 통해 데이터를 탐색하고 인사이트를 얻을 수 있습니다. 아니면 Jupyter Notebook과 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 템플릿을 참조하십시오.

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 Events의 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 워크플로를 참조하십시오. 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

실행할 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: `myOtherDataset`가 콘텐츠 생성을 완료할 때 이 데이터 세트 생성)일 수 있습니다. 또는 CreateDatasetContent API를 사용하여 데이터 세트 콘텐츠를 수동으로 생성할 수 있습니다.

Docker 컨테이너

자체 Docker 컨테이너를 생성하여 분석 도구를 패키징하거나 SageMaker AI가 제공하는 옵션을 사용할 수 있습니다. 자세한 내용은 Docker 컨테이너를 참조하십시오. 자체 Docker 컨테이너를 생성하여 분석 도구를 패키징하거나 SageMaker AI에서 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 컨테이너화를 참조하십시오.

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 `queryAction`에서 `filters`의 `deltaTime`을 설정해 델타 기간을 생성할 수 있습니다. 자세한 정보는 `CreateDataset`API를 참조하세요. 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 참조하십시오.

액세스 AWS IoT 분석

의 일부로 AWS IoT는 디바이스가 데이터를 생성하고 애플리케이션이 생성한 데이터와 상호 작용할 수 있도록 다음 인터페이스를 AWS IoT 분석 제공합니다.

AWS Command Line Interface (AWS CLI)

Windows, OS X 및 Linux AWS IoT 분석 에서에 대한 명령을 실행합니다. 이러한 명령을 사용하여 사물, 인증서, 규칙 및 정책을 생성하고 관리할 수 있습니다. 시작하려면 AWS Command Line

Interface 사용 설명서를 참조하십시오. 명령에 대한 자세한 내용은 AWS Command Line Interface 참조의 iot를 AWS IoT참조하세요.

> **Important**
>
> `aws iotanalytics` 명령을 사용하여와 상호 작용합니다 AWS IoT 분석. 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 IoT API 참조의 작업 섹션을 참조하십시오.

AWS SDKs

언어별 API를 사용하여 AWS IoT 분석 애플리케이션을 빌드합니다. APIs 이러한 SDK는 HTTP와 HTTPS API를 래핑하고 지원되는 언어로 프로그램을 작성할 수 있게 해줍니다. 자세한 내용은 AWS SDK 및 도구 섹션을 참조하십시오.

AWS IoT 디바이스 SDKs

메시지를 보내는 디바이스에서 실행되는 애플리케이션을 빌드합니다 AWS IoT 분석. 자세한 내용은 AWS IoT SDK를 참조하십시오.

AWS IoT 분석 콘솔

구성 요소를 구축하여 AWS IoT 분석 콘솔에서 결과를 시각화할 수 있습니다.

사용 사례

예측 유지 보수

AWS IoT 분석 는 예측 유지 관리 모델을 빌드하고 디바이스에 적용할 템플릿을 제공합니다. 예를 들어 AWS IoT 분석 를 사용하여 연결된 화물 차량에서 난방 및 냉방 시스템이 실패할 가능성이 있는 시기를 예측하여 차량의 경로를 변경하여 배송 손상을 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 분석 를 사용하면 인벤토리를 실시간으로 모니터링할 수 있는 IoT 애플리케이션을 구축할 수 있습니다. 예를 들어 식음료 회사에서 자동 판매기의 데이터를 분석하고 공급이 줄어들 때마다 상품을 사전 재주문할 수 있습니다.

프로세스 효율성 평가

를 사용하면 다양한 프로세스의 효율성을 지속적으로 모니터링하고 프로세스를 개선하기 위한 조치를 취하는 IoT 애플리케이션을 구축할 AWS IoT 분석수 있습니다. 예를 들어 광업 회사에서는 운행 시마다 적재량을 최대화하여 광석 운반 트럭의 효율성을 높일 수 있습니다. AWS IoT 분석를 사용하면 시간 경과에 따른 위치 또는 트럭의 가장 효율적인 적재를 식별한 다음 목표 적재와의 편차를 실시간으로 비교하고 효율성을 개선하기 위한 선행 지침을 더 잘 계획할 수 있습니다.

스마트 농업

AWS IoT 분석 는 AWS IoT 레지스트리 데이터 또는 퍼블릭 데이터 소스를 사용하여 컨텍스트 메타데이터로 IoT 디바이스 데이터를 보강하여 시간, 위치, 온도, 고도 및 기타 환경 조건의 분석 요소를 강화할 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 분석 지원 종료

신중한 고려 후 2025년 12월 15 AWS IoT 분석일부터에 대한 지원을 종료하기로 결정했습니다. AWS IoT 분석 는 2024년 7월 24일부터 더 이상 신규 고객을 받지 않습니다. 2024년 7월 23일 이전에 서비스에 가입한 계정이 있는 기존 고객은 AWS IoT 분석 기능을 계속 사용할 수 있습니다. 2025년 12월 15일 이후에는 더 이상를 사용할 수 없습니다 AWS IoT 분석.

2025년 12월 15일에 end-of-service가가 AWS IoT 분석 까워지면 고객이 마이그레이션 옵션을 이해하는 것이 중요합니다. 이 페이지에서는의 주요 기능에 대한 개요를 AWS IoT 분석 제공하고 기능을 복제하는 데 사용되는 대체 AWS 서비스에 매핑합니다. 이러한 대체 서비스의 기능을 이해하면 고객은 원활한 마이그레이션을 계획하고 실행하여 AWS IoT 데이터 분석 워크플로를 중단 없이 계속할 수 있습니다.

주제

마이그레이션 옵션

마이그레이션을 고려할 때는이 전환의 이점과 이유를 이해하는 AWS IoT 분석것이 중요합니다. 아래 표에는 대체 옵션과 기존 AWS IoT 분석 기능에 대한 매핑이 나와 있습니다.

작업	AWS IoT 분석	대체 서비스	이유
수집	AWS IoT 분석 를 사용하면 `BatchPutMessage` API를 사용하여 AWS IoT Core 또는 다른 소스에서 직접 데이터를 쉽게 수집할 수 있습니다. 이 통합을 통해 디바이스에서 분석 플랫폼으로 데이터를 원활하게 흐를 수 있습니다.	• Amazon Kinesis Data Streams • Amazon Data Firehose	Amazon Kinesis Data Streams는 강력한 솔루션을 제공합니다. Kinesis는 데이터를 실시간으로 스트리밍하여 즉각적인 처리 및 분석을 가능하게 하며, 이는 실시간 인사이트와 이상 탐지가 필요한 애플리케이션에 매우 중요합니다.

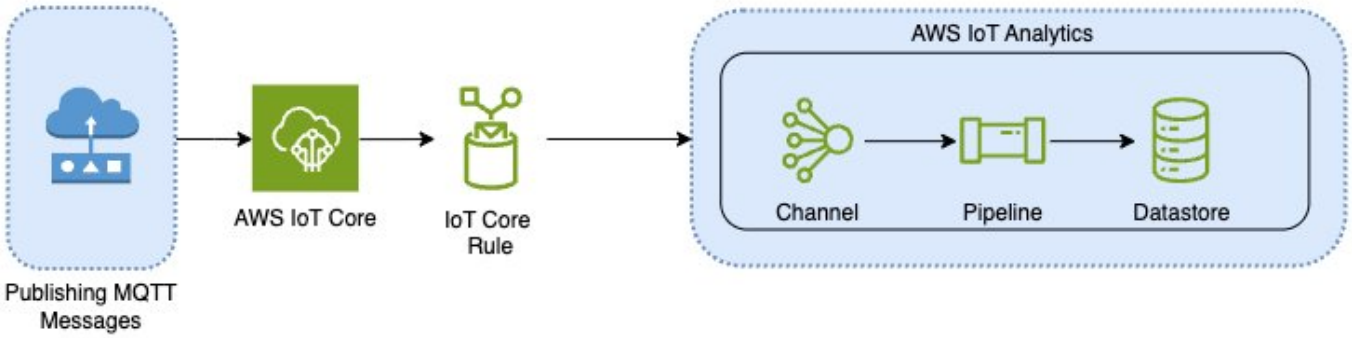
작업	AWS IoT 분석	대체 서비스	이유
			Amazon Data Firehose는 스트리밍 데이터가 Amazon S3에 도착하기 전에 스트리밍 데이터를 캡처하고 변환하는 프로세스를 간소화하여 데이터 처리량에 맞게 자동으로 규모를 조정합니다.
프로세스	에서 데이터를 처리 AWS IoT 분석 하려면 외부 소스로 데이터를 정리, 필터링, 변환 및 보강해야 합니다.	• Amazon Managed Service for Apache Flink • Amazon Data Firehose	Amazon Managed Service for Apache Flink는 정교한 AWS IoT 분석 시나리오에 필수적인 패턴 일치 및 집계와 같은 복잡한 이벤트 처리를 지원합니다. Amazon Data Firehose는 더 간단한 변환을 처리하고 사용자 지정 처리를 위해 AWS Lambda 함수를 호출하여 Flink의 복잡성 없이 유연성을 제공합니다.

작업	AWS IoT 분석	대체 서비스	이유
저장	AWS IoT 분석 는 데이터 보존 정책 및 액세스 관리와 같은 기능을 포함하는 AWS IoT 데이터에 최적화된 시계열 데이터 스토어를 사용합니다.	• Amazon S3 • Amazon Timestream	Amazon S3는 확장 가능하고 내구성이 뛰어나며 비용 효율적인 스토리지 솔루션을 제공합니다. Amazon S3는 다른 AWS 서비스와 통합되어 대규모 데이터 세트의 장기 저장 및 분석에 매우 적합합니다. Amazon Timestream은 특별히 구축된 시계열 데이터베이스입니다. Amazon S3에서 데이터를 일괄 로드할 수 있습니다.
분석	AWS IoT 분석 는 호스팅된 Jupyter Notebook에 대한 기본 제공 SQL 쿼리 기능, 시계열 분석 및 지원을 제공하므로 고급 분석 및 기계 학습을 쉽게 수행할 수 있습니다.	• AWS Glue • Amazon Athena	AWS Glue 는 ETL 프로세스를 간소화하여 데이터를 쉽게 추출, 변환 및 로드하는 동시에 쿼리를 용이하게 하기 위해 Athena와 통합되는 데이터 카탈로그를 제공합니다. Amazon Athena는 인프라를 관리할 필요 없이 Amazon S3에 저장된 데이터에 대해 SQL 쿼리를 직접 실행할 수 있도록 하여이 작업을 한 단계 더 진행합니다.

작업	AWS IoT 분석	대체 서비스	이유
시각화	AWS IoT 분석 는 Quick Suite와 통합되어 풍부한 시각화 및 대시보드를 생성할 수 있습니다.	• Amazon Quick Suite	Amazon S3와 같이 사용하려는 대체 데이터 스토어에 따라 Quick Suite를 계속 사용합니다.

마이그레이션 가이드

현재 아키텍처에서 AWS IoT 데이터는 AWS IoT Core 규칙을 AWS IoT Core AWS IoT 분석 통해에서로 흐릅니다. AWS IoT 분석 는 수집, 변환 및 스토리지를 처리합니다.



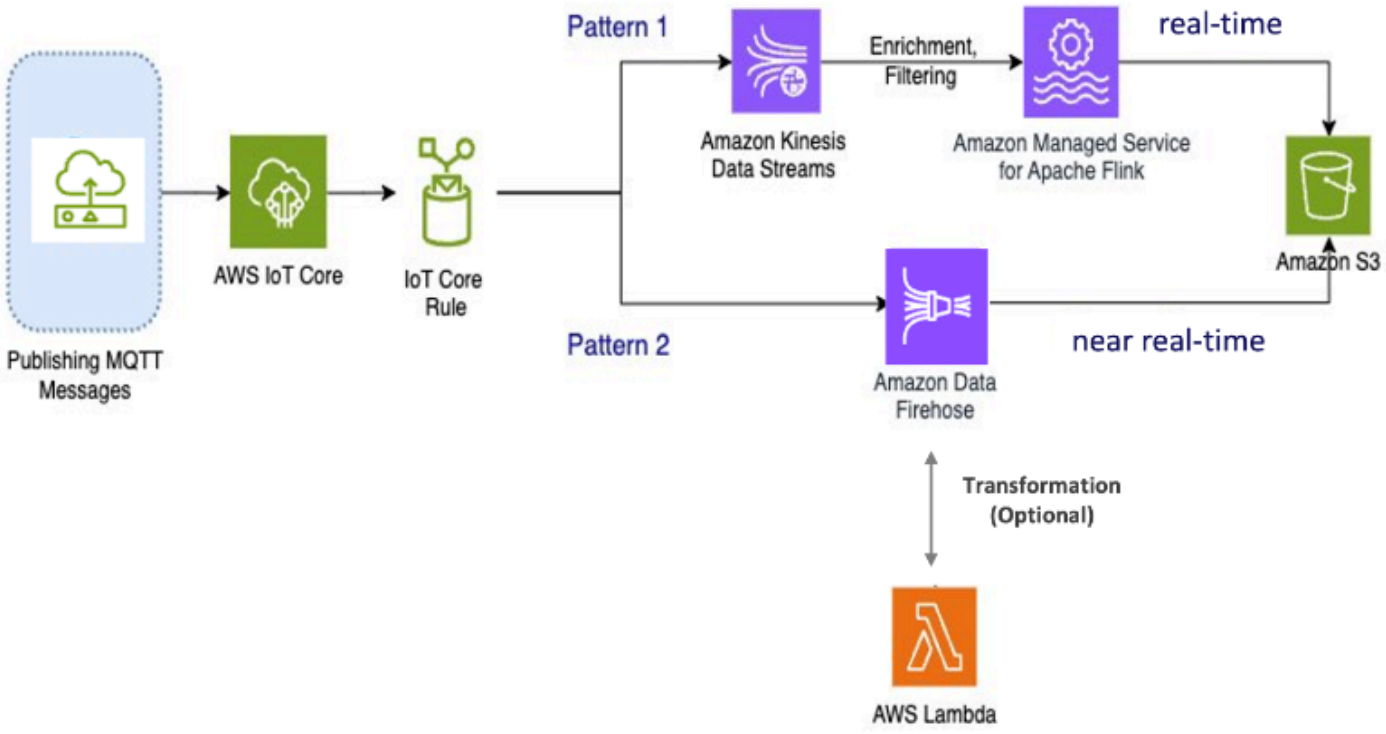
마이그레이션을 완료하려면 다음 두 단계를 따릅니다.

주제

- 1단계: 진행 중인 데이터 수집 리디렉션
- 2단계: 이전에 수집된 데이터 내보내기
- 두 패턴 모두에 대해 온디맨드 쿼리 실행
- 요약

1단계: 진행 중인 데이터 수집 리디렉션

마이그레이션의 첫 번째 단계는 진행 중인 데이터 수집을 새 서비스로 리디렉션하는 것입니다. 특정 사용 사례에 따라 두 가지 패턴을 사용하는 것이 좋습니다.



패턴 1: Amazon Managed Service for Apache Flink를 사용한 Amazon Kinesis Data Streams

이 패턴에서는가 Amazon Kinesis Data Streams와 통합 AWS IoT Core 되는 데이터를 게시하여 대량의 데이터를 실시간으로 수집, 처리 및 분석할 수 있습니다.

지표 및 분석

1. 수집 데이터: AWS IoT 데이터는 Amazon Kinesis Data Streams에 실시간으로 수집됩니다. Amazon Kinesis Data Streams는 수백만 개의 AWS IoT 디바이스에서 높은 처리량의 데이터를 처리할 수 있으므로 실시간 분석 및 이상 탐지가 가능합니다.
2. 데이터 처리: Amazon Managed Service for Apache Flink를 사용하여 Amazon Kinesis Data Streams에서 데이터를 처리, 보강 및 필터링합니다. Flink는 집계, 조인 및 임시 작업과 같은 복잡한 이벤트 처리를 위한 강력한 기능을 제공합니다.
3. 데이터 저장: Flink는 처리된 데이터를 Amazon S3에 출력하여 저장 및 추가 분석을 수행합니다. 그런 다음 Amazon Athena를 사용하여이 데이터를 쿼리하거나 다른 AWS 분석 서비스와 통합할 수 있습니다.

애플리케이션에 고대역폭 스트리밍 데이터가 포함되어 있고 패턴 일치 또는 윈도우 설정과 같은 고급 처리가 필요한 경우이 패턴을 사용하는 것이 가장 적합합니다.

패턴 2: Amazon Data Firehose 사용

이 패턴에서는 Amazon Data Firehose와 통합 AWS IoT Core되는에 데이터가 게시되므로 Amazon S3에 직접 데이터를 저장할 수 있습니다. 이 패턴은를 사용한 기본 변환도 지원합니다 AWS Lambda.

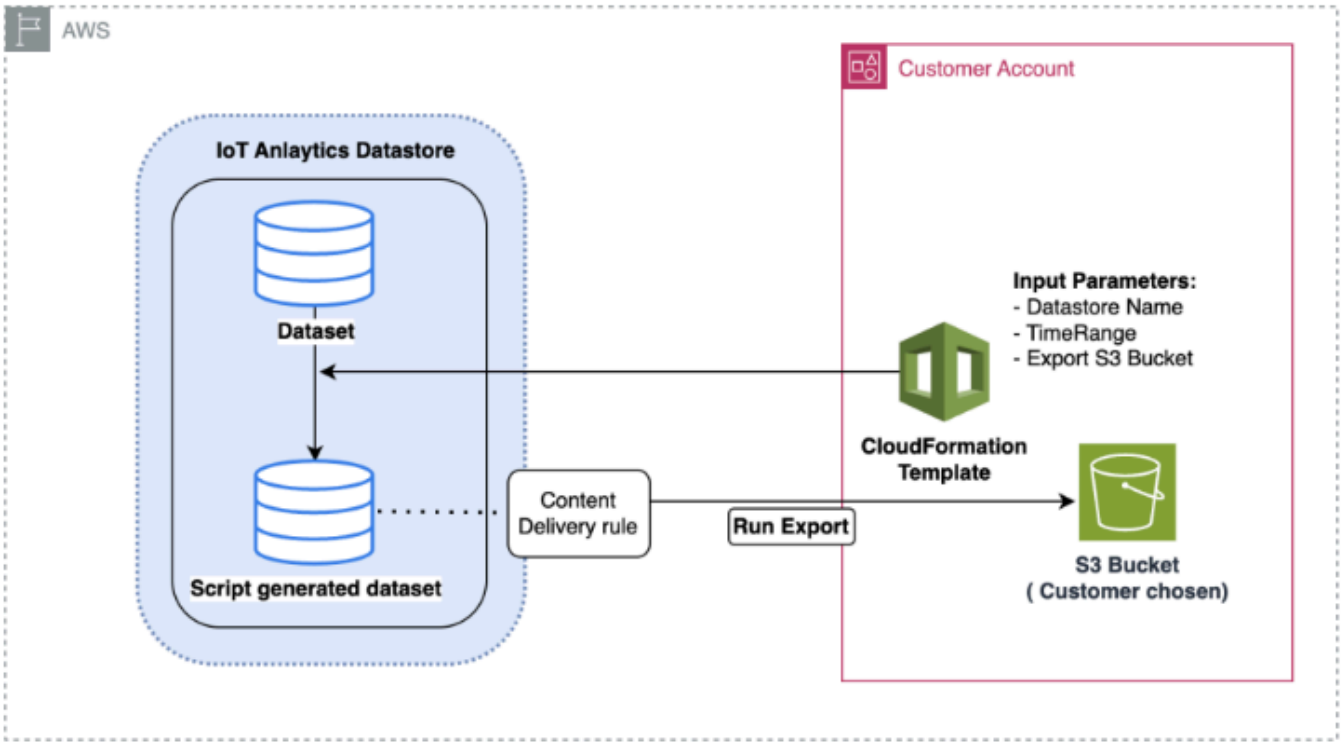
지표 및 분석

1. 수집 데이터: AWS IoT 데이터는 디바이스에서 직접 수집되거나 Amazon Data Firehose AWS IoT Core 로 수집됩니다.
2. 프로세스 데이터: Amazon Data Firehose는 형식 변환 및 보강과 같은 데이터에 대한 기본 변환 및 처리를 수행합니다. 수신 소스 데이터를 대상으로 전송하기 전에 변환하도록 AWS Lambda 함수를 호출하도록 구성하여 Firehose 데이터 변환을 활성화할 수 있습니다.
3. 데이터 저장: 처리된 데이터는 거의 실시간으로 Amazon S3로 전송됩니다. Amazon Data Firehose는 수신 데이터의 처리량에 맞게 자동으로 규모를 조정하여 안정적이고 효율적인 데이터 전송을 보장합니다.

기본 변환 및 처리가 필요한 워크로드에이 패턴을 사용합니다. 또한 Amazon Data Firehose는 Amazon S3에 저장된 데이터에 대한 데이터 버퍼링 및 동적 파티셔닝 기능을 제공하여 프로세스를 간소화합니다.

2단계: 이전에 수집된 데이터 내보내기

이전에 수집하여에 저장한 데이터의 경우 Amazon S3로 내보내 AWS IoT 분석야 합니다. 이 프로세스를 간소화하기 위해 템플릿을 사용하여 CloudFormation 전체 데이터 내보내기 워크플로를 자동화할 수 있습니다. 부분(시간 범위 기반) 데이터 추출에 스크립트를 사용할 수 있습니다.



CloudFormation Amazon S3로 데이터를 내보내는 템플릿

위 다이어그램은 CloudFormation 템플릿을 사용하여 동일한 AWS IoT 분석 데이터 스토어 내에 데이터 세트를 생성하여 타임스탬프를 기반으로 선택을 활성화하는 프로세스를 보여줍니다. 이를 통해 사용자는 원하는 기간 내에 특정 데이터 포인트를 검색할 수 있습니다. 또한 데이터를 Amazon S3 버킷으로 내보내는 콘텐츠 전송 규칙이 생성됩니다.

아래 절차는 단계를 보여줍니다.

1. CloudFormation 템플릿을 준비하고 YAML 파일로 저장합니다. 예를 들어 `migrate-datasource.yaml`입니다.

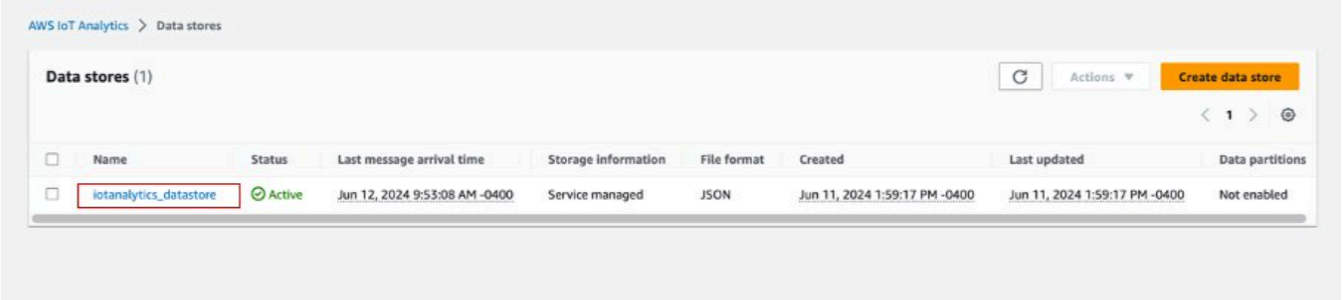
```
# Cloudformation Template to migrate an AWS IoT Analytics datastore to an external
 dataset
AWSTemplateFormatVersion: 2010-09-09
Description: Migrate an AWS IoT Analytics datastore to an external dataset
Parameters:
  DatastoreName:
    Type: String
    Description: The name of the datastore to migrate.
```

```
    AllowedPattern: ^[a-zA-Z0-9_]+$
  TimeRange:
    Type: String
    Description: |
      This is an optional argument to split the source data into multiple files.
      The value should follow the SQL syntax of WHERE clause.
      E.g. WHERE DATE(Item_TimeStamp) BETWEEN '09/16/2010 05:00:00' and '09/21/2010
 09:00:00'.
    Default: ''
  MigrationS3Bucket:
    Type: String
    Description: The S3 Bucket where the datastore will be migrated to.
    AllowedPattern: (?!(^xn--|.+-s3alias$))^[a-z0-9][a-z0-9-]{1,61}[a-z0-9]$
  MigrationS3BucketPrefix:
    Type: String
    Description: The prefix of the S3 Bucket where the datastore will be migrated
 to.
    Default: ''
    AllowedPattern: (^([a-zA-Z0-9.\-_]*\/)*$)|(^$)
Resources:
  # IAM Role to be assumed by the AWS IoT Analytics service to access the external
 dataset
  DatastoreMigrationRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: iotanalytics.amazonaws.com
            Action: sts:AssumeRole
      Policies:
        - PolicyName: AllowAccessToExternalDataset
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - s3:GetBucketLocation
                  - s3:GetObject
                  - s3:ListBucket
                  - s3:ListBucketMultipartUploads
                  - s3:ListMultipartUploadParts
```

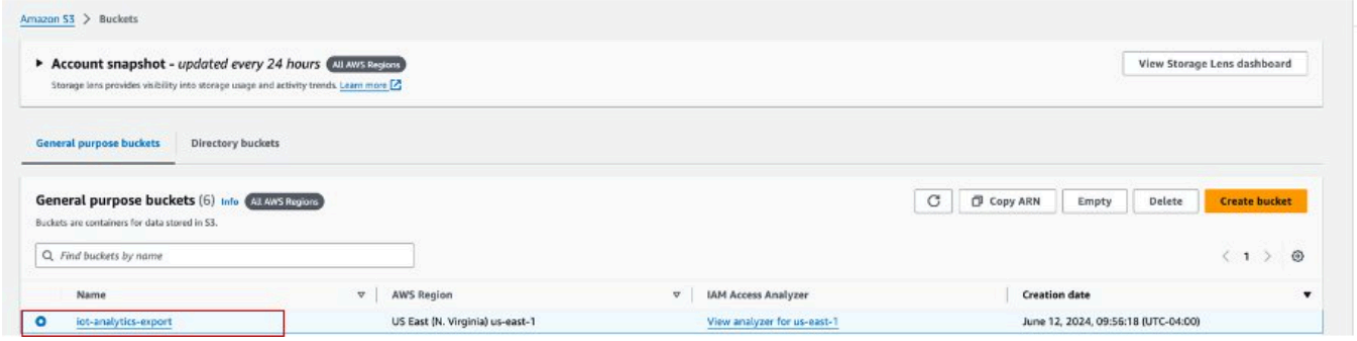
```
                  - s3:AbortMultipartUpload
                  - s3:PutObject
                  - s3:DeleteObject
                Resource:
                  - !Sub arn:aws:s3:::${MigrationS3Bucket}
                  - !Sub arn:aws:s3:::${MigrationS3Bucket}/
${MigrationS3BucketPrefix}*

  # This dataset that will be created in the external S3 Export
  MigratedDataset:
    Type: AWS::IoTAnalytics::Dataset
    Properties:
      DatasetName: !Sub ${DatastoreName}_generated
      Actions:
        - ActionName: SqlAction
          QueryAction:
            SqlQuery: !Sub SELECT * FROM ${DatastoreName} ${TimeRange}
      ContentDeliveryRules:
        - Destination:
            S3DestinationConfiguration:
              Bucket: !Ref MigrationS3Bucket
              Key: !Sub ${MigrationS3BucketPrefix}${DatastoreName}/!
{iotanalytics:scheduleTime}/!{iotanalytics:versionId}.csv
              RoleArn: !GetAtt DatastoreMigrationRole.Arn
      RetentionPeriod:
        Unlimited: true
      VersioningConfiguration:
        Unlimited: true
```

2. AWS IoT 분석 데이터를 내보내야 하는 데이터 스토어를 결정합니다. 이 가이드에서는 라는 샘플 데이터 스토어를 사용합니다`iot_analytics_datastore`.

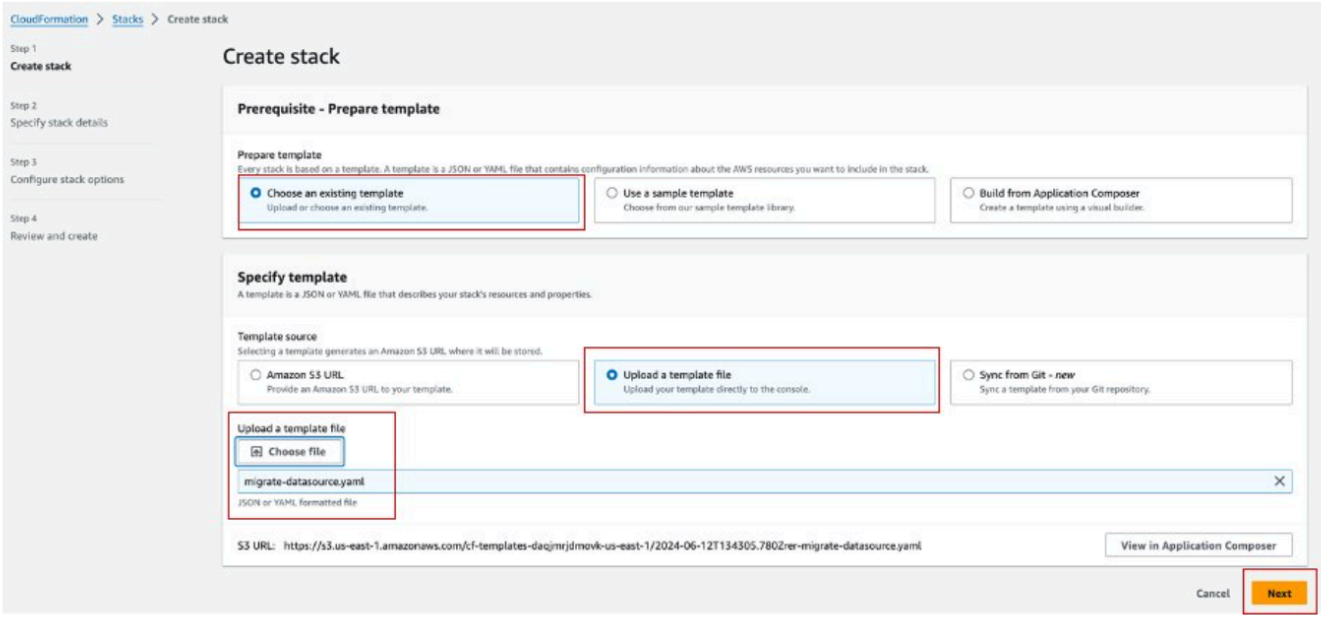


3. 데이터를 내보낼 Amazon S3 버킷을 생성하거나 식별합니다. 이 가이드에서는 `iot-analytics-export` 버킷을 사용합니다.



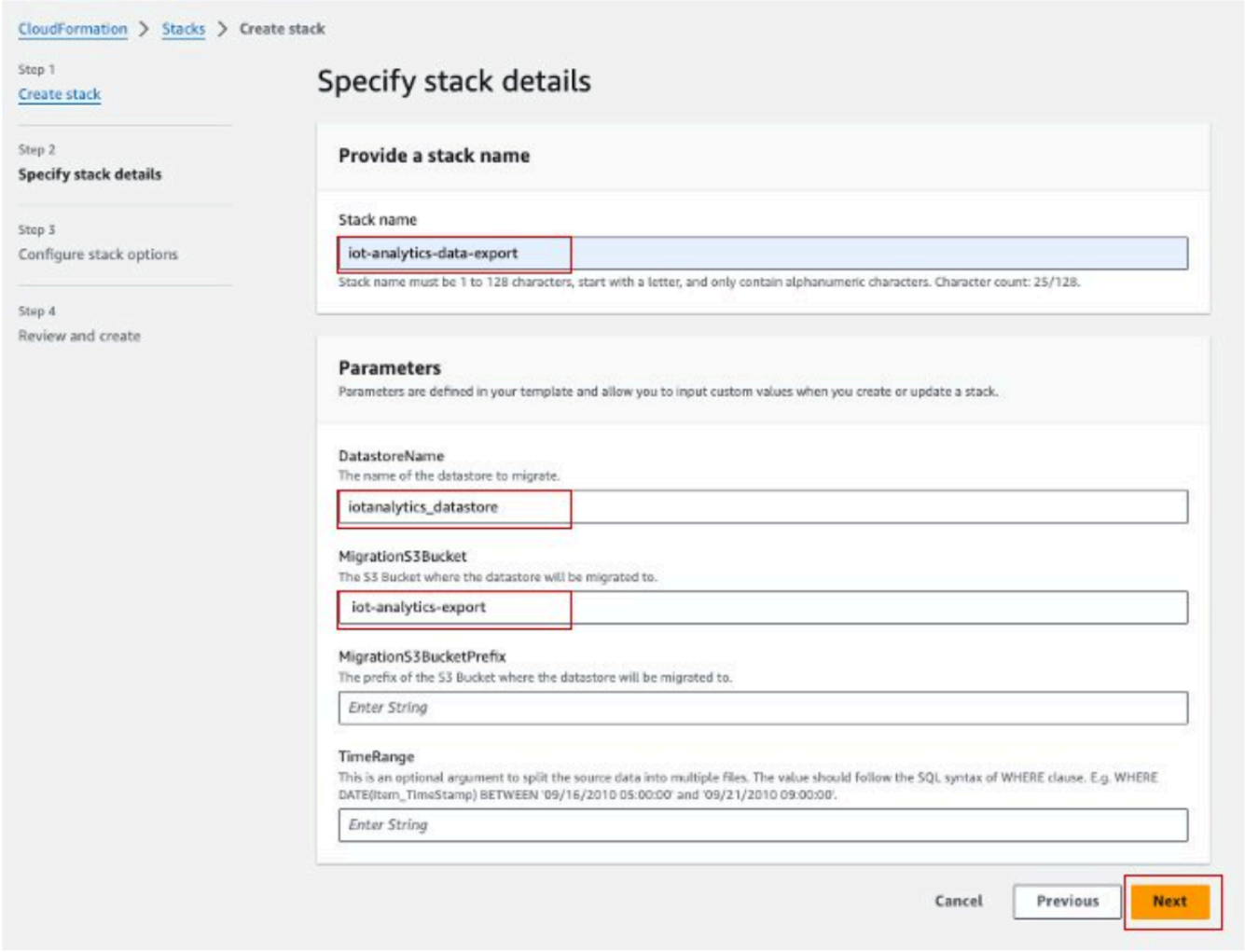
4. CloudFormation 스택을 생성합니다.

   - https://console.aws.amazon.com/cloudformation 이동합니다.
   - 스택 생성을 클릭하고 새 리소스 사용(표준)을 선택합니다.
   - `migrate-datasource.yaml` 파일을 업로드합니다.

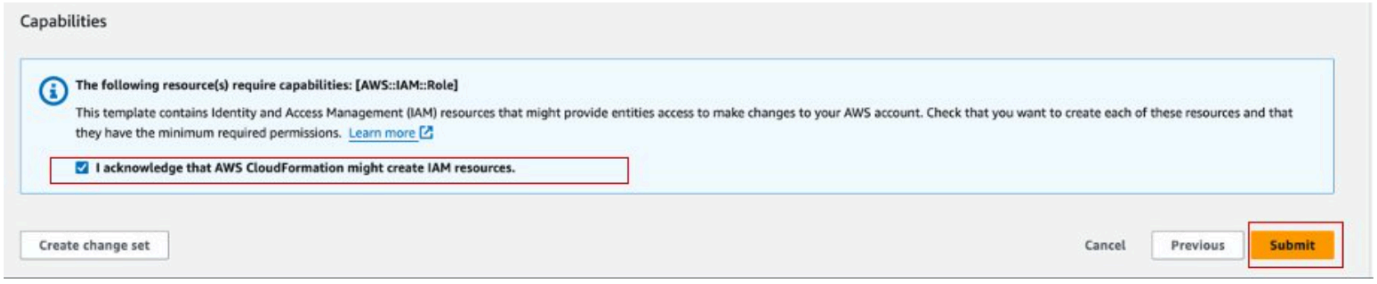


5. 스택 이름을 입력하고 다음 파라미터를 제공합니다.

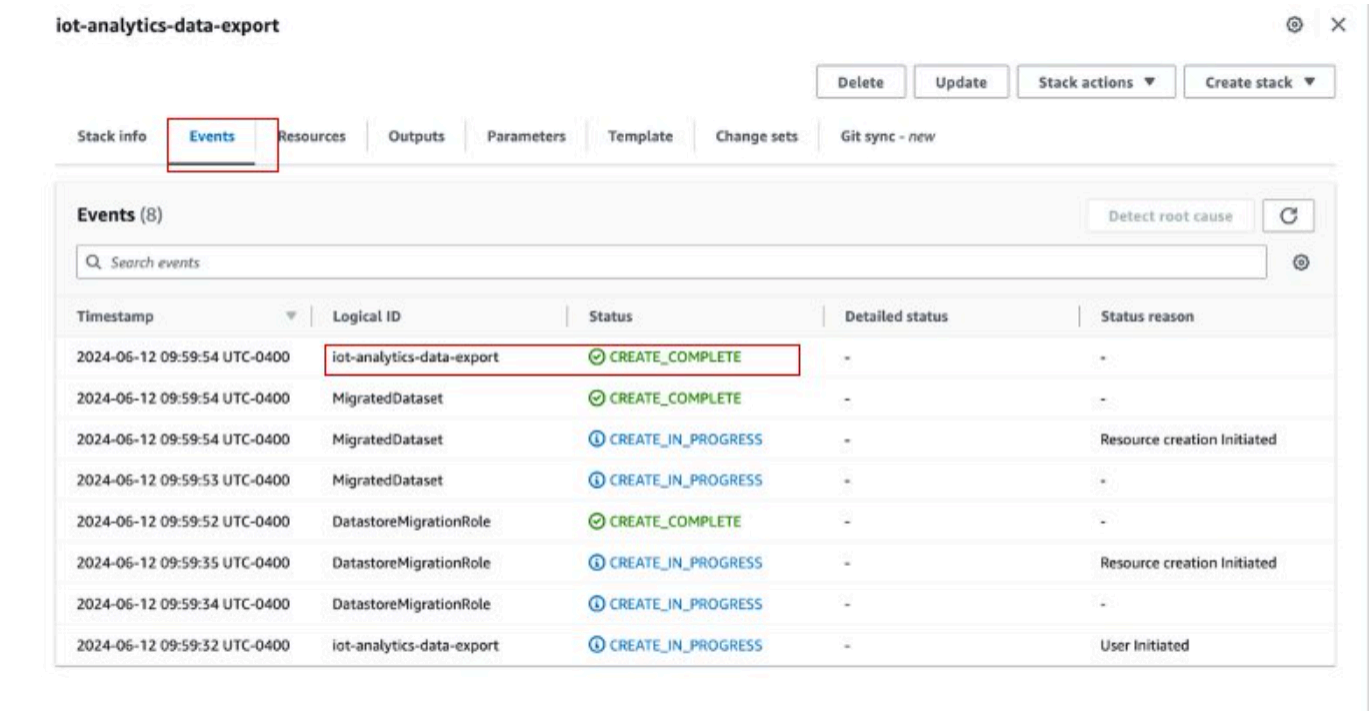
   - DatastoreName: 마이그레이션하려는 AWS IoT 분석 데이터 스토어의 이름입니다.
   - MigrationS3Bucket: 마이그레이션된 데이터가 저장되는 Amazon S3 버킷입니다.
   - MigrationS3BucketPrefix(선택 사항): Amazon S3 버킷의 접두사입니다.
   - TimeRange(선택 사항) : 내보내는 데이터를 필터링하여 지정된 시간 범위에 따라 소스 데이터를 여러 파일로 분할할 수 있는 `SQL WHERE` 절입니다.



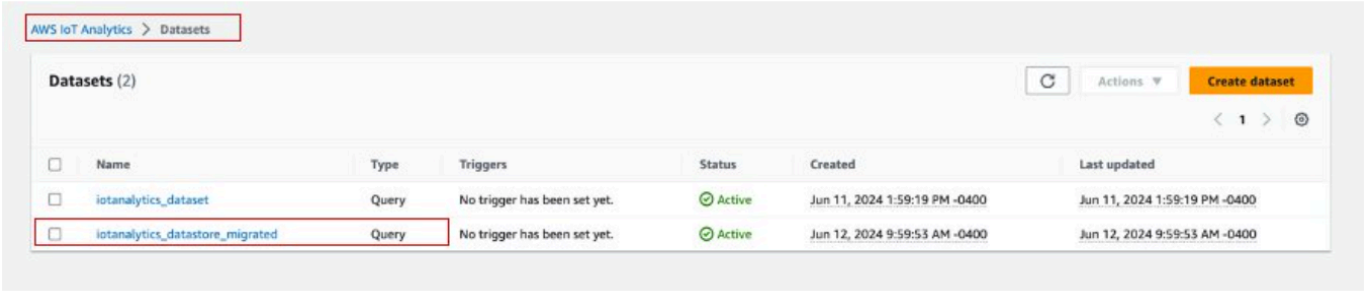
6. 스택 옵션 구성 화면에서 다음을 클릭합니다.
7. 확인란을 선택하여 IAM 리소스 생성을 확인하고 제출을 클릭합니다.



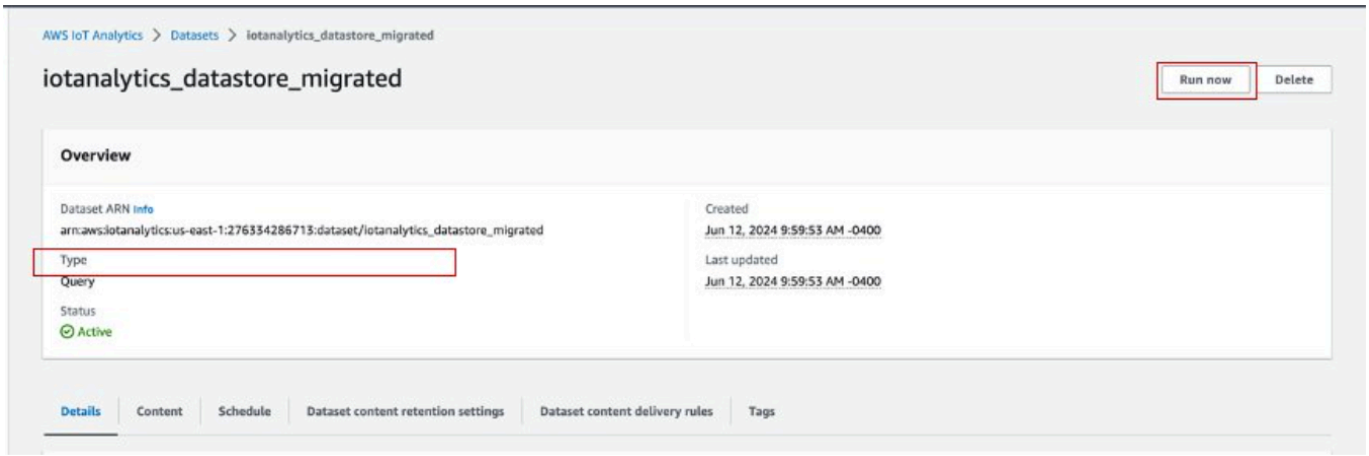
8. 이벤트 탭에서 스택 생성을 검토하여 완료합니다.



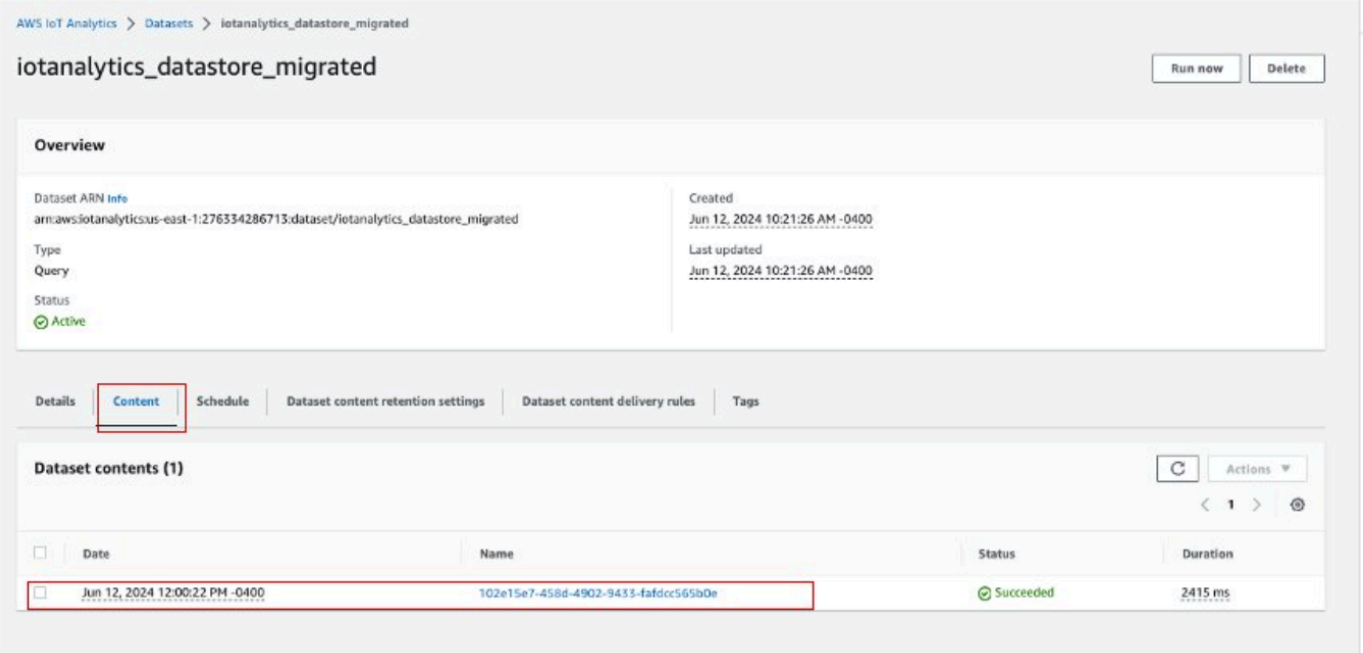
9. 스택이 성공적으로 완료되면 AWS IoT 분석 → 데이터 세트로 이동하여 마이그레이션된 데이터 세트를 확인합니다.



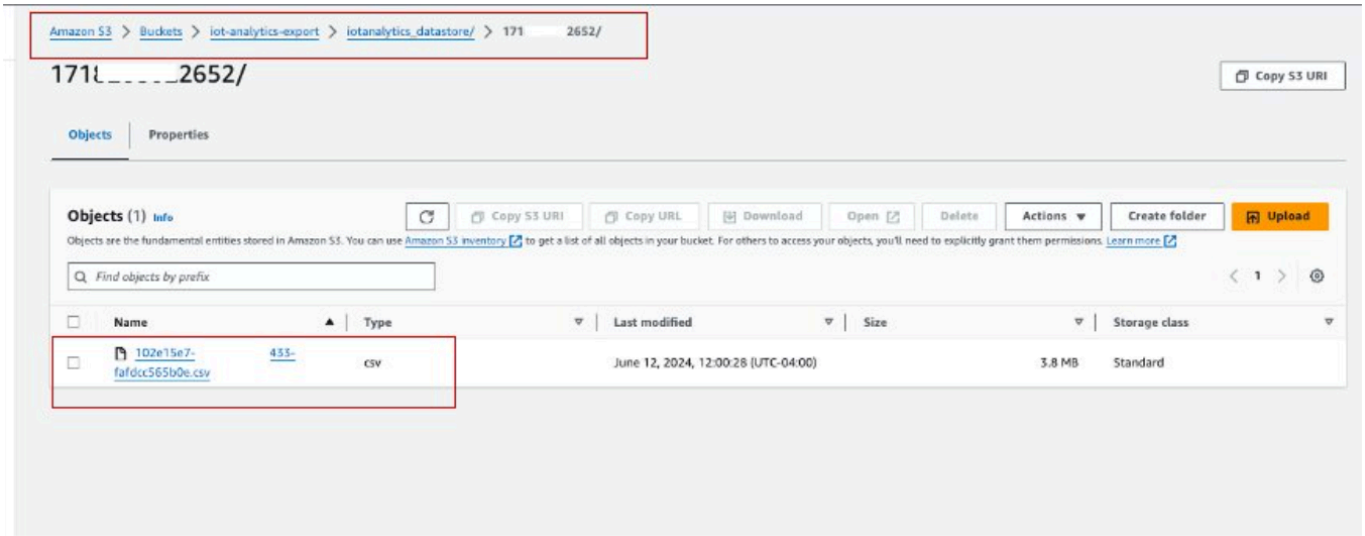
10. 생성된 데이터 세트를 선택하고 지금 실행을 클릭하여 데이터 세트를 내보냅니다.



11. 콘텐츠는 데이터 세트의 콘텐츠 탭에서 볼 수 있습니다.



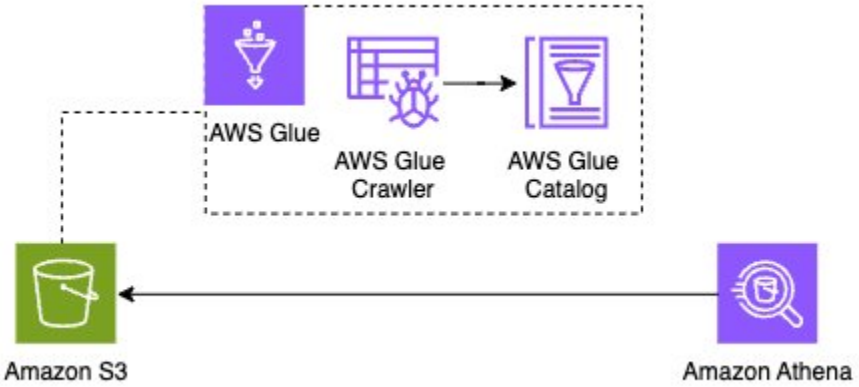
12. 마지막으로 Amazon S3 콘솔에서 iot-analytics-export 버킷을 열어 내보낸 콘텐츠를 검토합니다.



두 패턴 모두에 대해 온디맨드 쿼리 실행

AWS IoT 분석 워크로드를 Amazon Kinesis Data Streams 또는 Amazon Data Firehose로 마이그레이션할 때 AWS Glue 및 Amazon Athena를 활용하면 데이터 분석 프로세스를 더욱 간소화할 수 있습니다.는 데이터 준비 및 변환 AWS Glue 을 간소화하는 반면 Amazon Athena는 데이터를 빠르고 서버리

스 쿼리할 수 있습니다. 또한 AWS IoT 데이터를 분석하기 위한 강력하고 확장 가능하며 비용 효율적인 솔루션을 제공합니다.



요약

AWS IoT 분석 워크로드 AWS IoT 분석 를에서 Amazon Kinesis Data Streams, Amazon S3로 마이그레이션하면 대규모의 복잡한 AWS IoT 데이터를 처리하는 기능이 향상됩니다. 이 아키텍처는 확장 가능하고 내구성이 뛰어난 스토리지와 강력한 분석 기능을 제공하므로 IoT 데이터에서 실시간으로 심층적인 인사이트를 얻을 수 있습니다.

마이그레이션이 완료된 후 예기치 않은 비용을 방지하려면를 사용하여 생성된 리소스를 정리하는 것이 CloudFormation 중요합니다.

데이터 마이그레이션과 AWS IoT 분석 관련된 비용은 요금 페이지를 참조하세요. 불필요한 비용을 방지하려면 완료 시 새로 생성된 데이터 세트를 삭제하는 것이 좋습니다.

전체 데이터 세트 내보내기: 시간 기반 분할 없이 전체 데이터 세트를 내보내려면 AWS IoT 분석 콘솔을 사용하고 그에 따라 콘텐츠 전송 규칙을 설정할 수도 있습니다.

마이그레이션 가이드에 따라 데이터 수집 및 처리 파이프라인을 원활하게 전환하여 지속적이고 안정적인 데이터 흐름을 보장할 수 있습니다. AWS Glue 및 Amazon Athena를 활용하면 데이터 준비 및 쿼리가 더욱 간소화되므로 인프라를 관리하지 않고도 정교한 분석을 수행할 수 있습니다.

이 접근 방식을 사용하면 AWS IoT 분석 노력을 효과적으로 확장할 수 있으므로 증가하는 비즈니스 요구 사항에 더 쉽게 적응하고 AWS IoT 데이터에서 최대 가치를 추출할 수 있습니다.

시작하기 AWS IoT 분석 (콘솔)

이 자습서를 사용하여 IoT 디바이스 데이터에 대한 유용한 인사이트를 검색하는 데 필요한 AWS IoT 분석 리소스(구성 요소라고도 함)를 생성합니다.

Notes

- 다음 자습서에서 대문자를 입력하면 AWS IoT 분석 가 자동으로 대문자를 소문자로 변경합니다.
- AWS IoT 분석 콘솔에는 채널, 파이프라인, 데이터 스토어 및 데이터 세트를 생성하는 원클릭 시작하기 기능이 있습니다. AWS IoT 분석 콘솔에 로그인할 때 이 기능을 찾을 수 있습니다.
  - 이 자습서에서는 AWS IoT 분석 리소스를 생성하는 각 단계를 안내합니다.

아래 지침에 따라 AWS IoT 분석 채널, 파이프라인, 데이터 스토어 및 데이터 세트를 생성합니다. 또한 이 자습서에서는 AWS IoT Core 콘솔을 사용하여 수집될 메시지를 보내는 방법을 보여줍니다 AWS IoT 분석.

주제

AWS IoT 분석 콘솔에 로그인

시작하려면 AWS 계정이 있어야 합니다. 이미 AWS 계정이 있는 경우 https://console.aws.amazon.com/iotanalytics/ 이동합니다.

AWS 계정이 없는 경우 다음 단계에 따라 계정을 생성합니다.

AWS 계정을 생성하려면

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

   등록 절차 중 전화 또는 텍스트 메시지를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

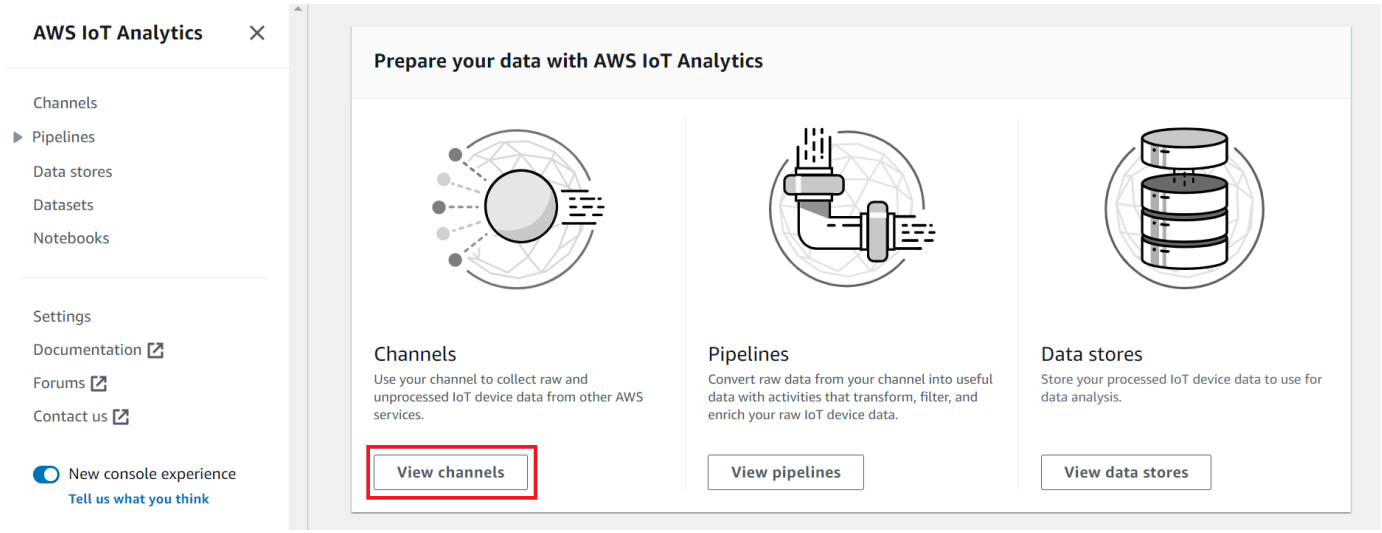
   에 가입하면 AWS 계정AWS 계정 루트 사용자이 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.
3. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iotanalytics/ 이동합니다.

채널 생성

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

채널 생성

1. https://console.aws.amazon.com/iotanalytics/의 AWS IoT 분석로 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 분석에 대해 생성하는 리소스를 식별하는 데 도움이 됩니다.
   c. 다음을 선택합니다.
4. AWS IoT 분석 는 처리되지 않은 원시 IoT 디바이스 데이터를 Amazon Simple Storage Service(Amazon S3) 버킷에 저장합니다. 액세스 및 관리할 수 있는 자체 Amazon S3 버킷을 선택하거나 Amazon S3 버킷을 관리할 AWS IoT 분석 수 있습니다.
   a. 이 튜토리얼에서는 스토리지 유형에서 서비스 관리형 스토리지를 선택합니다.
   b. 원시 데이터를 저장할 기간 선택에서 무기한을 선택합니다.
   c. 다음을 선택합니다.
5. 소스 구성 페이지에서가 메시지 데이터를 수집할 AWS IoT 분석 에 대한 정보를 입력합니다 AWS IoT Core.
   a. AWS IoT Core 주제 필터를 입력합니다. 예: `update/environment/dht1`. 이 튜토리얼 후반부에서는 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. 다음을 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 분석로 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 분석에 대해 생성하는 리소스를 식별하는 데 도움이 됩니다.

   c. 다음을 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. 다음을 선택합니다.

5. AWS IoT 분석 데이터 스토어는 JSON 및 Parquet 파일 형식을 지원합니다. 데이터 스토어 데이터 형식으로 JSON 또는 Parquet을 선택합니다. AWS IoT 분석 지원 파일 유형에 대한 자세한 내용은 파일 형식을 참조하십시오.

다음을 선택합니다.

6. (선택 사항) 데이터 스토어에서 사용자 지정 파티션을 AWS IoT 분석 지원하므로 정리된 데이터를 쿼리하여 지연 시간을 개선할 수 있습니다. 지원되는 사용자 지정 파티션에 대한 자세한 내용은 사용자 지정 파티션 단원을 참조하십시오.

   다음을 선택합니다.

7. 선택 사항을 검토한 다음 데이터 스토어 생성을 선택합니다.
8. 새 데이터 스토어가 데이터 스토어 페이지에 나타나는지 확인하십시오.

파이프라인 생성

채널을 데이터 스토어에 연결하려면 파이프라인을 생성해야 합니다. 기본 파이프라인은 데이터를 수집하는 채널만 지정하고 메시지가 전송되는 데이터 스토어를 식별합니다. 자세한 내용은 파이프라인 활동을 참조하십시오.

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

이 단계에 따라 파이프라인을 생성합니다.

파이프라인을 만들려면

1. https://console.aws.amazon.com/iotanalytics/의 AWS IoT 분석로 데이터 준비 섹션에서 파이프라인 보기를 선택합니다.

   > **Tip**
   >
   > 탐색 창에서 파이프라인을 선택할 수도 있습니다.

2. 파이프라인 페이지에서 파이프라인 생성을 선택합니다.
3. 파이프라인에 대한 세부 정보를 입력합니다.

   a. 파이프라인 ID 및 소스 설정에 파이프라인 이름을 입력합니다.
   b. 파이프라인의 소스를 선택합니다.이 소스는 파이프라인이 메시지를 읽을 AWS IoT 분석 채널입니다.
   c. 처리된 메시지 데이터가 저장되는 데이터 스토어인 파이프라인의 출력을 지정합니다.
   d. (선택 사항) 태그에서 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 확인하십시오.

> **Note**
>
> 다음을 수행할 수 있도록 AWS IoT 분석 리소스를 생성했습니다.
>
> - 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 분석 SQL 데이터 세트를 생성하여 IoT 디바이스에 대한 유용한 인사이트를 찾습니다.

데이터세트 생성

> **Note**
>
> 데이터 세트는 일반적으로 테이블 형식으로 구성되거나 구성되지 않을 수 있는 데이터 모음입니다. 반대로는 SQL 쿼리를 데이터 스토어의 데이터에 적용하여 데이터 세트를 AWS IoT 분석 생성합니다.

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 표현식과 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
   ```

   에서 지원되는 SQL 기능에 대한 자세한 내용은 섹션을 AWS IoT 분석참조하세요의 SQL 표현식 AWS IoT 분석.

   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 `__dt`만 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 참조하십시오.

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

6. (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.

7. AWS IoT 분석 는이 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.

8. (선택사항) 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.

9. 선택 사항을 검토한 다음 데이터 세트 생성을 선택합니다.

10. 데이터 세트 페이지에 새 데이터 세트가 나타나는지 확인하십시오.

를 사용하여 메시지 데이터 전송 AWS IoT

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

- AWS IoT 메시지 브로커를 사용합니다.
- AWS IoT 분석 BatchPutMessage API 작업을 사용합니다.

다음 단계에서는가이 데이터를 수집할 AWS IoT 분석 수 있도록 콘솔의 AWS IoT 메시지 브로커 AWS IoT Core 에서 메시지 데이터를 전송합니다.

> **Note**
>
> 메시지에 대한 주제 이름을 생성할 때 다음 사항에 유의하십시오.
>
> - 주제 이름은 대/소문자를 구분하지 않습니다. 동일한 페이로드에서 `example` 및 `EXAMPLE`라는 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 마십시오.
> - AWS IoT Core 는 AWS 계정 또는 AWS 리전 간에 메시지를 보낼 수 없습니다.

를 사용하여 메시지 데이터를 보내려면 AWS IoT

1. AWS IoT 콘솔에 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/dht1`를 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. 게시를 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/ 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 분석 는 데이터 세트 콘텐츠 페이지에 Jupyter Notebook의 HTML 부분을 포함할 수 있습니다. 자세한 내용은 콘솔을 사용하여 AWS IoT 분석 데이터 시각화 단원을 참조하십시오.

데이터 탐색

데이터 저장, 분석 및 시각화를 위한 몇 가지 옵션이 있습니다.

Amazon Simple Storage Service(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 작업의 `contentDeliveryRules::destination::s3DestinationConfiguration` 필드를 참조하십시오.

AWS IoT Events

데이터 세트 콘텐츠를 입력으로 전송할 수 있습니다.이 AWS IoT Events서비스를 사용하면 디바이스 또는 프로세스에서 장애 또는 작업 변경을 모니터링하고 이러한 이벤트가 발생할 때 추가 작업을 시작할 수 있습니다.

이렇게 하려면 CreateDataset 작업을 사용하여 데이터 세트를 생성하고 필드에 AWS IoT Events 입력을 지정합니다`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. 또한를 실행할 수 있는 AWS IoT 분석 권한을 부여하는 역할`roleArn`의 도 지정해야 합니다`iotevents:BatchPutMessage`. 데이터 세트 콘텐츠가 생성될 때마다 AWS IoT 분석 는 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 다음는 다음과 같은 필드가 포함된 메시지를 AWS IoT 분석 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, 중 하나 이상whodt)를 인식하는 AWS IoT Events 입력을 생성하고 이벤트에서 이러한 입력 필드를 사용하여 작업을 트리거하거나 내부 변수를 설정하는 감지기 모델을 생성하는 AWS IoT Events 것이 좋습니다.

Jupyter Notebook

Jupyter Notebook은 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 분석 는 Amazon SageMaker AI 노트북 인스턴스를 사용하여 Jupyter Notebook을 호스팅합니다. 노트북 인스턴스를 생성하기 전에 AWS IoT 분석 와 Amazon SageMaker AI 간의 관계를 생성해야 합니다.

1. SageMaker AI 콘솔로 이동하여 노트북 인스턴스를 생성합니다.
   a. 세부 정보를 입력하고 새 역할 생성을 선택합니다. 역할의 ARN을 기록해 둡니다.
   b. 노트북 인스턴스를 생성합니다.
2. IAM 콘솔로 이동하여 SageMaker AI 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 새로 생성된 역할에 데이터 세트를 읽을 수 있는 권한이 부여됩니다 AWS IoT 분석.

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 SageMaker AI에서 생성한 노트북 인스턴스를 선택합니다.
4. 선택 사항을 검토한 후 노트북 생성을 선택합니다.
5. 노트북 페이지의 Amazon SageMaker AI 콘솔에서 노트북 인스턴스가 열립니다.

노트북 템플릿

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

AWS IoT 분석 콘솔에서 다음 노트북 템플릿을 찾을 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.

시작하기 AWS IoT 분석

이 섹션에서는를 사용하여 디바이스 데이터를 수집, 저장, 처리 및 쿼리하는 데 사용하는 기본 명령에 대해 설명합니다 AWS IoT 분석. 여기에 표시된 예제에서는 AWS Command Line Interface ()를 사용합니다AWS CLI. 에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 AWS CLI참조하세요. 사용 가능한 CLI 명령에 대한 자세한 내용은 AWS Command Line Interface 참조의 iot를 AWS IoT참조하세요.

> **Important**
>
> `aws iotanalytics` 명령을 사용하여를 AWS IoT 분석 사용하여와 상호 작용합니다 AWS CLI. `aws iot` 명령을 사용하여 AWS CLI를 사용하는 IoT 시스템의 다른 부분과 상호 작용할 수 있습니다.

> **Note**
>
> 다음 예제에서 AWS IoT 분석 개체(채널, 데이터 세트, 데이터 스토어 및 파이프라인)의 이름을 입력할 때 사용하는 대문자는 시스템에서 자동으로 소문자로 변경됩니다. 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 수집하려는 경우 AWS IoT 규칙 엔진 규칙을 생성하여이 채널로 메시지를 보낼 AWS IoT 분석수 있습니다. 이 내용은 에 데이터 수집 AWS IoT 분석의 뒷부분에서 설명합니다. 채널로 데이터를 가져오는 또 다른 방법은 AWS IoT 분석 명령을 사용하는 것입니다`BatchPutMessage`.

이미 생성한 채널을 나열하려면:

```
aws iotanalytics list-channels
```

채널에 관한 추가 정보를 가져오려면.

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

처리되지 않은 채널 메시지는에서 관리하는 Amazon S3 버킷 AWS IoT 분석또는 사용자가 관리하는 버킷에 저장됩니다. channelStorage 파라미터를 사용하여 저장할 버킷을 지정합니다. 기본값은 서비스 관리형 Amazon S3 버킷입니다. 관리하는 Amazon S3 버킷에 채널 메시지를 저장하도록 선택한 경우 사용자를 대신하여 Amazon S3 버킷에서 (버킷 위치 확인) s3:GetBucketLocation (스토어), s3:PutObject (s3:GetObject읽기), s3:ListBucket (재처리) 작업을 수행할 수 있는 AWS IoT 분석 권한을 부여해야 합니다.

Example

JSON

```
{
    "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:*:*:*your-iot-analytics-bucket*",
                "arn:aws:s3:*:*:*your-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 재처리를 참조하십시오.

데이터 스토어 생성

데이터 스토어는 메시지를 수신하고 저장합니다. 데이터 스토어는 데이터베이스가 아니라 확장 및 쿼리가 가능한 메시지 리포지토리입니다. 여러 데이터 스토어를 생성하여 서로 다른 디바이스 또는 위치에서 오는 메시지를 저장하거나가 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 분석 리소스에 대한 Amazon S3 정책

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

관리하는 Amazon S3 버킷에 데이터 스토어 메시지를 저장하도록 선택한 경우 Amazon S3 버킷에서 이러한 작업을 수행할 수 있는 AWS IoT 분석 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 부여하는 Amazon S3 버킷 정책을 설정합니다. AWS IoT 분석

> **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 제한됩니

다. 혼동된 대리자 문제에 관한 자세한 내용은 the section called “교차 서비스 혼동된 대리인 방지”를 참조하십시오.

다음은 이러한 필수 권한을 부여하는 버킷 정책의 예입니다.

JSON

```
{
    "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:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset",
                        "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastore"
                    ]
                }
            }
```

```
        }
    ]
}
```

자세한 내용은 Amazon Athena 사용 설명서의 크로스 계정 액세스를 참조하십시오.

> **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 재처리를 참조하십시오.

파일 형식

AWS IoT 분석 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 분석 데이터 스토어의 파일 형식을 구성하려면 데이터 스토어를 생성할 때 `FileFormatConfiguration` 객체를 사용할 수 있습니다.

`fileFormatConfiguration`

파일 형식의 구성 정보를 포함합니다. AWS IoT 분석 데이터 스토어는 JSON 및 Parquet을 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 Glue 개발자 안내서의 일반적인 데이터 유형을 참조하십시오.

길이 제한: 1~131072자

AWS IoT 분석 는 `DECIMAL(`*`precision`*`, `*`scale`*`)` - 를 제외하고 Amazon Athena의 데이터 형식 페이지에 나열된 모든 데이터 형식을 지원합니다*precision*.

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/ 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 분석에 대해 생성하는 리소스를 식별하는 데 도움이 됩니다.

c. 다음을 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. 다음을 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 분석 quotas를 참조하십시오.

- 기존 열의 데이터 유형을 변경할 수 있습니다. 지원되는 데이터 유형에 대한 자세한 내용을 알아보려면 AWS Glue 개발자 안내서의 일반적인 데이터 유형을 참조하십시오.

  > **Note**
  > 데이터 스토어를 생성한 후에는 기존 열의 데이터 유형을 변경할 수 없습니다.

- 기존 열을 제거하려면 열 제거를 선택합니다.

f. 다음을 선택합니다.

7. (선택 사항) 데이터 스토어에서 사용자 지정 파티션을 AWS IoT 분석 지원하므로 잘라낸 데이터를 쿼리하여 지연 시간을 개선할 수 있습니다. 지원되는 사용자 지정 파티션에 대한 자세한 내용은 사용자 지정 파티션 단원을 참조하십시오.

   다음을 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 확인하십시오.

사용자 지정 파티션

AWS IoT 분석 는 데이터 분할을 지원하므로 데이터 스토어에서 데이터를 구성할 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 활성화하십시오. 하나 이상의 데이터 파티션 차원을 지정하고 분할된 데이터 스토어를 AWS IoT 분석 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 분석 콘솔에 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. 태그는 생성한 리소스를 식별하는 데 도움이 될 수 있습니다 AWS IoT 분석.

   c. 다음을 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. 다음을 선택합니다.
6. 데이터 형식 구성 페이지에서 데이터 레코드의 구조 및 형식을 정의합니다.

   a. 데이터 스토어 데이터 형식 분류에서 JSON 또는 Parquet을 선택합니다. AWS IoT 분석 지원되는 파일 유형에 대한 자세한 내용은 섹션을 참조하세요파일 형식.

   > **Note**
   >
   > 데이터 스토어를 생성한 후에는 이 파일 형식을 변경할 수 없습니다.

   b. 다음을 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 선택하고이 데이터 스토어에 대한 메시지를 수집하는 AWS IoT 분석 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 분석 는 선택한 메시지 샘플 속성의 이름을 속성 파티션 차원의 식별자로 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 경우는 이름이 , `__year`, `__month`, 인 다음 네 가지 `__day`차원을 AWS IoT 분석 생성합니다`__hour`.
- 정렬 - 파티션 차원을 재배열하여 쿼리 지연 시간을 개선합니다.

타임스탬프 형식의 경우 메시지 데이터에서 수집된 타임스탬프와 일치시켜 타임스탬프 파티션의 형식을 지정합니다. AWS IoT 분석 나열된 형식 옵션 중 하나를 선택하거나 데이터 형식과 일치하는 옵션을 지정할 수 있습니다. 날짜 시간 포맷터 지정에 대해 자세히 알아보십시오.

메시지 속성이 아닌 새 차원을 추가하려면 새 파티션 추가를 선택합니다.

e. 다음을 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 확인하십시오.

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 활동을 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
```

에 데이터 수집 AWS IoT 분석

쿼리할 수 있는 데이터 스토어에 데이터를 저장하는 파이프라인으로 데이터를 라우팅하는 채널이 있으면 AWS IoT 분석로 메시지 데이터를 보낼 준비가 된 것입니다. 여기에서는 AWS IoT 분석로 데이터를 가져오는 두 가지 방법을 보여줍니다. 메시지 브로커 또는 `BatchPutMessage` API를 사용하여 AWS IoT 메시지를 보낼 수 있습니다 AWS IoT 분석 .

항목

- AWS IoT 메시지 브로커 사용
- BatchPutMessage API 사용

AWS IoT 메시지 브로커 사용

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

IAM 역할 생성

AWS IoT 메시지를 AWS IoT 분석 채널로 라우팅하려면 규칙을 설정합니다. 그러나 먼저 AWS IoT 분석 채널로 메시지 데이터를 전송할 수 있는 권한을 해당 규칙에게 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.

JSON

```
{
    "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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotanalytics:BatchPutMessage",
            "Resource": [
                "arn:aws:iotanalytics:us-east-1:111122223333:channel/your-
channel"
            ]
        }
    ]
}
```

AWS IoT 규칙 생성

채널로 메시지를 보내는 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/test`를 라우팅해야 하는 메시지의 MQTT 주제로 바꿉니다. 채널 이름과 역할을 이전 단원에서 생성한 이름과 역할로 바꿉니다.

에 MQTT 메시지 전송 AWS IoT 분석

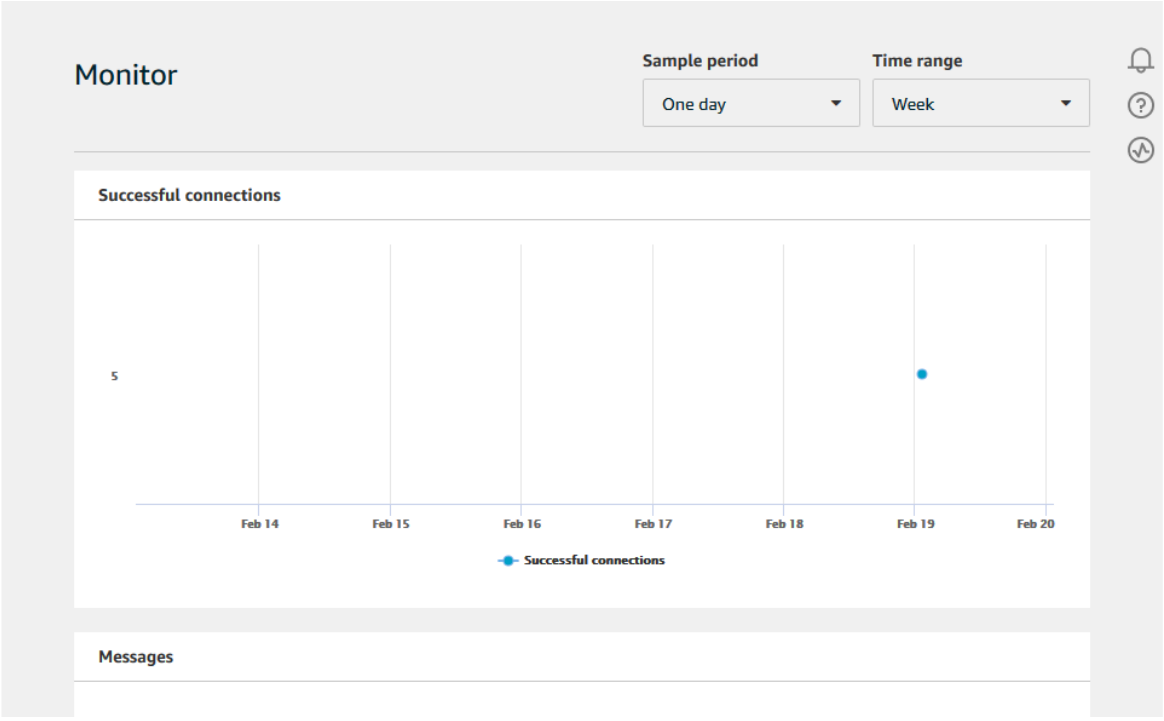
규칙을 채널에 조인하고, 채널을 파이프라인에 조인하고, 파이프라인을 데이터 스토어에 조인한 후에는 규칙과 일치하는 모든 데이터가 이제 쿼리할 준비가 된 데이터 스토어 AWS IoT 분석 로 전달됩니다. 이를 테스트하기 위해 AWS IoT 콘솔을 사용하여 메시지를 보낼 수 있습니다.

> **Note**
>
> 보내는 메시지 페이로드(데이터)의 필드 이름입니다 AWS IoT 분석.
>
> - 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.

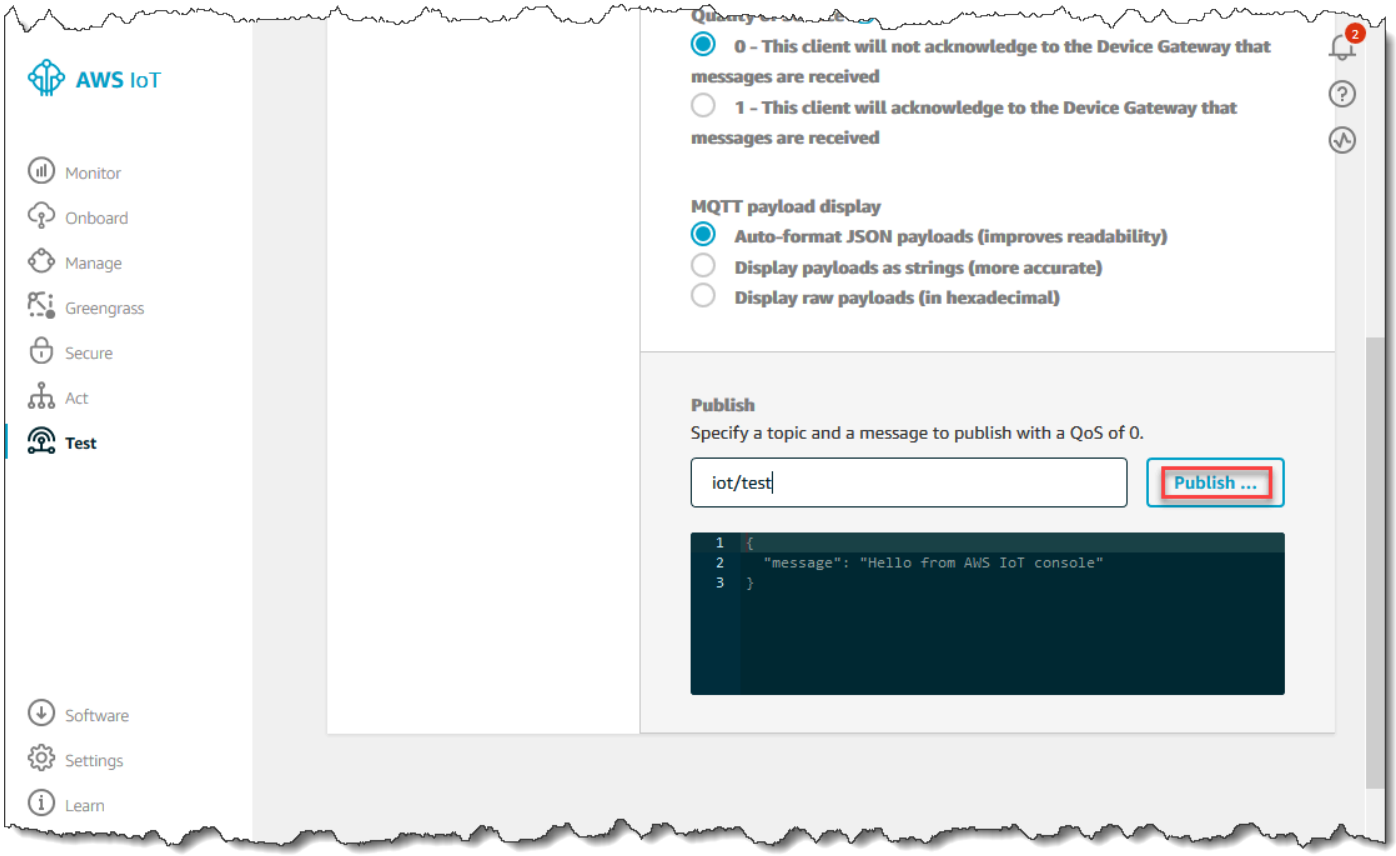
1. AWS IoT 콘솔의 왼쪽 탐색 창에서 테스트를 선택합니다.



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

메시지 데이터를 로 가져오는 또 다른 방법은 `BatchPutMessage` API 명령을 사용하는 AWS IoT 분석 것입니다. 이 방법을 사용하면 특정 주제가 포함된 메시지를 채널로 라우팅하는 AWS IoT 규칙을 설정할 필요가 없습니다. 그러나 채널로 데이터/메시지를 전송하는 디바이스는 AWS SDK로 생성된 소프트웨어를 실행할 수 있거나를 사용하여를 호출 AWS CLI 할 수 있어야 합니다`BatchPutMessage`.

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

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

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
```

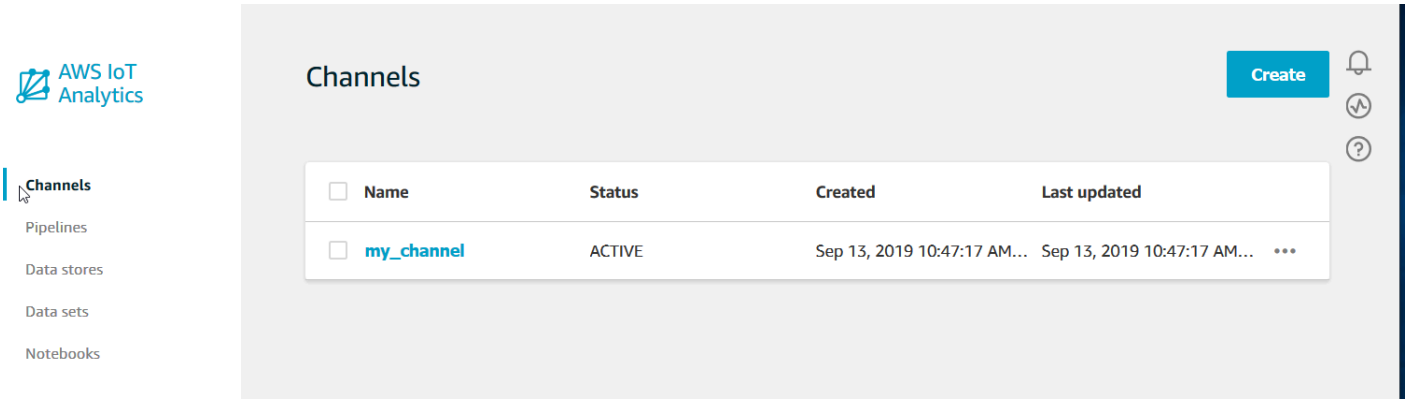
오류가 없다면, 다음과 같은 출력을 확인할 수 있습니다.

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

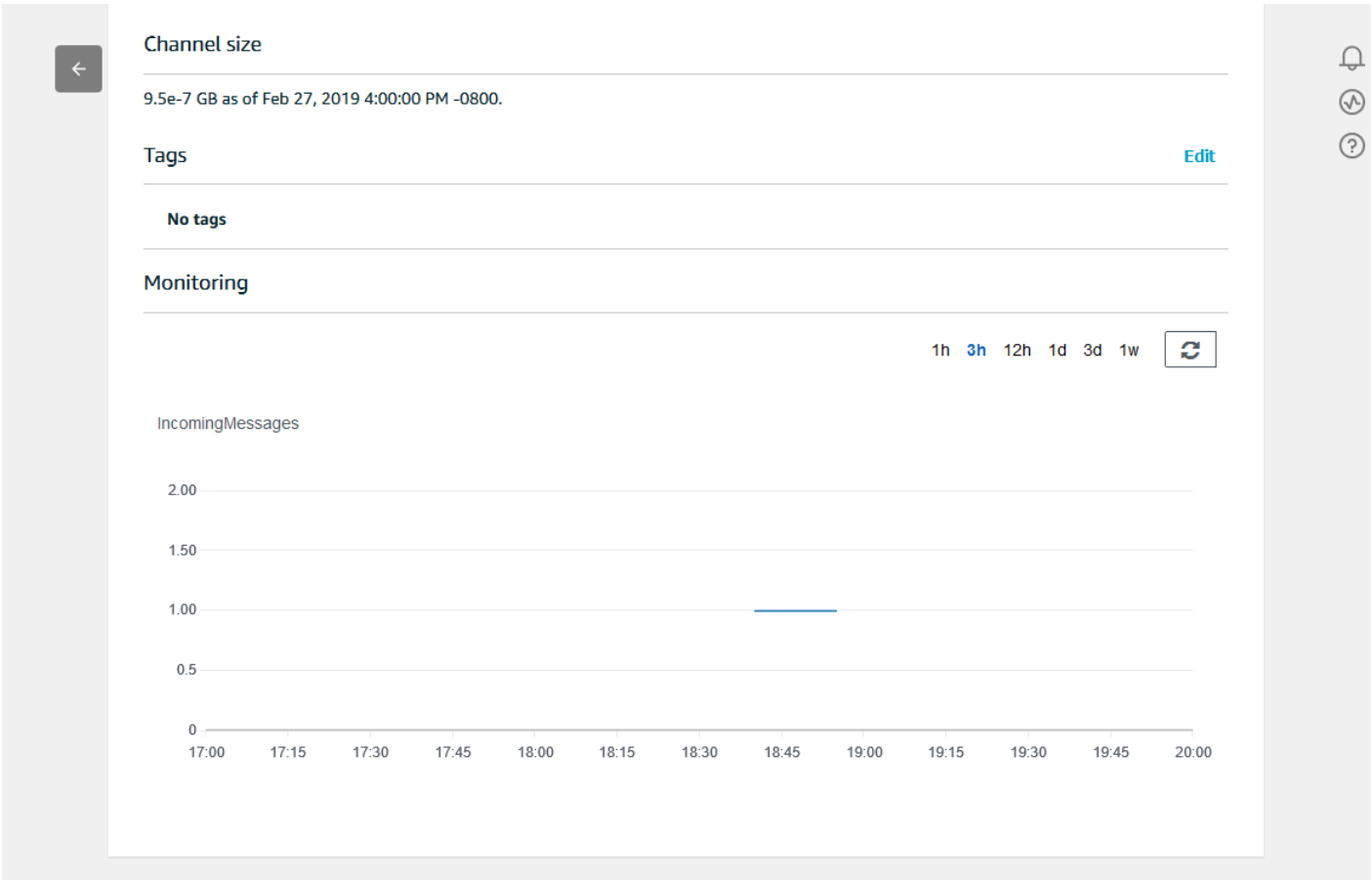
수집된 데이터 모니터링

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

1. AWS IoT 분석 콘솔의 왼쪽 탐색 창에서 준비를 선택하고(필요한 경우) 채널을 선택한 다음 이전에 생성한 채널 이름을 선택합니다.

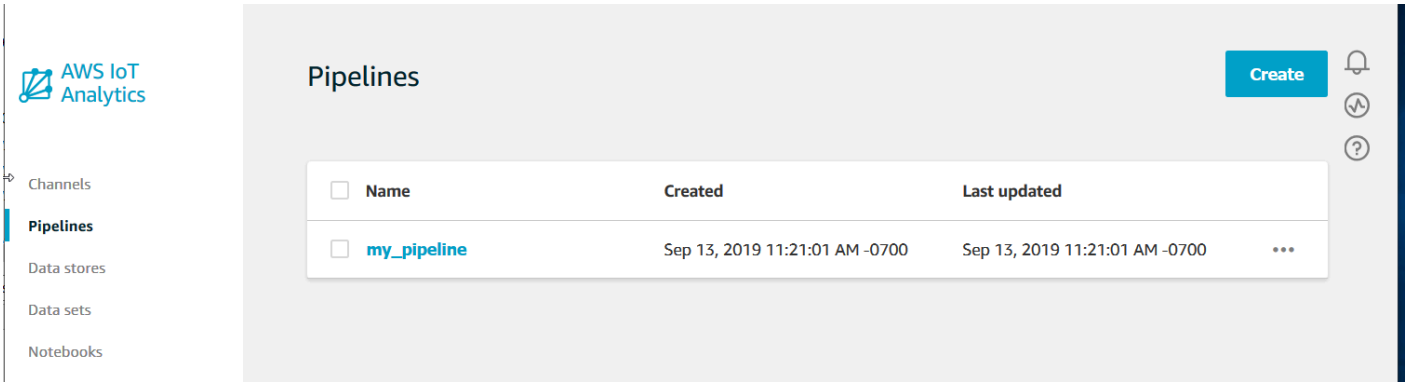


2. 채널 세부 정보 페이지에서 모니터링 섹션까지 아래로 스크롤합니다. 기간 표시기(1시간, 3시간, 12시간, 1일, 3일, 1주) 중 하나를 선택하여 필요에 따라 표시된 기간을 조정합니다. 지정된 기간 동안 이 채널에서 수집된 메시지 수를 나타내는 그래프 선이 표시되어야 합니다.

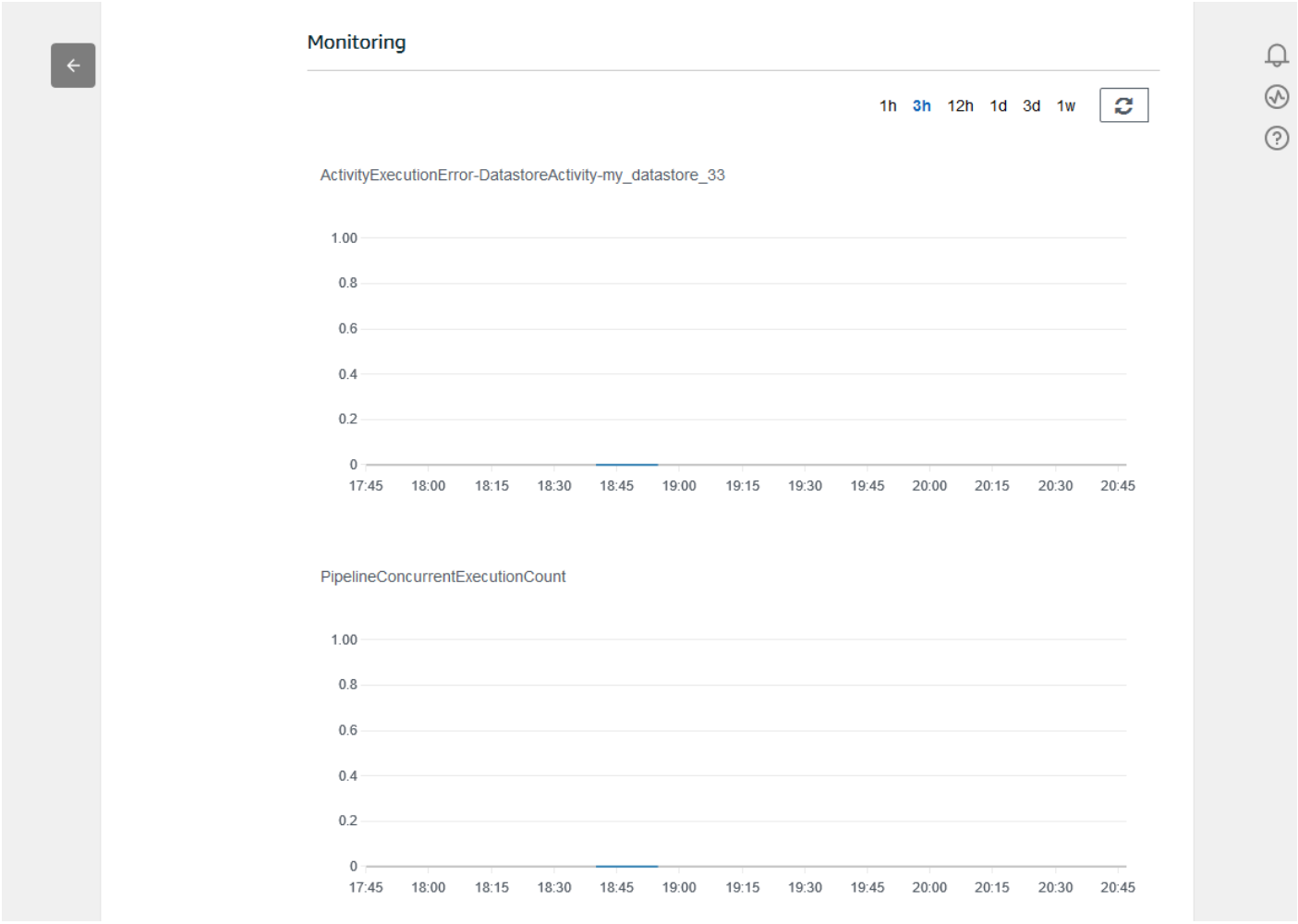


파이프라인 활동 실행을 확인하기 위한 유사한 모니터링 기능이 있습니다. 파이프라인의 세부 정보 페이지에서 활동 실행 오류를 모니터링할 수 있습니다. 파이프라인의 일부로 활동을 지정하지 않은 경우 실행 오류가 0개 표시되어야 합니다.

1. AWS IoT 분석 콘솔의 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.는 데이터를 쿼리하여 분석 질문에 답할 AWS IoT 분석 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 표현식과 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 분석 데이터 탐색

AWS IoT 분석 데이터를 저장, 분석 및 시각화하는 몇 가지 옵션이 있습니다.

이 페이지의 주제:

- Amazon S3
- AWS IoT Events
- Quick Suite
- Jupyter Notebook

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. CreateDataset의 `contentDeliveryRules::destination::s3DestinationConfiguration` 필드를 참조하십시오.

AWS IoT Events

데이터 세트 콘텐츠를 입력으로에 전송할 수 있습니다.이 AWS IoT Events서비스를 사용하면 디바이스 또는 프로세스에서 장애 또는 작업 변경을 모니터링하고 이러한 이벤트가 발생할 때 추가 작업을 트리거할 수 있습니다.

이렇게 하려면 CreateDataset을 사용하여 데이터 세트를 생성하고 필드에 AWS IoT Events 입력을 지정합니다`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. 또한 "iotevents:BatchPutMessage"를 실행할 수 있는 AWS IoT 분석 권한을 부여하는 역할`roleArn`의를 지정해야 합니다. 데이터 세트의 콘텐츠가 생성될 때마다 AWS IoT 분석 는 각 데이터 세트 콘텐츠 항목을 지정된 AWS IoT Events 입력에 메시지로 전송합니다. 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"
...
```

그러면 AWS IoT 분석 가 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`, 중 하나 이상`dt`)를 인식하는 AWS IoT Events 입력을 생성하고 이벤트에서 이러한 입력 필드를 사용하여 작업을 트리거하거나 내부 변수를 설정하는 AWS IoT Events 감지기 모델을 생성하려고 `who`합니다.

Quick Suite

AWS IoT 분석 는 Quick Suite와의 직접 통합을 제공합니다. Quick Suite는 시각화를 구축하고, 임시 분석을 수행하고, 데이터에서 비즈니스 인사이트를 빠르게 얻는 데 사용할 수 있는 빠른 비즈니스 분석 서비스입니다. Quick Suite를 사용하면 조직이 수십만 명의 사용자로 확장할 수 있으며 강력한 인 메모리 엔진(SPICE)을 사용하여 응답 성능을 제공할 수 있습니다. Quick Suite는 이러한 리전에서 사용할 수 있습니다.

Jupyter Notebook

AWS IoT 분석 Jupyter Notebook은 데이터 세트를 직접 사용하여 고급 분석 및 데이터 탐색을 수행할 수도 있습니다. Jupyter Notebook 오픈 소스 솔루션입니다. http://jupyter.org/install.html에서 다운로드하여 설치할 수 있습니다. Amazon 호스팅 노트북 솔루션인 SageMaker AI와의 추가 통합도 사용할 수 있습니다.

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 [지정 안 함]	retentionPeriod: unlimited = TRUE, numberOfDays = 설정 안 함	retentionPeriod: 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.

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

메시지 페이로드 구문

보내는 메시지 페이로드(데이터)의 필드 이름 AWS IoT 분석:

- 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 SiteWise 는 대규모 산업 장비에서 데이터를 수집, 모델링, 분석 및 시각화하는 데 사용할 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 소비할 산업 장비 데이터와 데이터를 복잡한 지표로 처리하는 방법을 정의할 수 있습니다. AWS 클라우드에서 데이터를 수집하고 처리하도록 자산 모델을 구성할 수 있습니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서를 참조하세요.

AWS IoT 분석 는와 통합되어 AWS IoT SiteWise 데이터에 대한 SQL 쿼리 AWS IoT SiteWise 를 실행하고 예약할 수 있습니다. AWS IoT SiteWise 데이터 쿼리를 시작하려면 AWS IoT SiteWise 사용 설명서의 스토리지 설정 구성의 절차에 따라 데이터 스토어를 생성합니다. 그런 다음 AWS IoT SiteWise 데이터로 데이터 세트 생성(콘솔) 또는의 단계에 따라 AWS IoT 분석 데이터 세트를 AWS IoT SiteWise 데이터로 데이터 세트 생성(AWS CLI) 생성하고 산업 데이터에 대해 SQL 쿼리를 실행합니다.

항목

- AWS IoT SiteWise 데이터로 AWS IoT 분석 데이터 세트 생성
- 데이터 세트 콘텐츠 액세스
- 자습서:에서 AWS IoT SiteWise 데이터 쿼리 AWS IoT 분석

AWS IoT SiteWise 데이터로 AWS IoT 분석 데이터 세트 생성

AWS IoT 분석 데이터 세트에는 지정한 날짜 및 시간에 쿼리를 반복하는 선택적 일정과 함께 데이터 스토어의 데이터를 쿼리하는 데 사용하는 SQL 문 및 표현식이 포함되어 있습니다. 이 일정 옵션은 Amazon CloudWatch 일정 표현식과 유사한 표현식을 사용하여 생성할 수 있습니다.

> **Note**
>
> 데이터 세트는 일반적으로 테이블 형식으로 구성되거나 구성되지 않을 수 있는 데이터 모음입니다. 반대로는 SQL 쿼리를 데이터 스토어의 데이터에 적용하여 데이터 세트를 AWS IoT 분석 생성합니다.

다음 단계에 따라 AWS IoT SiteWise 데이터에 대한 데이터 세트 생성을 시작합니다.

항목

- AWS IoT SiteWise 데이터로 데이터 세트 생성(콘솔)
- AWS IoT SiteWise 데이터로 데이터 세트 생성(AWS CLI)

AWS IoT SiteWise 데이터로 데이터 세트 생성(콘솔)

다음 단계에 따라 AWS IoT 분석 콘솔에서 AWS IoT SiteWise 데이터에 대한 데이터 세트를 생성합니다.

데이터세트 생성

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
      ```

      에서 지원되는 SQL 기능에 대한 자세한 내용은 섹션을 AWS IoT 분석참조하세요의 SQL 표현식 AWS IoT 분석. 또는 데이터에 대한 통찰력을 제공할 수 있는 통계 쿼리의 예인 자습서: 에서 AWS IoT SiteWise 데이터 쿼리 AWS IoT 분석 항목을 참조하십시오.

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

      > **Note**
      >
      > Amazon Athena 는 실행 중인 쿼리의 최대 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 참조하십시오.

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

6. (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**
   >
   > 의 데이터는 AWS IoT SiteWise 6 AWS IoT 분석 시간마다 로 수집됩니다. 빈도를 6시간 이상으로 선택하는 것이 좋습니다.

   빈도에서 옵션을 선택하고 다음을 선택합니다.

7. AWS IoT 분석 는이 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.

8. (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.

9. 선택 사항을 검토한 다음 데이터 세트 생성을 선택합니다.

10. 데이터 세트 페이지에 새 데이터 세트가 나타나는지 확인하십시오.

AWS IoT SiteWise 데이터로 데이터 세트 생성(AWS CLI)

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

여기에 표시된 예제에서는 AWS Command Line Interface ()를 사용합니다AWS CLI. 에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 AWS CLI참조하세요. 사용 가능한 CLI 명령에 대한 자세한 AWS IoT 분석내용은 AWS Command Line Interface 참조의 iotanalytics를 참조하세요.

데이터세트 생성

1. 다음 `create-dataset` 명령을 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"
            }
        }
    ]
}
```

에서 지원되는 SQL 기능에 대한 자세한 내용은 섹션을 AWS IoT 분석참조하세요의 SQL 표현식 AWS IoT 분석. 또는 데이터에 대한 통찰력을 제공할 수 있는 통계 쿼리의 예인 자습서:에서 AWS IoT SiteWise 데이터 쿼리 AWS IoT 분석 항목을 참조하십시오.

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.

항목

- 에서 데이터 세트 콘텐츠 액세스 AWS IoT 분석 (콘솔)
- AWS IoT 분석 (AWS CLI)에서 데이터 세트 콘텐츠 액세스

에서 데이터 세트 콘텐츠 액세스 AWS IoT 분석 (콘솔)

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

AWS IoT 분석 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 분석 (AWS CLI)에서 데이터 세트 콘텐츠 액세스

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

여기에 표시된 예제에서는 AWS Command Line Interface ()를 사용합니다AWS CLI. 에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서를 AWS CLI참조하세요. 사용 가능한 CLI 명령에 대한 자세한 AWS IoT 분석내용은 AWS Command Line Interface 참조의 iotanalytics를 참조하세요.

AWS IoT 분석 데이터 세트 결과에 액세스하려면(AWS CLI)

1. 다음 `get-dataset-content` 명령을 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.

자습서:에서 AWS IoT SiteWise 데이터 쿼리 AWS IoT 분석

이 자습서에서는에서 AWS IoT SiteWise 데이터를 쿼리하는 방법을 보여줍니다 AWS IoT 분석. 이 자습서에서는 풍력 발전소에 대한 샘플 데이터 세트를 AWS IoT SiteWise 제공하는의 데모 데이터를 사용합니다.

> **Important**
>
> 이 데모가 생성하고 사용하는 리소스에 대한 요금이 청구됩니다.

항목

사전 조건

이 튜토리얼을 이해하려면 다음 리소스가 필요합니다.

- AWS IoT SiteWise 및를 시작하려면 AWS 계정이 있어야 합니다 AWS IoT 분석. 계정이 없는 경우에는 계정 AWS 생성하기에서 절차에 따라 계정을 만드십시오.
- Windows, macOS, Linux 또는 Unix를 실행하며 AWS Management Console에 액세스할 수 있는 개발 컴퓨터. 자세한 내용은 AWS Management Console시작하기를 참조하세요.

- AWS IoT SiteWise AWS IoT SiteWise 모델과 자산을 정의하고 풍력 발전소 장비의 데이터를 나타내는 데이터를 스트리밍하는 데이터입니다. 데이터를 생성하려면 AWS IoT SiteWise 사용 설명서의 AWS IoT SiteWise 데모 생성 단계를 따르세요.
- 관리하는 기존 데이터 스토어의 AWS IoT SiteWise 데모 풍력 발전소 장비 데이터입니다. AWS IoT SiteWise 데이터에 대한 데이터 스토어를 생성하는 방법에 대한 자세한 내용은 AWS IoT SiteWise 사용 설명서의 스토리지 설정 구성을 참조하세요.

> **Note**
>
> AWS IoT SiteWise 메타데이터는 생성 직후 AWS IoT SiteWise 데이터 스토어에 표시되지만 원시 데이터가 표시되는 데 최대 6시간이 걸릴 수 있습니다. 그 동안 AWS IoT 분석 데이터 세트를 생성하고 메타데이터에서 쿼리를 실행할 수 있습니다.

다음 단계

데이터 로드 및 확인

데이터 로드 및 확인

이 자습서에서 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) 또는의 단계에 따라 AWS IoT SiteWise 데이터에 대한 AWS IoT 분석 데이터 세트를 AWS IoT SiteWise 데이터로 데이터 세트 생성(AWS CLI) 생성합니다.
2. 이 튜토리얼 전체에서 데이터 세트 쿼리를 업데이트하려면 다음을 수행합니다.

a. AWS IoT 분석 콘솔의 데이터 세트 페이지에서 이전 페이지에서 생성한 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 분석 콘솔에서 또는를 사용하여 데이터에 대해 다음 쿼리를 AWS CLI실행하여 `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 데이터가 생성되어 데이터 스토어에 로드되면 AWS IoT 분석 데이터 세트를 생성하고에서 SQL 쿼리를 실행 AWS IoT 분석 하여 자산에 대한 인사이트를 검색할 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 사용합니다.

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

4. 를 사용하면 다음 예제와 같이 데이터 스토어의 두 개 이상의 테이블에서 데이터를 조인 AWS IoT 분석할 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 사용하십시오.

```
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 탐색했으므로 산업 장비에 대한 귀중한 인사이트를 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_id`를 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 자체적으로 삭제됩니다. 데모 리소스의 사용을 마쳤으면 이전 데모를 삭제할 수 있습니다. 데모를 수동으로 삭제하려면 다음 단계를 수행하십시오.

1. CloudFormation 콘솔로 이동합니다.
2. Stacks 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. 데모가 다시 삭제되지 않으면 CloudFormation 콘솔의 단계에 따라 삭제에 실패한 리소스를 건너뛰고 다시 시도하세요.

데이터 스토어를 삭제하려면

- 관리형 데이터 스토어를 삭제하려면 다음 예와 같이 CLI 명령 `delete-datastore`를 실행합니다.

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

AWS IoT 분석 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 예는 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 배열을 수신하고 반환해야 합니다. 예제는 the section called “Lambda 함수 예시 1” 섹션을 참조하세요.

Lambda 함수를 호출할 수 있는 AWS IoT 분석 권한을 부여하려면 정책을 추가해야 합니다. 예를 들어 다음 CLI 명령을 실행하고 *`exampleFunctionName`*을 Lambda 함수의 이름으로 바꾸고, *`123456789012`*을 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\":\"iotanalyticsa\",\"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 사용을 참조하십시오.

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 살펴보십시오.

```
{
    "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 메시지가 있다고 가정합니다.

```
{
    "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-dddeeef-gghh-jjkk-llmmnnoopp"
    }
```

```
}
```

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

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeThing"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:thing/your-thingName"
            ]
        }
    ]
}
```

또 신뢰 정책은 다음과 같아야 합니다.

JSON

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

```
}
```

DeviceShadowEnrich 활동

deviceShadowEnrich 활동은 AWS IoT 디바이스 섀도우 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:GetThingShadow"
            ],
            "Resource": [
            "arn:aws:iot:us-east-1:123456789012:thing/your-thingName"
            ]
        }
    ]
}
```

또 신뢰 정책은 다음과 같아야 합니다.

JSON

```
{
    "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.

`+`	추가
`-`	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: `abs(-5)`= 5.

인수 유형	결과
`Int`	`Int`, 인수의 절대값
`Decimal`	`Decimal`, 인수의 절대값
`Boolean`	`Undefined` .
`String`	`Decimal`. 결과는 인수의 절대값입니다. 문자열을 변환할 수 없는 경우 결과는 `Undefined` 입니다.
Array	`Undefined` .
객체	`Undefined` .
Null	`Undefined` .

인수 유형	결과
정의되지 않음	Undefined .

acos(Decimal)

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

예: acos(0) = 1.5707963267948966

인수 유형	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.
Array	Undefined .
객체	Undefined .
Null	Undefined .
정의되지 않음	Undefined .

asin(Decimal)

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

예: asin(0) = 0.0

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

atan(Decimal)

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

예: atan(0) = 0.0

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

인수 유형	결과
String	Decimal(배정밀도), 인수의 역탄젠트. 문자열을 변환할 수 없는 경우 결과는 Undefined 입니다. 가상 결과는 Undefined 로 반환됩니다.
Array	Undefined .
객체	Undefined .
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: atan(1, 0) = 1.5707963267948966

인수 유형	인수 유형	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.

예시:

`ceil(1.2)` = 2

`ceil(11.2)` = -1

인수 유형	결과
`Int`	`Int`, 인수 값
`Decimal`	`Int`, 문자열은 `Decimal`로 변환된 후 가장 가까운 `Int`로 올림됩니다. 문자열을 `Decimal`로 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 유형	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 `Decimal`로 변환할 수 없는 경우 결과는 `Undefined` 입니다. 가상 결과는 `Undefined` 로 반환됩니다.
Array	`Undefined` .

인수 유형	결과
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 유형	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.
Array	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 유형	결과
`Int`	`Decimal`(배정밀도), e^인수.
`Decimal`	`Decimal`(배정밀도), e^인수
`String`	`Decimal`(배정밀도), e^인수. `String`을 `Decimal`로 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 유형	결과
`Int`	`Decimal`(배정밀도), 인수의 자연 로그.
`Decimal`	`Decimal`(배정밀도), 인수의 자연 로그
`Boolean`	`Undefined` .
`String`	`Decimal`(배정밀도), 인수의 자연 로그. 문자열을 `Decimal`로 변환할 수 없는 경우 결과는 `Undefined` 입니다.
Array	`Undefined` .
객체	`Undefined` .

인수 유형	결과
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 유형	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.
Array	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: mod(8, 3) = 2

왼쪽 피연산자	오른쪽 피연산자	결과
`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 `Decimals`로 변환될 경우 결과는 두 번째 인수를 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. `Decimal`인수는 함수 적용 전에 배정밀도로 반올림됩니다.

예: `power(2, 5)` = 32.0

인수 형식 1	인수 형식 2	결과
`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 `Decimals`로 변환됩니다. `String`을(를) `Decimal`(으)로 변환하지 못한 경우 결과는 `Undefined` 입니다.

인수 형식 1	인수 형식 2	결과
기타 값	기타 값	Undefined .

round(Decimal)

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

예시:

Round(1.2) = 1

Round(1.5) = 2

Round(1.7) = 2

Round(-1.1) = -1

Round(-1.5) = -2

인수 유형	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.

예시:

`sign(-7)` = -1

`sign(0)` = 0

`sign(13)` = 1

인수 유형	결과
`Int`	`Int`, `Int` 값의 부호.
`Decimal`	`Int`, `Decimal` 값의 부호.
`String`	`Int`, `Decimal` 값의 부호. 문자열은 `Decimal` 값으로 변환되고 `Decimal` 값의 부호가 반환됩니다. `String`을 `Decimal`로 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 유형	결과
`Int`	`Decimal`(배정밀도), 인수의 사인.
`Decimal`	`Decimal`(배정밀도), 인수의 사인.
`Boolean`	`Undefined` .
`String`	`Decimal`, 인수의 사인. 문자열을 `Decimal`로 변환할 수 없는 경우 결과는 `Undefined` 입니다.
`Array`	`Undefined` .
`Object`	`Undefined` .

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

인수 유형	결과
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 유형	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.

예시:

`trunc(2.3, 0)` = 2

`trunc(2.3123, 2)` = 2.31

`trunc(2.888, 2)` = 2.88

`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 `Int`가 아닐 경우 가까운 `Int`로 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.json파일을 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": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

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

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

채널 메시지 재처리

AWS IoT 분석 를 사용하면 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

를 사용하여 파이프라인을 통해 채널 메시지를 재처리 AWS IoT 분석하는 경우 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.jso`

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: 타임스탬프

`startTime`

재처리된 원시 메시지 데이터의 시작 시간(포함)입니다.

`startTime` 파라미터 값을 지정하는 경우 `channelMessages` 객체를 사용해서는 안 됩니다.

유형: 타임스탬프

`pipelineName`

재처리를 시작할 파이프라인의 이름입니다.

유형: 문자열

길이 제한: 1~128자.

채널 메시지 재처리(콘솔)

이 자습서에서는 AWS IoT 분석 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 분석 콘솔에 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 분석 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 콘솔을 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 유의하십시오.

- `startTime` 및 `endTime` 파라미터는 원시 데이터의 수집 시점을 지정하지만, 이것은 대강의 추정치에 불과합니다. 가장 가까운 시간 단위로 반올림할 수 있습니다. `startTime`은 포함되지만 `endTime`은 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 내용은AWS IoT Analytics API 참조의 StartPipelineReprocessing API를 참조하십시오.

채널 재처리 활동 취소

파이프라인 재처리 활동을 취소하려면 CancelPipelineReprocessing API를 사용하거나 AWS IoT 분석 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 참조하십시오.

워크플로우 자동화

AWS IoT 분석 는에 대한 고급 데이터 분석을 제공합니다 AWS IoT. 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

`CreateDataset`를 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 참조하십시오.

또 다른 데이터 세트가 완료되었을 때 데이터 세트 콘텐츠를 생성합니다

다른 데이터 세트의 콘텐츠 생성이 완료되면 새 데이터 세트 콘텐츠 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 콘텐츠 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. 를 사용하면 이러한 주기적 기간에서 중복되지 않는 데이터를 집계하고 최종 제품 품질에 대한 KPI 보고서를 생성하는 분석을 자동화 AWS IoT 분석할 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. 를 AWS IoT 분석 사용하면 분석을 오케스트레이션하고 각 실행에 대한 KPI 및 보고서를 생성한 다음 향후 분석을 위해 데이터를 저장할 수 있습니다.

자동 이상 탐지

AWS IoT 분석 를 사용하면 데이터 스토어에 도착한 새 데이터에 대해 15분마다 수동으로 실행해야 하는 이상 탐지 워크플로를 자동화할 수 있습니다. 또한 지정한 기간 내의 디바이스 사용량과 최고 사용자를 표시하도록 대시보드를 자동화할 수 있습니다.

산업 제어 결과 예측

산업 생산 라인이 있습니다. 사용 가능한 프로세스 측정값을 AWS IoT 분석포함하여 로 전송된 데이터를 사용하여 분석 워크플로를 운영하여 프로세스 결과를 예측할 수 있습니다. 모델 데이터는 M x N 매트릭스로 정렬할 수 있으며, 각 행에는 실험실 샘플을 채취하는 다양한 시점의 데이터가 포함됩니다. AWS IoT 분석 는 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 설정하는 데 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 분석 를 사용하면 변수를 사용하여 컨테이너화된 애플리케이션의 입력 데이터 소스와 Docker 컨테이너의 출력 데이터 대상을 정의할 수 있습니다. (사용자 지정 Docker 컨테이너 입력/출력 변수에는 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. AWS IoT 분석 채널로 메시지 데이터를 전송할 수 있는 권한을 부여하는 IAM 역할 생성(`iam: CreateRole`.)
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 사용하여 AWS SDK 또는를 사용할 수 있는 디바이스에서 채널로 직접 메시지를 보낼 수 있습니다 AWS CLI.
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 데이트 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`).

SQL 쿼리 및 델타 창을 사용하여 데이터 세트를 생성하고 애플리케이션이 있는 컨테이너를 사용하면는 델타 창에서 데이터에 지정한 간격으로 실행되는 예약된 프로덕션 데이터 세트를 AWS IoT 분석 생성하여 원하는 출력을 생성하고 알림을 보냅니다.

선택에 따라, 프로덕션 데이터 세트 애플리케이션을 일시 중지 및 다시 시작할 수 있습니다. 프로덕션 데이터 세트 애플리케이션을 재개하면 AWS IoT 분석기본적으로는 마지막 실행 이후 도착했지만 아직 분석되지 않은 모든 데이터를 포착합니다. 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": {
       "OutputUris": {
           "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. OutputUris 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 목록으로 로드해야 합니다. 액세스 목록은 업로드된 출력에 대한 AWS IoT 분석 서비스 제어 권한을 부여합니다. 이 예에서는 `params` 파일의 `custom_output`로 정의한 Amazon S3 위치에 `example_var`의 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. 한 역할은 노트북을 컨테이너화하기 위해 SageMaker AI 인스턴스를 시작할 수 있는 권한을 부여합니다. 컨테이너 실행에 또 다른 역할이 필요합니다.

첫 번째 역할을 자동 또는 수동으로 생성할 수 있습니다. AWS IoT 분석 콘솔을 사용하여 새 SageMaker AI 인스턴스를 생성하는 경우 SageMaker AI 인스턴스를 실행하고 노트북을 컨테이너화하는 데 필요한 모든 권한을 부여하는 새 역할을 자동으로 생성할 수 있는 옵션이 제공됩니다. 또는 수동으로 이런 권한을 가진 역할을 생성할 수도 있습니다. 이렇게 하려면 `AmazonSageMakerFullAccess` 정책이 연결된 역할을 생성한 후 다음 정책을 추가합니다.

JSON

```
{
  "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 분석 콘솔을 사용하여 첫 번째 역할을 자동으로 생성했더라도이 작업을 수행해야 합니다. 다음 정책과 신뢰 정책이 연결된 역할을 생성합니다.

JSON

```
{
    "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.

JSON

```
{
  "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 `CreateDatasetContent`를 호출해서 수동으로, 또는 지정한 `trigger`에 따라 자동으로 채울 수 있습니다. 자세한 내용은 CreateDataset 및 `CreateDatasetContent`를 참조하십시오.

항목

- 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: <datatsetName>, DatasetArn: <datatsetARN>, 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: <datatsetName>, DatasetArn: <datatsetARN>, 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: <datatsetName>, DatasetArn: <datatsetARN>, 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: <datatsetName>, DatasetArn: <datatsetARN>}
```

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": "<datatsetARN>",
    "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": "<datatsetARN>",
}
```

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 분석 콘솔에 표시하여 HTML을 표시하는 컴퓨터에 잠재적 공격 벡터를 제공할 수 있습니다. 타사 노트북을 사용하기 전에 해당 노트북의 작성자를 신뢰할 수 있는지 확인하십시오.

고급 분석 기능을 실행하는 한 가지 옵션은 Jupyter 노트북을 사용하는 것입니다. Jupyter Notebook은 기계 학습과 각종 통계 분석을 수행할 수 있는 강력한 데이터 과학 도구입니다. 자세한 내용은 노트북 템플릿을 참조하십시오. (JupyterLab 내부의 컨테이너화는 현재 지원되지 않습니다.) Jupyter Notebook 및 라이브러리를 정의한 델타 기간 AWS IoT 분석 동안가 수신하는 새 데이터 배치에서 주기적으로 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.

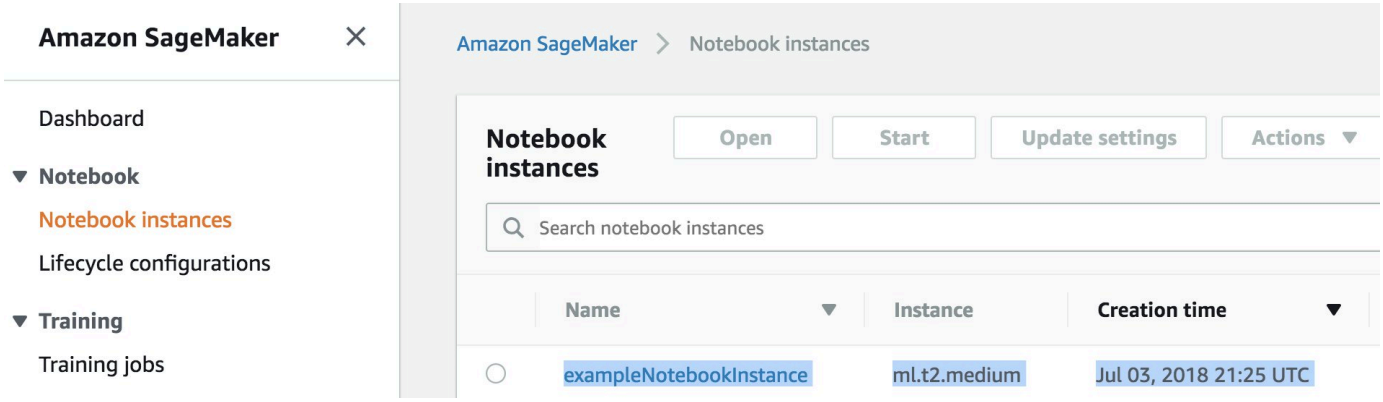
2018년 8월 23일 이후에 AWS IoT 분석 콘솔을 사용하여 SageMaker AI 인스턴스를 생성한 경우 컨테이너화 확장 설치가 자동으로 완료되고 컨테이너화된 이미지 생성을 시작할 수 있습니다. 그렇지 않으면이 섹션에 나열된 단계에 따라 SageMaker AI 인스턴스에서 노트북 컨테이너화를 활성화합니다. 다음과 같이 Amazon EC2에 컨테이너 이미지를 업로드하고 컨테이너화 확장을 설치할 수 있도록 SageMaker AI 실행 역할을 수정합니다.

AWS IoT 분석 콘솔을 통해 생성되지 않은 노트북 인스턴스의 컨테이너화 활성화

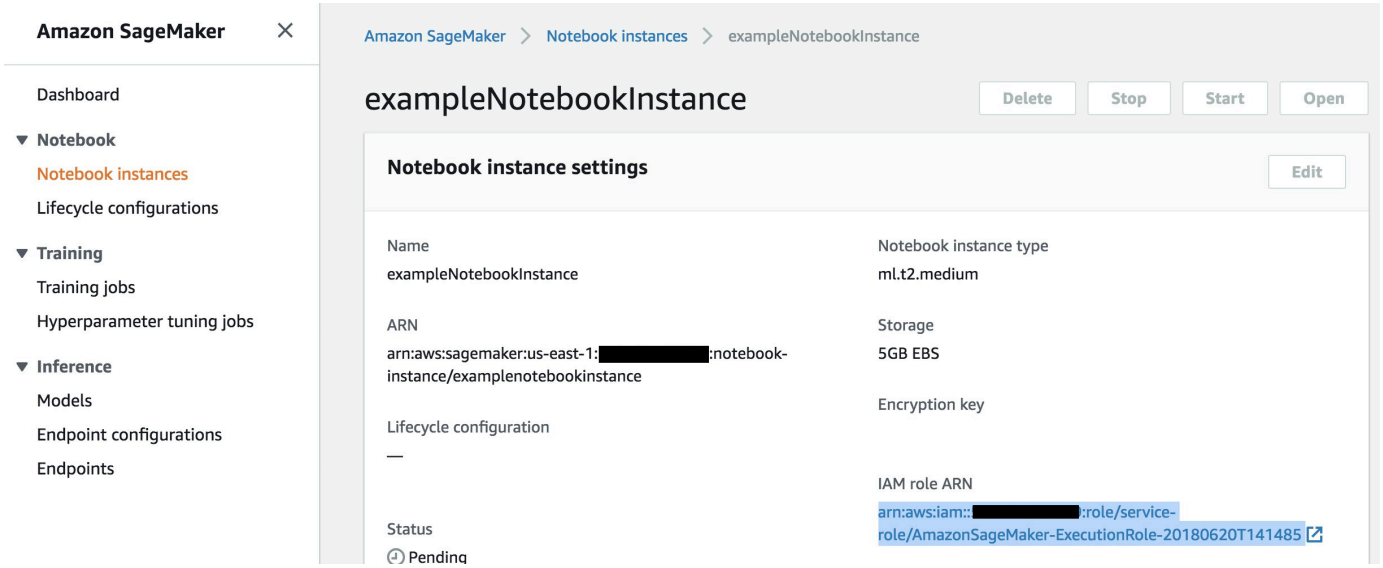
다음 단계를 따르는 대신 AWS IoT 분석 콘솔을 통해 새 SageMaker AI 인스턴스를 생성하는 것이 좋습니다. 새 인스턴스는 자동으로 컨테이너화를 지원합니다.

여기에 표시된 대로 컨테이너화를 활성화한 후 SageMaker AI 인스턴스를 다시 시작하는 경우 IAM 역할 및 정책을 다시 추가할 필요가 없지만 마지막 단계에 표시된 대로 확장을 다시 설치해야 합니다.

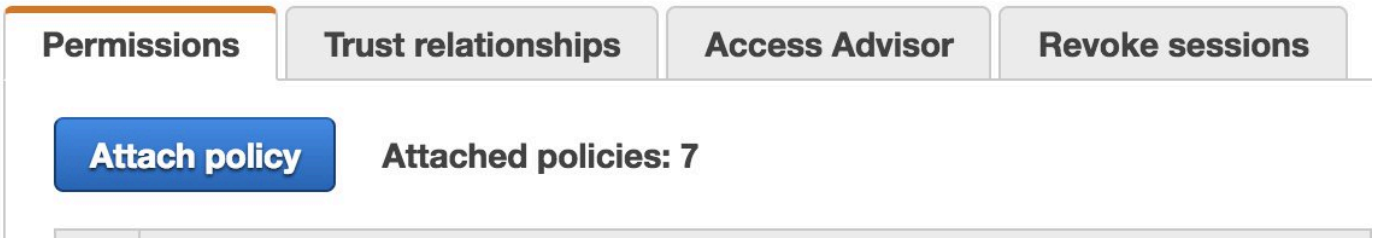
1. 노트북 인스턴스에 Amazon ECS에 대한 액세스 권한을 부여하려면 SageMaker AI 페이지에서 SageMaker AI 인스턴스를 선택합니다.



2. IAM 역할 ARN에서 SageMaker AI 실행 역할을 선택합니다.

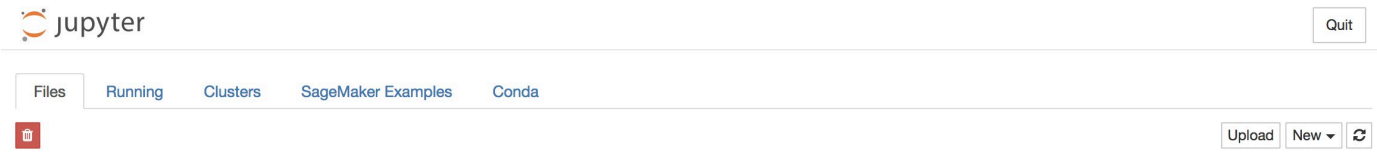


3. Attach Policy(정책 연결)을 선택한 후, 권한에 표시된 것처럼 정책을 정의하고 연결합니다. 아직 연결되어 있지 않은 경우 `AmazonSageMakerFullAccess` 정책을 연결합니다.



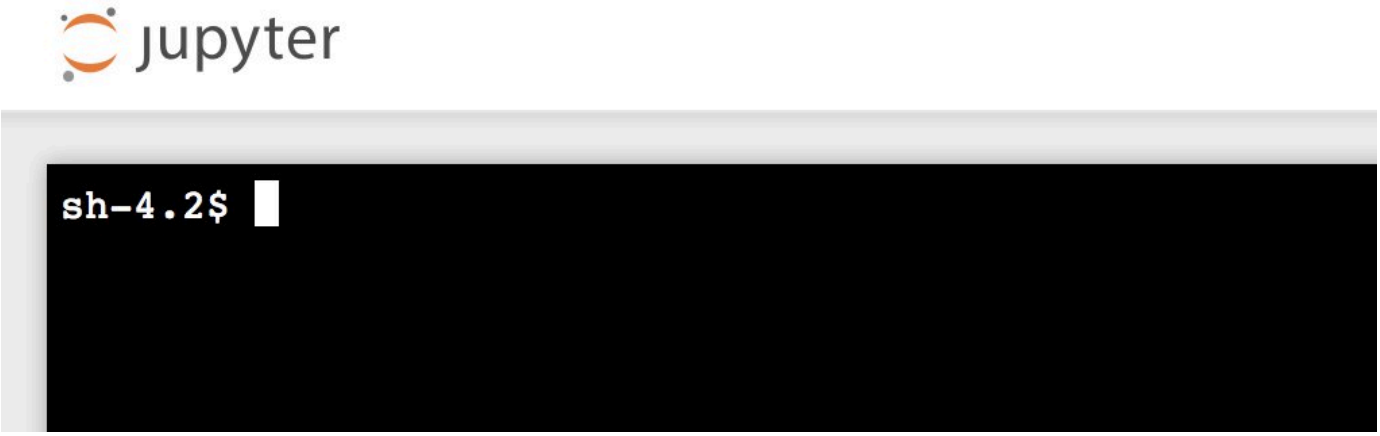
또한 Amazon S3에서 컨테이너화 코드를 다운로드하여 노트북 인스턴스에 설치해야 합니다. 첫 번째 단계는 SageMaker AI 인스턴스의 터미널에 액세스하는 것입니다.

1. Jupyter에서 새로 만들기를 선택합니다.



2. 표시되는 메뉴에서 터미널을 선택합니다.

3. 터미널에서 다음 명령을 입력해 코드를 다운로드해서 압축을 푼 후 설치합니다. 이러한 명령은이 SageMaker AI 인스턴스의 노트북에서 실행 중인 모든 프로세스를 종료합니다.



```
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.

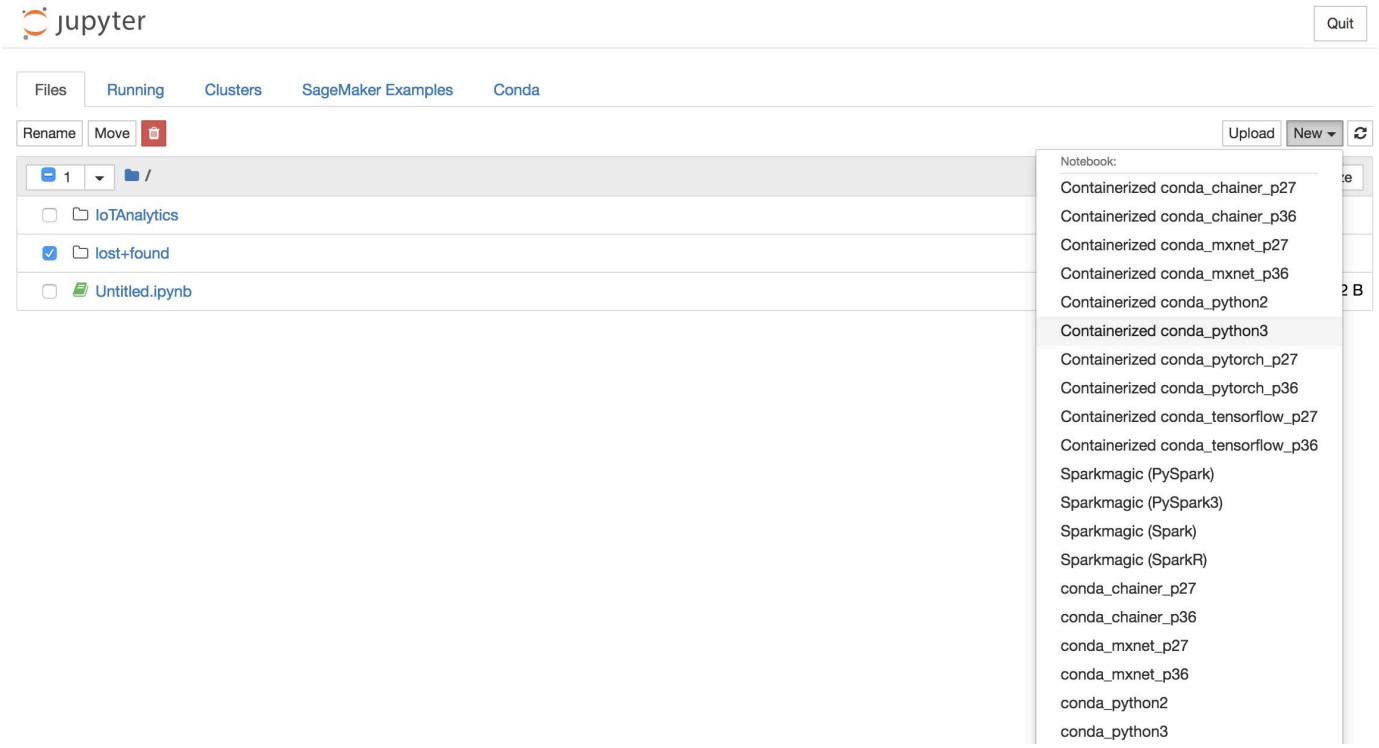
노트북 컨테이너화 확장 업데이트

2018년 8월 23일 이후에 AWS IoT 분석 콘솔을 통해 SageMaker AI 인스턴스를 생성한 경우 컨테이너화 확장이 자동으로 설치되었습니다. SageMaker AI 콘솔에서 인스턴스를 다시 시작하여 확장을 업데이트할 수 있습니다. 확장 프로그램을 수동으로 설치한 경우 AWS IoT 분석 콘솔을 통해 생성되지 않은 노트북 인스턴스의 컨테이너화 활성화에 나열된 터미널 명령을 다시 실행하여 업데이트할 수 있습니다.

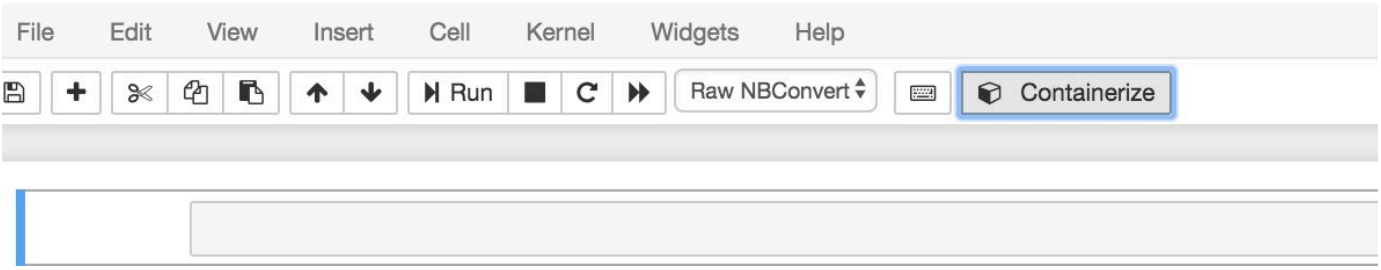
컨테이너화된 이미지 생성

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

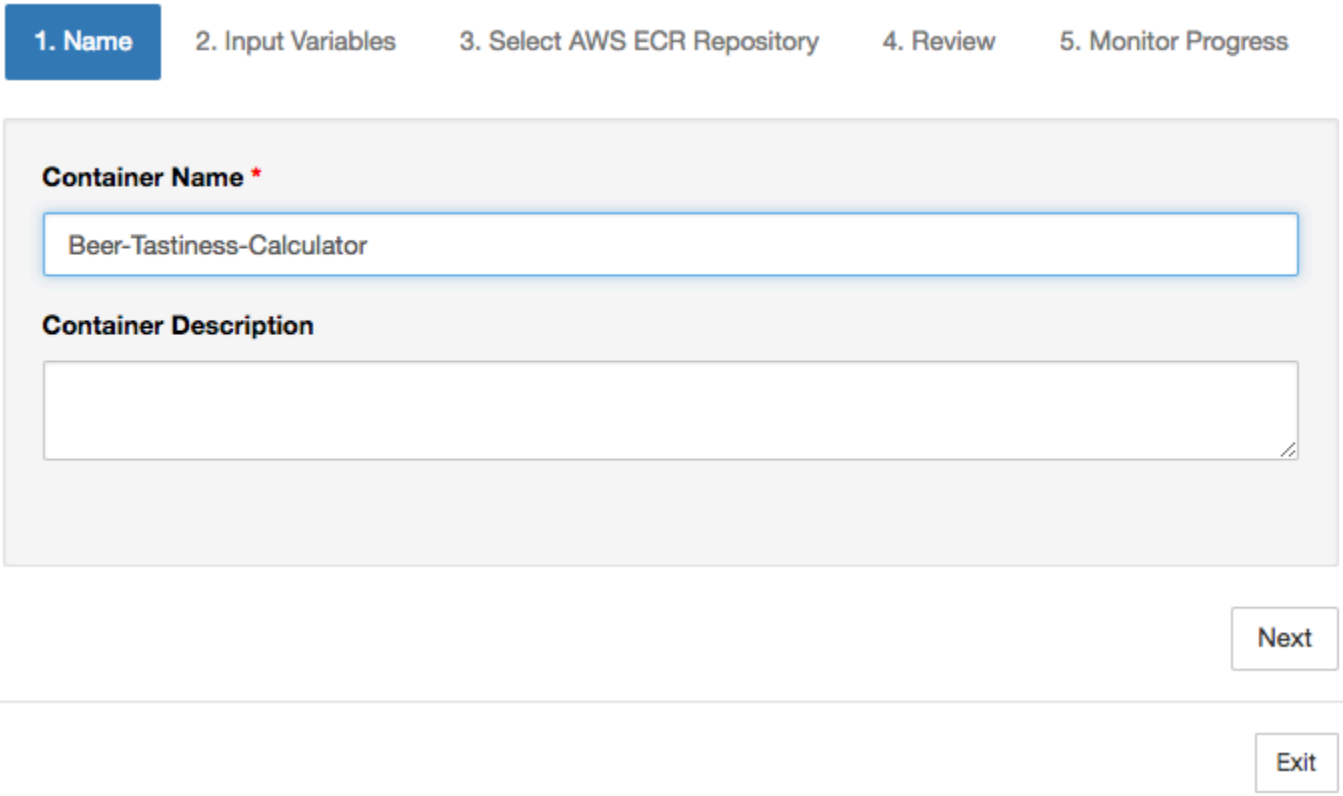
1. Jupyter 노트북에서 새로 만들기를 선택한 후, 드롭다운 목록에서 원하는 커널 유형을 선택합니다. (커널 유형은 “Containerized”로 시작하고 다른 방법으로는 선택했을 커널로 끝나야 합니다. 예를 들어, “conda_python3"과 같은 평범한 Python 3.0 환경을 원한다면 “Containerized conda_python3”을 선택합니다.



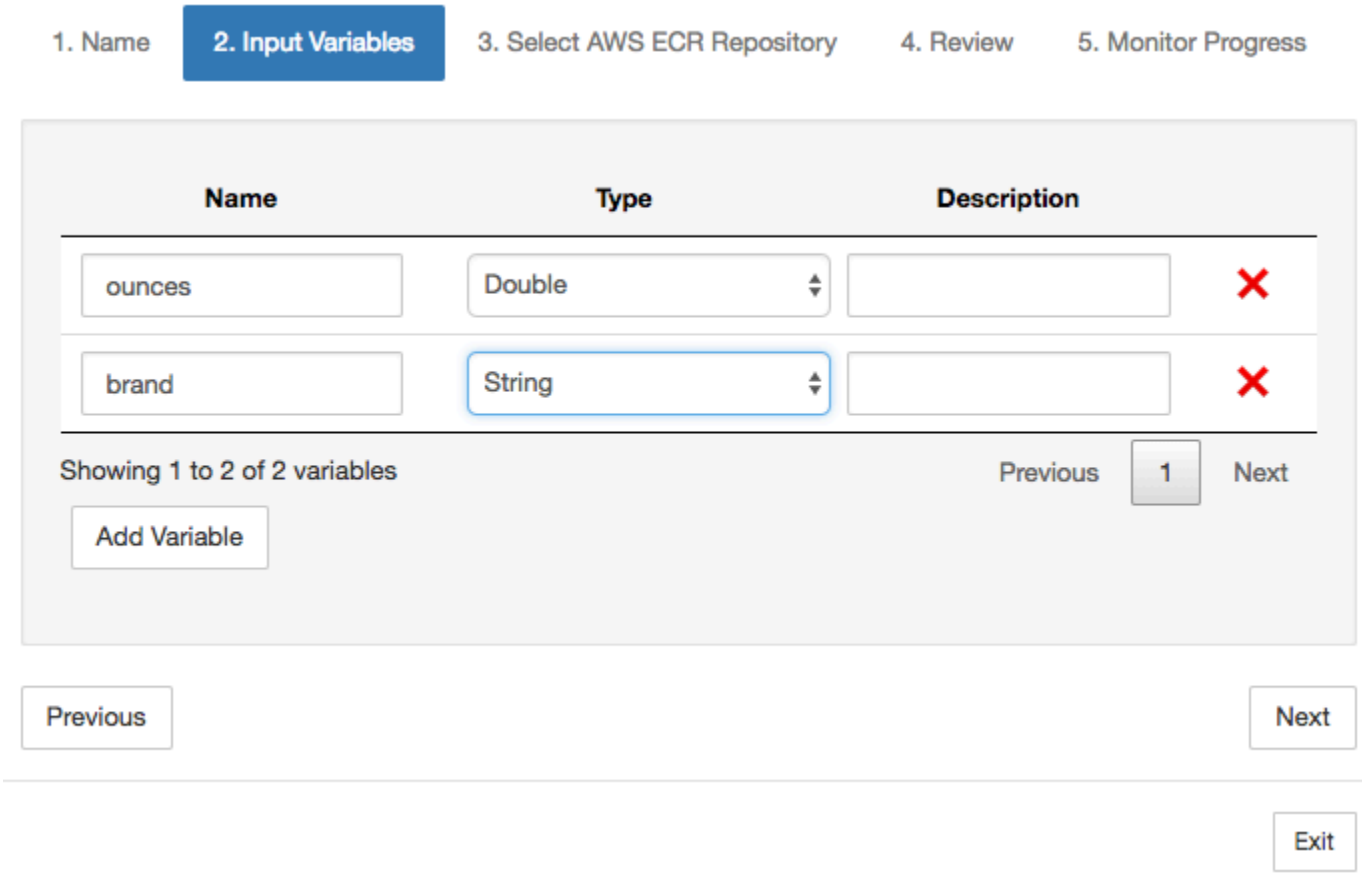
2. 노트북에서 작업을 완료하고 컨테이너화를 원하면 컨테이너화 버튼을 선택합니다.



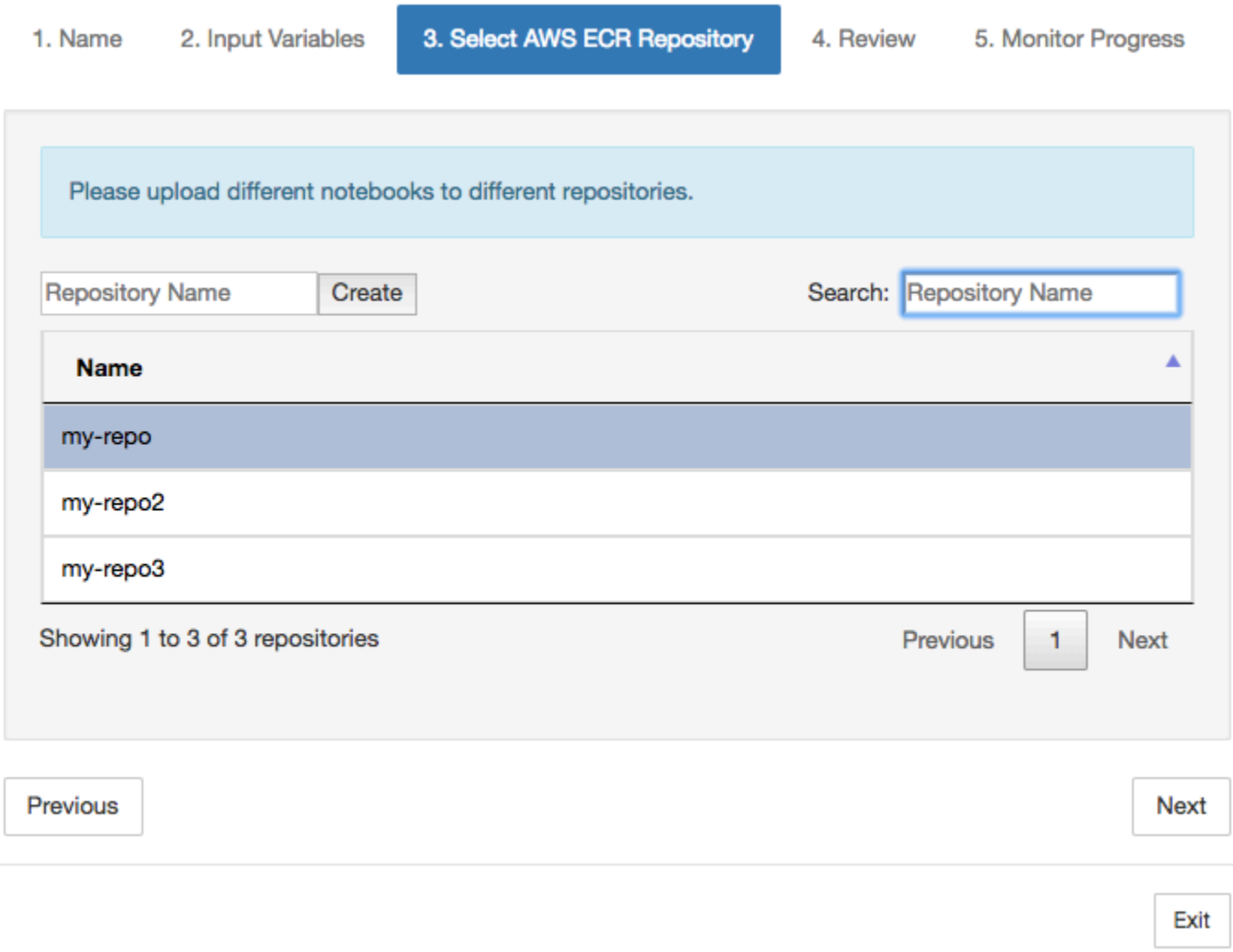
3. 컨테이너화된 노트북의 이름을 입력합니다. 또한 설명을 입력할 수 있습니다(선택 사항).



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

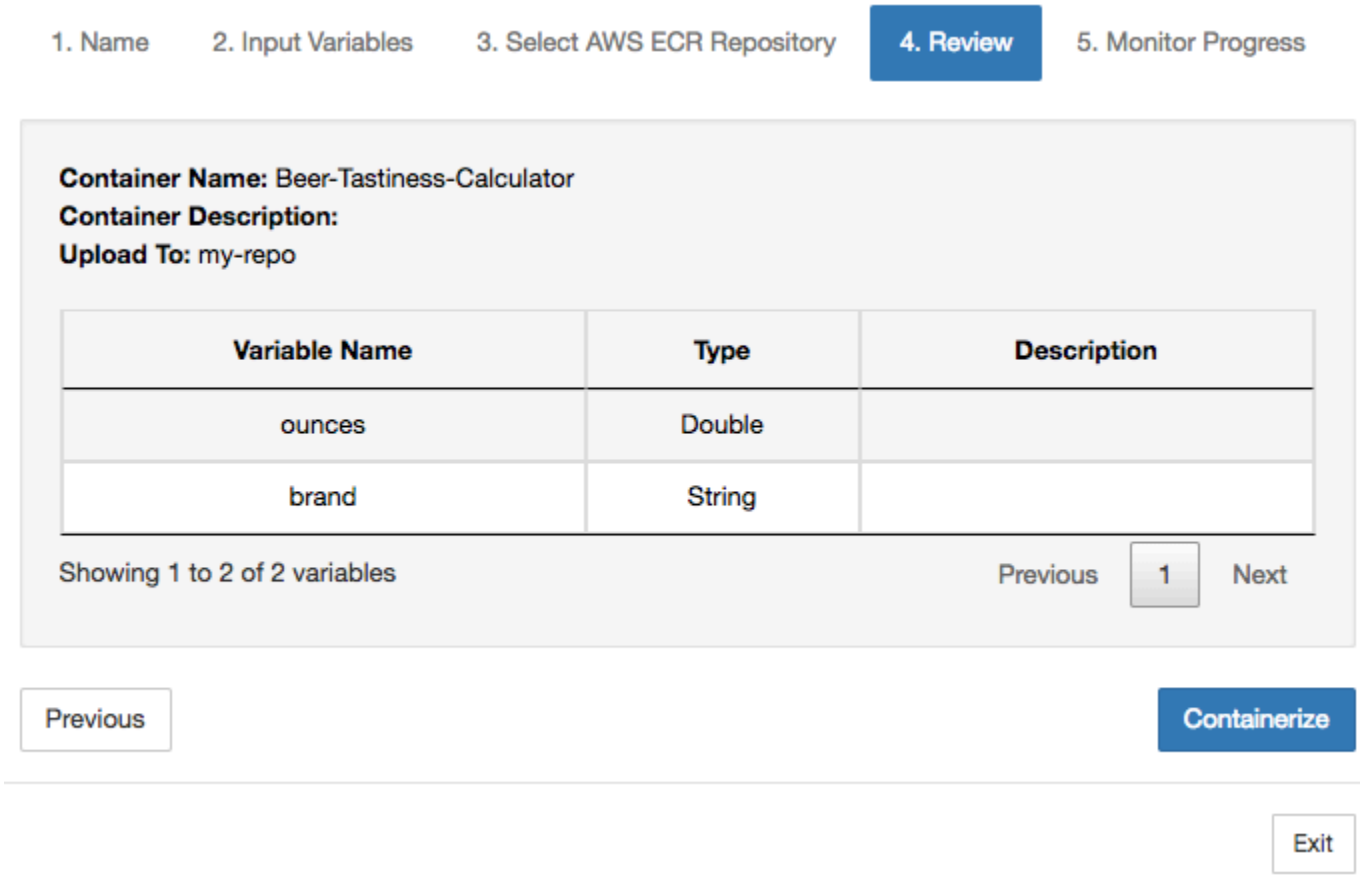


5. 노트북에서 생성한 이미지를 업로드할 Amazon ECR 리포지토리를 선택합니다.

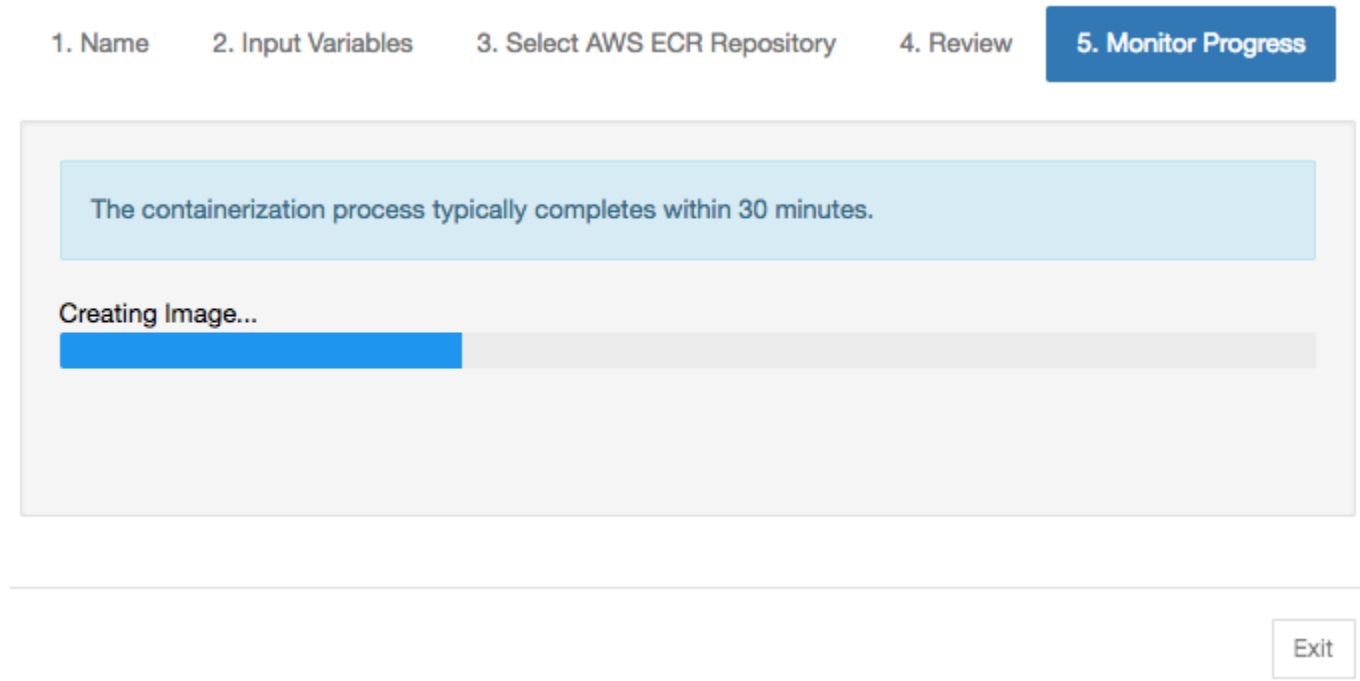


6. 컨테이너화를 선택해 프로세스를 시작합니다.

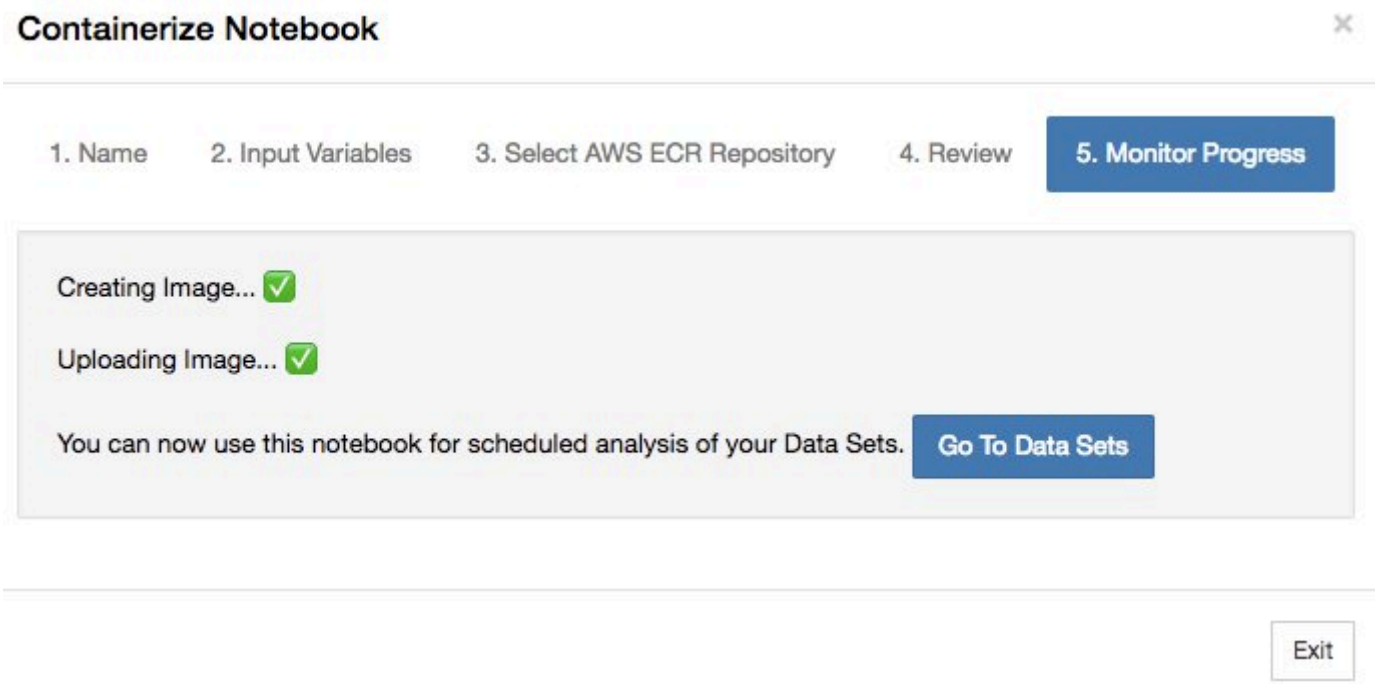
   자신의 입력을 요약한 개요가 제공됩니다. 프로세스를 시작하면 이를 취소할 수 없다는 점에 유의해주십시오. 프로세스는 최대 1시간까지 지속될 수 있습니다.



7. 다음 페이지에 진행률이 표시됩니다.



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



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 분석 콘솔에 표시하여 HTML을 표시하는 컴퓨터에 잠재적 공격 벡터를 제공할 수 있습니다. 타사 노트북을 사용하기 전에 해당 노트북의 작성자를 신뢰할 수 있는지 확인하십시오.

자체 사용자 지정 컨테이너를 생성하고 AWS IoT 분석 서비스와 함께 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 분석 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` - JSON을 구문 분석하고 AWS IoT 분석, Octave 스크립트를 실행하고, 아티팩트를 Amazon S3에 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 cd를 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.txt를 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 바꾸고 *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 대리인 방지” 단원을 참조하십시오.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "sagemaker.amazonaws.com",
                    "iotanalytics.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset"
```

```
                }
            }
        }
    ]
}
```

2. `role1.json` 다운로드한 파일을 AWS IoT 분석사용하여 SageMaker AI 및에 대한 액세스 권한을 부여하는 역할을 생성합니다.

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

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

   JSON

```
{
    "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-123456789012/*"
            ]
        },
        {
            "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 간접 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. `DatasetContentState`이 `SUCCEEDED`가 되기까지 몇 분 정도 기다려야 할 수도 있습니다.

8단계: 출력을 Ocatave로 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
```

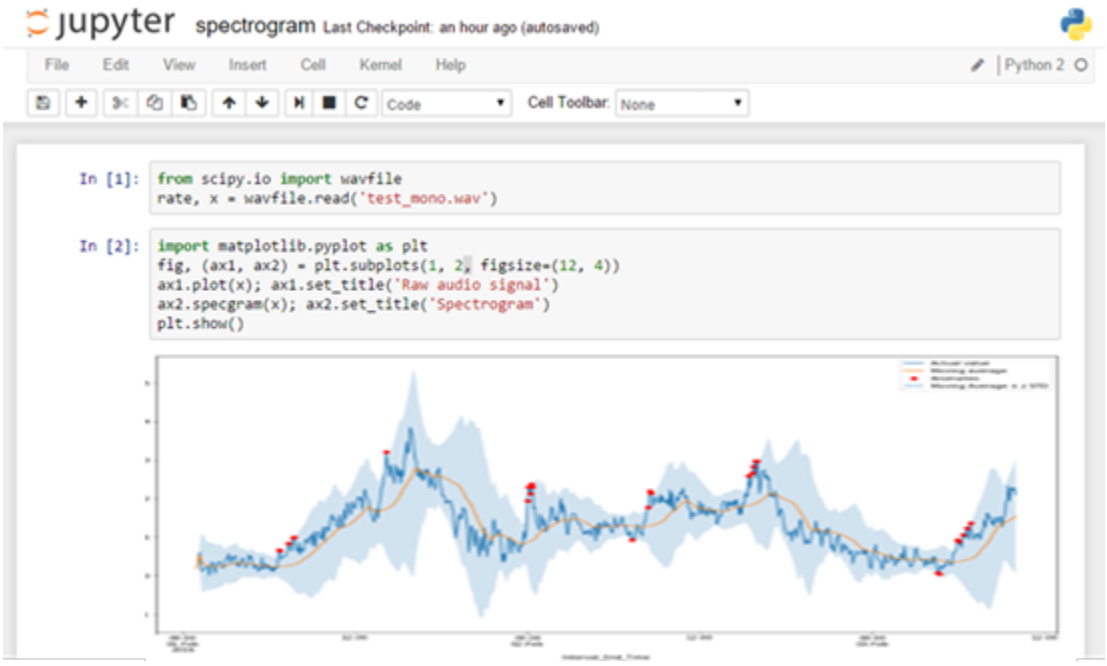
AWS IoT 분석 데이터 시각화

AWS IoT 분석 콘솔 또는 Quick Suite를 사용하여 AWS IoT 분석 데이터를 시각화할 수 있습니다.

주제

콘솔을 사용하여 AWS IoT 분석 데이터 시각화

AWS IoT 분석 는 AWS IoT 분석 콘솔의 컨테이너 데이터 세트 콘텐츠 페이지에 컨테이너 데이터 세트 (파일에 있음`output.html`)의 HTML 출력을 포함할 수 있습니다. 예를 들어, Jupyter Notebook 실행하는 컨테이너 데이터 세트를 정의하고 Jupyter 노트북에서 시각화를 작성하는 경우, 데이터 세트는 다음과 같을 수 있습니다.



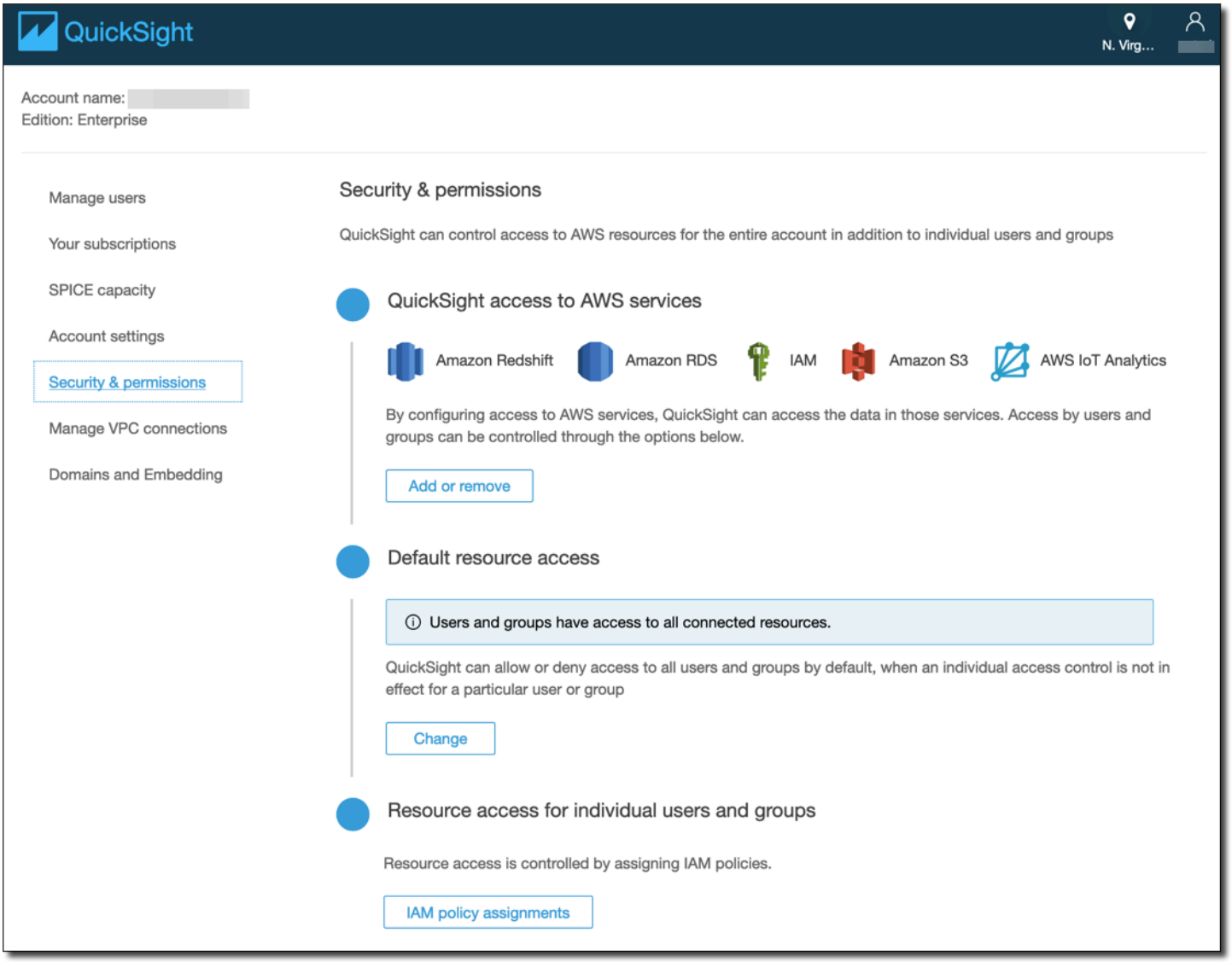
컨테이너 데이터 세트 콘텐츠가 작성되면 콘솔의 데이터 세트 콘텐츠 페이지에서 이 시각화를 볼 수 있습니다.

Jupyter Notebook 실행하는 컨테이너 데이터 세트 작성에 대한 자세한 내용은 내 워크플로우 자동화를 참조하십시오.

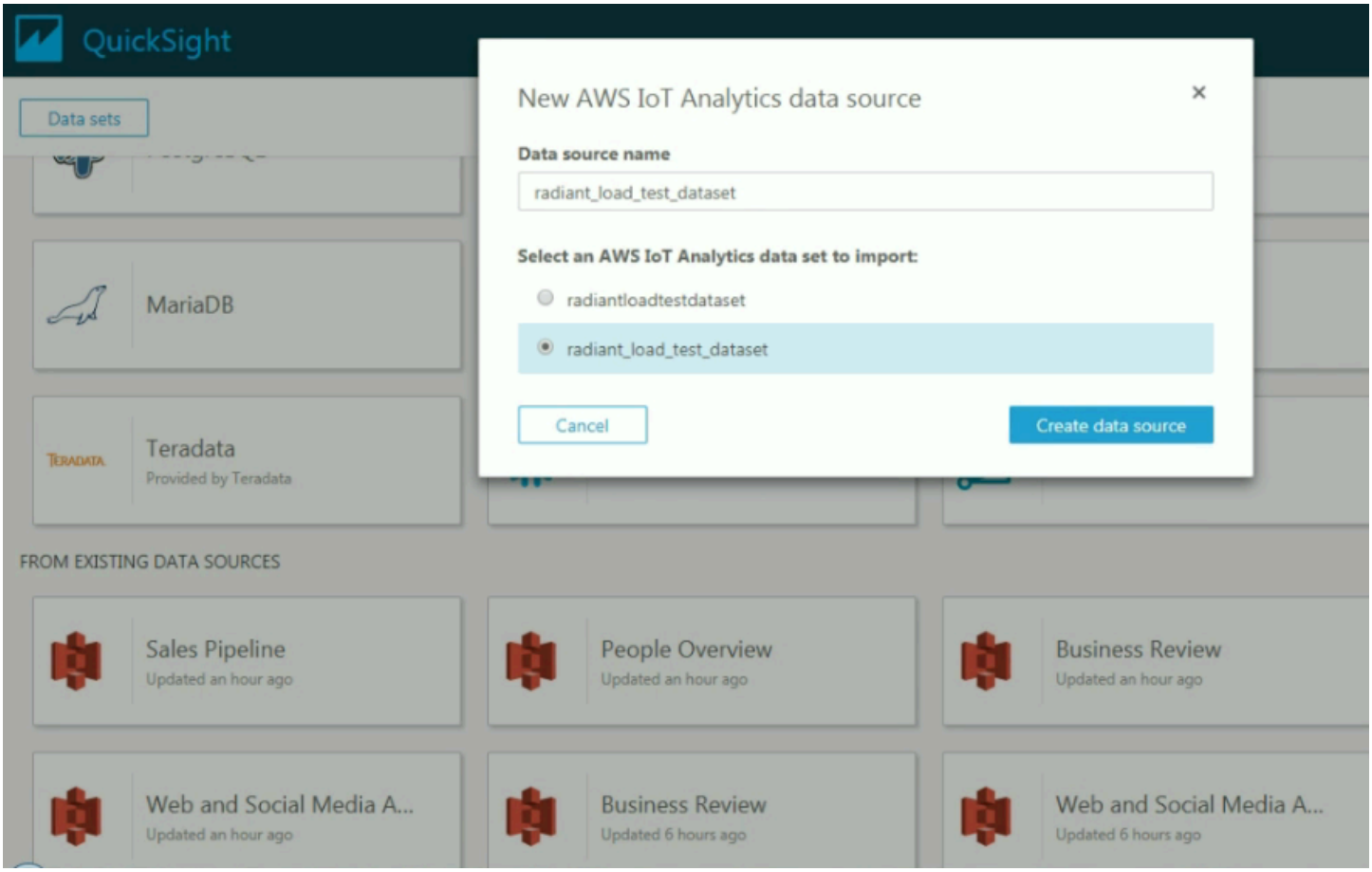
Quick Suite를 사용하여 AWS IoT 분석 데이터 시각화

AWS IoT 분석 는 Quick Suite와의 직접 통합을 제공합니다. Quick Suite는 시각화를 구축하고, 임시 분석을 수행하고, 데이터에서 비즈니스 인사이트를 빠르게 얻는 데 사용할 수 있는 빠른 비즈니스 분석 서비스입니다. Quick Suite를 사용하면 조직이 수십만 명의 사용자로 확장할 수 있으며 강력한 인 메모리 엔진(SPICE)을 사용하여 응답 성능을 제공할 수 있습니다. Quick Suite 콘솔에서 AWS IoT 분석 데이터 세트를 선택하고 대시보드 및 시각화 생성을 시작할 수 있습니다. Quick Suite는 이러한 리전에서 사용할 수 있습니다.

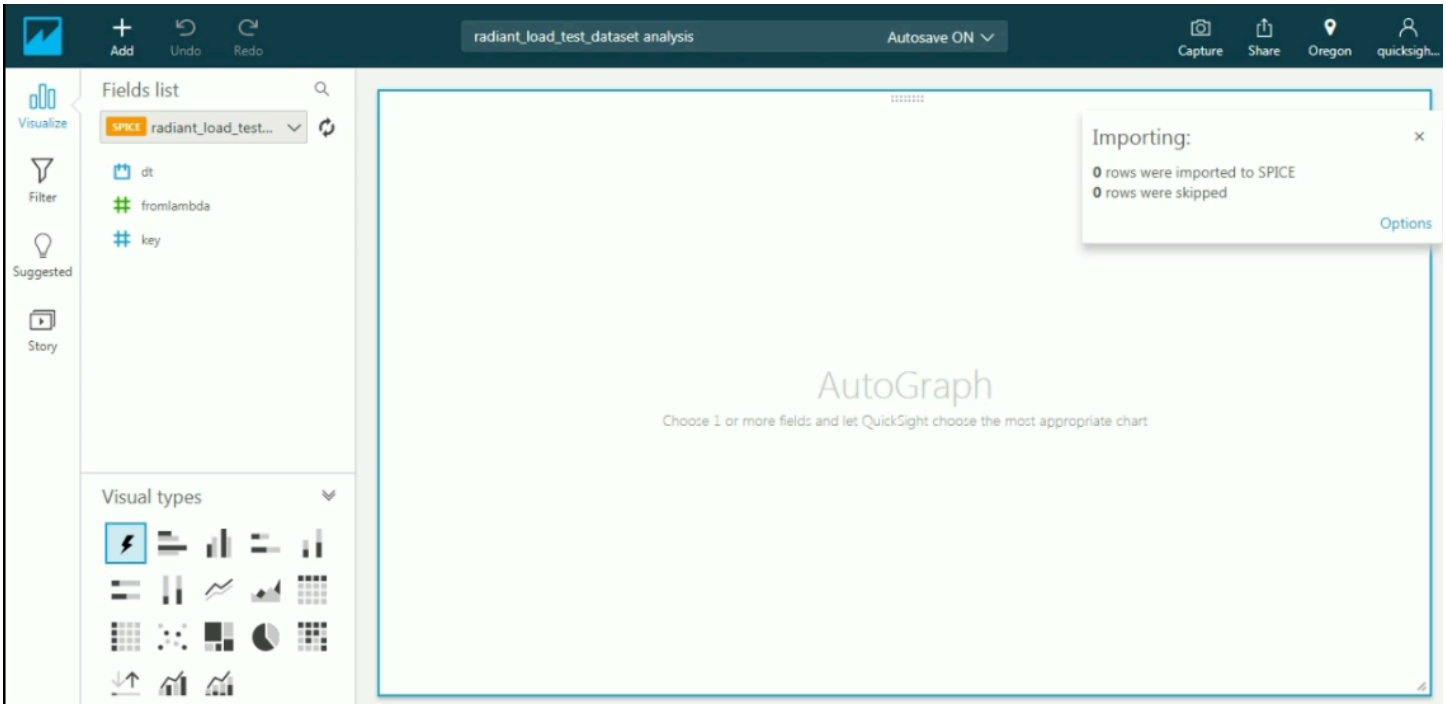
Quick Suite 시각화를 시작하려면 Quick Suite 계정을 생성해야 합니다. 계정을 설정할 때 Quick Suite에 AWS IoT 분석 데이터에 대한 액세스 권한을 부여해야 합니다. 이미 계정이 있는 경우 관리, QuickSight 관리, 보안 및 권한을 선택하여 Quick Suite에 AWS IoT 분석 데이터에 대한 액세스 권한을 부여합니다. AWS 서비스에 대한 QuickSight 액세스에서 추가 또는 제거를 선택한 다음 옆의 확인란을 선택하고 업데이트를 AWS IoT 분석 선택합니다.



계정을 설정한 후 관리자 Quick Suite 콘솔 페이지에서 새 분석 및 새 데이터 세트를 선택한 다음 소스 AWS IoT 분석 로를 선택합니다. 데이터 소스의 이름을 입력하고, 가져올 데이터 세트를 선택한 다음 데이터 소스 생성을 선택합니다.



데이터 소스를 생성한 후 Quick Suite에서 시각화를 생성할 수 있습니다.



Quick Suite 대시보드 및 데이터 세트에 대한 자세한 내용은 Quick Suite 설명서를 참조하세요.

AWS IoT 분석 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

태그를 사용하면 용도, 소유자 또는 환경 등 다양한 방식으로 AWS IoT 분석 리소스를 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, AWS 에서 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 설명서의 비용 할당 태그 사용을 참조하십시오.

콘솔에서 태그를 생성하고 관리하는 중앙 통합 방법을 AWS 결제 및 비용 관리 제공하는 Tag Editor를 사용하면 쉽게 사용할 수 있습니다. 자세한 내용은 AWS Management Console시작하기에서 태그 편집기로 작업을 참조하십시오.

AWS CLI 및 AWS IoT 분석 API를 사용하여 태그로 작업할 수도 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 분석 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 DescribeLoggingOptions의 사용이 허용/거부되지 않습니다.

자세한 내용은 IAM 사용 설명서의 태그를 사용한 액세스 제어를 참조하십시오. 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 참조하십시오).

JSON

```
{
  "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 제한 사항은 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.
- `aws: prefix` 태그 이름 또는 값은 사용을 위해 예약되어 있으므로를 AWS 사용하지 마십시오. 이 접두사가 지정된 태그 이름이나 값은 편집하거나 삭제할 수 없습니다. 이 접두사가 지정된 태그는 소스당 태그 수 제한에 포함되지 않습니다.
- 태그 지정 스키마를 여러 서비스와 리소스에서 사용하는 경우 다른 서비스에서 허용되는 문자에 제한이 있을 수 있음에 유의하세요. 일반적으로 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 공백 및 숫자와 특수 문자(+ - = . _ : / @)입니다.

의 SQL 표현식 AWS IoT 분석

데이터 세트는 데이터 스토어의 데이터에 SQL 표현식을 사용하여 생성됩니다.는 Amazon Athena와 동일한 SQL 쿼리, 함수 및 연산자를 AWS IoT 분석 사용합니다.

AWS IoT 분석 는 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 파라미터를 참조하십시오.

AWS IoT 분석 및 Amazon Athena는 다음을 지원하지 않습니다.

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

주제

- 에서 지원되는 SQL 기능 AWS IoT 분석
- AWS IoT 분석에서 SQL 쿼리와 관련된 일반적인 문제 해결

에서 지원되는 SQL 기능 AWS IoT 분석

데이터 세트는 데이터 스토어의 데이터에 SQL 표현식을 사용하여 생성됩니다. 에서 실행하는 쿼리는 Presto 0.217을 AWS IoT 분석 기반으로 합니다.

지원되는 데이터 유형

AWS IoT 분석 및 Amazon Athena는 이러한 데이터 유형을 지원합니다.

- primitive_type
  - `TINYINT`
  - `SMALLINT`
  - `INT`
  - `BIGINT`
  - `BOOLEAN`
  - `DOUBLE`
  - `FLOAT`
  - `STRING`
  - `TIMESTAMP`
  - `DECIMAL(precision, scale)`
  - `DATE`
  - `CHAR`(길이가 지정된 고정 길이 문자 데이터)
  - `VARCHAR`(길이가 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 분석 및 Amazon Athena는 일부 데이터 유형을 지원하지 않습니다.

지원되는 함수

Amazon Athena 및 AWS IoT 분석 SQL 기능은 Presto 0.217을 기반으로 합니다. 관련된 함수, 연산자, 표현식에 대한 자세한 내용은 Presto 설명서에서 함수 및 연산자 및 다음의 특정 단원을 참조하십시오.

- 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
- 윈도 함수
- 색상 함수
- 배열 함수와 연산자
- 맵 함수와 연산자
- 람다 식과 함수
- Teradata 함수

> **Note**
>
> AWS IoT 분석 및 Amazon Athena는 사용자 정의 함수(UDFs 또는 UDAFs 지원하지 않습니다.

AWS IoT 분석에서 SQL 쿼리와 관련된 일반적인 문제 해결

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

- 작은 따옴표를 이스케이프 처리하려면 작은 따옴표 앞에 다른 작은 따옴표를 추가합니다. 이를 큰 따옴표와 혼동하지 마십시오.

  Example예제

  ```
  SELECT 'O''Reilly'
  ```

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

  Example예제

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

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

  Example예제

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

- 예약 키워드를 이스케이프 처리하려면 예약 키워드를 큰따옴표로 묶어야 합니다. 자세한 내용은 SQL SELECT 문의 예약 키워드 목록을 참조하십시오.

의 보안 AWS IoT 분석

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 AWS 규정 준수 프로그램별 범위 내 서비스를 AWS IoT 분석참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 데이터의 민감도, 조직의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다 AWS IoT 분석. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 AWS IoT 분석 를 구성하는 방법을 보여줍니다. 또한 AWS IoT 분석 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

AWS Identity and Access Management in AWS IoT 분석

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 누가 AWS IoT 분석 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한을 요청합니다(AWS IoT 분석 자격 증명 및 액세스 문제 해결 참조).
- 서비스 관리자 - 사용자 액세스를 결정하고 권한 요청을 제출합니다(AWS IoT 분석 에서 IAM을 사용하는 방법 참조).

- IAM 관리자 - 액세스를 관리하기 위한 정책을 작성합니다(AWS IoT 분석 자격 증명 기반 정책 예제 참조).

ID를 통한 인증

인증은 AWS 자격 증명으로에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증되어야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS Sign-In 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 자격 증명입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 권한 관리를 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 위탁자가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로, 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 이를 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 해당 권한을 정의합니다.

ID 기반 정책

자격 증명 기반 정책은 자격 증명(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

자격 증명 기반 정책은 인라인 정책(단일 자격 증명에 직접 포함) 또는 관리형 정책(여러 자격 증명에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

기타 정책 타입

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - 자격 증명 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) - 계정 내 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.

- 세션 정책 - 역할 또는 페더레이션 사용자에 대해 임시 세션을 생성할 때 파라미터 자격으로 전달되는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS IoT 분석 에서 IAM을 사용하는 방법

IAM을 사용하여에 대한 액세스를 관리하기 전에 사용할 수 있는 IAM 기능을 이해해야 AWS IoT 분석 합니다 AWS IoT 분석. AWS IoT 분석 및 기타 AWS 서비스에서 IAM을 사용하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

이 페이지의 주제:

- AWS IoT 분석 자격 증명 기반 정책
- AWS IoT 분석 리소스 기반 정책
- AWS IoT 분석 태그 기반 권한 부여
- AWS IoT 분석 IAM 역할

AWS IoT 분석 자격 증명 기반 정책

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부된 작업 및 리소스와 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다.는 특정 작업, 리소스 및 조건 키를 AWS IoT 분석 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

작업

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

의 정책 작업은 작업 앞에 다음 접두사를 AWS IoT 분석 사용합니다. `iotanalytics:` 예를 들어 API 작업으로 채널을 생성할 AWS IoT 분석 수 있는 AWS IoT 분석 `CreateChannel` 권한을 부여하려면 해당 정책에 `iotanalytics:BatchPuMessage` 작업을 포함합니다. 정책 설명에는 `Action` 또는

NotAction 요소가 포함되어야 합니다.는이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 세트를 AWS IoT 분석 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, Describe라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

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

AWS IoT 분석 작업 목록을 보려면 IAM 사용 설명서의에서 정의한 작업을 AWS IoT 분석 참조하세요.

Resources

Resource 요소는 작업이 적용되는 객체를 지정합니다. 문에는 Resource또는 NotResource요소가 반드시 추가되어야 합니다. ARN을 사용하거나 문이 모든 리소스에 적용됨을 표시하는 와일드카드(*)를 사용하여 리소스를 지정합니다.

AWS IoT 분석 데이터 세트 리소스에는 다음과 같은 ARN이 있습니다.

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

ARN 형식에 대한 자세한 내용은 Amazon 리소스 이름(ARN) 및 AWS 서비스 네임스페이스 단원을 참조하세요.

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/*"
```

리소스를 생성하기 위한 작업과 같은 일부 AWS IoT 분석 작업은 특정 리소스에서 수행할 수 없습니다. 이러한 경우, 와일드카드(*)를 사용해야 합니다.

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

일부 AWS IoT 분석 API 작업에는 여러 리소스가 포함됩니다. 예를 들어 `CreatePipeline`은 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 분석 리소스 유형 및 해당 ARNs 목록을 보려면 IAM 사용 설명서에서 정의한 리소스를 AWS IoT 분석 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 AWS IoT 분석가 정의한 작업을 참조하십시오.

조건 키

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같음, 미만 등 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 빌드할 수 있습니다.

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

조건을 지정할 때 자리 표시자 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 분석 는 서비스별 조건 키를 제공하지 않지만 일부 전역 조건 키 사용을 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

예시

자격 AWS IoT 분석 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS IoT 분석 자격 증명 기반 정책 예제.

AWS IoT 분석 리소스 기반 정책

AWS IoT 분석 는 리소스 기반 정책을 지원하지 않습니다. 자세한 리소스 기반 정책 페이지의 예를 보려면 AWS Lambda 개발자 안내서에서 AWS Lambda에 대해 리소스 기반 정책 사용하기를 참조하십시오.

AWS IoT 분석 태그 기반 권한 부여

AWS IoT 분석 리소스에 태그를 연결하거나 요청에 태그를 전달할 수 있습니다 AWS IoT 분석. 태그를 기반으로 액세스를 제어하려면 `iotanalytics:ResourceTag/{key-name}`, `aws:RequestTag/{key-name}` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다. AWS IoT 분석 리소스 태그 지정에 대한 자세한 내용은 리소스 태그 지정을 참조하세요 AWS IoT 분석.

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하기 위한 자격 증명 기반 정책 예제를 보려면 태그를 기반으로 AWS IoT 분석 채널 보기를 참조하세요.

AWS IoT 분석 IAM 역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정 내 엔터티입니다.

에서 임시 자격 증명 사용 AWS IoT 분석

임시 보안 인증을 사용하여 페더레이션을 통해 로그인하거나, IAM 역할을 맡거나, 교차 계정 역할을 맡을 수 있습니다. AssumeRole 또는 GetFederationToken 같은 AWS Security Token Service (AWS STS) API 작업을 호출하여 임시 보안 인증을 가져옵니다.

AWS IoT 분석 는 임시 자격 증명 사용을 지원하지 않습니다.

서비스 연결 역할

서비스 라인 역할을 사용하면 AWS 서비스가 다른 서비스의 리소스에 액세스하여 사용자를 대신하여 작업을 완료할 수 있습니다. 서비스 연결 역할은 IAM 계정에 나타나고 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수 없습니다.

AWS IoT 분석 는 서비스 연결 역할을 지원하지 않습니다.

서비스 역할

이 기능을 사용하면 서비스가 사용자를 대신하여 서비스 역할을 수임할 수 있습니다. 이 역할을 사용하면 서비스가 다른 서비스의 리소스에 액세스해 사용자를 대신해 작업을 완료할 수 있습니다. 서비스 역

할은 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 분석 는 서비스 역할을 지원합니다.

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS교차 서비스 가장은 혼동된 대리자 문제를 초래할 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 AWS 에서는 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 리소스에 다른 서비스를 AWS IoT Analytics 부여하는 권한이 제한됩니다. 두 전역 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. 예를 들어 `arn:aws:`*`iotanalytics`*`::`*`123456789012`*`:*`입니다.

주제

- Amazon S3 버킷에 대한 예방
- Amazon CloudWatch Logs를 통한 예방
- 고객 관리형 AWS IoT 분석 리소스에 대한 혼동된 대리자 방지

Amazon S3 버킷에 대한 예방

AWS IoT 분석 데이터 스토어에 고객 관리형 Amazon S3 스토리지를 사용하는 경우 데이터를 저장하는 Amazon S3 버킷이 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 분석 데이터 스토어에 대한 정보가 저장됩니다. 서비스 AWS IoT 분석 보안 주체가 자신을 대신하여 *`DOC-EXAMPLE-BUCKET`*을 쿼리할 수 있도록 하는 정

책을 지정합니다. Nikki의 동료인 Li Juan은 자신의 계정에서 *DOC-EXAMPLE-BUCKET*을 쿼리하고 결과가 포함된 데이터 세트를 만듭니다. 따라서 AWS IoT 분석 서비스 보안 주체는 Li가 자신의 계정에서 쿼리를 실행했더라도 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 - 버킷 정책의 다음 예제에서는 Nikki 계정(*123456789012*)의 AWS IoT 분석 리소스만 *DOC-EXAMPLE-BUCKET*에 액세스할 수 있도록 지정합니다.

JSON

```
{
    "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:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
```

```
    ]
}
```

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

JSON

```
{
    "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:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset",
                        "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-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 `aws:SourceAccount` 인
- 의 `sts:AssumeRole` 요청과 연결된 고객 리소스 AWS IoT 분석

다음 예제에서 *123456789012*을 AWS 계정 ID로 바꾸고 *us-east-1*을 AWS IoT 분석 계정의 리전으로 바꿉니다.

JSON

```
{
    "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 참조하십시오.

고객 관리형 AWS IoT 분석 리소스에 대한 혼동된 대리자 방지

AWS IoT 분석 리소스에 대한 작업을 수행할 수 있는 AWS IoT 분석 권한을 부여하면 리소스가 혼동된 대리자 문제에 노출될 수 있습니다. 혼동된 대리자 문제를 방지하려면 다음 예제 리소스 정책을 AWS IoT 분석 사용하여에 부여된 권한을 제한할 수 있습니다.

주제

- AWS IoT 분석 채널 및 데이터 스토어에 대한 방지
- AWS IoT 분석 데이터 세트 콘텐츠 전송 규칙에 대한 교차 서비스 혼동된 대리자 방지

AWS IoT 분석 채널 및 데이터 스토어에 대한 방지

IAM 역할을 사용하여가 사용자를 대신하여 액세스할 AWS IoT 분석 수 있는 AWS 리소스를 제어합니다. 역할이 혼동된 대리자 문제에 노출되지 않도록 하려면 `aws:SourceAccount` 요소에 AWS 계정을 지정하고 역할에 연결하는 신뢰 정책의 `aws:SourceArn` 요소에 AWS IoT 분석 리소스의 ARN을 지정할 수 있습니다.

다음 예제에서 *123456789012*을 AWS 계정 ID로 바꾸고 *arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-EXAMPLE-CHANNEL*을 AWS IoT 분석 채널 또는 데이터 스토어의 ARN으로 바꿉니다.

JSON

```
{
    "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:us-
east-1:123456789012:channel/your-channel"
                }
            }
        }
    ]
}
```

채널 및 데이터 스토어의 고객 관리형 S3 스토리지 옵션에 대한 자세한 내용은 AWS IoT 분석 API 참조의 CustomerManagedChannelS3Storage 및 CustomerManagedDatastoreS3Storage 항목을 참조하십시오.

AWS IoT 분석 데이터 세트 콘텐츠 전송 규칙에 대한 교차 서비스 혼동된 대리자 방지

가 Amazon S3 또는에 데이터 세트 쿼리 결과를 전달하기 위해 AWS IoT 분석 수임하는 IAM 역할은 혼동된 대리자 문제에 노출될 AWS IoT Events 수 있습니다. 혼동된 대리자 문제를 방지하려면 `aws:SourceAccount` 요소에 AWS 계정을 지정하고 역할에 연결하는 신뢰 정책의 `aws:SourceArn` 요소에 AWS IoT 분석 리소스의 ARN을 지정합니다.

JSON

```
{
    "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:us-
east-1:123456789012:dataset/your-dataset"
                }
            }
```

```
        }
    ]
}
```

데이터 세트 콘텐츠 전송 규칙을 구성하는 방법에 대한 자세한 내용은 AWS IoT 분석 API 참조의 `contentDeliveryRules` 항목을 참조하십시오.

AWS IoT 분석 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할에는 AWS IoT 분석 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console AWS CLI또는 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 생성을 참조하십시오.

이 페이지의 주제:

- 정책 모범 사례
- AWS IoT 분석 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- 하나의 AWS IoT 분석 입력에 액세스
- 태그를 기반으로 AWS IoT 분석 채널 보기

정책 모범 사례

자격 증명 기반 정책은 매우 강력합니다. 계정에서 사용자가 AWS IoT 분석 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 - AWS IoT 분석 빠르게 사용을 시작하려면 AWS 관리형 정책을 사용하여 직원에게 필요한 권한을 부여합니다. 이 정책은 이미 계정에서 사용할 수 있으며 AWS에 의해 유지 관리 및 업데이트됩니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책으로 권한 사용 시작하기를 참조하세요.
- 최소 권한 부여 - 사용자 지정 정책을 생성할 때 오직 작업을 수행하는 데 필요한 권한만을 부여합니다. 최소한의 권한 세트로 시작하여 필요에 따라 추가 권한을 부여합니다. 처음부터 권한을 많이 부

여한 후 나중에 줄이는 방법보다 이 방법이 안전합니다. 자세한 정보는 IAM 사용 설명서의 최소 권한 부여를 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 설명서의 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. 또한 지정된 날짜 또는 시간 범위 내에서만 요청을 허용하거나 SSL 또는 MFA를 사용해야 하는 조건을 작성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하십시오.

AWS IoT 분석 콘솔 사용

AWS IoT 분석 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은의 AWS IoT 분석 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

이러한 엔터티가 AWS IoT 분석 콘솔을 계속 사용할 수 있도록 하려면 다음 AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

JSON

```
{
     "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:aws:iotanalytics:us-
east-1:123456789012:channel/your-channel-name",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-datasetName",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastoreName",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:pipeline/your-pipelineName"
        }
    ]
}
```

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 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 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": [
                "arn:aws:iam:*:*:user/username"
            ]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

하나의 AWS IoT 분석 입력에 액세스

이 예제에서는 AWS IoT 분석 채널 중 하나인에 대한 AWS 계정 액세스 권한을의 사용자에게 부여하려고 합니다`exampleChannel`. 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 제어를 참조하십시오.

JSON

```
{
   "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 분석 채널 보기

자격 증명 기반 정책의 조건을 사용하여 태그를 기반으로 AWS IoT 분석 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.

JSON

```
{
    "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 사용자가를 보richard-roe려고 하면 AWS IoT 분석 channel에 태그를 지정해야 channel 합니다Owner=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: 조건을 참조하십시오.

AWS IoT 분석 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 작업 시 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다 AWS IoT 분석.

항목

- 에서 작업을 수행할 권한이 없음 AWS IoT 분석
- iam:PassRole을 수행할 권한이 없음
- 내 외부의 AWS 계정 사람이 내 AWS IoT 분석 리소스에 액세스하도록 허용하고 싶습니다.

에서 작업을 수행할 권한이 없음 AWS IoT 분석

에서 작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 문의하여 도움을 받아야 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 분석에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 AWS IoT 분석에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

```
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 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 AWS 계정 사람이 내 AWS IoT 분석 리소스에 액세스하도록 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.

- 에서 이러한 기능을 AWS IoT 분석 지원하는지 여부를 알아보려면 AWS IoT 분석 에서 IAM을 사용하는 방법을 참조하세요.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 리소스에 대한 액세스 권한을 타사에 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 소유에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하십시오.
- 교차 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하십시오.

에서 로깅 및 모니터링 AWS IoT 분석

AWS 는 모니터링에 사용할 수 있는 도구를 제공합니다 AWS IoT 분석. 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 문제 발생 시 모니터링하고 AWS IoT 보고할 수 있습니다.

- Amazon CloudWatch Logs - AWS CloudTrail 또는 기타 소스에서 로그 파일을 모니터링, 저장 및 액세스합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 AWS CloudTrail로그 파일 모니터링은 무엇입니까를 참조하십시오.
- AWS CloudTrail 로그 모니터링 - 계정 간에 로그 파일을 공유하고, CloudTrail 로그 파일을 CloudWatch Logs로 전송하여 실시간으로 모니터링하고, Java로 로그 처리 애플리케이션을 작성하고, CloudTrail에서 전송한 후 로그 파일이 변경되지 않았는지 확인합니다. 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 로그 파일 작업을 참조하십시오.

수동 모니터링 도구

모니터링의 또 다른 중요한 부분은 CloudWatch 경보 AWS IoT 가 다루지 않는 항목을 수동으로 모니터링하는 것입니다. AWS IoT, CloudWatch 및 기타 AWS 서비스 콘솔 대시보드는 AWS 환경 상태를 at-a-glance 볼 수 있습니다. 로그 파일도 확인하는 것이 좋습니다 AWS IoT 분석.

- AWS IoT 분석 콘솔에 다음이 표시됩니다.
  - 채널
  - Pipelines
  - 데이터 스토어
  - 데이터 세트
  - Notebooks
  - Settings
  - 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.
  - 맞춤 대시보드를 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 분석 는 Amazon CloudWatch를 사용한 로깅을 지원합니다. PutLoggingOptions API 작업을 사용하여 AWS IoT 분석 에 대한 Amazon CloudWatch 로깅을 활성화하고 구성할 수 있습니다. 이 섹션에서는 AWS Identity and Access Management (IAM)와 PutLoggingOptions 함께를 사용하여 Amazon CloudWatch 로깅을 구성하고 활성화하는 방법을 설명합니다 AWS IoT 분석.

CloudWatch Logs에 대한 자세한 내용은 Amazon CloudWatch Logs User Guide를 참조하십시오. AWS IAM에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서를 참조하세요.

> **Note**
>
> AWS IoT 분석 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 참조하십시오.

IAM 역할을 생성하여 로깅을 활성화합니다

Amazon CloudWatch에 대한 로깅을 활성화하기 위한 IAM 역할을 만들려면

1. AWS IAM 콘솔 또는 다음 AWS IAM CLI 명령 CreateRole을 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*을 AWS 계정 ID로 바꾸고 *aws-region*을 AWS 리소스의 AWS 리전으로 바꿉니다. 자세한 내용은 the section called “교차 서비스 혼동된 대리인 방지” 단원을 참조하십시오.

JSON

```
{
    "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:us-
east-1:123456789012:*"
                }
            }
        }
    ]
}
```

나중에 명령을 호출할 때이 역할의 ARN을 AWS IoT 분석 `PutLoggingOptions` 사용합니다.

2. AWS IAM PutRolePolicy를 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.

   JSON

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

```
    {
     "Effect": "Allow",
     "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream"
     ],
     "Resource": [
      "arn:aws:logs:*:*:*"
     ]
    }
   ]
}
```

3. Amazon CloudWatch에 로깅 이벤트를 넣을 수 있는 AWS IoT 분석 권한을 부여하려면 Amazon CloudWatch 명령 PutResourcePolicy를 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 대리인 방지”를 참조하십시오.

```
aws logs put-resource-policy --policy-in-json
exampleResourcePolicy.json
```

`exampleResourcePolicy.json` 파일에는 다음 리소스 정책이 포함됩니다.

JSON

```
{
    "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 분석에 대한 Amazon CloudWatch 로깅을 구성 및 활성화합니다. `loggingOptions` 필드의 `roleArn`은 이전 섹션에서 생성이 완료된 역할의 ARN이어야 합니다. 또한 `DecribeLoggingOptions` 명령을 사용하여 로깅 옵션 설정을 점검할 수 있습니다.

PutLoggingOptions

AWS IoT 분석 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 단원을 참조하십시오.

DescribeLoggingOptions

AWS IoT 분석 로깅 옵션의 현재 설정을 검색합니다. 자세한 내용은 DescribeLoggingOptions 섹션을 참조하세요.

네임스페이스, 지표 및 측정기준

AWS IoT 분석 는 다음 지표를 Amazon CloudWatch 리포지토리에 넣습니다.

네임스페이스
AWS/IoTAnalytics

지표	Description
ActionExecution	실행된 작업의 수입니다.
ActionExecutionThrottled	제한된 작업 수.
ActivityExecutionError	파이프라인 활동을 실행하는 동안 생성된 오류의 수입니다.
IncomingMessages	채널로 들어오는 메시지의 수입니다.
PipelineConcurrentExecutionCount	동시에 실행된 파이프라인 활동 수.

차원	Description
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 분석 는 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 참조하십시오.

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 간접 호출하여 Amazon S3 객체에 저장된 채널 메시지를 처리할 수 있습니다. Amazon CloudWatch Events는 다음과 같은 다양한 유형의 대상을 지원합니다.

- Amazon Kinesis 스트림
- 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 대상을 참조하십시오.

CloudWatch Events 리소스 및 관련 대상은 AWS IoT 분석 리소스를 생성한 AWS 리전에 있어야 합니다. 자세한 내용은 AWS 일반 참조의 서비스 엔드포인트 및 할당량을 참조하십시오.

AWS Lambda 활동의 런타임 오류에 대해 Amazon CloudWatch Events로 전송된 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 QueryFilter 때에 대한 deltaTime 오프셋을 지정할 수 있습니다. AWS IoT 분석 는 여전히 델타 시간 내에 도착하는 데이터를 처리하고 데이터 세트 콘텐츠에는 시간 지연이 있습니다. 지연 데이

터 알림 기능을 사용하면 델타 시간 이후에 데이터가 도착할 때 AWS IoT 분석 가 Amazon CloudWatch Events를 통해 알림을 보낼 수 있습니다.

AWS IoT 분석 콘솔, API, AWS Command Line Interface (AWS CLI) 또는 AWS SDK를 사용하여 데이터 세트에 대한 지연 데이터 규칙을 지정할 수 있습니다.

AWS IoT 분석 API에서 `LateDataRuleConfiguration` 객체는 데이터 세트의 지연 데이터 규칙 설정을 나타냅니다. 이 객체는 `CreateDataset` 및 `UpdateDataset` API 작업과 관련된 `Dataset` 객체에 포함되어 있습니다.

파라미터

를 사용하여 데이터 세트에 대한 지연 데이터 규칙을 생성할 때 다음 정보를 지정 AWS IoT 분석해야 합니다.

`ruleConfiguration` (**`LateDataRuleConfiguration`**)

지연 데이터 규칙의 구성 정보를 포함하는 구조입니다.

`deltaTimeSessionWindowConfiguration`

델타 시간 세션 기간의 구성 정보를 포함하는 구조입니다.

DeltaTime은 시간 간격을 지정합니다. `DeltaTime`을 사용하면 마지막 실행 이후 데이터 스토어에 도착한 데이터로 데이터 세트 콘텐츠를 생성할 수 있습니다. `DeltaTime`의 예제를 알아보려면 델타 기간으로 SQL 데이터 세트 생성(CLI)을 참조하십시오.

`timeoutInMinutes`

시간 간격입니다. AWS IoT 분석 가 마지막 실행 이후 생성된 지연 데이터 알림을 일괄 처리할 수 `timeoutInMinutes` 있도록를 사용할 수 있습니다. AWS IoT 분석 는 한 번에 하나의 알림 배치를 CloudWatch Events로 보냅니다.

유형: 정수

유효 범위: 1~60

`ruleName`

지연 데이터 규칙의 이름입니다.

유형: 문자열

> **Important**
>
> `lateDataRules`를 지정하려면 데이터 세트에 `DeltaTime` 필터를 사용해야 합니다.

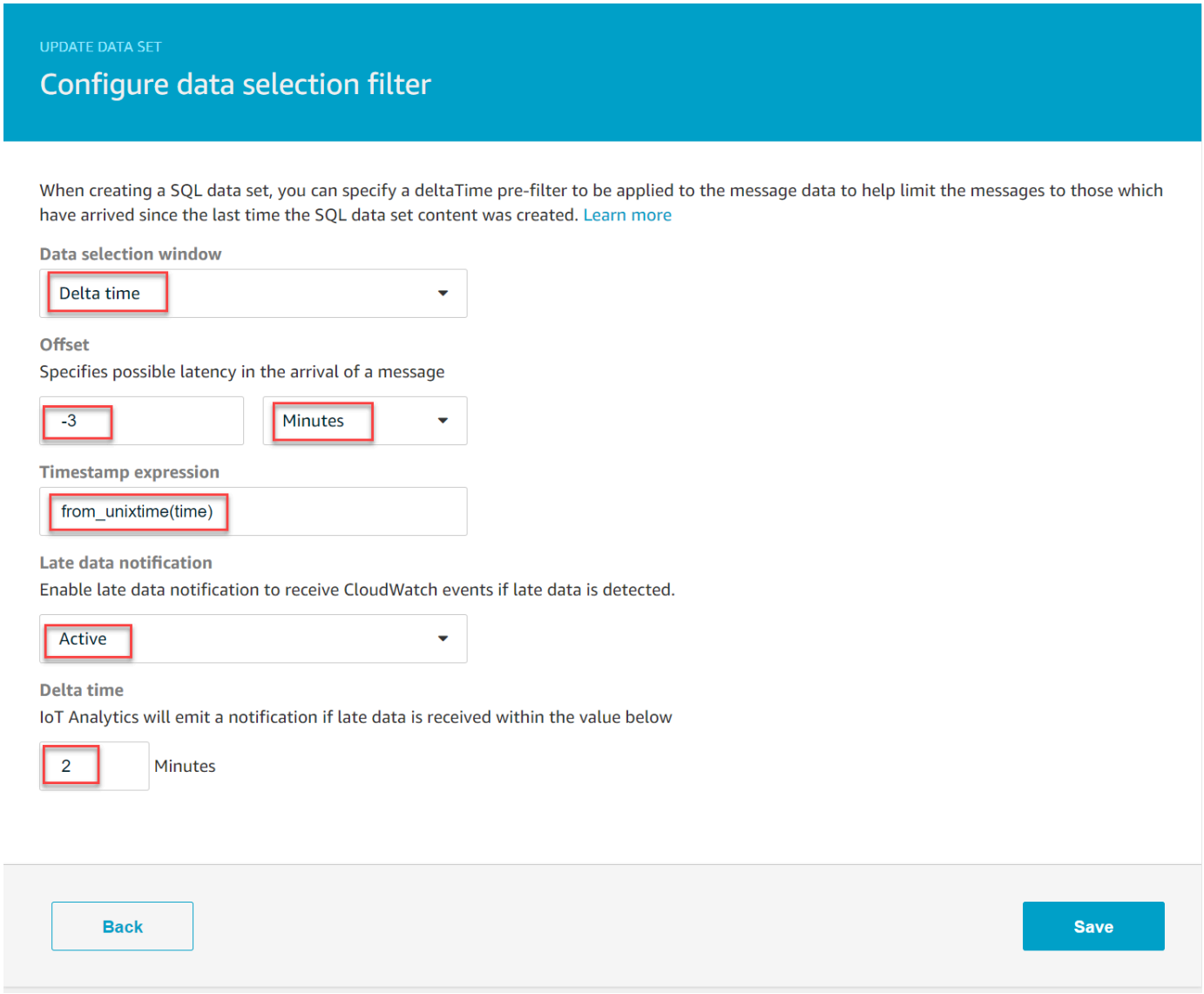
지연 데이터 규칙 구성(콘솔)

다음 절차는 AWS IoT 분석 콘솔에서 데이터 세트의 지연 데이터 규칙을 구성하는 방법을 보여줍니다.

최신 데이터 규칙을 구성하려면

1. AWS IoT 분석 콘솔에 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 기간을 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 연산자를 참조하십시오.
    d. 지연 데이터 알림의 경우 활성을 선택합니다.
    e. 델타 시간에는 정수를 입력합니다. 값의 범위는 1~60입니다.
    f. 저장을 선택합니다.



지연 데이터 규칙(CLI) 구성

AWS IoT 분석 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 AWS SDK를 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 연산자를 참조하십시오.
- *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 분석에서 전송된 지연 데이터 알림을 처리하는 방법을 정의하는 규칙을 CloudWatch Events에서 만들 수 있습니다. CloudWatch Events는 알림을 수신하면 규칙에 정의된 지정된 대상 작업을 간접 호출합니다.

CloudWatch Events 규칙 생성을 위한 사전 조건

에 대한 CloudWatch Events 규칙을 생성하기 전에 다음을 수행해야 AWS IoT 분석합니다.

- CloudWatch Events의 이벤트, 규칙 및 대상을 익힙니다.
- CloudWatch Events 규칙에 의해 간접 호출되는 대상을 생성하고 구성해야 합니다. 규칙은 다음과 같은 다양한 유형의 대상을 간접 호출할 수 있습니다.
  - Amazon Kinesis 스트림
  - AWS Lambda 함수
  - Amazon Simple Notification Service(Amazon SNS) 주제
  - Amazon Simple Queue Service(Amazon SQS) 대기열

  CloudWatch Events 규칙 및 연결된 대상은 AWS IoT 분석 리소스를 생성한 AWS 리전에 있어야 합니다. 자세한 내용은 AWS 일반 참조의 서비스 엔드포인트 및 할당량을 참조하십시오.

자세한 정보는 Amazon CloudWatch Events 사용 설명서의 Amazon CloudWatch Events란 무엇인가요? 및 Amazon CloudWatch Events 시작하기를 참조하십시오.

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 대기열로 AWS IoT 분석 지연 데이터 알림을 보내는 규칙을 생성하는 방법을 보여줍니다.

CloudWatch Events 규칙을 생성하려면

1. Amazon CloudWatch 콘솔에 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.

를 사용하여 AWS IoT 분석 API 호출 로깅 AWS CloudTrail

AWS IoT 분석 는 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다 AWS IoT 분석. CloudTrail은 AWS IoT 분석 콘솔의 호출 및 API에 대한 코드 호출을 포함하여에 대한 API 호출의 하위 집합을 이벤트 AWS IoT 분석 로 캡처합니다. AWS IoT 분석 APIs 추적을 생성하면 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다 AWS IoT 분석. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 수행된 요청, 요청이 수행된 AWS IoT 분석 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

AWS IoT 분석 의 정보 AWS CloudTrail

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. 에서 활동이 발생하면 AWS IoT 분석해당 활동은 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 내용은 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 AWS IoT 분석생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 지역에 추적이 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- 추적 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

AWS IoT 분석 는 다음 작업을 CloudTrail 로그 파일에 이벤트로 로깅할 수 있도록 지원합니다.

- 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 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

AWS IoT 분석 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 직접 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 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```
{
"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"
}
```


에 대한 규정 준수 검증 AWS IoT 분석

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 범위 내를 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 Downloading Reports inDownloading AWS Artifact을 참조하세요.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다. 사용 시 규정 준수 책임에 대한 자세한 내용은 AWS 보안 설명서를 AWS 서비스참조하세요.

의 복원력 AWS IoT 분석

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전은 물리적으로 분리되고 격리된 다수의 가용 리전을 제공하며 이러한 가용 리전은 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 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

의 인프라 보안 AWS IoT 분석

관리형 서비스인는 AWS 글로벌 네트워크 보안으로 보호 AWS IoT Analytics 됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요 AWS .

AWS 에서 게시한 API 호출을 사용하여 네트워크를 AWS IoT 분석 통해에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(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.

AWS IoT 분석 할당량

이 AWS 일반 참조 안내서에서는 AWS IoT 분석 AWS 계정의 기본 할당량을 제공합니다. 지정하지 않는 한 각 할당량은 AWS 리전당입니다. 자세한 내용은 AWS 일반 참조 안내서의 AWS IoT 분석 엔드포인트 및 할당량 및 AWS 서비스 할당량을 참조하십시오.

서비스 할당량 증가를 요청하려면 지원 센터 콘솔에서 지원 사례를 제출하십시오. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하십시오.

AWS IoT 분석 명령

이 주제를 읽고 지원되는 웹 서비스 프로토콜에 대한 샘플 요청 AWS IoT 분석, 응답 및 오류를 포함하여의 API 작업에 대해 알아봅니다.

AWS IoT 분석 작업

AWS IoT 분석 API 명령을 사용하여 IoT 데이터를 수집, 처리, 저장 및 분석할 수 있습니다. 자세한 내용은 AWS IoT 분석 API 참조의 AWS IoT 분석 에서 지원하는 작업을 참조하세요.

AWS CLI 명령 참조의 AWS IoT 분석 섹션에는 관리 및 조작에 사용할 수 있는 AWS CLI 명령이 포함되어 있습니다 AWS IoT 분석.

AWS IoT 분석 데이터

AWS IoT 분석 데이터 API 명령을 사용하여 AWS IoT 분석 `channel`, `pipeline`, `datastore`및를 사용하여 고급 활동을 수행할 수 있습니다`dataset`. 자세한 내용은 AWS IoT 분석 API 참조의 데이터에서 지원하는 데이터 유형을 참조하세요. AWS IoT 분석

문제 해결 AWS IoT 분석

오류를 해결하고 문제를 해결할 수 있는 및 가능한 해결 방법을 찾으려면 다음 섹션을 참조하세요 AWS IoT 분석.

항목

제 메시지가 AWS IoT 분석로 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 2를 참조하십시오.

- Lambda 활동으로 간접 호출된 Lambda 함수에 권한이 부족합니다.

  Lambda 활동의 각 Lambda 함수에 AWS IoT 분석 서비스에서 호출할 권한이 있는지 확인합니다. 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 분석를 사용하도록 노트북 인스턴스를 올바르게 구성하려면 어떻게 해야 합니까?

노트북 인스턴스를 만들 때 사용한 IAM 역할에 필요한 권한이 있는지 다음 단계에 따라 확인하십시오.

1. SageMaker AI 콘솔로 이동하여 노트북 인스턴스를 생성합니다.
2. 세부 정보를 입력하고 새 역할 생성을 선택합니다. 역할 ARN을 기록해 둡니다.
3. 노트북 인스턴스를 생성합니다. 이렇게 하면 SageMaker AI가 사용할 수 있는 역할도 생성됩니다.
4. IAM 콘솔로 이동하여 새로 생성된 SageMaker AI 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. 이제 새로 생성된 역할에 데이터 세트를 읽을 수 있는 정책 권한이 있습니다 AWS IoT 분석.
7. AWS IoT 분석 콘솔로 이동하여 노트북 인스턴스에서 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 확인하십시오.

Quick Suite에서 데이터 세트를 볼 수 없는 이유는 무엇인가요?

Quick Suite에는 AWS IoT 분석 데이터 세트 콘텐츠를 읽을 수 있는 권한이 필요할 수 있습니다. 권한을 부여하려면 다음 단계를 수행합니다.

1. Quick Suite의 오른쪽 상단 모서리에서 계정 이름을 선택하고 QuickSight 관리를 선택합니다.
2. 왼쪽 탐색 창에서 보안 & 권한을 선택합니다. QuickSight access to AWS services에서 AWS IoT 분석에 액세스 권한이 부여되었는지 확인합니다.

a. AWS IoT 분석 에 액세스 권한이 없는 경우 추가 또는 제거를 선택합니다.

b. AWS IoT 분석 옆에 있는 상자를 선택한 다음 업데이트를 선택합니다. 이렇게 하면 Quick Suite에 데이터 세트 콘텐츠를 읽을 수 있는 권한이 부여됩니다.

3. 데이터를 다시 시각화합니다.

AWS IoT 분석 및 Quick Suite 모두에 대해 동일한 AWS 리전을 선택해야 합니다. 그렇지 않으면 AWS 리소스에 액세스하는 데 문제가 있을 수 있습니다. 지원되는 리전 목록은 AWS IoT 분석 의 엔드포인트 및 할당량과 Quick Suite 엔드포인트 및 할당량을 참조하세요Amazon Web Services 일반 참조.

기존 Jupyter Notebook에서 컨테이너화 버튼을 볼 수 없는 이유가 무엇입니까?

- 이는 AWS IoT 분석 컨테이너화 플러그인 누락으로 인해 발생합니다. 2018년 8월 23일 이전에 SageMaker 노트북 인스턴스를 생성한 경우, 노트북 컨테이너화에 있는 지침에 따라 수동으로 플러그인을 설치해야 합니다.
- AWS IoT 분석 콘솔에서 SageMaker 노트북 인스턴스를 생성하거나 수동으로 설치한 후 컨테이너화 버튼이 표시되지 않으면 AWS IoT 분석 기술 지원에 문의하세요.

컨테이너화 플러그인 설치가 실패하는 이유가 무엇입니까?

- 일반적으로 SageMaker 노트북 인스턴스에 권한이 누락되어 있어 플러그인 설치가 실패합니다. 노트북 인스턴스의 필수 권한은 권한을 참조하고, 노트북 인스턴스 역할의 필수 권한을 추가하십시오. 문제가 지속되면 AWS IoT 분석 콘솔에서 새 노트북 인스턴스를 생성합니다.
- 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 권한을 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 분석 컨테이너화 플러그인은 “컨테이너화된” 커널을 사용하여 노트북을 실행한 후 노트북의 모든 변수를 자동으로 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 분석 컨테이너화 Jupyter 플러그인은 노트북의 변수를 자동으로 인식하고 컨테이너화 프로세스의 일부로 사용할 수 있도록 하여이 프로세스를 간소화합니다.

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 분석 Jupyter 플러그인을 사용해 컨테이너화한 경우, containerDataset의 executionRole과 동일한 역할로 새 SageMaker 노트북 인스턴스를 생성한 후 수동으로 노트북을 실행합니다. Jupyter 플러그인 밖에서 컨테이너를 생성한 경우, 수동으로 코드를 실행하고 권한을 executionRole로 제한합니다.

문서 이력

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
지원 종료 알림	지원 종료 알림: 2025년 12월 15일에 AWS 에 대한 지원이 종료됩니다 AWS IoT 분석. 2025년 12월 15일 이후에는 AWS IoT 분석 콘솔 또는 AWS IoT 분석 리소스에 더 이상 액세스할 수 없습니다. 자세한 내용은 AWS IoT 분석 지원 종료를 참조하세요.	2025년 5월 20일
AWS IoT 분석 신규 고객은를 더 이상 사용할 수 없습니다.	AWS IoT 분석 는 더 이상 신규 고객이 사용할 수 없습니다. 의 기존 고객은 평소와 같이 서비스를 계속 사용할 AWS IoT 분석 수 있습니다. 자세히 알아보기	2024년 8월 8일
리전 출시	AWS IoT 분석 이제 아시아 태평양(뭄바이) 리전에서를 사용할 수 있습니다.	2021년 8월 18일
`JOIN`로 쿼리하기	이 업데이트를 사용하면를 사용하여 AWS IoT 분석 데이터 세트를 쿼리`JOIN`할 수 있습니다.	2021년 7월 27일
와 통합 AWS IoT SiteWise	이제를 사용하여 AWS IoT SiteWise 데이터를 쿼리 AWS IoT 분석 할 수 있습니다.	2021년 7월 27일

사용자 지정 파티션	AWS IoT 분석 는 이제 일반적으로 메시지 속성 또는 파이프라인 활동을 통해 추가된 속성에 따라 데이터 파티셔닝을 지원합니다.	2021년 6월 14일
채널 메시지 재처리	이 업데이트를 통해 지정된 Amazon S3 객체의 채널 데이터를 재처리할 수 있습니다.	2020년 12월 15일
Parquet 스키마	AWS IoT 분석 이제 데이터 스토어는 Parquet 파일 형식을 지원합니다.	2020년 12월 15일
CloudWatch 이벤트로 모니터링	AWS IoT 분석 는 AWS Lambda 활동 중에 런타임 오류가 발생하면 Amazon CloudWatch Events에 이벤트를 자동으로 게시합니다.	2020년 12월 15일
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) AWS IoT 분석 에서 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) 리전 AWS IoT 분석 에서 출시되었습니다.	2020년 7월 16일
업데이트	재구성된 설명서.	2020년 5월 7일

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