



사용자 가이드

AWS IoT SiteWise



AWS IoT SiteWise: 사용자 가이드

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

무엇입니까 AWS IoT SiteWise?	1
작동 방식	2
산업 데이터 수집	2
자산을 모델링하여 수집된 데이터를 상황에 맞게 조정합니다.	3
쿼리, 알람 및 예측을 사용하여 분석합니다.	4
운영을 시각화하세요.	4
데이터 저장	4
다른 서비스와 통합	5
개념	5
사용 사례	9
제조	10
식음료	10
에너지/공공 설비	10
시작하기	11
요구 사항	11
설정 AWS 계정	12
가입하기 AWS 계정	12
관리자 액세스 권한이 있는 사용자 생성	12
빠른 시작 데모 사용	13
데모 만들기 AWS IoT SiteWise	14
데모 삭제 AWS IoT SiteWise	16
튜토리얼	17
OEE 계산	17
필수 조건	17
OEE 계산 방법	18
사물의 데이터 수집 AWS IoT	20
사전 조건	21
1단계: 정책 생성	21
2단계: AWS IoT 사물 만들기	23
3단계: 기기 자산 모델 만들기	25
4단계: 디바이스 플릿 생성	27
5단계: 기기 표현하기	28
6단계: 디바이스 플릿 표현	29
7단계: 기기로 데이터 전송	30

8단계: 기기 클라이언트 스크립트	33
9단계: 리소스 정리	40
모니터에서 데이터 시각화 및 공유 SiteWise	41
필수 조건	42
1단계: 포털 생성	43
2단계: 포털에 로그인	46
3단계: 프로젝트 생성	48
4단계: 대시보드 만들기	51
5단계: 포털 둘러보기	58
6단계: 리소스 정리	59
Amazon DynamoDB에 속성 값 업데이트 게시	61
사전 조건	62
1단계: 속성 값 업데이트를 AWS IoT SiteWise 게시하도록 구성	62
2단계: 규칙 생성	65
3단계: DynamoDB 테이블 생성	67
4단계: 규칙 작업 구성	69
5단계: 데이터 탐색	70
6단계: 리소스 정리	71
에 데이터 수집 AWS IoT SiteWise	75
데이터 스트림 관리	75
데이터 스트림 관리	76
API 사용 AWS IoT SiteWise	84
규칙 사용 AWS IoT Core	87
필수 액세스 권한 부여	87
규칙 작업 구성	88
Basic Ingest를 통한 비용 절감	96
액션 사용 AWS IoT Events	97
AWS IoT Greengrass 스트림 관리자 사용	98
API 사용 CreateBulkImportJob	98
대량 가져오기 작업 생성(AWS CLI)	100
대량 가져오기 작업 설명(AWS CLI)	103
대량 가져오기 작업 목록(AWS CLI)	104
SiteWise 에지 게이트웨이 사용	106
요구 사항	106
요구 사항	107
SiteWise 에지 게이트웨이 생성	110

SiteWise 에지 게이트웨이 생성	110
로컬 장치에 SiteWise Edge 게이트웨이 소프트웨어 설치	112
엣지 데이터 처리 활성화	114
엣지 기능 설정하기	115
엣지에서 데이터 처리	117
퍼블리셔 구성	118
데이터 소스 구성	121
OPC-UA 소스 설정	122
데이터 소스 인증 구성	143
소스 서버 데이터의 목적지 선택	147
파트너 데이터 소스 추가	150
보안	150
파트너 데이터 소스 추가	151
에지 게이트웨이에 docker 설정 SiteWise	152
파트너 데이터 소스	153
팩 사용	153
팩 업그레이드하기	153
SiteWise 에지 게이트웨이 관리	154
콘솔을 AWS IoT SiteWise 통한 SiteWise 에지 게이트웨이 관리	155
다음을 사용하여 SiteWise AWS OpsHub 에지 게이트웨이를 관리합니다. AWS IoT SiteWise	155
로컬 운영 체제 자격 SiteWise 증명을 사용하여 에지 게이트웨이에 액세스	157
SiteWise 에지 게이트웨이 인증서 관리	159
SiteWise Edge 게이트웨이 구성 요소 팩의 버전 변경	160
지멘스 SiteWise 인더스트리얼 엣지의 러닝 엣지	161
필수 조건	161
보안	162
구성 파일 생성	162
문제 해결	163
문의처	164
자산 필터링	165
엣지 필터링 설정	165
API 사용	166
AWS IoT SiteWise 엣지 디바이스에 사용할 수 있는 모든 API	166
엣지 전용 API	167
자습서: 자산 모델 목록 가져오기	170

백업 및 복원 SiteWise 에지 게이트웨이	179
지표 데이터의 일일 백업	179
SiteWise 에지 게이트웨이 복원	180
AWS IoT SiteWise 데이터 복원	181
백업 및 복원의 성공 여부 확인	182
SiteWise 에지 게이트웨이 설정 (AWS IoT Greengrass Version 1)	184
AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 장치 선택	185
AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 구성	186
AWS IoT Greengrass V1 SiteWise 에지 게이트웨이에서 데이터 소스 구성	204
산업 자산 모델링	224
자산 및 모델 상태	225
자산의 상태 확인	226
자산 모델 또는 구성 요소 모델의 상태 확인	228
사용자 지정 복합 모델 (구성 요소)	230
인라인 사용자 지정 복합 모델	231
Component-model-based 사용자 지정 복합 모델	232
경로를 사용하여 사용자 지정 복합 모델 속성 참조	234
개체 ID로 작업하기	235
개체 UUID로 작업하기	236
외부 ID 사용	236
자산 모델 및 구성 요소 모델 생성	238
자산 모델 생성	239
컴포넌트 모델 생성	253
데이터 속성 정의	257
사용자 지정 복합 모델 생성 (구성 요소)	332
자산 만들기	336
자산 생성(콘솔)	336
자산 만들기 (AWS CLI)	337
새 자산 구성	339
자산 검색	339
필수 조건	339
고급 검색 커짐 AWS IoT SiteWise Console	339
산업 데이터 스트림을 자산 속성에 매핑	342
속성 별칭 설정(콘솔)	344
속성 별칭 설정 (AWS CLI)	345
속성 값 업데이트	347

자산 연결 및 연결 해제	350
자산 연결 및 연결 해제(콘솔)	350
자산 연결 및 연결 해제 (AWS CLI)	352
자산 및 모델 업데이트	353
자산 업데이트	354
자산 모델 및 구성 요소 모델 업데이트	355
커스텀 복합 모델 (컴포넌트) 업데이트	359
자산 및 모델 삭제	362
자산 삭제	362
자산 모델 삭제	365
자산 및 모델을 사용한 대량 작업	366
주요 개념 및 용어	367
지원되는 기능	368
대량 작업 사전 요구 사항	368
대량 가져오기 작업 실행	371
대량 내보내기 작업 실행	373
작업 진행 상황 추적 및 오류 처리	376
메타데이터 가져오기 예제	381
메타데이터 내보내기 예제	395
AWS IoT SiteWise 메타데이터 전송 작업 스키마	398
경보로 데이터 모니터링	417
경보 유형	417
경보 상태	418
경보 상태 속성	419
자산 모델에서 경보 정의	422
AWS IoT Events 알람 정의	425
외부 경보 정의	459
자산에 대한 경보 구성	461
임계값 구성(콘솔)	461
임계값 구성 (AWS CLI)	462
알림 설정 구성(콘솔)	464
알림 설정 구성(CLI)	464
경보에 대응	466
경보에 대응(콘솔)	467
경보에 응답(API)	470
외부 경보 상태 수집	470

외부 경보 상태 스트림 매핑	471
경보 상태 데이터 수집	472
웹 포털에서 데이터 모니터링	474
SiteWise 모니터 역할	475
SAML 페더레이션	476
SiteWise 모니터 개념	477
시작하기	479
포털 생성	479
포털 구성	480
관리자 초대	484
포털 사용자 추가	486
대시보드 생성(CLI)	490
포털의 경보 활성화	495
옛지에서 포털 활성화	499
포털 관리	499
포털 속성 변경	501
포털 관리자 추가 또는 제거	501
포털 관리자에게 이메일 초대 보내기	504
포털 사용자 추가 또는 제거	504
포털 삭제	507
IoT 대시보드 애플리케이션을 통한 데이터 모니터링	510
쿼리 데이터: AWS IoT SiteWise	511
현재 자산 가치 쿼리	511
자산 속성의 현재 값 쿼리 (콘솔)	512
자산 속성의 현재 값 쿼리 (AWS CLI)	512
과거 자산 자산 값 쿼리	513
자산 자산의 가치 기록 쿼리 (AWS CLI)	514
자산 속성 애그리게이트 쿼리	515
자산 자산에 대한 집계 (API)	515
자산 자산의 집계 (AWS CLI)	517
AWS IoT SiteWise 쿼리 언어	518
필수 조건	518
쿼리 언어 참조	519
다른 서비스와의 상호 작용	527
자산 속성의 MQTT 주제 이해	527
자산 속성 알림 사용하기	528

자산 속성 알림 활성화(콘솔)	528
자산 자산 알림 활성화 (AWS CLI)	529
자산 속성 알림 메시지 쿼리	531
Amazon S3로 데이터 내보내기	533
AWS CloudFormation 스택 생성	535
Amazon S3에서 데이터 보기	536
내보낸 데이터 분석	538
템플릿 리소스가 생성되었습니다.	546
Grafana와 통합	549
AWS IoT TwinMaker과 통합	550
통합 활성화	551
AWS IoT SiteWise 및 AWS IoT TwinMaker 통합	551
장비 이상 감지	552
예측 정의 추가 (콘솔)	554
예측 학습 (콘솔)	557
예측에 대한 추론 시작 또는 중지 (콘솔)	558
예측 정의 추가 (CLI)	558
예측 학습 및 시작 추론 (CLI)	562
예측 학습 (CLI)	563
예측에 대한 추론 시작 또는 중지 (CLI)	565
데이터 스토리지 관리	567
스토리지 설정 구성	568
데이터 보존에 미치는 영향	568
웜 티어의 스토리지 설정을 구성합니다 (콘솔).	569
웜 티어의 스토리지 설정을 구성합니다 (AWS CLI)	570
콜드 티어의 스토리지 설정을 구성합니다 (콘솔).	573
콜드 티어의 스토리지 설정을 구성합니다 (AWS CLI).	575
스토리지 설정 문제 해결	581
오류: 버킷이 존재하지 않습니다	581
오류: Amazon S3 경로에 대한 액세스가 거부되었습니다.	581
오류: 역할 ARN을 맡을 수 없습니다	582
오류: 리전 간 Amazon S3 버킷에 액세스하지 못했습니다.	582
콜드 티어에 저장된 데이터의 파일 경로 및 스키마	582
장비 데이터(측정값)	582
지표, 변환, 집계	586
자산 메타데이터	590

자산 계층 구조 메타데이터	593
스토리지 데이터 인덱스 파일	595
보안	597
데이터 보호	598
인터넷워크 트래픽 개인 정보	599
데이터 암호화	599
저장 중 암호화	599
전송 중 암호화	602
키 관리	603
자격 증명 및 액세스 관리	605
고객	605
ID를 통한 인증	606
IAM의 AWS IoT SiteWise 작동 방식	609
관리형 정책	627
서비스 연결 역할	630
경보에 대한 권한 설정	644
교차 서비스 혼동된 대리인 방지	650
문제 해결	651
규정 준수 확인	653
복원력	654
인프라 보안	655
구성 및 취약성 분석	655
VPC 엔드포인트	656
지원되는 API 작업	656
인터페이스 VPC 엔드포인트 생성	659
인터페이스 VPC 엔드포인트를 AWS IoT SiteWise 통한 액세스	660
VPC 엔드포인트 정책 생성	661
보안 모범 사례	662
OPC-UA 서버에서 인증 보안 인증 사용	662
OPC-UA 서버에 암호화된 통신 모드 사용	663
구성 요소를 최신 상태로 유지	663
SiteWise 에지 게이트웨이의 파일 시스템을 암호화하세요.	663
엣지 구성에 대한 보안 액세스	663
SiteWise 모니터 사용자에게 가능한 최소 권한을 부여하십시오.	663
민감한 정보 노출 금지	664
AWS IoT Greengrass 보안 베스트 프랙티스를 따르세요.	664

다음 사항도 참조하세요.	664
로깅 및 모니터링	665
서비스 로그 모니터링	665
로그인 관리 AWS IoT SiteWise	667
예: AWS IoT SiteWise 로그 파일 항목	668
SiteWise 에지 게이트웨이 로그 모니터링	669
아마존 CloudWatch 로그 사용	669
서비스 로그 사용	671
이벤트 로그 사용	672
Amazon CloudWatch 메트릭을 사용한 모니터링	675
AWS IoT Greengrass Version 2 게이트웨이 지표	676
AWS IoT Greengrass Version 1 게이트웨이 메트릭	683
AWS CloudTrail을 사용하여 API 호출 로깅	688
AWS IoT SiteWise 에 대한 정보 CloudTrail	688
AWS IoT SiteWise 의 데이터 이벤트 CloudTrail	689
AWS IoT SiteWise 의 관리 이벤트 CloudTrail	692
예: AWS IoT SiteWise 로그 파일 항목	692
리소스에 태그 지정	694
에서 태그 사용 AWS IoT SiteWise	694
를 사용하여 태그를 지정하십시오. AWS Management Console	694
API로 AWS IoT SiteWise 태그 지정	694
IAM 정책에 태그 사용	696
문제 해결	698
대량 가져오기 및 내보내기 문제 해결	698
포털 문제 해결	699
사용자와 관리자가 AWS IoT SiteWise 포털에 접근할 수 없음	699
게이트웨이의 문제 해결	700
SiteWise 에지 게이트웨이 로그 구성 및 액세스	701
SiteWise 에지 게이트웨이 문제 해결	701
문제 AWS IoT Greengrass 해결	704
AWS IoT SiteWise 규칙 조치 문제 해결	704
AWS IoT Core 로그 구성	704
재게시 오류 작업 구성	705
문제 해결	707
규칙 문제 해결	709
규칙 문제 해결	710

엔드포인트 및 할당량	715
엔드포인트	715
data.iotsitewise.region.amazonaws.com	715
api.iotsitewise.region.amazonaws.com	715
iotsitewise.region.amazonaws.com	716
model.iotsitewise.region.amazonaws.com	716
edge.iotsitewise.region.amazonaws.com	716
monitor.iotsitewise.region.amazonaws.com	716
할당량	717
예외 항목 탐지를 위한 할당량	728
사용 설명서 기록	729
AWS 용어집	745
.....	dccxlvii

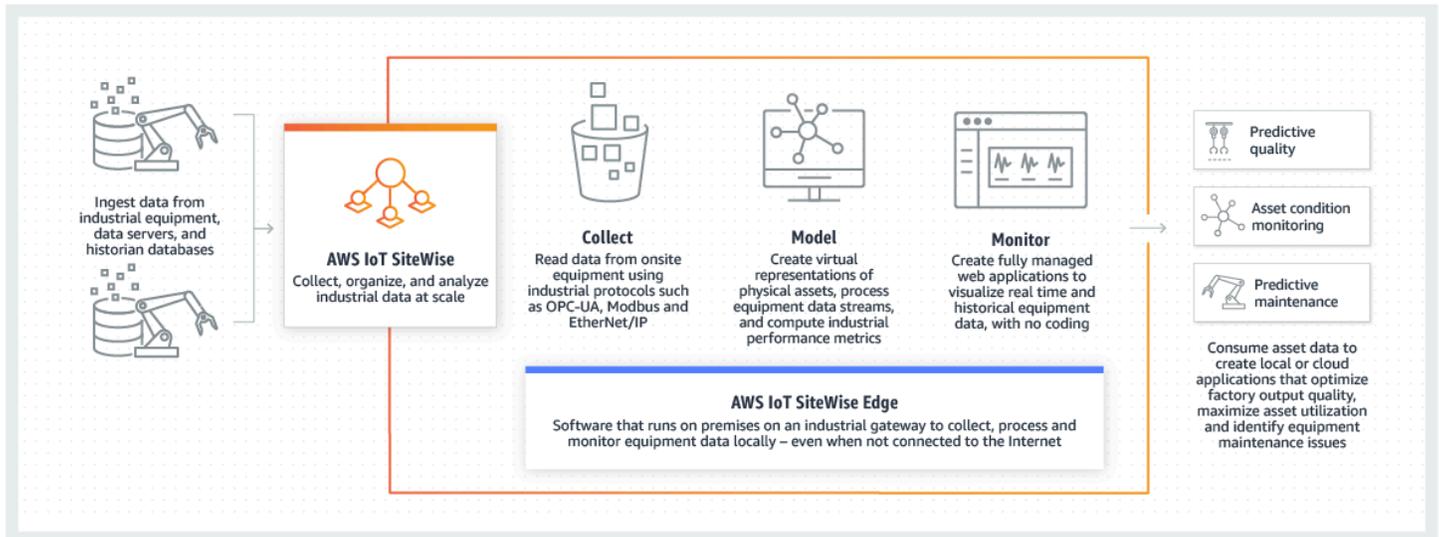
무엇입니까 AWS IoT SiteWise?

AWS IoT SiteWise 산업 장비의 데이터를 대규모로 쉽게 수집, 저장, 구성 및 모니터링하여 더 나은 데이터 기반 의사 결정을 내리는 데 도움이 되는 관리형 서비스입니다. AWS IoT SiteWise 이를 사용하여 시설 전반의 운영을 모니터링하고, 일반적인 산업 성능 지표를 신속하게 계산하고, 산업 장비 데이터를 분석하여 비용이 많이 드는 장비 문제를 예방하고 생산 격차를 줄이는 애플리케이션을 만들 수 있습니다.

AWS IoT SiteWise Monitor 운영 사용자가 실시간으로 산업 데이터를 보고 분석할 수 있는 웹 애플리케이션을 빠르게 만들 수 있습니다. 평균 고장 시간 및 전반적인 장비 효율성(OEE) 등의 지표를 구성하고 모니터링하여 산업 조직에 대한 통찰을 얻을 수 있습니다.

AWS IoT SiteWise Edge는 로컬 장치에서 데이터를 수집, 저장 및 처리할 수 있는 AWS IoT SiteWise 있는 구성 요소입니다. 이는 인터넷 액세스가 제한적이거나 데이터를 비공개로 유지해야 하는 경우에 유용합니다.

다음 다이어그램은 다음의 기본 아키텍처를 보여줍니다 AWS IoT SiteWise.



주제

- [AWS IoT SiteWise 작동 방식](#)
- [AWS IoT SiteWise 개념](#)
- [사용 사례 AWS IoT SiteWise](#)

AWS IoT SiteWise 작동 방식

AWS IoT SiteWise 산업용 장치, 프로세스 및 시설을 표현하는 데 사용할 수 있는 리소스 모델링 프레임워크를 제공합니다. 예서는 장비와 프로세스의 표현을 자산 모델이라고 합니다. AWS IoT SiteWise 자산 모델을 사용하면 사용할 원시 데이터와 이를 유용한 지표로 처리하는 방법을 정의합니다. [AWS IoT SiteWise 콘솔에서](#) 산업 운영을 위한 자산과 모델을 구축하고 시각화하세요. 에지 또는 AWS 클라우드에서 데이터를 수집하고 처리하도록 자산 모델을 구성할 수도 있습니다.

주제

- [산업 데이터 수집](#)
- [자산을 모델링하여 수집된 데이터를 상황에 맞게 조정합니다.](#)
- [쿼리, 알람 및 예측을 사용하여 분석합니다.](#)
- [운영을 시각화하세요.](#)
- [데이터 저장](#)
- [다른 서비스와 통합](#)

산업 데이터 수집

산업 데이터를 AWS IoT SiteWise 수집하여 사용을 시작하십시오. 데이터 수집은 여러 가지 방법 중 하나로 이루어집니다.

- 현장 서버에서 직접 수집: OPC-UA와 같은 프로토콜을 활용하여 현장 장치에서 직접 데이터를 읽습니다. 일반 산업용 게이트웨이 또는 가상 서버와 같은 다양한 플랫폼에 AWS IoT Greengrass V2호환되는 SiteWise Edge 게이트웨이 소프트웨어를 배포하십시오. 단일 게이트웨이에 최대 100대의 OPC-UA 서버를 연결할 수 있습니다. AWS IoT SiteWise 자세한 정보는 [SiteWise 에지 게이트웨이 요구 사항](#)을 참조하세요.

Modbus TCP 및 이더넷/IP (EIP) 와 같은 프로토콜은 다음과 같은 맥락에서 당사와의 파트너십을 통해 지원된다는 점에 유의하십시오. Domatica AWS IoT Greengrass V2

- 팩을 통한 에지 데이터 처리: 포괄적인 SiteWise 엣지 기능을 지원하는 팩을 추가하여 에지 게이트웨이를 개선하십시오. 예서 AWS IoT Greengrass V2사용할 수 있는 SiteWise Edge를 사용하면 데이터 처리가 현장에서 직접 실행되고 AWS IoT Greengrass 스트림을 사용하여 AWS 클라우드로 안전하게 전송됩니다. 자세한 정보는 [팩 사용](#)을 참조하세요.
- Amazon S3를 통한 적응형 수집 (대량 작업 포함): 많은 수의 자산 또는 자산 모델로 작업하는 경우 대량 작업을 사용하여 Amazon S3 버킷에서 리소스를 대량으로 가져오고 내보냅니다. 자세한 정보는 [자산 및 모델을 사용한 대량 작업](#)을 참조하세요.

- AWS IoT 핵심 규칙이 포함된 MQTT 메시지: MQTT 메시지를 보내는 AWS IoT Core에 연결된 디바이스의 경우 AWS IoT Core 규칙 엔진을 사용하여 해당 메시지를 전달하십시오. [MQTT](#) 메시지를 보내는 AWS IoT Core에 연결된 디바이스가 있는 AWS IoT SiteWise 경우 Core 규칙 엔진을 사용하여 해당 메시지를 라우팅하십시오. AWS IoT SiteWise 자세한 정보는 [규칙을 사용한 AWS IoT Core 데이터 수집](#)을 참조하십시오.
- 이벤트 트리거 데이터 수집: AWS IoT Events 작업을 사용하여 이벤트 발생 시 데이터를 AWS IoT Events 전송하도록 IoT SiteWise 작업을 구성합니다. AWS IoT SiteWise 자세한 정보는 [에서 데이터 수집 AWS IoT Events](#)을 참조하십시오.
- AWS IoT SiteWise API: 엣지 또는 클라우드의 애플리케이션이 데이터를 직접 전송할 수 있습니다. AWS IoT SiteWise 자세한 정보는 [API를 사용한 데이터 수집 AWS IoT SiteWise](#)을 참조하십시오.

자산을 모델링하여 수집된 데이터를 상황에 맞게 조정합니다.

데이터를 수집한 후에는 물리적 운영 모델을 구축하여 데이터를 사용하여 자산, 프로세스 및 시설을 가상으로 표현할 수 있습니다. 장치 또는 프로세스를 나타내는 자산은 데이터 스트림을 클라우드로 전송합니다. AWS 자산은 논리적 장치 그룹화를 의미할 수도 있습니다. 계층은 자산을 연결하여 복잡한 작업을 반영함으로써 형성됩니다. 이러한 계층 구조를 통해 자산은 관련 하위 자산의 데이터에 액세스할 수 있습니다. 자산은 자산 모델에서 생성됩니다. 자산 모델은 자산 형식을 표준화하는 선언적 구조입니다. 모델의 구성 및 유지 관리를 위해 자산 구성 요소를 재사용하십시오. 자세한 정보는 [산업 자산 모델링](#)을 참조하십시오.

를 사용하면 수신되는 데이터를 상황에 맞는 지표로 변환하고 변환하도록 자산을 구성할 수 있습니다. AWS IoT SiteWise

- 변환은 장비 데이터를 수신할 때 작동합니다.
- 지표는 사용자가 정의한 간격으로 계산됩니다.

지표와 변환은 개별 자산 또는 여러 자산 모두에 적용할 수 있습니다. AWS IoT SiteWise 장비 데이터, 지표 및 변환과 관련된 다양한 시간대에 걸쳐 일반적으로 사용되는 통계 집계 (예: 평균, 합계, 개수) 를 자동으로 계산합니다.

를 사용하여 자산을 동기화할 수 있습니다. AWS IoT TwinMaker 자세한 정보는 [AWS IoT SiteWise 및 AWS IoT TwinMaker 통합](#)을 참조하십시오.

쿼리, 알람 및 예측을 사용하여 분석합니다.

쿼리를 실행하고 경보를 AWS IoT SiteWise 설정하여 수집된 날짜를 분석합니다. 또한 Amazon Lookout을 사용하여 지표 내에서 이상을 자동으로 감지하고 근본 원인을 식별할 수 있습니다.

- 장비나 프로세스가 최적의 성능을 벗어날 경우 팀에 알리도록 특정 경보를 설정하여 신속하게 문제를 식별하고 해결할 수 있습니다. 자세한 정보는 [경보로 데이터 모니터링](#)을 참조하세요.
- AWS IoT SiteWise API 작업을 사용하여 특정 기간 동안 자산 자산의 현재 값, 과거 값 및 집계를 쿼리할 수 있습니다. 자세한 정보는 [에서 데이터 쿼리 AWS IoT SiteWise](#)을 참조하세요.
- Amazon Lookout for Equipment에서 이상 탐지를 사용하여 장비 또는 작동 조건의 변화를 식별하고 시각화할 수 있습니다. 이상 탐지를 통해 운영에 대한 예방적 유지 관리 조치를 결정할 수 있습니다. 이 통합을 통해 고객은 Amazon Lookout for AWS IoT SiteWise Equipment 간에 데이터를 동기화할 수 있습니다. 자세한 정보는 [Amazon Lookout for Equipment를 사용하여 장비 이상을 감지합니다.](#)을 참조하세요.

운영을 시각화하세요.

SiteWise Monitor를 설정하여 운영 직원을 위한 웹 애플리케이션을 만드세요. 웹 애플리케이션은 직원들이 운영을 시각화하는 데 도움이 됩니다. IAM ID 센터 또는 IAM을 사용하여 직원의 다양한 액세스 수준을 처리할 수 있습니다. 각 직원이 전체 산업 운영의 특정 하위 집합을 볼 수 있도록 고유한 로그인 및 권한을 구성하세요. AWS IoT SiteWise 해당 직원들이 모니터 사용 SiteWise 방법을 배울 수 있도록 [애플리케이션 가이드](#)를 제공합니다.

작업을 시각화하는 방법에 대한 자세한 내용은 [을 참조하십시오](#) [를 사용하여 데이터 모니터링 AWS IoT SiteWise Monitor](#).

데이터 저장

시계열 스토리지를 산업용 데이터 레이크와 통합할 수 있습니다. AWS IoT SiteWise 산업 데이터를 위한 세 가지 스토리지 계층이 있습니다.

- 실시간 애플리케이션에 최적화된 핫 스토리지 티어입니다.
- 분석 워크로드에 최적화된 웜 스토리지 티어입니다.
- 지연 시간이 긴 운영 데이터 애플리케이션을 위해 Amazon S3를 사용하는 고객 관리형 콜드 스토리지 티어입니다.

AWS IoT SiteWise 최신 데이터를 핫 스토리지 tier에 보관하여 스토리지 비용을 관리할 수 있습니다. 그런 다음 데이터 보존 정책을 정의하여 기록 데이터를 워밍 tier 또는 콜드 tier 스토리지로 이동합니다. 자세한 정보는 [데이터 스토리지 관리](#)를 참조하세요.

자산 메타데이터를 가져오고 내보낼 수도 있습니다. 자세한 내용은 [자산 메타데이터](#) 단원을 참조하세요.

다른 서비스와 통합

AWS IoT SiteWise 여러 AWS 서비스와 통합하여 AWS 클라우드에서 완벽한 AWS IoT 솔루션을 개발합니다. 자세한 내용은 [다른 AWS 서비스와의 상호작용](#)(를) 참조하세요.

AWS IoT SiteWise 개념

의 핵심 개념은 다음과 같습니다 AWS IoT SiteWise.

Aggregate

집계는 모든 시계열 데이터를 AWS IoT SiteWise 자동으로 계산하는 기본 지표 또는 측정값입니다. 자세한 정보는 [자산 속성 집계 쿼리](#)를 참조하세요.

자산

산업 AWS IoT SiteWise 장비에서 데이터를 입력하거나 수집하면 장치, 장비 및 프로세스가 각각 자산으로 표시됩니다. 각 자산에는 관련 데이터가 있습니다. 예를 들어, 장비에는 일련 번호, 위치, 제조업체 및 모델, 설치 날짜가 있을 수 있습니다. 가용성, 성능, 품질, 온도, 압력 등에 대한 시계열 값도 있을 수 있습니다. 자산을 계층 구조로 그룹화하여 자산이 하위 자산에 저장된 데이터에 액세스할 수 있도록 합니다. 자세한 정보는 [산업 자산 모델링](#)을 참조하세요.

자산 계층 구조

자산 계층 구조를 설정하여 산업 운영을 논리적으로 표현하세요. 이렇게 하려면 자산 모델에서 계층 구조를 정의하고 해당 모델에서 생성된 자산을 지정된 계층 구조와 연관시키십시오. 상위 자산의 지표는 하위 자산의 속성 데이터를 결합할 수 있으므로 전체 운영 또는 특정 부분에 대한 통찰력을 제공하는 지표를 계산할 수 있습니다. 자세한 정보는 [자산 모델 계층 구조 정의](#)을 참조하세요.

자산 모델

모든 자산은 자산 모델을 사용하여 만들어집니다. 자산 모델은 자산 형식을 정의하고 표준화하는 구조입니다. 동일한 유형의 여러 자산에 대해 일관된 정보를 보장하므로 장치 그룹을 나타내는 자산의 데이터를 처리할 수 있습니다. 각 자산 모델에서 [특성](#), 시계열 입력([측정](#)), 시계열 변환([변환](#)),

시계열 집계([지표](#)) 및 [자산 계층 구조](#)를 정의할 수 있습니다. 자세한 정보는 [산업 자산 모델링](#)을 참조하세요.

에지용으로 에셋 모델을 구성하여 자산 모델의 속성이 처리되는 위치를 결정하십시오. 이 기능을 활용하여 로컬 장치의 자산 데이터를 처리하고 모니터링할 수 있습니다.

자산 속성

자산 속성은 산업 데이터를 보관하는 각 자산 내의 구조입니다. 각 자산에는 데이터 유형이 있으며 단위도 가질 수 있습니다. 속성은 [특성](#), [측정치](#), [변환](#) 또는 [지표](#)가 될 수 있습니다. 자세한 정보는 [데이터 속성 정의](#)을 참조하세요.

에지에서 계산하도록 자산 속성을 구성하십시오. 엣지에서 데이터 프로세싱에 대한 자세한 내용은 [the section called “엣지 데이터 처리 활성화”](#) 단원을 참조하십시오.

특성

속성은 기기 제조업체 또는 기기 위치와 같이 일반적으로 일정하게 유지되는 자산의 속성입니다. 속성에는 사전 설정된 값이 있을 수 있습니다. 자산 모델에서 생성되는 모든 자산에는 해당 모델에 정의된 속성의 기본값이 포함됩니다. 자세한 정보는 [정적 데이터\(속성\) 정의](#)을 참조하세요.

대시보드

각 프로젝트에는 대시보드 집합이 포함되어 있습니다. 대시보드는 자산 집합의 값에 대한 시각화 집합을 제공합니다. 프로젝트 소유자는 대시보드와 대시보드에 포함된 시각화를 생성합니다. 프로젝트 소유자가 대시보드 집합을 공유할 준비가 되면 소유자는 프로젝트에 뷰어를 초대하여 프로젝트의 모든 대시보드에 대한 액세스 권한을 부여할 수 있습니다. 다른 대시보드에 대해 다른 뷰어 집합을 사용하려면 프로젝트 간에 대시보드를 나누어야 합니다. 뷰어는 대시보드를 볼 때 특정 데이터를 볼 수 있도록 시간 범위를 사용자 지정할 수 있습니다.

데이터 스트림

자산 모델 및 자산을 생성하기 AWS IoT SiteWise 전에도 산업 데이터를 입력하거나 수집할 수 있습니다. AWS IoT SiteWise 데이터 스트림을 자동으로 생성하여 장비에서 원시 데이터 스트림을 수집합니다.

데이터 스트림 별칭

데이터 스트림 별칭을 사용하면 데이터 스트림을 쉽게 식별할 수 있습니다. 예를 들어 별칭은 풍력 발전소 #3 터빈의 #7 온도 값을 `server1-windfarm/3/turbine/7/temperature` 나타냅니다. `server1`이 용어는 OPC-UA 서버를 식별하는 데 도움이 되는 데이터 원본 `server1-` 이름이며 이 OPC-UA 서버에서 보고되는 모든 데이터 스트림에 붙는 접두사입니다.

데이터 스트림 연결

자산 모델 및 자산을 생성한 후에는 데이터 스트림을 자산에 정의된 자산 속성과 연결하여 데이터를 구조화하십시오. AWS IoT SiteWise 그런 다음 자산 모델 및 자산을 사용하여 데이터 스트림에서 들어오는 데이터를 처리할 수 있습니다. 자산 속성에서 데이터 스트림을 연결 해제할 수도 있습니다. 자세한 내용은 [데이터 스트림 관리](#) 단원을 참조하십시오.

공식

각 [변환](#) 및 [메트릭](#) 속성에는 속성이 데이터를 변환하거나 집계하는 방법을 설명하는 공식이 함께 제공됩니다. 이러한 공식에는 에서 제공하는 속성 입력, 연산자 및 함수가 포함됩니다. AWS IoT SiteWise 자세한 정보는 [공식 표현식 사용](#)을 참조하세요.

측정

측정값은 장치 또는 장비의 원시 센서 시계열 데이터 스트림을 나타내는 자산의 속성입니다. 자세한 정보는 [장비의 데이터 스트림 정의\(측정값\)](#)을 참조하세요.

지표

지표는 집계된 시계열 데이터를 나타내는 자산의 속성입니다. 각 지표에는 데이터 포인트를 집계하는 방법과 해당 집계를 계산하는 데 필요한 시간 간격을 설명하는 [수식\(수식\)](#) 이 함께 제공됩니다. 지표는 지정된 각 시간 간격에 대해 단일 데이터 포인트를 생성합니다. 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)을 참조하세요.

팩

SiteWise 에지 게이트웨이는 팩을 사용하여 데이터를 수집, 처리 및 라우팅하는 방법을 결정합니다. 현재 데이터 수집 팩과 데이터 처리 팩을 AWS IoT SiteWise 지원합니다. SiteWise Edge 게이트웨이에 사용할 수 있는 팩에 대한 자세한 내용은 [the section called “팩 사용”](#)을 참조하십시오.

데이터 수집 팩

데이터 수집 팩을 사용하면 SiteWise 에지 게이트웨이가 산업 데이터를 수집하여 선택한 AWS 목적지로 라우팅할 수 있습니다. 이 팩은 SiteWise 에지 게이트웨이에 자동으로 추가되며 제거할 수 없습니다.

데이터 처리 팩

데이터 처리 팩을 사용하면 엣지에서 데이터를 처리하고 로컬 애플리케이션에서 사용할 수 있도록 30일간 보존할 수 있습니다.

Portal

AWS IoT SiteWise Monitor 포털은 AWS IoT SiteWise 데이터를 시각화하고 공유하는 데 사용할 수 있는 웹 애플리케이션입니다. 포털에는 관리자가 한 명 이상 있으며 프로젝트가 0개 이상 포함되어 있습니다.

포털 관리자

각 SiteWise 모니터 포털에는 한 명 이상의 포털 관리자가 있습니다. 포털 관리자는 포털을 사용하여 자산 및 대시보드 컬렉션이 포함된 프로젝트를 생성합니다. 그런 다음 포털 관리자가 각 프로젝트에 자산과 소유자를 할당합니다. 프로젝트에 대한 액세스를 제어하면 포털 관리자가 프로젝트 소유자와 뷰어가 볼 수 있는 자산을 제한할 수 있습니다.

프로젝트

각 SiteWise 모니터 포털에는 프로젝트 세트가 포함되어 있습니다. 각 프로젝트에는 연결된 AWS IoT SiteWise 자산의 하위 집합이 있습니다. 프로젝트 소유자는 하나 이상의 대시보드를 생성하여 해당 자산과 연결된 데이터를 일관된 방식으로 볼 수 있습니다. 프로젝트 소유자는 프로젝트에 뷰어를 초대하여 프로젝트의 자산 및 대시보드를 보도록 허용할 수 있습니다. 프로젝트는 SiteWise Monitor 내 공유의 기본 단위입니다. 프로젝트 소유자는 AWS 관리자로부터 포털 접근 권한을 부여 받은 사용자를 초대할 수 있습니다. 해당 포털에 있는 프로젝트를 해당 사용자와 공유하려면 먼저 사용자에게 포털에 대한 액세스 권한이 있어야 합니다.

프로젝트 소유자

각 SiteWise Monitor 프로젝트에는 소유자가 있습니다. 프로젝트 소유자는 운영 데이터를 일관된 방식으로 나타내기 위해 대시보드 형태로 시각화를 생성합니다. 대시보드를 공유할 준비가 되면 프로젝트 소유자가 프로젝트에 뷰어를 초대할 수 있습니다. 프로젝트 소유자는 프로젝트에 다른 소유자를 할당할 수도 있습니다. 프로젝트 소유자는 경보에 대한 임계값 및 알림 설정을 구성할 수 있습니다.

프로젝트 뷰어

각 SiteWise 모니터 프로젝트에는 뷰어가 있습니다. 프로젝트 뷰어는 포털에 연결하여 프로젝트 소유자가 생성한 대시보드를 볼 수 있습니다. 각 대시보드에서 프로젝트 뷰어는 시간 범위를 조정하여 운영 데이터를 더 잘 이해할 수 있습니다. 프로젝트 뷰어는 액세스 권한이 있는 프로젝트의 대시보드만 볼 수 있습니다. 프로젝트 뷰어는 경보를 확인하고 일시 중지할 수 있습니다.

속성 별칭

OPC-UA 서버 데이터 스트림 경로 (예: /company/windfarm/3/터빈/7/temperature) 와 같은 자산 속성에 별칭을 생성할 수 있어 자산 데이터를 수집 또는 검색하는 동안 자산 자산을 쉽게 식별할 수 있습니다.

습니다. [SiteWise Edge 게이트웨이를 사용하여 서버에서 데이터를 수집하는 경우 속성 별칭은 원시 데이터 스트림의 경로와 일치해야 합니다.](#) 자세한 정보는 [산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.

속성 알림

자산 속성에 대한 속성 알림을 활성화하면 해당 속성이 새 값을 받을 AWS IoT Core 때마다 MQTT 메시지를 AWS IoT SiteWise 게시합니다. 메시지 페이로드에는 해당 속성 값 업데이트에 대한 세부 정보가 포함됩니다. 자산 가치 알림을 사용하여 산업 데이터를 다른 AWS IoT SiteWise AWS 서비스와 연결하는 솔루션을 만드세요. 자세한 정보는 [다른 AWS 서비스와의 상호작용](#)을 참조하세요.

SiteWise 에지 게이트웨이

SiteWise 에지 게이트웨이는 고객 구내에 위치하여 데이터를 수집, 처리 및 전달합니다. SiteWise 엣지 게이트웨이는 [OPC-UA](#) 프로토콜을 통해 산업 데이터 소스에 연결하여 데이터를 수집 및 처리한 후 AWS 클라우드로 전송합니다. SiteWise 에지 게이트웨이는 [파트너](#) 데이터 소스에도 연결할 수 있습니다. SiteWise 에지 게이트웨이는 데이터 수집, 에지 처리 등에 팩을 사용합니다. 사용 가능한 팩에 대한 자세한 내용은 [the section called “팩 사용”](#) 단원을 참조하십시오.

실행 AWS IoT Greengrass가 가능한 모든 장치 또는 플랫폼에서 SiteWise 에지 게이트웨이를 유연하게 만들 수 있습니다. 자세한 정보는 [SiteWise 에지 게이트웨이 사용](#)을 참조하세요.

변환

변환은 변환된 시계열 데이터를 나타내는 자산의 속성입니다. 모든 변환에는 데이터 포인트를 한 양식에서 다른 양식으로 변환하는 방법을 지정하는 [수식 \(수식\)](#) 이 함께 제공됩니다. 변환된 데이터 포인트는 입력 데이터 포인트와 one-to-one 관계를 유지합니다. 자세한 정보는 [데이터 변환\(변환\)](#)을 참조하세요.

시각화

각 대시보드에서 프로젝트 소유자는 프로젝트와 연결된 속성과 경보를 표시하는 방법을 결정합니다. 가용성은 꺾은선형 차트로 가장 잘 표현되고 다른 값은 막대 차트나 핵심 성능 지표(KPI)로 표시될 수 있습니다. 경보는 상태 그리드 및 상태 타임라인으로 표시하는 것이 가장 좋습니다. 프로젝트 소유자는 각 시각화를 사용자 지정하여 해당 자산의 데이터를 가장 잘 이해할 수 있도록 합니다.

사용 사례 AWS IoT SiteWise

AWS IoT SiteWise 다양한 산업 분야에서 다양한 산업 데이터 수집 및 분석 애플리케이션에 사용됩니다.

모든 소스에서 일관되게 데이터를 수집하여 문제를 신속하게 해결할 수 있습니다. AWS IoT SiteWise 현장에서 직접 데이터를 수집하거나 여러 시설의 여러 소스에서 데이터를 수집할 수 있는 원격 모니터링을 제공합니다. AWS IoT SiteWise 산업용 IoT 데이터 솔루션에 필요한 유연성을 제공합니다.

제조

AWS IoT SiteWise 장비에서 데이터를 수집하고 활용하는 프로세스를 단순화하여 비효율성을 정확히 찾아내고 최소화하여 산업 운영을 개선할 수 있습니다. AWS IoT SiteWise 제조 라인 및 장비에서 데이터를 수집하는 데 도움이 됩니다. 이를 사용하여 데이터를 AWS 클라우드로 전송하고 특정 장비 및 프로세스에 대한 성능 지표를 작성할 수 있습니다. AWS IoT SiteWise 생성된 지표를 사용하여 운영의 전반적인 효율성을 파악하고 혁신 및 개선 기회를 식별할 수 있습니다. 또한 제조 프로세스를 보고 장비 및 프로세스 결함, 생산 격차 또는 제품 결함을 식별할 수 있습니다.

식음료

식음료 산업 시설에서는 곡물 제분, 도축/육류 포장, 간편 조리 식품의 조합/조리/냉동을 비롯하여 다양한 식품 처리를 진행합니다. 식품 가공 공장은 여러 위치에 분산되어 있는 경우가 많으며, 공장 및 장비 운영자가 중앙 위치에 배치되어 공정과 장비를 모니터링합니다. 예를 들어 냉동 설비는 재료 취급 및 유통기한을 평가합니다. 시설 전반의 폐기물 생성을 모니터링하여 운영 효율성을 보장합니다. 이를 통해 AWS IoT SiteWise 여러 위치의 센서 데이터 스트림을 생산 라인 및 시설별로 그룹화하여 공정 엔지니어가 시설 전반을 더 잘 이해하고 개선할 수 있습니다.

에너지/공공 설비

AWS IoT SiteWise를 사용하면 장비 문제를 더 쉽고 효율적으로 해결할 수 있습니다. 자산 성능을 원격 및 실시간으로 모니터링할 수 있습니다. 어디서든 과거 장비 데이터에 액세스하여 잠재적 문제를 찾아내고, 정확한 리소스를 파견하고, 문제를 더 빠르게 예방하고 해결할 수 있습니다.

시작하기 AWS IoT SiteWise

를 사용하면 데이터를 수집, 구성, 분석 및 시각화할 수 있습니다. AWS IoT SiteWise

AWS IoT SiteWise 실제 데이터 소스를 구성하지 않고도 서비스를 탐색하는 데 사용할 수 있는 데모를 제공합니다. 자세한 정보는 [AWS IoT SiteWise 데모 사용](#)을 참조하세요.

다음 자습서를 완료하여 특정 기능을 탐색할 수 있습니다. AWS IoT SiteWise

- [사물의 데이터 수집 AWS IoT](#)
- [모니터에서 SiteWise 풍력 발전 단지 데이터 시각화 및 공유](#)
- [Amazon DynamoDB에 속성 값 업데이트 게시](#)

자세히 AWS IoT SiteWise알아보려면 다음 주제를 참조하십시오.

- [예 데이터 수집 AWS IoT SiteWise](#)
- [산업 자산 모델링](#)
- [엣지 데이터 처리 활성화](#)
- [를 사용하여 데이터 모니터링 AWS IoT SiteWise Monitor](#)
- [에서 데이터 쿼리 AWS IoT SiteWise](#)
- [다른 AWS 서비스와의 상호작용](#)

주제

- [요구 사항](#)
- [설정 AWS 계정](#)
- [AWS IoT SiteWise 데모 사용](#)

요구 사항

AWS 계정 시작하려면 가 있어야 합니다 AWS IoT SiteWise. 계정이 없는 경우 [설정 AWS 계정](#) 단원을 참조하십시오.

가능한 지역을 AWS IoT SiteWise 사용하세요. 자세한 내용은 [AWS IoT SiteWise 엔드포인트 및 할당량](#)을 참조하세요. 의 지역 선택기를 사용하여 이러한 지역 중 하나로 AWS Management Console 전환할 수 있습니다.

설정 AWS 계정

주제

- [가입하기 AWS 계정](#)
- [관리자 액세스 권한이 있는 사용자 생성](#)

가입하기 AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

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

등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 [루트 사용자 액세스 권한이 필요한 작업](#)을 수행하는 것입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 <https://aws.amazon.com/>으로 가서 내 계정(My Account)을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

등록한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오 AWS 계정 루트 사용자.

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 [AWS Management Console](#)소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면AWS 로그인 사용 설명서의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

지침은 IAM [사용 설명서의 AWS 계정 루트 사용자 \(콘솔\)에 대한 가상 MFA 디바이스 활성화를 참조](#)하십시오.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center 설정](#)을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

를 ID 소스로 사용하는 방법에 대한 자습서는 사용 [설명서의 기본값으로 IAM Identity Center 디렉터리 사용자 액세스 구성](#)을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM IDentity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM Identity Center 사용자를 사용하여 [로그인하는 데 도움이 필요하다면 사용 설명서의 AWS 액세스 포털에 로그인](#)을 참조하십시오. AWS 로그인

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Create a permission set](#)을 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Add groups](#)를 참조하세요.

AWS IoT SiteWise 데모 사용

AWS IoT SiteWise 데모를 AWS IoT SiteWise 사용하여 쉽게 탐색할 수 있습니다. AWS IoT SiteWise 데모를 AWS CloudFormation 템플릿으로 제공하여 자산 모델, 자산 및 SiteWise 모니터 포털을 만들고 최대 1주일 동안 샘플 데이터를 생성할 수 있도록 배포할 수 있습니다.

⚠ Important

데모를 만들고 나면 이 데모에서 생성하고 사용하는 리소스에 대한 요금이 부과되기 시작합니다.

주제

- [데모 만들기 AWS IoT SiteWise](#)
- [데모 삭제 AWS IoT SiteWise](#)

데모 만들기 AWS IoT SiteWise

AWS IoT SiteWise 콘솔에서 AWS IoT SiteWise 데모를 만들 수 있습니다.

i Note

데모에서는 Lambda 함수, CloudWatch 하나의 이벤트 규칙 및 AWS Identity and Access Management 데모에 필요한 (IAM) 역할을 생성합니다. 에서 이러한 리소스를 볼 수 있습니다. AWS 계정데모를 완료할 때까지 이러한 리소스를 보관하는 것이 좋습니다. 이러한 리소스를 삭제하면 데모가 제대로 작동하지 않을 수 있습니다.

AWS IoT SiteWise 콘솔에서 데모를 만들려면

1. [AWS IoT SiteWise 콘솔로](#) 이동하여 페이지 오른쪽 상단에서 SiteWise 데모를 찾을 수 있습니다.
2. (선택 사항) SiteWise 데모에서 데모 자산 보관 날짜 필드를 변경하여 삭제하기 전에 데모를 보관할 기간 (일) 을 지정하십시오.
3. (선택 사항) 샘플 데이터를 모니터링할 SiteWise 모니터 포털을 만들려면 다음과 같이 하십시오.

i Note

이 데모에서 생성하고 사용하는 SiteWise 모니터 리소스에 대한 요금이 부과됩니다. 자세한 내용은 AWS IoT SiteWise 가격에서 SiteWise [모니터를](#) 참조하십시오.

- a. 모니터 리소스를 선택합니다.

- b. 권한을 선택합니다.
- c. 페더레이션된 IAM 사용자에게 포털 액세스 권한을 부여하는 기존 IAM 역할을 선택합니다.

⚠ Important

IAM 역할에 다음 권한이 있어야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:Describe*",
        "iotsitewise:List*",
        "iotsitewise:Get*",
        "cloudformation:DescribeStacks",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedRolePolicies",
        "sso:DescribeRegisteredRegions",
        "organizations:DescribeOrganization"
      ],
      "Resource": "*"
    }
  ]
}
```

SiteWise Monitor를 사용하는 방법에 대한 자세한 내용은 [무엇입니까 AWS IoT SiteWise Monitor?](#) 를 참조하십시오. AWS IoT SiteWise Monitor 애플리케이션 가이드에서

4. Create demo(데모 생성)를 선택합니다.

데모를 만드는 데 약 3분이 걸립니다. 데모를 만들지 못하는 경우, 계정에 사용 권한이 부족하기 때문일 수 있습니다. 관리 권한이 있는 계정으로 전환하거나 다음 단계에 따라 데모를 삭제하고 다시 시도합니다.

- a. Delete demo(데모 삭제)를 선택합니다.

데모를 삭제하는 데 약 15분이 걸립니다.

- b. 데모가 삭제되지 않으면 [AWS CloudFormation 콘솔](#)을 열고 SiteWiseDemoAssetsIoT라는 스택을 선택한 다음 오른쪽 상단에서 삭제를 선택합니다.
 - c. 데모를 다시 삭제하지 못하면 AWS CloudFormation 콘솔의 단계에 따라 삭제에 실패한 리소스를 건너뛰고 다시 시도하세요.
5. 데모가 성공적으로 생성되면 [AWS IoT SiteWise 콘솔](#)에서 데모 자산 및 데이터를 탐색할 수 있습니다.

데모 삭제 AWS IoT SiteWise

데모는 일주일 후, 또는 AWS CloudFormation 콘솔에서 AWS IoT SiteWise 데모 스택을 생성한 경우 선택한 일수가 지나면 자동으로 삭제됩니다. 데모 리소스의 사용을 마쳤으면 그 전에 데모를 삭제할 수 있습니다. 데모를 만드는 데 실패한 경우에도 데모를 삭제할 수 있습니다. 데모를 수동으로 삭제하려면 다음 단계를 수행하십시오.

데모를 삭제하려면 AWS IoT SiteWise

1. [AWS CloudFormation 콘솔](#)로 이동합니다.
2. 스택 목록에서 IoTSiteWiseDemoAssets을 선택합니다.
3. 삭제를 선택합니다.

스택을 삭제하면 데모용으로 생성된 모든 리소스가 삭제됩니다.

4. 확인 대화 상자에서 스택 삭제를 선택합니다.

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

AWS IoT SiteWise 튜토리얼

AWS IoT SiteWise 튜토리얼 페이지에 오신 것을 환영합니다. 점점 늘어나는 튜토리얼 컬렉션을 통해 복잡한 내용을 탐색하는 데 필요한 지식과 기술을 습득할 수 있습니다. AWS IoT SiteWise이 자습서는 사용자의 요구에 맞는 다양한 기본 주제를 제공합니다. 자습서를 자세히 살펴보면서 다양한 측면에 대한 귀중한 통찰력을 발견하십시오. AWS IoT SiteWise

각 자습서에서는 특정 장비 예제를 사용합니다. 이 자습서는 테스트 환경을 대상으로 하며 가상의 회사 이름, 모델, 자산, 속성 등을 사용합니다. 그 목적은 일반적인 지침을 제공하는 것입니다. 자습서는 조직의 고유한 요구 사항에 맞게 주의 깊게 검토하고 조정하지 않으면 프로덕션 환경에서 직접 사용할 수 없습니다.

주제

- [OEE 계산 기준 AWS IoT SiteWise](#)
- [사물의 데이터 수집 AWS IoT](#)
- [모니터에서 SiteWise 풍력 발전 단지 데이터 시각화 및 공유](#)
- [Amazon DynamoDB에 속성 값 업데이트 게시](#)

OEE 계산 기준 AWS IoT SiteWise

이 자습서에서는 제조 프로세스에 대한 전반적인 장비 효율성(OEE)을 계산하는 방법에 대한 예가 수록되어 있습니다. 결과적으로, OEE 계산 또는 공식이 여기에 표시된 것과 다를 수도 있습니다. 일반적으로 OEE는 Availability * Quality * Performance로 정의됩니다. OEE 계산에 대한 자세한 내용은 Wikipedia의 [전체 장비 효율성](#)을 참조하십시오.

필수 조건

이 자습서를 완료하려면 다음과 같은 세 가지 데이터 스트림이 있는 디바이스에 대한 데이터 수집을 구성해야 합니다.

- Equipment_State – 유휴, 결함, 계획된 중지 또는 정상 작동과 같이 시스템 상태를 나타내는 숫자 코드입니다.
- Good_Count – 각 데이터 포인트가 마지막 데이터 포인트 이후 성공한 작업 수를 포함하는 데이터 스트림입니다.
- Bad_Count – 각 데이터 포인트가 마지막 데이터 포인트 이후 실패한 작업 수를 포함하는 데이터 스트림입니다.

데이터 수집을 구성하려면 [예 데이터 수집 AWS IoT SiteWise](#) 단원을 참조하십시오. 사용 가능한 산업 조직이 없는 경우 AWS IoT SiteWise API를 통해 샘플 데이터를 생성하고 업로드하는 스크립트를 작성할 수 있습니다.

OEE 계산 방법

이 자습서에서는 Equipment_State, Good_Count 및 Bad_Count 등, 세 가지 데이터 입력 스트림에서 OEE를 계산하는 자산 모델을 생성합니다. 이 예에서는 설탕, 감자 칩 또는 페인트 포장에 사용되는 일반 포장기를 살펴보겠습니다. [AWS IoT SiteWise 콘솔에서](#) 다음과 같은 측정, 변환 및 지표를 사용하여 AWS IoT SiteWise 자산 모델을 생성합니다. 그런 다음 포장 기계를 나타내는 자산을 만들고 OEE AWS IoT SiteWise 계산 방법을 관찰할 수 있습니다.

포장기의 원시 데이터 스트림을 나타내는 다음 [측정](#)을 정의합니다.

측정

- Equipment_State – 포장기의 현재 상태를 숫자 코드로 나타내는 데이터 스트림(또는 측정)입니다.
 - 1024 – 머신이 유휴 상태입니다.
 - 1020 – 오류나 지연 같은 결함입니다.
 - 1000 – 계획된 중지입니다.
 - 1111 – 정상적인 작업입니다.
- Good_Count – 각 데이터 포인트가 마지막 데이터 포인트 이후 성공한 작업 수를 포함하는 데이터 스트림입니다.
- Bad_Count – 각 데이터 포인트가 마지막 데이터 포인트 이후 실패한 작업 수를 포함하는 데이터 스트림입니다.

Equipment_State 측정 데이터 스트림 및 포함된 코드를 사용하여 다음 [변형](#)(또는 파생 측정)을 정의합니다. 변환은 원시 측정값과 one-to-one 관계가 있습니다.

변형

- Idle = eq(Equipment_State, 1024) – 머신의 유휴 상태를 포함하는 변형된 데이터 스트림입니다.
- Fault = eq(Equipment_State, 1020) – 머신의 결함 상태를 포함하는 변형된 데이터 스트림입니다.

- $Stop = eq(Equipment_State, 1000)$ – 머신의 계획된 중지 상태를 포함하는 변형된 데이터 스트림입니다.
- $Running = eq(Equipment_State, 1111)$ – 머신의 정상 작동 상태를 포함하는 변형된 데이터 스트림입니다.

원시 측정과 변형된 측정을 사용하여 지정된 시간 간격 동안 머신 데이터를 집계하는, 다음과 같은 [지표](#)를 정의합니다. 이 섹션에서 지표를 정의할 때 각 지표에 동일한 시간 간격을 선택합니다.

지표

- $Successes = sum(Good_Count)$ – 지정된 시간 간격 동안 성공적으로 채워진 패키지 수입니다.
- $Failures = sum(Bad_Count)$ – 지정된 시간 간격 동안 성공적으로 채워지지 않은 패키지 수입니다.
- $Idle_Time = statetime(Idle)$ – 지정된 시간 간격당 머신의 총 유휴 시간(초)입니다.
- $Fault_Time = statetime(Fault)$ – 지정된 시간 간격당 머신의 총 결함 시간(초)입니다.
- $Stop_Time = statetime(Stop)$ – 지정된 시간 간격당 머신의 총 계획된 중지 시간(초)입니다.
- $Run_Time = statetime(Running)$ – 지정된 시간 간격당 문제 없이 실행되는 머신의 총 시간(초)입니다.
- $Down_Time = Idle_Time + Fault_Time + Stop_Time$ - 지정된 시간 간격 동안 머신의 총 가동 중지 시간(초)으로, 이외의 머신 상태의 합계로 계산됩니다 Run_Time .
- $Availability = Run_Time / (Run_Time + Down_Time)$ – 지정된 시간 간격 동안 머신의 가동 시간 또는 머신을 작동에 사용할 수 있는 예약 시간의 백분율입니다.
- $Quality = Successes / (Successes + Failures)$ – 지정된 시간 간격 동안 머신의 성공적으로 채워진 패키지 백분율입니다.
- $Performance = ((Successes + Failures) / Run_Time) / Ideal_Run_Rate$ – 지정된 시간 간격 동안 프로세스의 이상적인 실행 속도(초) 대비 백분율로 나타낸 머신 성능입니다.

예를 들어, $Ideal_Run_Rate$ 은 분당 60개의 패키지(초당 1개 패키지)가 될 수 있습니다. 분당 또는 시간당 $Ideal_Run_Rate$ 인 경우 Run_Time 은 초 단위이므로 적절한 단위 변환 인수로 나누어야 합니다.

- $OEE = Availability * Quality * Performance$ – 지정된 시간 간격 동안 머신의 전반적인 장비 효율성. 이 공식은 OEE를 1의 분수로 계산합니다.

사물의 데이터 수집 AWS IoT

이 튜토리얼에서 디바이스 새도우를 사용하여 다양한 AWS IoT SiteWise AWS IoT 사물로부터 데이터를 수집하는 방법을 알아보십시오. 기기 새도는 기기의 현재 상태 정보를 저장하는 JSON 객체입니다 AWS IoT . 자세한 내용은 AWS IoT 개발자 가이드의 [디바이스 새도우 서비스](#) 단원을 참조하십시오.

이 자습서를 완료한 후에는 AWS IoT 사물을 AWS IoT SiteWise 기반으로 작업을 설정할 수 있습니다. AWS IoT 사물을 사용하여 작업을 의 다른 유용한 기능과 통합할 수 AWS IoT 있습니다. 예를 들어 다음과 같은 작업을 수행하도록 AWS IoT 기능을 구성할 수 있습니다.

- [Amazon DynamoDB](#) 및 기타 장치로 데이터를 스트리밍하기 위한 [AWS IoT Events](#) 추가 규칙을 구성합니다. AWS 서비스 자세한 내용은 AWS IoT 개발자 가이드의 [규칙](#)을 참조하십시오.
- AWS IoT 플릿 인덱싱 서비스를 사용하여 디바이스 데이터를 인덱싱, 검색 및 집계하십시오. 자세한 내용은 AWS IoT 개발자 가이드의 [플릿 인덱싱](#)을 참조하십시오.
- 를 사용하여 디바이스를 감사하고 보호하세요. AWS IoT Device Defender 자세한 내용은 AWS IoT 개발자 안내서의 [AWS IoT Device Defender](#) 섹션을 참조하십시오.

이 자습서에서는 사물의 디바이스 새도우에서 에셋까지 AWS IoT 데이터를 수집하는 방법을 알아봅니다. AWS IoT SiteWise 이렇게 하려면 하나 이상의 AWS IoT 항목을 만들고 CPU 및 메모리 사용 데이터로 각 사물의 디바이스 새도를 업데이트하는 스크립트를 실행합니다. 이 자습서에서는 CPU 및 메모리 사용량 데이터를 사용하여 실제 센서 데이터를 모방합니다. 그런 다음 사물의 기기 새도가 업데이트될 AWS IoT SiteWise 때마다 이 데이터를 자산으로 보내는 AWS IoT SiteWise 동작이 포함된 규칙을 만듭니다. 자세한 정보는 [규칙을 사용한 AWS IoT Core 데이터 수집](#)을 참조하세요.

주제

- [사전 조건](#)
- [1단계: AWS IoT 정책 생성](#)
- [단계 2: AWS IoT 사물 생성 및 구성](#)
- [단계 3: 디바이스 자산 모델 생성](#)
- [단계 4: 디바이스 플릿 자산 모델 생성](#)
- [단계 5: 디바이스 자산 생성 및 구성](#)
- [6단계: 디바이스 플릿 자산 생성 및 구성](#)
- [7단계: AWS IoT Core에서 장치 자산으로 데이터를 보내는 규칙 만들기](#)
- [8단계: 디바이스 클라이언트 스크립트 실행](#)
- [9단계: 자습서 완료 후 리소스 정리](#)

사전 조건

이 튜토리얼을 완료하려면 다음이 필요합니다.

- An AWS 계정. 계정이 없는 경우 [설정 AWS 계정](#) 단원을 참조하십시오.
- Windows, macOSLinux, 또는 를 Unix 실행하여 액세스할 수 있는 개발 컴퓨터 AWS Management Console. 자세한 내용은 [AWS Management Console 시작하기](#)를 참조하십시오.
- 관리자 권한이 있는 AWS Identity and Access Management (IAM) 사용자
- Python3은 개발 컴퓨터에 설치되거나 사물로 등록하려는 장치에 설치됩니다. AWS IoT

1단계: AWS IoT 정책 생성

이 절차에서는 AWS IoT 사물이 이 자습서에 사용된 리소스에 액세스할 수 있도록 허용하는 AWS IoT 정책을 만드세요.

AWS IoT 정책을 만들려면

1. [AWS Management Console](#)에 로그인합니다.
2. 지원되는 [AWS 지역](#)을 AWS IoT SiteWise 검토하세요. 필요한 경우 지원되는 리전 중 하나로 전환합니다.
3. [AWS IoT 콘솔](#)로 이동합니다. Connect 장치 버튼이 나타나면 해당 버튼을 선택합니다.
4. 왼쪽 탐색 창에서 보안을 선택하고 정책을 선택합니다.
5. 생성을 선택하세요.
6. AWS IoT 정책 이름 (예: **SiteWiseTutorialDevicePolicy**) 을 입력합니다.
7. 정책 문서에서 JSON을 선택하여 JSON 형식으로 다음 정책을 입력합니다. **##** 및 **account-id**를 사용자 리전 및 계정 ID로 바꿉니다(예: **us-east-1** 및 **123456789012**)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:region:account-id:client/SiteWiseTutorialDevice*"
    },
    {
      "Effect": "Allow",
```

```

    "Action": "iot:Publish",
    "Resource": [
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update",
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete",
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/get"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iot:Receive",
    "Resource": [
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
      "arn:aws:iot:region:account-id:topic/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iot:Subscribe",
    "Resource": [
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
    ]
  },
  {
    "Effect": "Allow",

```

```

    "Action": [
      "iot:GetThingShadow",
      "iot:UpdateThingShadow",
      "iot:DeleteThingShadow"
    ],
    "Resource": "arn:aws:iot:region:account-id:thing/SiteWiseTutorialDevice*"
  }
]
}

```

이 정책을 사용하면 AWS IoT 장치가 MQTT 메시지를 사용하여 연결을 설정하고 장치 새도우와 통신할 수 있습니다. MQTT 메시지에 대한 자세한 내용은 [MQTT란 무엇입니까?](#) 를 참조하십시오. 디바이스 새도우와 상호 작용하기 위해 AWS IoT 사물은 다음으로 시작하는 주제에 대한 MQTT 메시지를 게시하고 수신합니다. `$aws/things/thing-name/shadow/` 이 정책에는 로 알려진 사물 정책 변수가 통합되어 있습니다. `${iot:Connection.Thing.ThingName}` 이 변수는 각 주제에 있는 연결된 사물의 이름을 대체합니다. 이 `iot:Connect` 명령문은 연결을 설정할 수 있는 장치에 제한을 설정하여 사물 정책 변수가 `SiteWiseTutorialDevice` 시작하는 이름만 대체할 수 있도록 합니다.

자세한 내용은 AWS IoT 개발자 가이드의 [사물 정책 변수](#)를 참조하십시오.

Note

이 정책은 이름이 `SiteWiseTutorialDevice`로 시작하는 사물에 적용됩니다. 사물에 다른 이름을 사용하려면 그에 따라 정책을 업데이트해야 합니다.

8. 생성을 선택하세요.

단계 2: AWS IoT 사물 생성 및 구성

이 절차에서는 사물을 생성하고 AWS IoT 구성합니다. 개발용 컴퓨터를 사물로 지정할 수 있습니다 AWS IoT . 진행하면서 여기서 배우는 원리를 실제 프로젝트에 적용할 수 있다는 점을 기억하세요. FreeRTOS를 포함하여 AWS IoT Greengrass SDK를 실행할 수 있는 모든 기기에서 유연하게 만들고 설정할 AWS IoT 수 있습니다. 자세한 내용은 AWS IoT 개발자 가이드의 [AWS IoT SDK](#) 단원을 참조하십시오.

사물을 생성하고 구성하려면 AWS IoT

1. 명령줄을 열고 다음 명령을 실행하여 이 자습서의 디렉터리를 생성합니다.

```
mkdir iot-sitewise-rule-tutorial
cd iot-sitewise-rule-tutorial
```

- 다음 명령을 실행하여 사물의 인증서 디렉토리를 생성합니다.

```
mkdir device1
```

추가 사물을 생성하는 경우 그에 따라 디렉터리 이름의 번호를 늘려 어떤 인증서가 어떤 사물에 속하는지 추적하십시오.

- [AWS IoT 콘솔](#)로 이동합니다.
- 왼쪽 탐색 창의 관리 섹션에서 모든 장치를 선택합니다. 그런 다음 사물을 선택합니다.
- 아직 사물이 없습니다 대화 상자가 나타나면 사물 생성을 선택합니다. 또는 사물 생성을 선택합니다.
- 사물 생성 페이지에서 단일 사물 생성을 선택한 후 다음을 선택합니다.
- 사물 속성 지정 페이지에서 AWS IoT 사물의 이름(예: **SiteWiseTutorialDevice1**)을 입력하고 다음을 선택합니다. 추가 사물을 생성하는 경우 그에 따라 사물 이름의 번호를 늘립니다.

Important

사물 이름은 1단계: 정책 생성에서 생성한 정책에 사용된 이름과 일치해야 합니다. AWS IoT 그렇지 않으면 디바이스를 연결할 수 없습니다 AWS IoT.

- 디바이스 인증서 구성 - 선택 사항 페이지에서 신규 인증서 자동 생성(권장)을 선택하고 다음을 선택합니다. 인증서를 사용하면 AWS IoT 장치를 안전하게 식별할 수 있습니다.
- 인증서에 정책 연결 - 옵션 페이지에서 1단계: 정책 생성에서 생성한 AWS IoT 정책을 선택하고 사물 생성을 선택합니다.
- 인증서 및 키 다운로드 대화 상자에서 다음 작업을 수행합니다.
 - 다운로드 링크를 선택하여 사물의 인증서, 퍼블릭 키 및 프라이빗 키를 다운로드합니다. 사물의 인증서(예: `iot-sitewise-rule-tutorial/device1`)에 대해 생성한 디렉터리에 세 개 파일을 모두 저장합니다.

⚠ Important

디바이스를 AWS IoT에 성공적으로 연결하는 데 필요한 사물의 인증서와 키를 다운로드할 수 있는 유일한 시간입니다.

- b. 다운로드 링크를 선택하여 루트 CA 인증서를 다운로드합니다. 루트 CA 인증서를 `iot-sitewise-rule-tutorial1`에 저장합니다. Amazon Root CA 1을 다운로드하는 것이 좋습니다.

11. 완료를 선택합니다.

이제 컴퓨터에 AWS IoT 사물이 등록되었습니다. 다음 단계 중 하나를 수행하세요.

- 계속해서 3단계: 추가 AWS IoT 항목을 만들지 않고 기기 자산 모델 만들기를 진행하십시오. 하나의 사물만으로 이 자습서를 완료할 수 있습니다.
- 다른 컴퓨터 또는 디바이스에서 이 섹션의 단계를 반복하여 더 많은 AWS IoT 사물을 생성합니다. 이 자습서에서는 여러 디바이스에서 고유한 CPU 및 메모리 사용량 데이터를 수집할 수 있도록 이 옵션을 따르는 것이 좋습니다.
- 동일한 디바이스(컴퓨터)에서 이 섹션의 단계를 반복하여 더 많은 AWS IoT 사물을 생성합니다. 각 AWS IoT 사물은 컴퓨터로부터 비슷한 CPU 및 메모리 사용량 데이터를 수신하므로 이 접근 방식을 사용하여 여러 기기에서 고유하지 않은 데이터를 수집하는 방법을 보여주세요.

단계 3: 디바이스 자산 모델 생성

이 절차에서는 CPU 및 메모리 사용 데이터를 스트리밍하는 장치를 나타내는 자산 모델을 만듭니다. AWS IoT SiteWise 기기 그룹을 나타내는 자산의 데이터를 처리하기 위해 자산 모델은 동일한 유형의 여러 자산에 대해 일관된 정보를 적용합니다. 자세한 정보는 [산업 자산 모델링](#)을 참조하세요.

디바이스를 나타내는 자산 모델 생성

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 모델을 선택합니다.
3. Create model(모델 생성)을 선택합니다.
4. 모델 세부 정보에서 모델 이름을 입력합니다. 예를 들어 **SiteWise Tutorial Device Model**입니다.
5. 측정 정의에서 다음을 수행합니다.

- a. 이름에 **CPU Usage**를 입력합니다.
- b. 단위에 %를 입력합니다.
- c. 데이터 유형을 Double로 둡니다.

측정 속성은 디바이스의 원시 데이터 스트림을 나타냅니다. 자세한 내용은 [장비의 데이터 스트림 정의\(측정값\)](#) 단원을 참조하십시오.

6. 새 측정 추가를 선택하여 두 번째 측정 속성을 추가합니다.
7. 측정 정의 아래의 두 번째 행에서 다음을 수행합니다.
 - a. 이름에 **Memory Usage**를 입력합니다.
 - b. 단위에 %를 입력합니다.
 - c. 데이터 유형을 Double로 둡니다.
8. 지표 정의에서 다음을 수행합니다.
 - a. 이름에 **Average CPU Usage**를 입력합니다.
 - b. 공식에 **avg(CPU Usage)**를 입력합니다. 자동 완성 목록에서 CPU Usage가 나타나면 선택합니다.
 - c. 시간 간격에 **5 minutes**를 입력합니다.

지표 속성은 한 간격 동안 모든 입력 데이터 포인트를 처리하고 간격당 단일 데이터 포인트를 출력하는 집계 계산을 정의합니다. 이 지표 속성은 각 디바이스의 평균 CPU 사용량을 5분마다 계산합니다. 자세한 내용은 [속성과 기타 자산의 데이터 집계\(지표\)](#) 단원을 참조하십시오.

9. 새 지표 추가를 선택하여 두 번째 지표 속성을 추가합니다.
10. 지표 정의 아래의 두 번째 행에서 다음을 수행합니다.
 - a. 이름에 **Average Memory Usage**를 입력합니다.
 - b. 공식에 **avg(Memory Usage)**를 입력합니다. 자동 완성 목록에서 Memory Usage가 나타나면 선택합니다.
 - c. 시간 간격에 **5 minutes**를 입력합니다.

이 지표 속성은 각 디바이스의 평균 메모리 사용량을 5분마다 계산합니다.

11. (선택 사항) 디바이스별로 계산하려는 다른 지표를 추가합니다. 몇 가지 흥미로운 함수는 min 및 max를 포함합니다. 자세한 내용은 [공식 표현식 사용](#) 단원을 참조하십시오. 단계 4: 디바이스 플릿

자산 모델 생성에서는 전체 디바이스 플릿의 데이터를 사용하여 지표를 계산할 수 있는 상위 자산을 생성합니다.

12. Create model(모델 생성)을 선택합니다.

단계 4: 디바이스 플릿 자산 모델 생성

이 절차에서는 장치 컬렉션을 AWS IoT SiteWise 상징하는 자산 모델을 작성합니다. 이 자산 모델 내에서 여러 장치 자산을 하나의 중요한 플릿 자산에 연결할 수 있는 구조를 설정합니다. 그런 다음 연결된 모든 장치 자산의 데이터를 통합하기 위한 플릿 자산 모델의 메트릭을 간략하게 설명합니다. 이 접근 방식을 통해 전체 플릿의 집합적 성과에 대한 포괄적인 통찰력을 얻을 수 있습니다.

디바이스 플릿을 나타내는 자산 모델 생성

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 모델을 선택합니다.
3. Create model(모델 생성)을 선택합니다.
4. 모델 세부 정보에서 모델 이름을 입력합니다. 예를 들어 **SiteWise Tutorial Device Fleet Model**입니다.
5. 계층 구조 정의에서 다음을 수행하십시오.
 - a. 계층 구조 이름에 **Device**를 입력합니다.
 - b. 계층 구조 모델에서 디바이스 자산 모델(**SiteWise Tutorial Device Model**)을 선택합니다.

계층 구조는 상위(플릿) 자산 모델과 하위(디바이스) 자산 모델 간의 관계를 정의합니다. 상위 자산은 하위 자산의 속성 데이터에 액세스할 수 있습니다. 나중에 자산을 생성할 때는 상위 자산 모델의 계층 구조 정의에 따라 하위 자산을 상위 자산에 연결해야 합니다. 자세한 내용은 [자산 모델 계층 구조 정의](#) 단원을 참조하십시오.

6. 지표 정의에서 다음을 수행합니다.
 - a. 이름에 **Average CPU Usage**를 입력합니다.
 - b. 공식에 **avg(Device | Average CPU Usage)**를 입력합니다. 자동 완성 목록이 나타나면 Device를 선택하여 계층 구조를 선택한 다음 Average CPU Usage를 선택하여 이전에 생성한 디바이스 자산에서 지표를 선택합니다.
 - c. 시간 간격에 **5 minutes**를 입력합니다.

이 지표 속성은 **Device** 계층 구조를 통해 플릿 자산과 연결된 모든 디바이스 자산의 평균 CPU 사용량을 계산합니다.

7. 새 지표 추가를 선택하여 두 번째 지표 속성을 추가합니다.
8. 지표 정의 아래의 두 번째 행에서 다음을 수행합니다.
 - a. 이름에 **Average Memory Usage**를 입력합니다.
 - b. 공식에 **avg(Device | Average Memory Usage)**를 입력합니다. 자동 완성 목록이 나타나면 **Device**를 선택하여 계층 구조를 선택한 다음 **Average Memory Usage**를 선택하여 이전에 생성한 디바이스 자산에서 지표를 선택합니다.
 - c. 시간 간격에 **5 minutes**를 입력합니다.

이 지표 속성은 **Device** 계층 구조를 통해 플릿 자산과 연결된 모든 디바이스 자산의 평균 메모리 사용량을 계산합니다.

9. (선택 사항) 디바이스 플릿에서 계산하려는 다른 지표를 추가합니다.
10. Create model(모델 생성)을 선택합니다.

단계 5: 디바이스 자산 생성 및 구성

이 절차에서는 장치 자산 모델을 기반으로 장치 자산을 생성합니다. 그런 다음 각 측정 속성에 대한 속성 별칭을 정의합니다. 속성 별칭은 자산 속성을 식별하는 고유한 문자열입니다. 나중에 자산 ID 및 속성 ID 대신 별칭을 사용하여 데이터 업로드에 사용할 속성을 식별할 수 있습니다. 자세한 정보는 [산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.

디바이스 자산을 생성하고 속성 별칭 정의

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 자산을 선택합니다.
3. 자산 생성을 선택합니다.
4. 모델 정보에서 디바이스 자산 모델인 **SiteWise Tutorial Device Model**을 선택합니다.
5. 자산 정보에서 자산 이름을 입력합니다. 예를 들어 **SiteWise Tutorial Device 1**입니다.
6. 자산 생성을 선택합니다.
7. 새 디바이스 자산에 대해 편집을 선택합니다.

8. CPU Usage에서 속성 별칭으로 `/tutorial/device/SiteWiseTutorialDevice1/cpu`를 입력합니다. 속성 별칭에 AWS IoT 사물 이름을 포함하면 단일 규칙을 사용하여 모든 기기에서 데이터를 수집할 수 있습니다. AWS IoT
9. Memory Usage에서 속성 별칭으로 `/tutorial/device/SiteWiseTutorialDevice1/memory`를 입력합니다.
10. 저장을 선택합니다.

이전에 여러 AWS IoT 항목을 만든 경우 기기마다 3~10단계를 반복하고 그에 따라 자산 이름과 속성 별칭의 수를 늘리십시오. 예를 들어 두 번째 디바이스 자산의 이름은 **SiteWise Tutorial Device 2**이고 속성 별칭은 `/tutorial/device/SiteWiseTutorialDevice2/cpu` 및 `/tutorial/device/SiteWiseTutorialDevice2/memory`여야 합니다.

6단계: 디바이스 플릿 자산 생성 및 구성

이 절차에서는 디바이스 플릿 자산 모델에서 파생된 디바이스 플릿 자산을 구성합니다. 그런 다음 개별 기기 자산을 플릿 자산에 연결합니다. 이 연결을 통해 플릿 자산의 메트릭 속성을 통해 여러 장치의 데이터를 컴파일하고 분석할 수 있습니다. 이 데이터를 통해 전체 플릿의 종합적인 성과를 통합적으로 확인할 수 있습니다.

디바이스 플릿 자산을 생성하고 디바이스 플릿 연결

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 자산을 선택합니다.
3. 자산 생성을 선택합니다.
4. 모델 정보에서 디바이스 플릿 자산 모델인 **SiteWise Tutorial Device Fleet Model**을 선택합니다.
5. 자산 정보에서 자산 이름을 입력합니다. 예를 들어 **SiteWise Tutorial Device Fleet 1**입니다.
6. 자산 생성을 선택합니다.
7. 새 디바이스 플릿 자산에 대해 편집을 선택합니다.
8. 이 자산과 관련된 자산에서 관련 자산 추가를 선택하고 다음을 수행하십시오.
 - a. 계층 구조에서 Device를 선택합니다. 이 계층 구조는 디바이스 자산과 디바이스 플릿 자산 간의 계층적 관계를 식별합니다. 이 자습서의 앞부분에서 디바이스 플릿 자산 모델에서 이 계층 구조를 정의했습니다.
 - b. 자산에서 디바이스 자산 SiteWise Tutorial Device 1을 선택합니다.

9. (선택 사항) 이전에 여러 디바이스 자산을 생성한 경우 생성한 각 디바이스 자산에 대해 8~10단계를 반복합니다.
10. 저장을 선택합니다.

이제 디바이스 자산이 계층 구조에 구성되어 있어야 합니다.

7단계: AWS IoT Core에서 장치 자산으로 데이터를 보내는 규칙 만들기

이 절차에서는 에서 규칙을 설정합니다 AWS IoT Core. 규칙은 디바이스 새도우의 알림 메시지를 해석하고 데이터를 디바이스 자산으로 전송하도록 설계되었습니다. 디바이스의 새도가 업데이트될 AWS IoT SiteWise때마다 MQTT 메시지가 AWS IoT 전송됩니다. MQTT 메시지를 기반으로 디바이스 새도우가 변경될 때 조치를 취하는 규칙을 생성할 수 있습니다. 이 경우 목표는 업데이트 메시지를 처리하고 속성 값을 추출하여 의 장치 자산으로 전송하는 것입니다. AWS IoT SiteWise

AWS IoT SiteWise 액션이 포함된 규칙 생성하기

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 메시지 라우팅을 선택한 다음 규칙을 선택합니다.
3. Create rule을 선택합니다.
4. 규칙의 이름과 설명을 입력하고 다음을 선택합니다.
5. 다음 SQL 문을 입력하고 다음을 선택합니다.

```
SELECT
  *
FROM
  '$aws/things/+/shadow/update/accepted'
WHERE
  startsWith(topic(3), "SiteWiseTutorialDevice")
```

이 규칙 쿼리 문은 디바이스 새도우 서비스가 \$aws/things/*thingName*/shadow/update/accepted에 새도우 업데이트를 게시하기 때문에 작동합니다. 디바이스 새도우에 대한 자세한 내용은 AWS IoT 개발자 가이드의 [디바이스 새도우 서비스](#)를 참조하십시오.

WHERE 절에서 이 규칙 쿼리 문은 topic(3) 함수를 사용하여 주제의 세 번째 세그먼트에서 사물 이름을 가져옵니다. 그런 다음 명령문은 자습서 디바이스의 이름과 일치하지 않는 이름을 가진 디바이스를 필터링합니다. AWS IoT SQL에 대한 자세한 내용은 AWS IoT 개발자 안내서의 [AWS IoT SQL 참조](#)를 참조하십시오.

6. 규칙 작업에서 AWS IoT SiteWise의 자산 속성에 메시지 데이터 보내기를 선택하고 다음을 수행하십시오.

- a. 속성 별칭 기준을 선택합니다.
- b. 속성 별칭에 `/tutorial/device/${topic(3)}/cpu`를 입력합니다.

`${...}`구문은 대체 템플릿입니다. AWS IoT 중괄호 안의 내용을 평가합니다. 이 대체 템플릿은 주제에서 사물 이름을 가져와서 각 사물에 고유한 별칭을 생성합니다. 자세한 내용은 AWS IoT 개발자 가이드의 [대체 템플릿](#)을 참조하십시오.

Note

대체 템플릿의 표현식은 SELECT 문과 별도로 평가되므로 대체 템플릿을 사용하면 AS 절을 사용하여 생성한 별칭은 참조할 수 없습니다. 지원되는 함수와 연산자 외에 원래 페이로드에 있는 정보만 참조할 수 있습니다.

- c. 항목 ID - 선택 사항에 `${concat(topic(3), "-cpu-", floor(state.reported.timestamp))}`을 입력합니다.

항목 ID는 각 값 입력 시도를 고유하게 식별합니다. 항목이 오류를 반환하는 경우 오류 출력에서 항목 ID를 찾아 문제를 해결할 수 있습니다. 이 항목 ID의 대체 템플릿은 사물 이름과 디바이스의 보고된 타임스탬프를 결합합니다. 예를 들어 결과 항목 ID는 SiteWiseTutorialDevice1-cpu-1579808494와 같을 수 있습니다.

- d. 초 단위 시간에 `${floor(state.reported.timestamp)}`를 입력합니다.

이 대체 템플릿은 디바이스의 보고된 타임스탬프에서 시간(초)을 계산합니다. 이 자습서에서 디바이스가 Unix 에포크 시간의 타임스탬프(초)를 부동 소수점 숫자로 보고합니다.

- e. 나노초 단위 오프셋 - 선택 사항에 `${floor((state.reported.timestamp % 1) * 1E9)}`를 입력합니다.

이 대체 템플릿은 디바이스의 보고된 타임스탬프의 소수 부분을 변환하여 초 단위 시간에서 나노초 단위 오프셋을 계산합니다.

Note

AWS IoT SiteWise 데이터에 Unix 에포크 타임의 현재 타임스탬프가 있어야 합니다. 디바이스가 시간을 정확하게 보고하지 않는 경우 `timestamp()`를 사용하여 AWS IoT

규칙 엔진에서 현재 시간을 가져올 수 있습니다. 이 함수는 시간을 밀리초 단위로 보고하므로 규칙 작업의 시간 파라미터를 다음 값으로 업데이트해야 합니다.

- 초 단위 시간에 $\{\text{floor}(\text{timestamp}() / 1\text{E}3)\}$ 를 입력합니다.
- 나노초 단위 오프셋에 $\{(\text{timestamp}() \% 1\text{E}3) * 1\text{E}6\}$ 를 입력합니다.

f. 데이터 유형에서 Double을 선택합니다.

이 데이터 유형은 자산 모델에서 정의한 자산 속성의 데이터 유형과 일치해야 합니다.

- g. 값에는 $\{\text{state.reported.cpu}\}$ 를 입력합니다. 대체 템플릿에서는 `.` 연산자를 사용하여 JSON 구조 내에서 값을 검색합니다.
- h. 항목 추가를 선택하여 메모리 사용량 속성에 대한 새 항목을 추가하고 해당 속성에 대해 다음 단계를 다시 완료합니다.
- 속성 별칭 기준을 선택합니다.
 - 속성 별칭에 `/tutorial/device/${topic(3)}/memory`를 입력합니다.
 - 항목 ID - 선택 사항에 $\{\text{concat}(\text{topic}(3), \text{"-memory-"}, \text{floor}(\text{state.reported.timestamp}))\}$ 을 입력합니다.
 - 초 단위 시간에 $\{\text{floor}(\text{state.reported.timestamp})\}$ 를 입력합니다.
 - 나노초 단위 오프셋 - 선택 사항에 $\{\text{floor}((\text{state.reported.timestamp} \% 1) * 1\text{E}9)\}$ 를 입력합니다.
 - 데이터 유형에서 Double을 선택합니다.
 - 값에는 $\{\text{state.reported.memory}\}$ 를 입력합니다.
- i. IAM 역할에서 새 역할 생성을 선택하여 이 규칙 작업에 대한 IAM 역할을 생성합니다. 이 역할을 통해 AWS IoT 디바이스 플릿 자산 및 해당 자산 계층 구조의 속성으로 데이터를 푸시할 수 있습니다.
- j. 역할 이름을 입력하고 생성을 선택합니다.
7. (선택 사항) 규칙 문제를 해결하는 데 사용할 수 있는 오류 작업을 구성합니다. 자세한 내용은 [규칙 문제 해결](#) 단원을 참조하십시오.
8. 다음을 선택합니다.
9. 설정을 검토하고 생성을 선택하여 규칙을 생성합니다.

8단계: 디바이스 클라이언트 스크립트 실행

이 가이드에서는 실제 기기를 사용하여 데이터를 보고하지는 않습니다. 대신 스크립트를 실행하여 CPU 및 메모리 사용량으로 AWS IoT 사물의 기기 새도를 업데이트하여 실제 센서 데이터를 모방합니다. 스크립트를 실행하려면 먼저 필수 Python 패키지를 설치해야 합니다. 이 절차에서는 필수 Python 패키지를 설치한 다음 장치 클라이언트 스크립트를 실행합니다.

디바이스 클라이언트 스크립트 구성 및 실행

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창 하단에서 설정을 선택합니다.
3. 디바이스 클라이언트 스크립트와 함께 사용할 사용자 지정 엔드포인트를 저장합니다. 이 엔드포인트를 사용하여 사물의 새도우와 상호 작용할 수 있습니다. 이 엔드포인트는 현재 리전의 계정에 고유합니다.

사용자 지정 엔드포인트는 다음 예제와 같아야 합니다.

```
identifier.iot.region.amazonaws.com
```

4. 명령줄을 열고 다음 명령을 실행하여 이전에 생성한 자습서 디렉터리로 이동합니다.

```
cd iot-sitewise-rule-tutorial
```

5. 다음 명령을 실행하여 AWS IoT Device SDK for Python를 설치합니다.

```
pip3 install AWSIoTPythonSDK
```

자세한 내용은 AWS IoT 개발자 가이드의 [AWS IoT Device SDK for Python](#) 단원을 참조하십시오.

6. 다음 명령을 실행하여 교차 플랫폼 프로세스 및 시스템 유틸리티 라이브러리인 psutil을 설치합니다.

```
pip3 install psutil
```

자세한 내용은 Python 패키지 인덱스의 [psutil](#)을 참조하십시오.

7. `iot-sitewise-rule-tutorial` 디렉터리에서 `thing_performance.py`이라는 파일을 생성한 후 다음 Python 코드를 파일에 복사합니다.

```
import AWSIoTPythonSDK.MQTTLib as AWSIoTPyMQTT
```

```
import json
import psutil
import argparse
import logging
import time

# Configures the argument parser for this program.
def configureParser():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "-e",
        "--endpoint",
        action="store",
        required=True,
        dest="host",
        help="Your AWS IoT custom endpoint",
    )
    parser.add_argument(
        "-r",
        "--rootCA",
        action="store",
        required=True,
        dest="rootCAPath",
        help="Root CA file path",
    )
    parser.add_argument(
        "-c",
        "--cert",
        action="store",
        required=True,
        dest="certificatePath",
        help="Certificate file path",
    )
    parser.add_argument(
        "-k",
        "--key",
        action="store",
        required=True,
        dest="privateKeyPath",
        help="Private key file path",
    )
    parser.add_argument(
```

```
        "-p",
        "--port",
        action="store",
        dest="port",
        type=int,
        default=8883,
        help="Port number override",
    )
    parser.add_argument(
        "-n",
        "--thingName",
        action="store",
        required=True,
        dest="thingName",
        help="Targeted thing name",
    )
    parser.add_argument(
        "-d",
        "--requestDelay",
        action="store",
        dest="requestDelay",
        type=float,
        default=1,
        help="Time between requests (in seconds)",
    )
    parser.add_argument(
        "-v",
        "--enableLogging",
        action="store_true",
        dest="enableLogging",
        help="Enable logging for the AWS IoT Device SDK for Python",
    )
    return parser

# An MQTT shadow client that uploads device performance data to AWS IoT at a
regular interval.
class PerformanceShadowClient:
    def __init__(
        self,
        thingName,
        host,
        port,
        rootCAPath,
```

```
privateKeyPath,
certificatePath,
requestDelay,
):
    self.thingName = thingName
    self.host = host
    self.port = port
    self.rootCAPath = rootCAPath
    self.privateKeyPath = privateKeyPath
    self.certificatePath = certificatePath
    self.requestDelay = requestDelay

    # Updates this thing's shadow with system performance data at a regular
    interval.
    def run(self):
        print("Connecting MQTT client for {}".format(self.thingName))
        mqttClient = self.configureMQTTClient()
        mqttClient.connect()
        print("MQTT client for {} connected".format(self.thingName))
        deviceShadowHandler = mqttClient.createShadowHandlerWithName(
            self.thingName, True
        )

        print("Running performance shadow client for {}...
\n".format(self.thingName))
        while True:
            performance = self.readPerformance()
            print("[{}].format(self.thingName))
            print("CPU:\t{}%".format(performance["cpu"]))
            print("Memory:\t{}%\n".format(performance["memory"]))
            payload = {"state": {"reported": performance}}
            deviceShadowHandler.shadowUpdate(
                json.dumps(payload), self.shadowUpdateCallback, 5
            )
            time.sleep(args.requestDelay)

    # Configures the MQTT shadow client for this thing.
    def configureMQTTClient(self):
        mqttClient = AWSIoTPyMQTT.AWSIoTMQTTShadowClient(self.thingName)
        mqttClient.configureEndpoint(self.host, self.port)
        mqttClient.configureCredentials(
            self.rootCAPath, self.privateKeyPath, self.certificatePath
        )
        mqttClient.configureAutoReconnectBackoffTime(1, 32, 20)
```

```
mqttClient.configureConnectDisconnectTimeout(10)
mqttClient.configureMQTTOperationTimeout(5)
return mqttClient

# Returns the local device's CPU usage, memory usage, and timestamp.
def readPerformance(self):
    cpu = psutil.cpu_percent()
    memory = psutil.virtual_memory().percent
    timestamp = time.time()
    return {"cpu": cpu, "memory": memory, "timestamp": timestamp}

# Prints the result of a shadow update call.
def shadowUpdateCallback(self, payload, responseStatus, token):
    print("{} {}".format(self.thingName))
    print("Update request {} {} \n".format(token, responseStatus))

# Configures debug logging for the AWS IoT Device SDK for Python.
def configureLogging():
    logger = logging.getLogger("AWSIoTPythonSDK.core")
    logger.setLevel(logging.DEBUG)
    streamHandler = logging.StreamHandler()
    formatter = logging.Formatter(
        "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
    )
    streamHandler.setFormatter(formatter)
    logger.addHandler(streamHandler)

# Runs the performance shadow client with user arguments.
if __name__ == "__main__":
    parser = configureParser()
    args = parser.parse_args()
    if args.enableLogging:
        configureLogging()
    thingClient = PerformanceShadowClient(
        args.thingName,
        args.host,
        args.port,
        args.rootCAPath,
        args.privateKeyPath,
        args.certificatePath,
        args.requestDelay,
    )
```

```
thingClient.run()
```

8. 다음 파라미터가 있는 명령줄에서 `thing_performance.py`를 실행합니다.
- `-n, --thingName` - 사물 이름(예: **SiteWiseTutorialDevice1**)입니다.
 - `-e, --endpoint` — 이 절차의 앞부분에서 저장한 사용자 지정 AWS IoT 엔드포인트.
 - `-r, --rootCA` — AWS IoT 루트 CA 인증서의 경로.
 - `-c, --cert` — AWS IoT 사물 인증서의 경로.
 - `-k, --key` — AWS IoT 사물 인증서 개인 키의 경로.
 - `-d, --requestDelay` - (선택 사항) 각 디바이스 새도우 업데이트 사이에 대기할 시간(초)입니다. 기본값은 1초입니다.
 - `-v, --enableLogging` - (선택 사항) 이 파라미터가 있으면 스크립트는 AWS IoT Device SDK for Python에서 디버그 메시지를 인쇄합니다.

명령은 다음 예제와 비슷해야 합니다.

```
python3 thing_performance.py \
  --thingName SiteWiseTutorialDevice1 \
  --endpoint identifier.iot.region.amazonaws.com \
  --rootCA AmazonRootCA1.pem \
  --cert device1/thing-id-certificate.pem.crt \
  --key device1/thing-id-private.pem.key
```

추가 항목에 대한 스크립트를 실행하는 경우 AWS IoT 사물 이름과 인증서 디렉토리를 적절히 업데이트하십시오.

9. 디바이스에서 프로그램을 열고 닫으면 CPU 및 메모리 사용량이 어떻게 변하는지 확인할 수 있습니다. 스크립트는 각 CPU 및 메모리 사용량 읽기를 인쇄합니다. 스크립트가 디바이스 새도우 서비스에 데이터를 성공적으로 업로드하면 스크립트의 출력은 다음 예제와 같아야 합니다.

```
[SiteWiseTutorialDevice1]
CPU:    24.6%
Memory: 85.2%

[SiteWiseTutorialDevice1]
Update request e6686e44-fca0-44db-aa48-3ca81726f3e3 accepted
```

10. 스크립트가 디바이스 새도우를 업데이트하는지 확인하려면 다음 단계를 따르십시오.

- a. [AWS IoT 콘솔](#)로 이동합니다.
- b. 왼쪽 탐색 창에서 모든 디바이스스를 선택한 후 사물을 선택합니다.
- c. 원하는 것을 선택하세요 SiteWiseTutorialDevice.
- d. 디바이스 새도우 탭을 선택하고 클래식 새도우를 선택한 다음 새도우 상태가 다음 예와 같은지 확인합니다.

```
{
  "reported": {
    "cpu": 24.6,
    "memory": 85.2,
    "timestamp": 1579567542.2835066
  }
}
```

사물의 새도우 상태가 비어 있거나 이전 예제와 다르게 보인다면 스크립트가 실행 중이고 제대로 연결되었는지 확인하세요 AWS IoT. 에 연결할 때 스크립트 제한 시간이 계속 초과되면 이 자습서에 따라 [사물 정책](#)이 구성되어 있는지 확인하세요. AWS IoT

11. 규칙 작업에서 데이터를 AWS IoT SiteWise에 전송하는지 확인하려면 다음 단계를 따르십시오.

- a. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
- b. 왼쪽 탐색 창에서 자산을 선택합니다.
- c. 디바이스 플릿 자산(SiteWise Tutorial Device Fleet 1 1) 옆에 있는 화살표를 선택하여 자산 계층 구조를 확장한 다음 디바이스 자산(SiteWise Tutorial Device 1)을 선택합니다.
- d. 측정을 선택합니다.
- e. 최신 값 셀에 CPU Usage 및 Memory Usage 속성에 대한 값이 있는지 확인합니다.

Measurements				
Name	Alias	Notification status	Notification topic	Latest value
CPU Usage	/tutorial/device/SiteWiseTutorialDevice1/cpu	⊖ Disabled	-	24.6
Memory Usage	/tutorial/device/SiteWiseTutorialDevice1/memory	⊖ Disabled	-	85.2

- f. CPU Usage 및 Memory Usage 속성에 최신 값이 없으면 페이지를 새로 고칩니다. 몇 분 후에도 값이 나타나지 않는 경우 [규칙 문제 해결](#) 단원을 참조하십시오.

12. 이 자습서를 완료했습니다. 데이터의 실시간 시각화를 탐색하려는 경우 AWS IoT SiteWise Monitor에서 포털을 구성할 수 있습니다. 자세한 정보는 [를 사용하여 데이터 모니터링 AWS IoT SiteWise Monitor](#)을 참조하세요. 그렇지 않으면 명령 프롬프트에서 CTRL+C를 눌러 디바이스 클

라이언트 스크립트를 중지할 수 있습니다. Python 프로그램으로 전송되는 메시지로 요금이 발생할 가능성은 적지만 사용 후에는 프로그램을 중지시키는 것이 좋습니다.

9단계: 자습서 완료 후 리소스 정리

AWS IoT 사물의 데이터 수집에 관한 튜토리얼을 완료한 후에는 추가 비용이 발생하지 않도록 리소스를 정리하세요.

에서 계층적 자산을 삭제하려면 AWS IoT SiteWise

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 자산을 선택합니다.
3. 에서 자산을 삭제할 때는 먼저 자산을 AWS IoT SiteWise 연결 해제해야 합니다.

디바이스 자산을 디바이스 플릿 자산에서 분리하려면 다음 단계를 완료하십시오.

- a. 디바이스 플릿 자산(SiteWise Tutorial Device Fleet 1)을 선택합니다.
- b. 편집을 선택합니다.
- c. 이 자산에 연결된 자산 아래에서 이 디바이스 플릿 자산과 연결된 각 디바이스 자산에 대해 연결 해제를 선택합니다.
- d. 저장을 선택합니다.

이제 디바이스 자산이 더 이상 계층 구조로 구성되어 있지 않음을 확인할 수 있습니다.

4. 디바이스 자산(SiteWise Tutorial Device 1)을 선택합니다.
5. 삭제를 선택합니다.
6. 확인 대화 상자에 **Delete**를 입력한 다음 삭제를 선택합니다.
7. 각 디바이스 자산 및 디바이스 플릿 자산(SiteWise Tutorial Device Fleet 1)에 대해 단계 4~6을 반복합니다.

에서 계층적 자산 모델을 삭제하려면 AWS IoT SiteWise

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 아직 디바이스 자산 및 디바이스 플릿 자산을 삭제하지 않았다면 삭제하십시오. 자세한 내용은 [이전 절차](#)를 참조하십시오. 해당 모델에서 생성된 자산이 있으면 모델을 삭제할 수 없습니다.
3. 왼쪽 탐색 창에서 모델을 선택합니다.
4. 디바이스 플릿 자산 모델(SiteWise Tutorial Device Fleet Model)을 선택합니다.

계층적 자산 모델을 삭제할 때는 먼저 상위 자산 모델을 삭제해야 합니다.

5. 삭제를 선택합니다.
6. 확인 대화 상자에 **Delete**를 입력한 다음 삭제를 선택합니다.
7. 디바이스 자산 모델(SiteWise Tutorial Device Model)에 대해 단계 4~6을 반복합니다.

에서 규칙을 비활성화하거나 삭제하려면 AWS IoT Core

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 메시지 라우팅을 선택한 다음 규칙을 선택합니다.
3. 규칙을 선택하고 삭제를 선택합니다.
4. 확인 대화 상자에서 규칙 이름을 입력한 다음 삭제를 선택합니다.

모니터에서 SiteWise 풍력 발전 단지 데이터 시각화 및 공유

이 자습서에서는 포털이라는 관리형 웹 애플리케이션을 통해 산업 데이터를 시각화하고 공유하는 AWS IoT SiteWise Monitor 방법을 설명합니다. 각 포털에는 프로젝트가 포함되므로 각 프로젝트 내에서 액세스할 수 있는 데이터를 유연하게 선택할 수 있습니다. 그런 다음 각 포털에 접근할 수 있는 조직 내 사용자를 지정하세요. 사용자는 AWS IAM Identity Center 계정을 사용하여 포털에 로그인하므로 기존 ID 저장소 또는 에서 관리하는 AWS스토어를 사용할 수 있습니다.

관리자와 충분한 권한이 있는 사용자는 각 프로젝트에서 대시보드를 생성하여 산업 데이터를 의미 있는 방식으로 시각화할 수 있습니다. 그런 다음 사용자가 이러한 대시보드를 통해 데이터에 대한 통찰력을 빠르게 확보하고 운영을 모니터링할 수 있습니다. 회사의 모든 사용자에게 각 프로젝트에 대한 관리 권한이나 읽기 전용 권한을 구성할 수 있습니다. 자세한 정보는 [를 사용하여 데이터 모니터링 AWS IoT SiteWise Monitor](#)을 참조하세요.

자습서 전체에서 풍력 발전 단지에 대한 샘플 데이터셋을 제공하여 AWS IoT SiteWise 데모를 개선합니다. SiteWise 모니터에서 포털을 구성하고, 프로젝트를 만들고, 대시보드를 생성하여 풍력 발전 단지 데이터를 시각화합니다. 이 자습서에서는 프로젝트 및 관련 대시보드를 소유하거나 볼 수 있는 권한을 할당하는 방법과 함께 추가 사용자를 생성하는 방법도 다룹니다.

Note

SiteWise Monitor를 사용하는 경우 포털에 로그인한 사용자당 요금이 부과됩니다 (월별). 이 자습서에서는 3명의 사용자를 생성하지만 한 명의 사용자로만 로그인하면 됩니다. 이 자습서를

완료하면 한 명의 사용자에게 대한 요금이 청구됩니다. 자세한 내용은 [AWS IoT SiteWise 요금](#)을 참조하십시오.

주제

- [필수 조건](#)
- [1단계: 모니터에서 SiteWise 포털 생성](#)
- [2단계: 포털에 로그인](#)
- [3단계: 풍력 발전 단지 프로젝트 생성](#)
- [4단계: 대시보드를 만들어 풍력 발전 단지 데이터를 시각화하세요.](#)
- [5단계: 포털 둘러보기](#)
- [6단계: 자습서 완료 후 리소스 정리](#)

필수 조건

이 튜토리얼을 완료하려면 다음이 필요합니다.

- An AWS 계정. 계정이 없는 경우 [설정 AWS 계정](#) 단원을 참조하십시오.
- Windows, macOSLinux, 또는 Unix 를 실행하는 개발 컴퓨터 AWS Management Console. 자세한 내용은 [AWS Management Console 시작하기](#)를 참조하십시오.
- 관리자 권한이 있는 AWS Identity and Access Management (IAM) 사용자.
- 실행 중인 AWS IoT SiteWise 풍력 발전 단지 데모. 데모를 설정하면 모델과 자산을 정의하고 데이터를 스트리밍하여 풍력 발전 단지를 표현합니다. AWS IoT SiteWise 자세한 정보는 [AWS IoT SiteWise 데모 사용](#)을 참조하세요.
- 계정에서 IAM Identity Center를 활성화한 경우 AWS Organizations 관리 계정에 로그인하십시오. 자세한 내용을 알아보려면 [AWS Organizations 용어 및 개념](#)을 참조하십시오. IAM Identity Center를 활성화하지 않은 경우 이 자습서에서 활성화하고 계정을 관리 계정으로 설정해야 합니다.

AWS Organizations 관리 계정에 로그인할 수 없는 경우 조직에 IAM Identity Center 사용자가 있는 한 자습서를 부분적으로 완료할 수 있습니다. 이 경우 포털과 대시보드를 생성할 수 있지만 프로젝트에 할당할 새 IAM Identity Center 사용자를 생성할 수는 없습니다.

1단계: 모니터에서 SiteWise 포털 생성

이 절차에서는 AWS IoT SiteWise Monitor에서 포털을 생성합니다. 각 포털은 관리되는 웹 애플리케이션으로, 사용자와 함께 AWS IAM Identity Center 계정으로 로그인할 수 있습니다. IAM Identity Center를 사용하면 회사의 기존 ID 저장소를 사용하거나 관리되는 AWS ID 저장소를 생성할 수 있습니다. 회사 직원은 별도로 AWS 계정만들지 않고도 로그인할 수 있습니다.

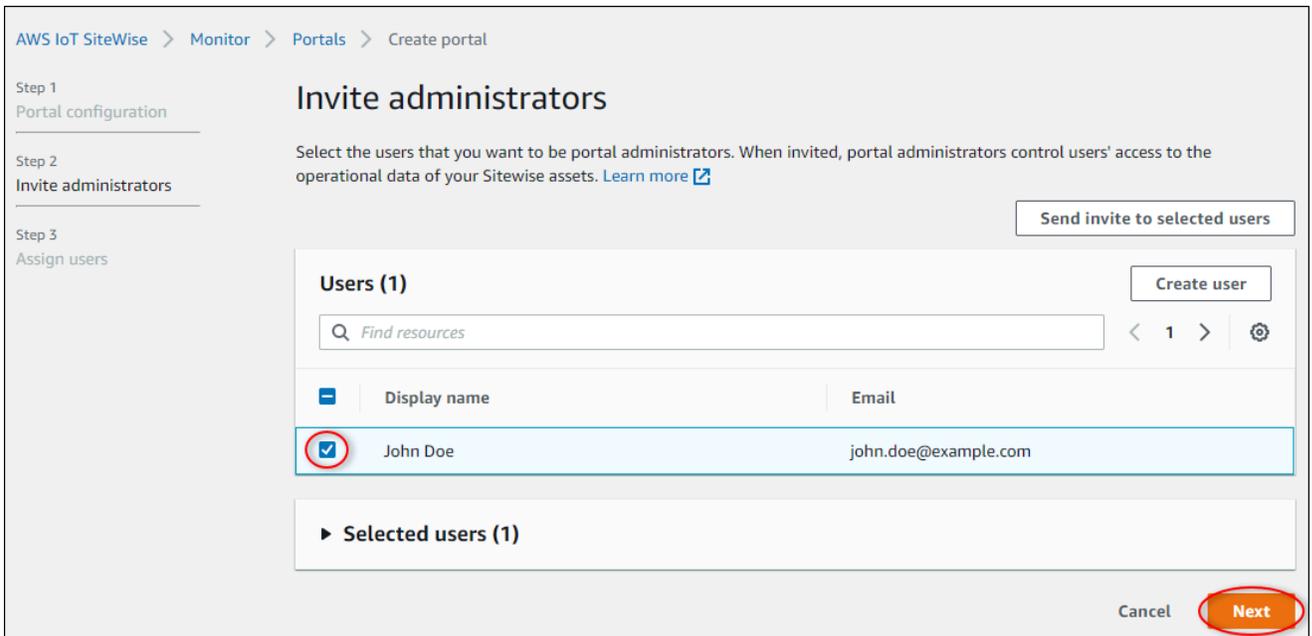
포털 생성

1. [AWS IoT SiteWise 콘솔](#)에 로그인합니다.
2. 지원되는 [AWS IoT SiteWise 엔드포인트와 할당량을](#) 검토하고 필요한 AWS IoT SiteWise 경우 지역을 전환하세요. 동일한 지역에서 AWS IoT SiteWise 데모를 실행해야 합니다.
3. 왼쪽 탐색 창에서 포털을 선택합니다.
4. 포털 생성을 선택합니다.
5. IAM Identity Center를 이미 활성화한 경우 6단계로 건너뛰십시오. 그렇지 않은 경우 다음 단계를 완료하여 IAM Identity Center를 활성화합니다.
 - a. 활성화 AWS IAM Identity Center (SSO) 페이지에서 이메일 주소, 이름, 성을 입력하여 포털 관리자가 될 IAM Identity Center 사용자를 생성합니다. 새 IAM Identity Center 사용자의 암호를 설정할 때 이메일을 받을 수 있도록 액세스할 수 있는 이메일 주소를 사용합니다.

포털에서 포털 관리자는 프로젝트를 생성하고 사용자를 프로젝트에 할당합니다. 나중에 더 많은 사용자를 만들 수 있습니다.

The screenshot shows the 'Create portal' page in the AWS IoT SiteWise Monitor console. The page title is 'Enable AWS Single Sign-On (SSO)'. Below the title, there is a brief explanation: 'AWS IoT SiteWise Monitor requires SSO to create a portal and invite users. Create your first user below to enable AWS Single-Sign On. Later in this process, you'll have the opportunity to create other users by using the AWS SSO console. [Learn more](#)'. The main form is titled 'Create a user' and contains three input fields: 'Email address' with the value 'john.doe@example.com', 'First name' with the value 'John', and 'Last name' with the value 'Doe'. At the bottom right of the form, there are two buttons: 'Cancel' and 'Create user', with the 'Create user' button highlighted by a red circle.

- b. 사용자 생성을 선택합니다.
6. Portal configuration(포털 구성) 페이지에서 다음 단계를 완료합니다.
 - a. 포털의 이름(예: **WindFarmPortal**)을 입력합니다.
 - b. (선택 사항) 포털에 대한 설명을 입력합니다. 포털이 여러 개인 경우 의미 있는 설명을 사용하여 각 포털에 포함된 내용을 추적합니다.
 - c. (선택 사항) 이미지를 업로드하여 포털에 표시합니다.
 - d. 포털 사용자가 포털에 문제가 있고 문제 해결을 위해 회사 AWS 관리자의 도움이 필요할 때 연락할 수 있는 이메일 주소를 입력합니다.
 - e. 포털 생성을 선택합니다.
 7. 관리자 초대 페이지에서 IAM Identity Center 사용자를 포털에 관리자로 할당할 수 있습니다. 포털 관리자는 포털 내에서 권한과 프로젝트를 관리합니다. 이 페이지에서 다음을 수행합니다.
 - a. 포털 관리자가 될 사용자를 선택합니다. 이 자습서의 앞 부분에서 IAM Identity Center를 활성화한 경우 생성한 사용자를 선택하십시오.



- b. (선택 사항) 선택한 사용자에게 초대장 보내기를 선택합니다. 이메일 클라이언트가 열리고 메시지 본문에 초대장이 표시됩니다. 이메일을 포털 관리자에게 보내기 전에 사용자 지정할 수 있습니다. 나중에 포털 관리자에게 이메일을 보낼 수도 있습니다. SiteWise Monitor를 처음 사용하는 경우 포털 관리자가 될 예정이라면 직접 이메일을 보내지 않아도 됩니다.
- c. 다음을 선택합니다.

8. 사용자 할당 페이지에서 IAM Identity Center 사용자를 포털에 할당할 수 있습니다. 포털 관리자는 나중에 포털 사용자를 프로젝트 소유자 또는 뷰어로 할당할 수 있습니다. 프로젝트 소유자는 프로젝트에 대시보드를 생성할 수 있습니다. 프로젝트 뷰어는 자신에게 배정된 프로젝트에 대한 읽기 전용 액세스 권한을 가집니다. 이 페이지에서 IAM Identity Center 사용자를 생성하여 포털에 추가할 수 있습니다.

Note

AWS Organizations 관리 계정에 로그인하지 않은 경우 IAM Identity Center 사용자를 생성할 수 없습니다. 사용자 할당을 선택하여 포털 사용자 없이 포털을 생성한 다음 이 단계를 건너뛰십시오.

이 페이지에서 다음을 수행합니다.

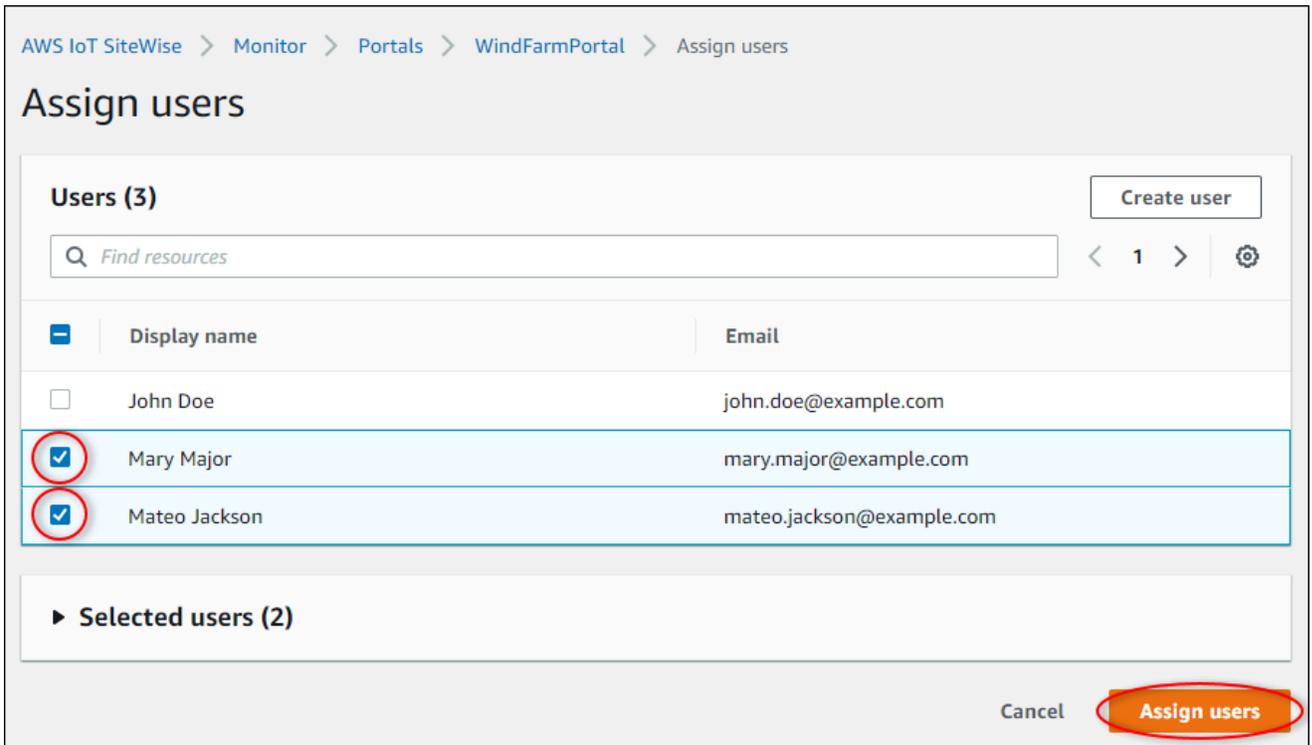
- a. 다음 단계를 두 번 완료하여 두 개의 IAM Identity Center 사용자를 생성합니다.
 - i. 사용자 생성을 선택하여 새 사용자에 대한 세부 정보를 입력할 수 있는 대화 상자를 엽니다.
 - ii. 새 사용자의 이메일 주소, 이름, 성을 입력합니다. IAM Identity Center는 사용자에게 암호 설정을 위한 이메일을 보냅니다. 이러한 사용자로 포털에 로그인하려면 액세스할 수 있는 이메일 주소를 선택하십시오. 각 이메일 주소는 고유해야 합니다. 사용자는 이메일 주소를 사용자 이름으로 사용하여 포털에 로그인합니다.

The screenshot shows a 'Create user' dialog box with the following fields and values:

- Email address:** mary.major@example.com
- First name:** Mary
- Last name:** Major

Buttons: Cancel, Create user

- iii. 사용자 생성을 선택합니다.
- b. 이전 단계에서 생성한 두 명의 IAM Identity Center 사용자를 선택합니다.



c. 사용자 할당을 선택하여 해당 사용자를 포털에 추가합니다.

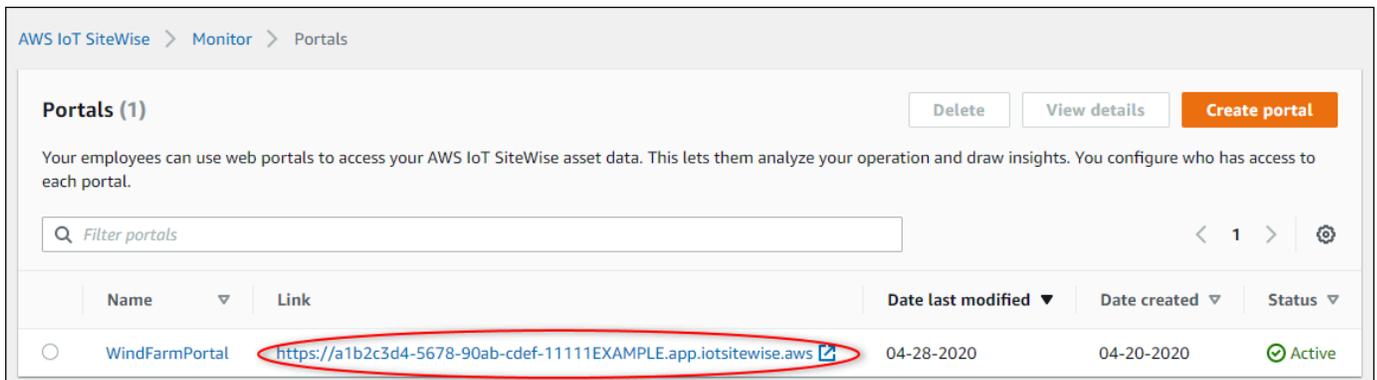
포털 페이지가 열리고 새 포털이 나열됩니다.

2단계: 포털에 로그인

이 절차에서는 포털에 추가한 AWS IAM Identity Center 사용자를 사용하여 새 포털에 로그인합니다.

포털 로그인

1. 포털 페이지에서 새 포털의 링크를 선택하여 새 탭에서 포털을 엽니다.



2. 자습서의 앞부분에서 첫 번째 IAM Identity Center 사용자를 생성한 경우 다음 단계를 따라 사용자의 암호를 생성합니다.
 - a. 이메일을 확인해 제목이 Invitation to join AWS IAM Identity Center인 이메일을 찾습니다.
 - b. 해당 초대 이메일을 열고 Accept invitation을 선택합니다.
 - c. 새 창에서 IAM Identity Center 사용자의 암호를 설정합니다.

이전에 생성한 두 번째 및 세 번째 IAM Identity Center 사용자로 나중에 포털에 로그인하려는 경우 다음 단계를 완료하여 해당 사용자의 암호를 설정할 수도 있습니다.

 Note

이메일을 받지 못한 경우 IAM Identity Center 콘솔에서 사용자의 암호를 생성할 수 있습니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 [사용자 암호 재설정](#)을 참조하십시오.

3. IAM Identity Center Username 및 Password를 입력합니다. 이 자습서의 앞 부분에서 IAM Identity Center 사용자를 생성한 경우 Username은 생성한 포털 관리자 사용자의 이메일 주소입니다.

포털 관리자를 포함한 모든 포털 사용자는 자신의 IAM Identity Center 사용자 자격 증명으로 로그인해야 합니다. 이러한 자격 증명은 일반적으로 AWS Management Console에 로그인하는 데 사용하는 자격 증명과 다릅니다.

4. Sign in을 선택합니다.

포털이 열립니다.

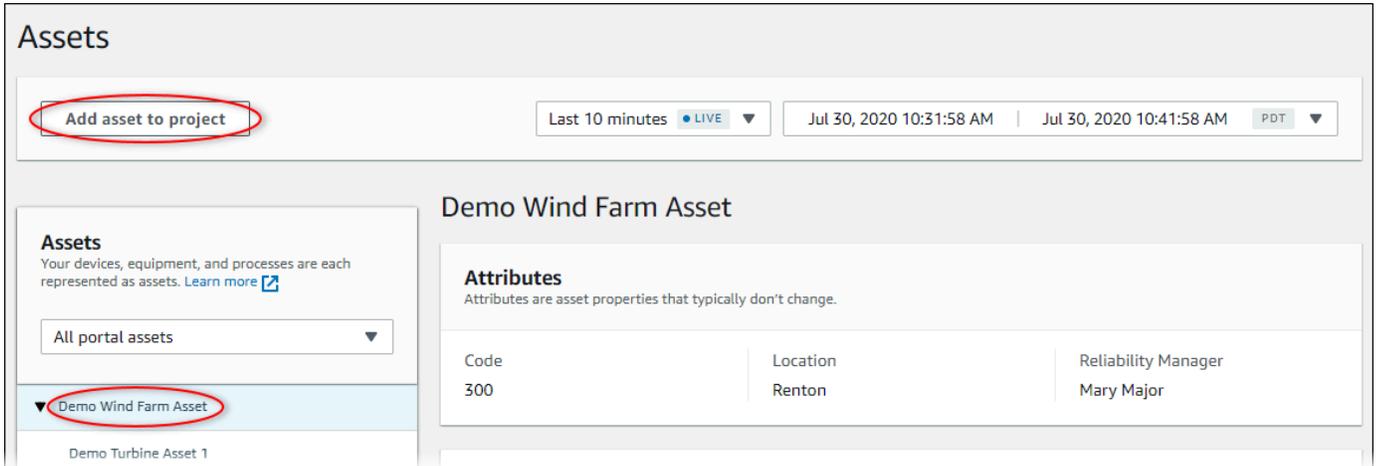
3단계: 풍력 발전 단지 프로젝트 생성

이 절차에서는 포털에 프로젝트를 생성합니다. 프로젝트는 일련의 권한, 자산 및 대시보드를 정의하는 리소스로, 해당 프로젝트의 자산 데이터를 시각화하도록 구성할 수 있습니다. 프로젝트를 통해 각 운영 하위 집합에 액세스할 수 있는 사용자와 해당 운영 하위 집합의 데이터를 시각화하는 방법을 정의합니다. 포털 사용자를 각 프로젝트의 소유자나 뷰어로 할당할 수 있습니다. 프로젝트 소유자는 대시보드를 생성하여 데이터를 시각화하고 다른 사용자와 프로젝트를 공유할 수 있습니다. 프로젝트 뷰어는 대시보드를 볼 수 있지만 편집할 수는 없습니다. SiteWise Monitor의 역할에 대한 자세한 내용은 [참조하십시오. SiteWise 모니터 역할](#)

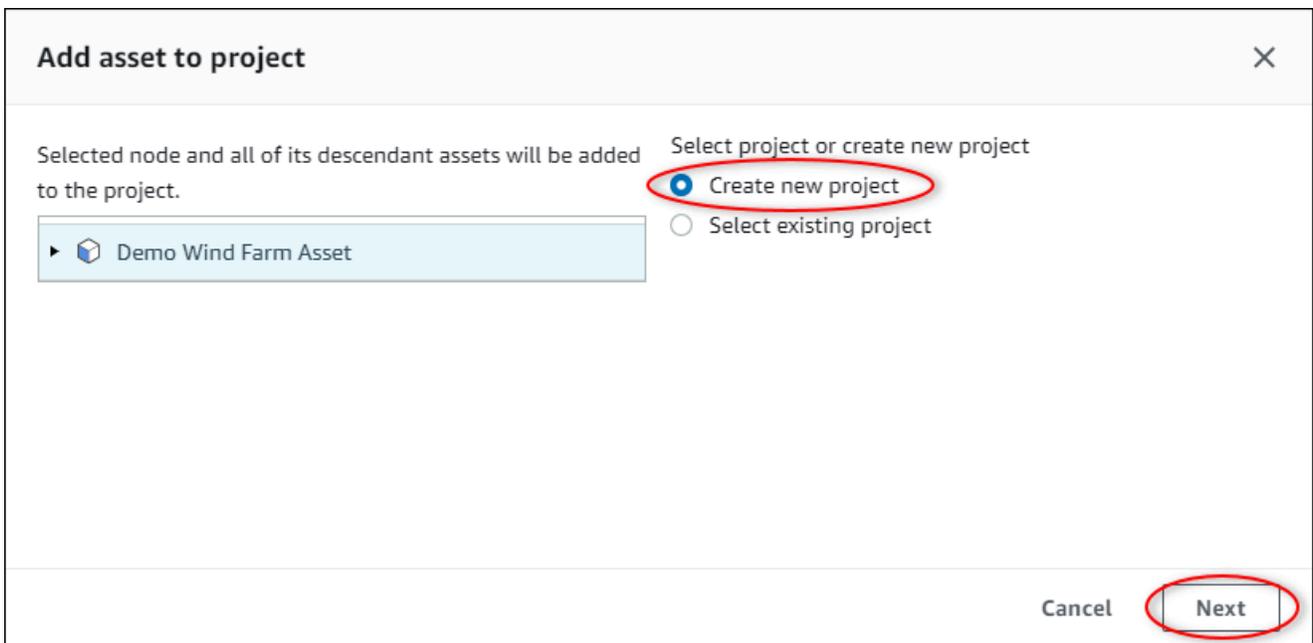
풍력 발전소 프로젝트 생성

1. 포털의 왼쪽 탐색 창에서 자산 탭을 선택합니다. 자산 페이지에서 포털에서 사용 가능한 모든 자산을 탐색하고 프로젝트에 자산을 추가할 수 있습니다.
2. 자산 브라우저에서 Demo Wind Farm Asset을 선택합니다. 자산을 선택하면 해당 자산의 실시간 데이터와 기록 데이터를 탐색할 수 있습니다. 버튼을 Shift 눌러 여러 자산을 선택하고 해당 데이터를 비교할 수도 side-by-side 있습니다.

3. 왼쪽 상단에서 프로젝트에 자산 추가를 선택합니다. 프로젝트에는 포털 사용자가 데이터를 탐색하기 위해 볼 수 있는 대시보드가 포함되어 있습니다. 각 프로젝트는 에서 AWS IoT SiteWise에셋의 일부에 액세스할 수 있습니다. 프로젝트에 자산을 추가하면 해당 프로젝트에 대한 액세스 권한이 있는 모든 사용자가 해당 자산과 하위 자산에 대한 데이터에도 액세스할 수 있습니다.



4. 프로젝트에 자산 추가 대화 상자에서 새 프로젝트 생성을 선택하고 다음을 선택합니다.



5. 새 프로젝트 생성 대화 상자에서 프로젝트의 프로젝트 이름과 프로젝트 설명을 입력한 다음 프로젝트에 자산 추가를 선택합니다.

Create new project ✕

Project name

 The project name can have up to 256 characters.

Project description

 The project description can have up to 2048 characters.

새 프로젝트 페이지가 열립니다.

6. 프로젝트 페이지에서 포털 사용자를 이 프로젝트의 소유자 또는 뷰어로 추가할 수 있습니다.

Note

AWS Organizations 관리 계정에 로그인하지 않은 경우 이 프로젝트에 할당할 포털 사용자가 없을 수 있으므로 이 단계를 건너뛰어도 됩니다.

이 페이지에서 다음을 수행합니다.

- a. 프로젝트 소유자에서 소유자 추가 또는 사용자 편집을 선택합니다.

Project owners Send invitations Remove owners **Edit owners**

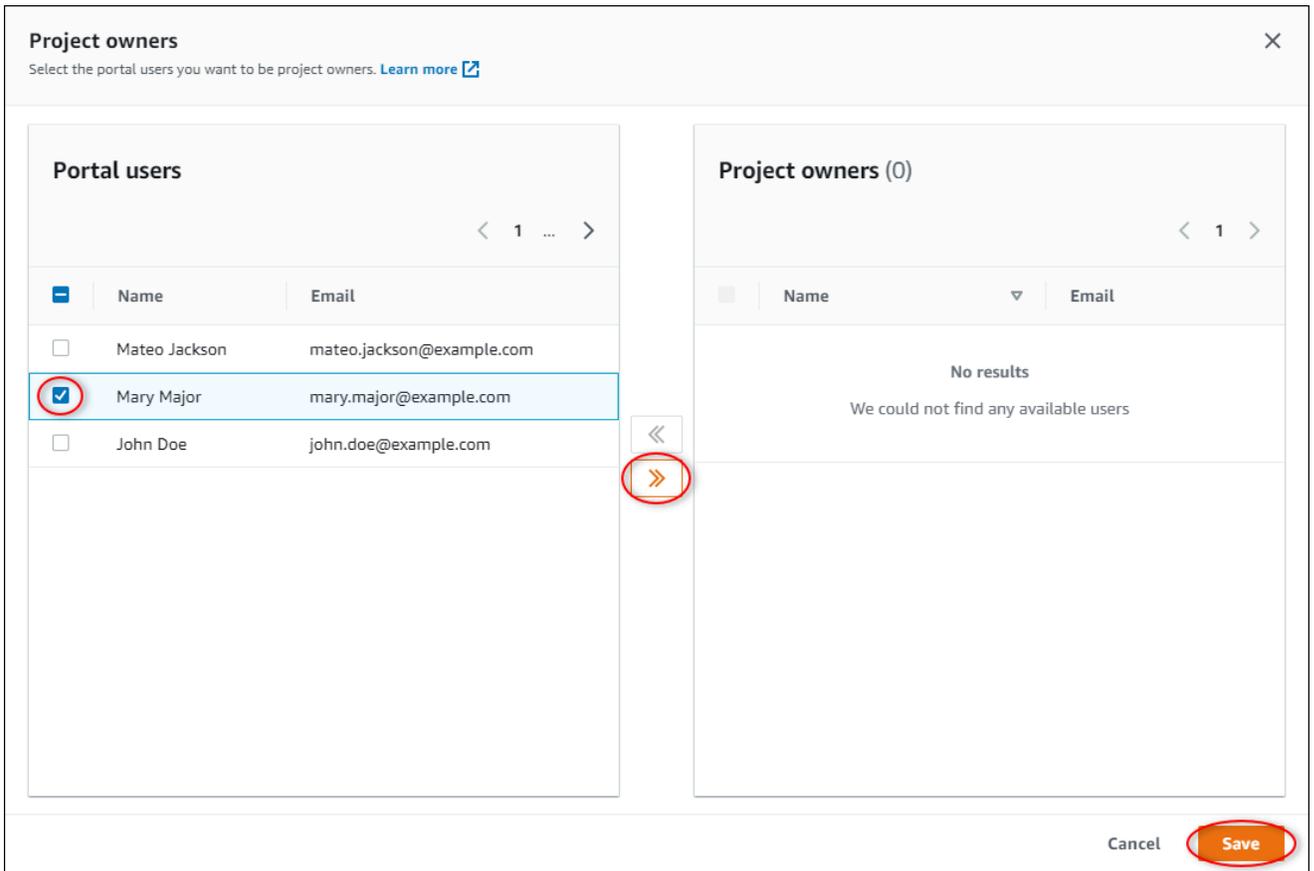
Project owners can create dashboards, view asset data, and invite other users to this project as owners or viewers.

< 1 >

Name	Email
You have not invited any other portal users to own this project. Project owners can modify and update dashboards and project viewers. Learn more	

Add owners

- b. 프로젝트 소유자로 추가할 사용자(예: Mary Major)를 선택한 다음 >> 아이콘을 선택합니다.



- c. 저장을 선택합니다.

IAM Identity Center 사용자 Mary Major는 이 포털에 로그인하여 이 프로젝트의 대시보드를 편집하고 이 프로젝트를 이 포털의 다른 사용자와 공유할 수 있습니다.

- d. 프로젝트 뷰어에서 뷰어 추가 또는 사용자 편집을 선택합니다.
 e. 프로젝트 뷰어로 추가할 사용자(예: Mateo Jackson)를 선택한 다음 >> 아이콘을 선택합니다.
 f. 저장을 선택합니다.

IAM Identity Center 사용자 Mateo Jackson은 이 포털에 로그인하여 풍력 발전소 프로젝트의 대시보드를 볼 수 있지만 편집할 수는 없습니다.

4단계: 대시보드를 만들어 풍력 발전 단지 데이터를 시각화하세요.

이 절차에서는 데모 풍력 발전소 데이터를 시각화하는 대시보드를 생성합니다. 대시보드에는 프로젝트 자산 데이터의 사용자 지정 가능한 시각화가 포함되어 있습니다. 각 시각화는 라인 차트, 막대 차트 또는 KPI (핵심 성과 지표) 디스플레이와 같은 다양한 유형을 가질 수 있습니다. 데이터에 가장 적합한

시각화 유형을 선택할 수 있습니다. 프로젝트 소유자는 대시보드를 편집할 수 있지만 프로젝트 뷰어는 대시보드만 보고 통찰력을 얻을 수 있습니다.

시각화가 포함된 대시보드 생성

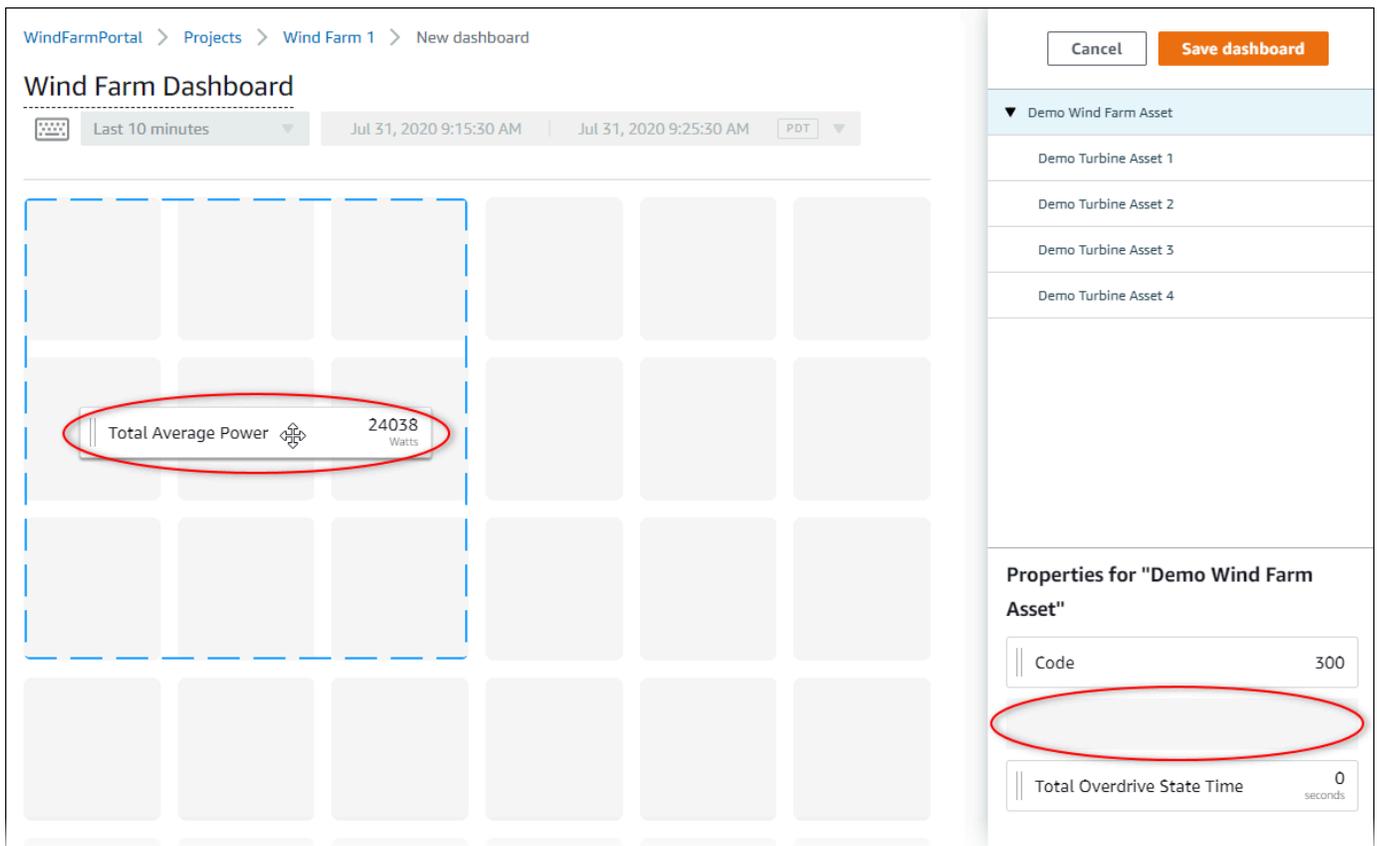
1. 새 프로젝트 페이지에서 대시보드 생성을 선택하여 대시보드를 만들고 편집 페이지를 엽니다.

대시보드의 편집 페이지에서 자산 계층 구조의 자산 속성을 대시보드로 끌어서 시각화를 만들 수 있습니다. 그런 다음 대시보드에서 각 시각화의 제목, 범례 제목, 유형, 크기 및 위치를 편집할 수 있습니다.

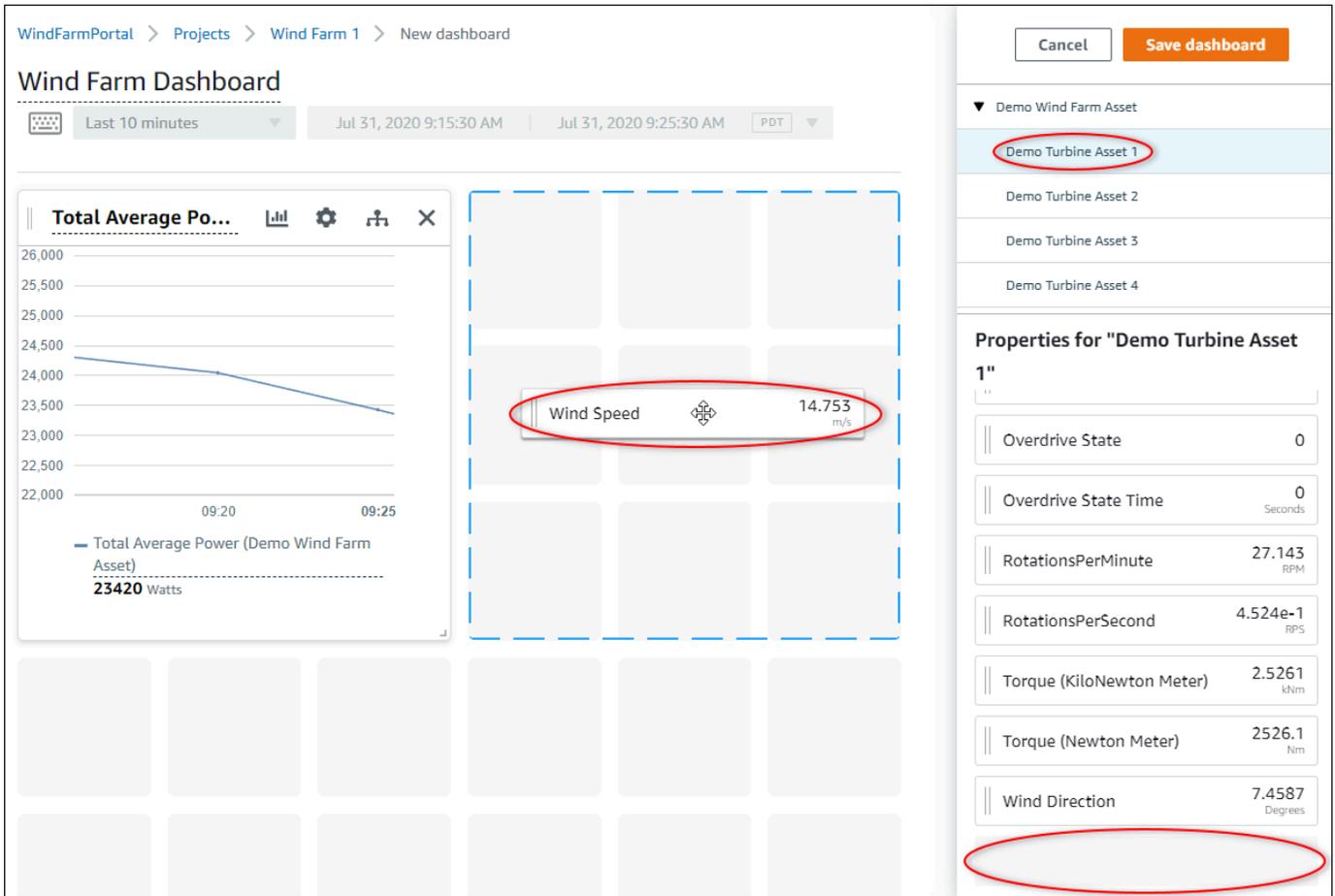
2. 대시보드 이름을 입력합니다.



3. Demo Wind Farm Asset에서 Total Average Power를 대시보드로 드래그하여 시각화를 생성합니다.

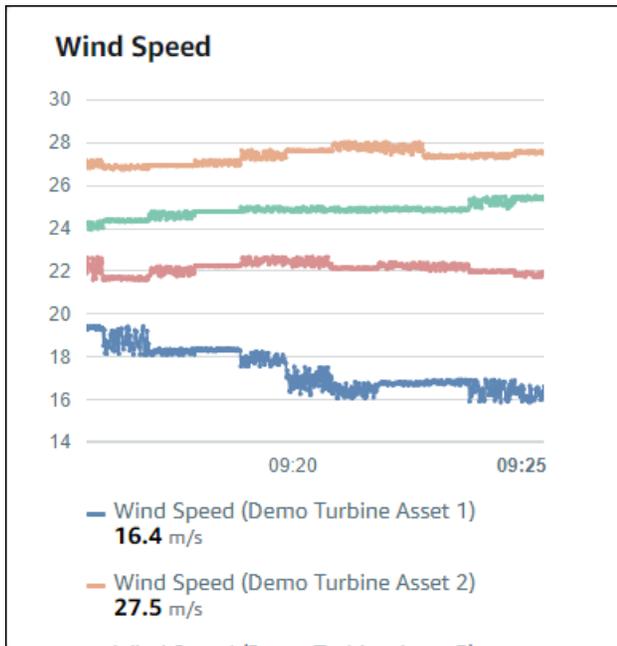


- 해당 자산의 속성을 표시하도록 Demo Turbine Asset 1을 선택한 다음 대시보드로 Wind Speed를 드래그하여 풍속을 시각화할 수 있습니다.

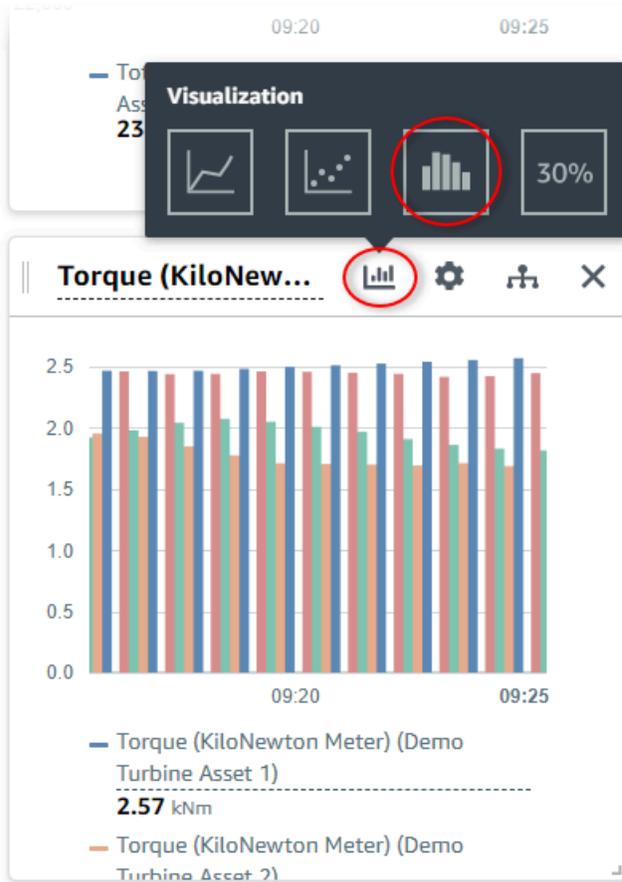


- 각 Demo Turbine Asset 2, 3, 4(순서대로)에 대한 새 풍속 시각화에 Wind Speed를 추가합니다.

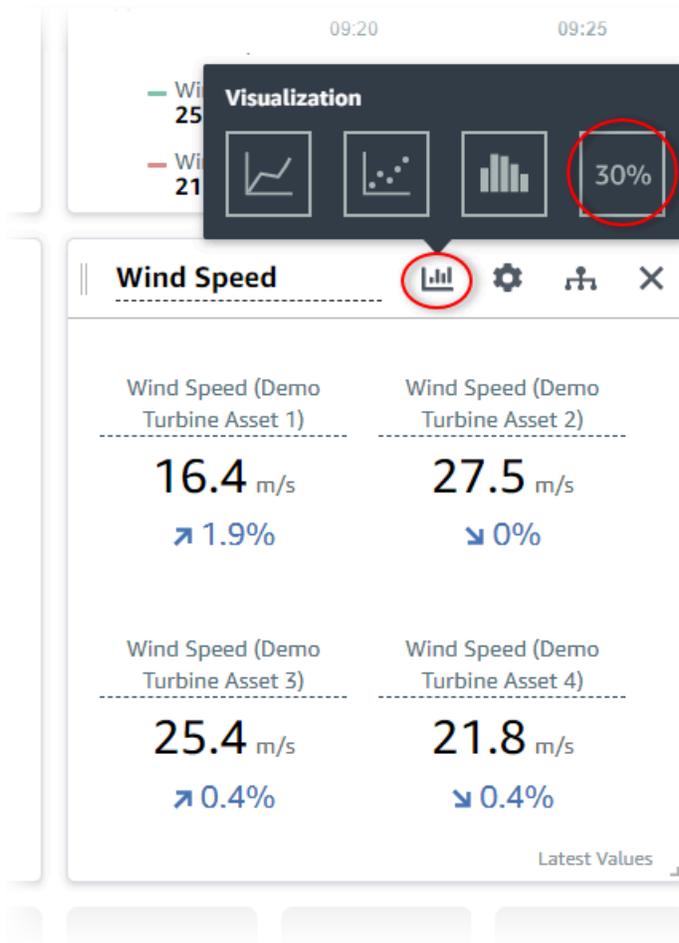
Wind Speed 시각화는 다음 스크린샷과 비슷한 모습이 되어야 합니다.



6. 풍력 터빈의 Torque (KiloNewton Meter) 속성에 대해 단계 4와 단계 5를 반복하여 풍력 터빈 토크에 대한 시각화를 만듭니다.
7. Torque (KiloNewton Meter) 시각화에 대한 시각화 유형 아이콘을 선택한 다음 막대 차트 아이콘을 선택합니다.

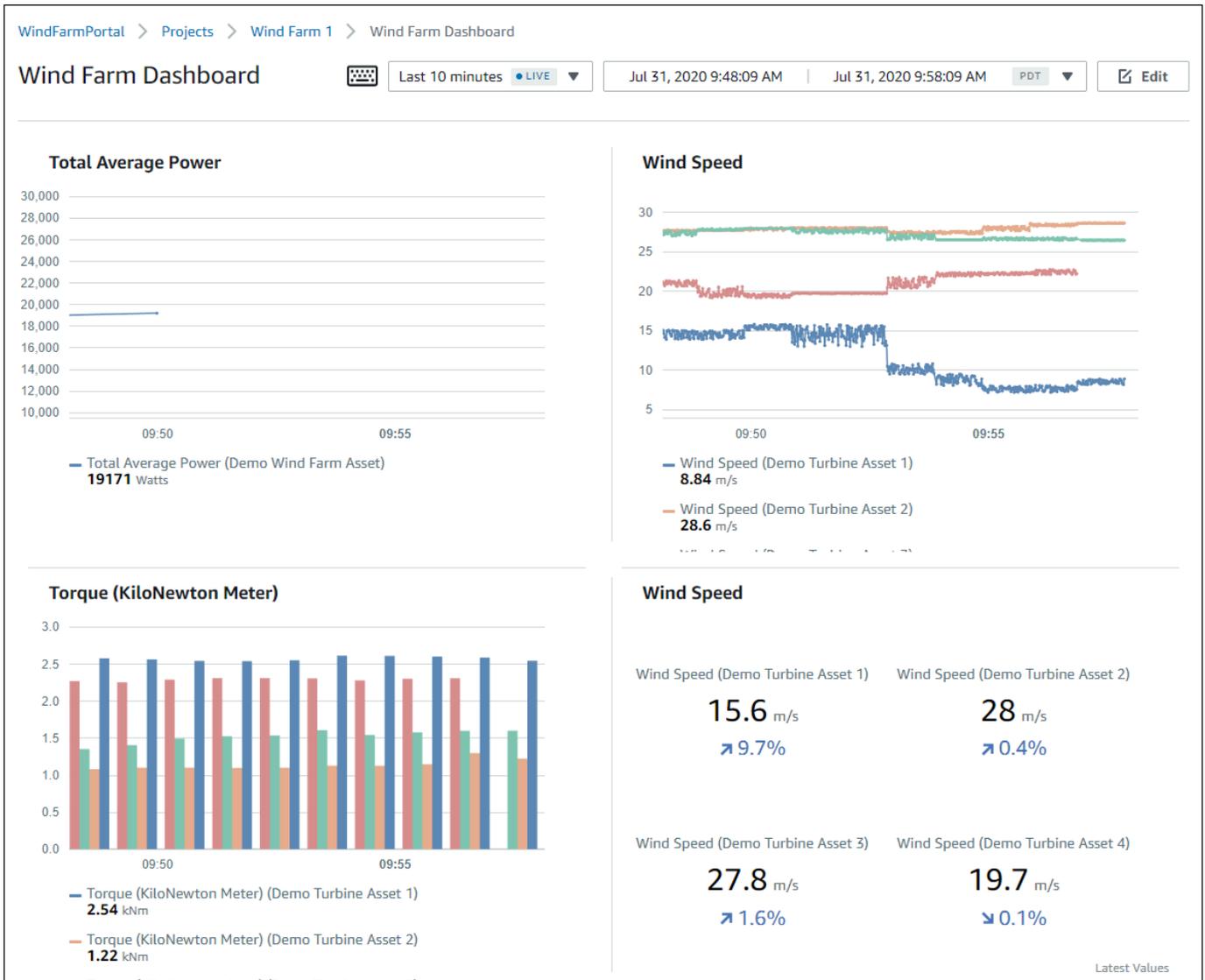


8. 풍력 터빈의 Wind Direction 속성에 대해 단계 4와 단계 5를 반복하여 풍향에 대한 시각화를 생성합니다.
9. Wind Direction 시각화에 대한 시각화 유형 아이콘을 선택한 다음 KPI 차트 아이콘(30%)을 선택합니다.



10. (선택 사항) 필요에 따라 각 시각화의 제목, 범례 제목, 유형, 크기 및 위치를 변경합니다.
11. 오른쪽 상단에 있는 대시보드 저장을 선택하여 대시보드를 저장합니다.

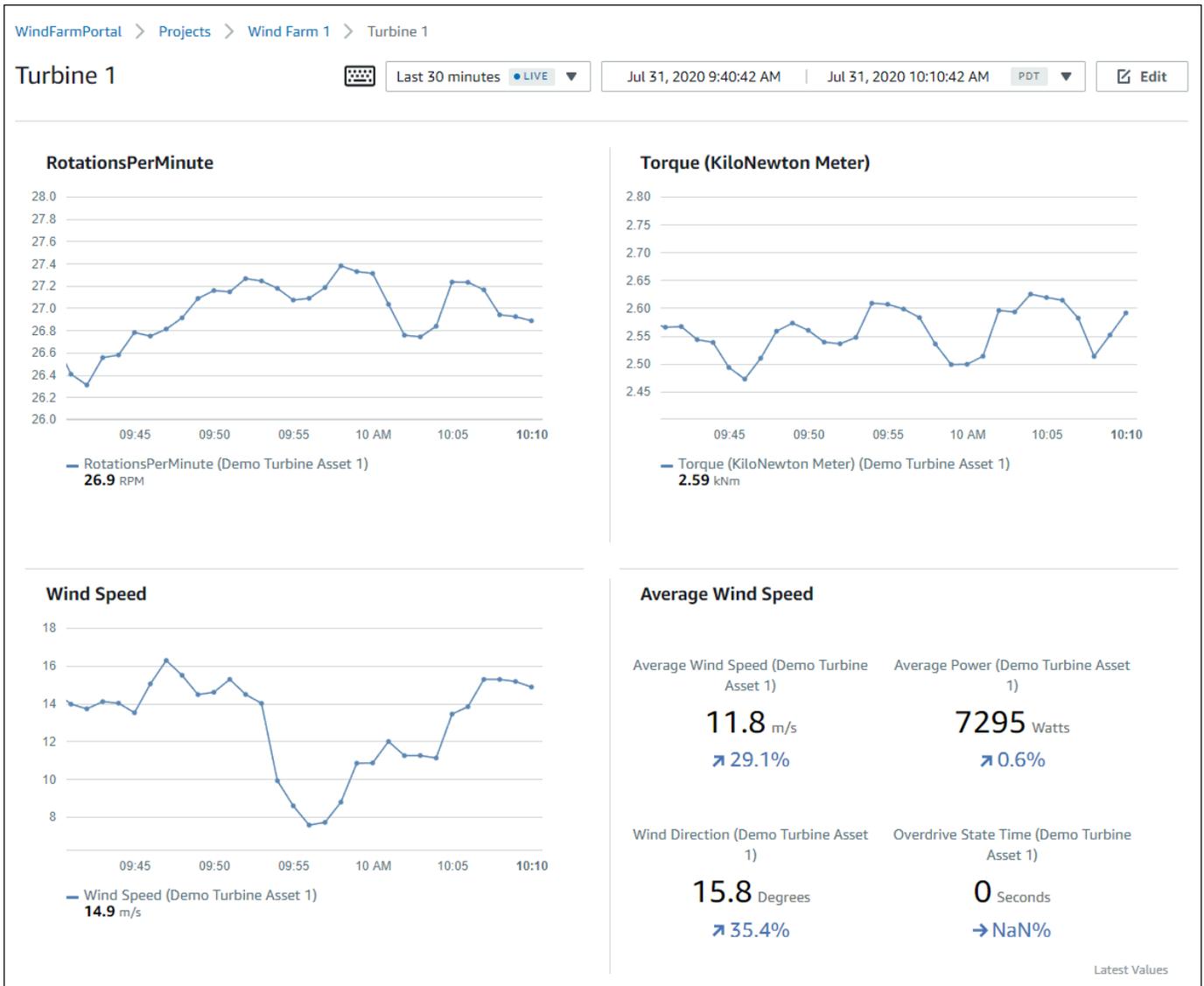
대시보드는 다음 스크린샷과 비슷한 모습이 되어야 합니다.



12. (선택 사항) 각 풍력 터빈 자산에 대해 추가 대시보드를 만듭니다.

프로젝트 뷰어가 각 개별 자산의 문제를 조사할 수 있도록 각 자산에 대한 대시보드를 만드는 것이 좋습니다. 각 시각화에는 최대 5개의 자산만 추가할 수 있으므로 다양한 시나리오에서 계층적 자산에 대한 여러 대시보드를 만들어야 합니다.

데모 풍력 터빈의 대시보드는 다음 스크린샷과 유사하게 보일 수 있습니다.



13. (선택 사항) 타임라인을 변경하거나 시각화에서 데이터 포인트를 선택하여 대시보드의 데이터를 탐색합니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 가이드의 [대시보드 보기](#)를 참조하십시오.

5단계: 포털 둘러보기

이 절차에서는 포털 관리자보다 권한이 적은 사용자로 포털을 탐색할 수 있습니다. AWS IoT SiteWise

포털 탐색 및 자습서 완료

- (선택 사항) 다른 사용자를 프로젝트에 소유자 또는 뷰어로 추가한 경우 해당 사용자로 포털에 로그인할 수 있습니다. 이렇게 하면 포털 관리자보다 권한이 적은 사용자로 포털을 탐색할 수 있습니다.

⚠ Important

포털에 로그인하는 각 사용자에게 대해 요금이 부과됩니다. 자세한 내용은 [AWS IoT SiteWise 요금](#)을 참조하십시오.

다른 사용자와 마찬가지로 포털을 탐색하려면 다음을 수행합니다.

- a. 포털 왼쪽 하단에서 로그아웃을 선택하여 웹 애플리케이션을 종료합니다.
- b. IAM Identity Center 애플리케이션 포털의 오른쪽 상단에서 로그아웃을 선택하여 IAM Identity Center 사용자에서 로그아웃합니다.
- c. 프로젝트 소유자 또는 프로젝트 뷰어로 할당된 IAM Identity Center 사용자로 포털에 로그인합니다. 자세한 내용은 [2단계: 포털에 로그인](#) 단원을 참조하십시오.

이 자습서를 완료했습니다. SiteWise Monitor에서 데모 풍력 발전 단지 탐색을 마치면 다음 절차에 따라 리소스를 정리하세요.

6단계: 자습서 완료 후 리소스 정리

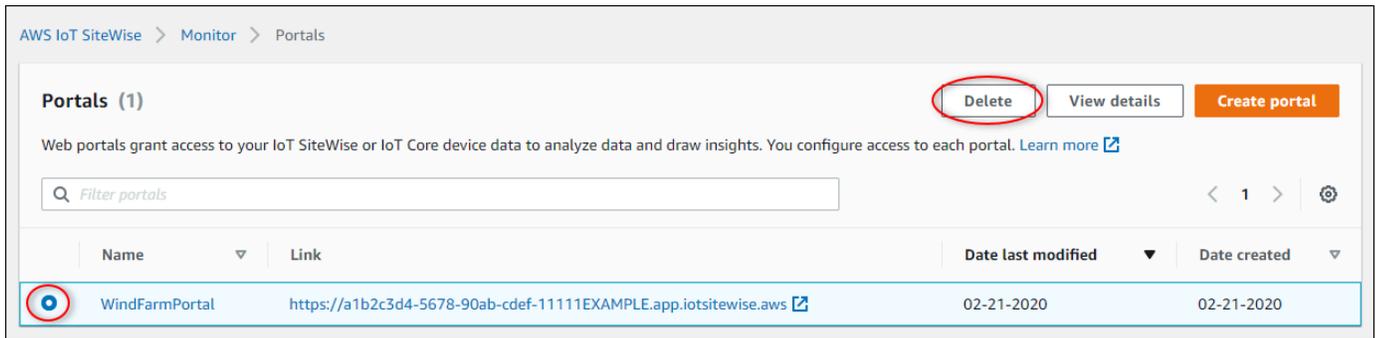
자습서를 완료한 후 리소스를 정리할 수 있습니다. 사용자가 포털에 로그인하지 않으면 AWS IoT SiteWise 에 대한 요금이 청구되지 않지만 포털과 AWS IAM Identity Center 디렉터리 사용자를 삭제할 수 있습니다. 데모 풍력 발전소 자산은 데모를 생성할 때 선택한 기간이 끝날 때 삭제되며, 데모를 수동으로 삭제할 수도 있습니다. 자세한 내용은 [데모 삭제 AWS IoT SiteWise](#) 단원을 참조하십시오.

포털 및 IAM Identity Center 사용자를 삭제하려면 다음 절차를 수행하십시오.

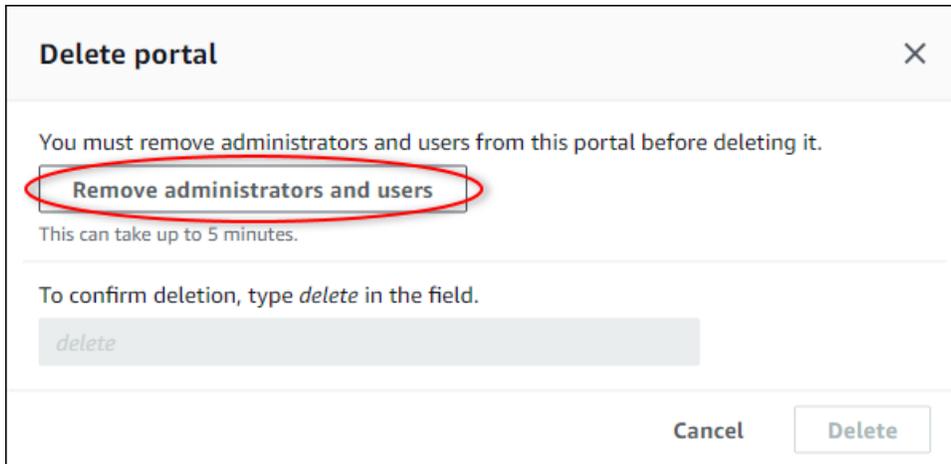
포털 삭제

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 포털을 선택합니다.
3. 포털을 WindFarmPortal선택한 다음 삭제를 선택합니다.

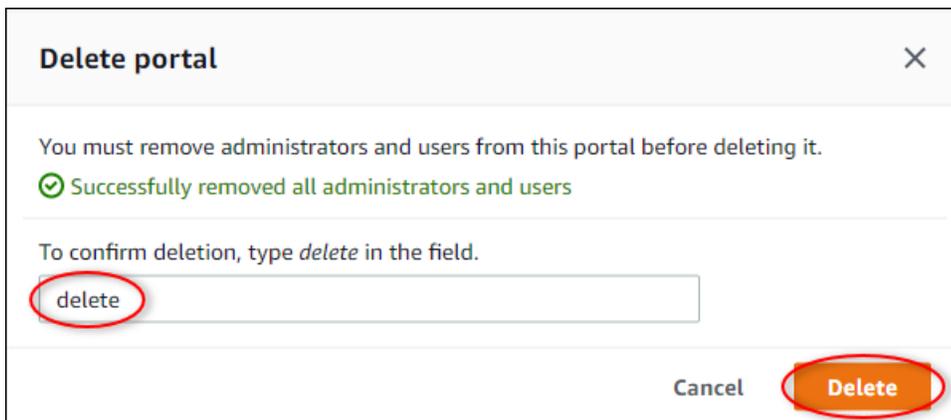
포털이나 프로젝트를 삭제할 때 삭제된 프로젝트와 연결된 자산은 영향을 받지 않습니다.



4. 포털 삭제 대화 상자에서 관리자 및 사용자 제거를 선택합니다.

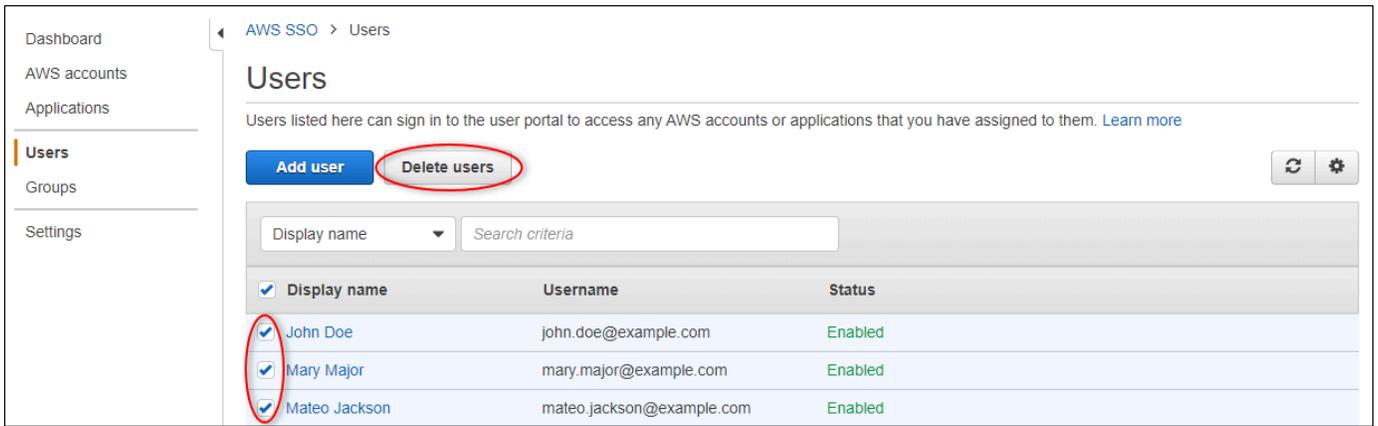


5. **delete**를 입력하여 삭제를 확인한 다음 삭제를 선택합니다.

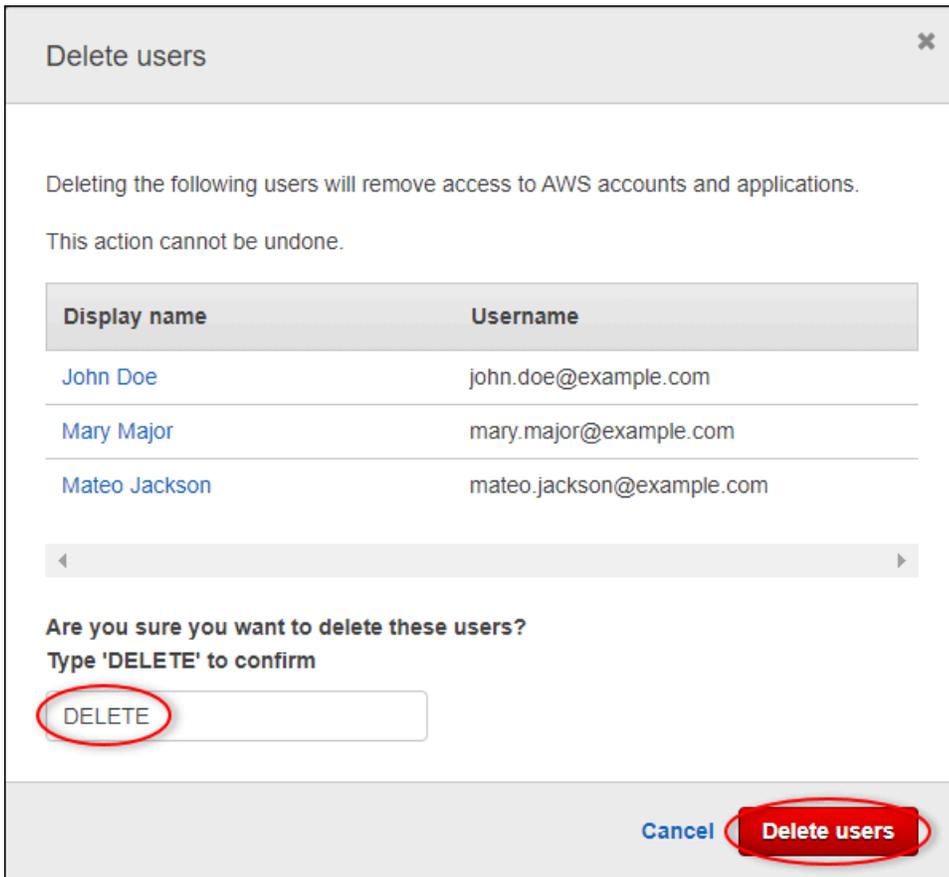


IAM Identity Center 사용자 삭제

1. [IAM Identity Center 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 사용자를 선택합니다.
3. 삭제할 각 사용자의 확인란을 선택한 다음 사용자 삭제를 선택합니다.



4. 사용자 삭제 대화 상자에 **DELETE**를 입력한 후 사용자 삭제를 선택합니다.



Amazon DynamoDB에 속성 값 업데이트 게시

이 자습서에서는 [Amazon DynamoDB](#)를 사용하여 데이터를 저장하는 편리한 방법을 소개합니다. 이를 통해 API를 반복적으로 쿼리하지 않고도 이전 자산 데이터에 쉽게 액세스할 수 있습니다. AWS IoT SiteWise 이 자습서를 완료한 후에는 전체 풍력 발전소의 풍속 및 풍향을 보여주는 실시간 지도와 같이 자산 데이터를 사용하는 사용자 지정 소프트웨어를 만들 수 있습니다. 맞춤형 소프트웨어 솔루션을

구현하지 않고 데이터를 모니터링하고 시각화하려면 [을 참조하십시오. 를 사용하여 데이터 모니터링 AWS IoT SiteWise Monitor](#)

이 자습서에서는 풍력 발전 단지에 대한 샘플 데이터 세트를 제공하는 AWS IoT SiteWise 데모를 기반으로 합니다. AWS IoT Core 규칙을 통해 데이터를 생성한 DynamoDB 테이블로 보내도록 풍력 발전소 데모에서 속성 값 업데이트를 구성합니다. 속성 값 업데이트를 활성화하면 MQTT 메시지로 AWS IoT Core 데이터가 AWS IoT SiteWise 전송됩니다. 그런 다음 해당 메시지의 내용에 따라 DynamoDB 작업과 같은 작업을 수행하는 AWS IoT 핵심 규칙을 정의합니다. 자세한 정보는 [다른 AWS 서비스와의 상호작용](#)을 참조하세요.

주제

- [사전 조건](#)
- [1단계: 속성 값 업데이트를 AWS IoT SiteWise 게시하도록 구성](#)
- [2단계: AWS IoT Core에서 규칙 만들기](#)
- [3단계: DynamoDB 테이블 생성](#)
- [4단계: DynamoDB 규칙 작업 구성](#)
- [5단계: DynamoDB에서 데이터 탐색](#)
- [6단계: 자습서 완료 후 리소스 정리](#)

사전 조건

이 튜토리얼을 완료하려면 다음이 필요합니다.

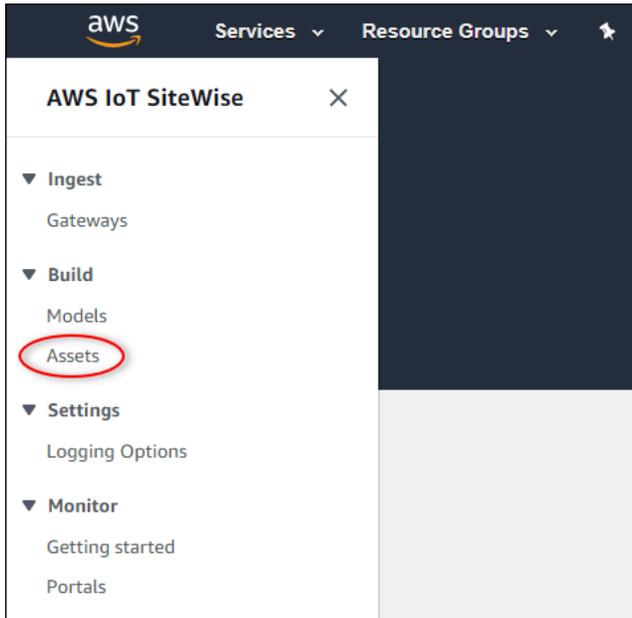
- 계정. AWS 계정이 없는 경우 [설정 AWS 계정](#) 단원을 참조하십시오.
- Windows, macOS, Linux 또는 Unix를 실행하여 액세스할 수 있는 개발 컴퓨터입니다. AWS Management Console 자세한 내용은 [AWS Management Console 시작하기](#)를 참조하십시오.
- 관리자 권한이 있는 IAM 사용자.
- 실행 중인 AWS IoT SiteWise 풍력 발전 단지 데모. 데모를 설정하면 모델과 자산을 정의하고 데이터를 스트리밍하여 풍력 발전 단지를 표현합니다. AWS IoT SiteWise 자세한 정보는 [AWS IoT SiteWise 데모 사용](#)을 참조하세요.

1단계: 속성 값 업데이트를 AWS IoT SiteWise 게시하도록 구성

이 절차에서는 데모 터빈 자산의 Wind Speed 속성에 대한 속성 값 알림을 활성화합니다. 속성 값 알림을 활성화하면 MQTT 메시지의 각 값 업데이트를 Core에 AWS IoT SiteWise 게시합니다. AWS IoT

자산 속성에 대한 속성 값 업데이트 알림 활성화

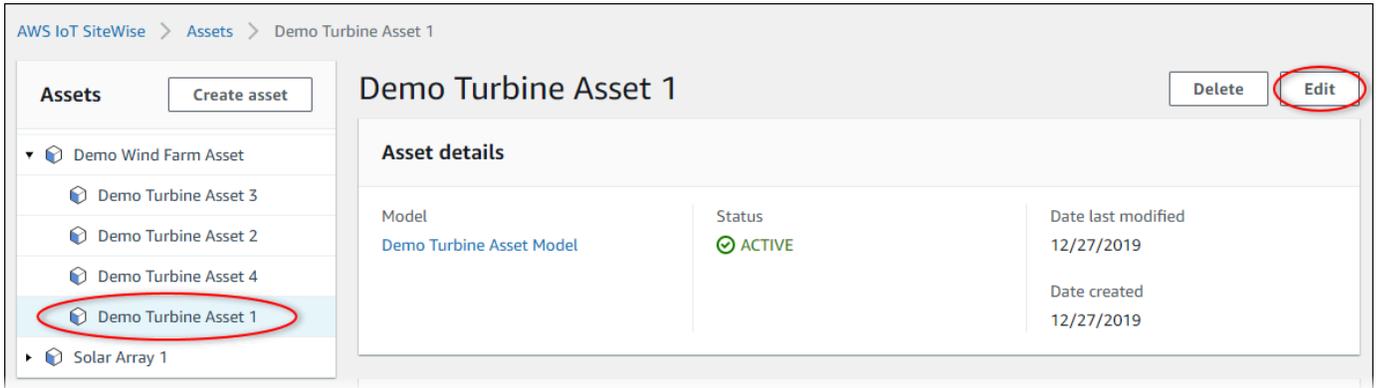
1. [AWS IoT SiteWise 콘솔](#)에 로그인합니다.
2. 지원되는 [AWS IoT SiteWise 엔드포인트와 할당량](#)을 검토하고 필요한 AWS IoT SiteWise 경우 지역을 전환하십시오 AWS . 데모를 실행 중인 지역으로 전환하십시오. AWS IoT SiteWise
3. 왼쪽 탐색 창에서 자산을 선택합니다.



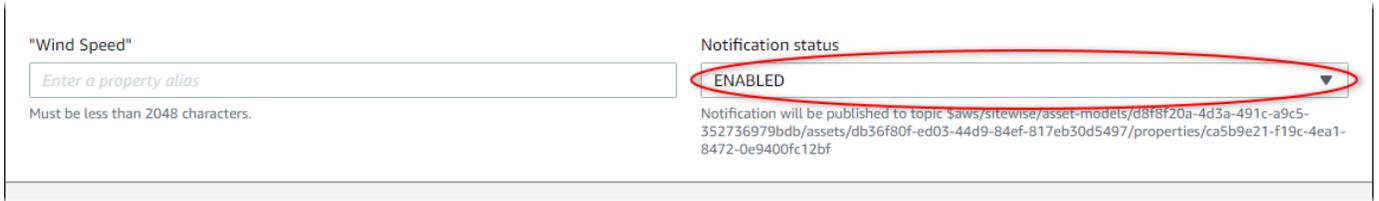
4. Demo Wind Farm Asset 옆에 있는 화살표를 선택하여 풍력 발전소 자산의 계층 구조를 확장합니다.



5. 데모 터빈을 선택하고 편집을 선택합니다.



6. Wind Speed 속성의 알림 상태를 활성화로 업데이트합니다.



7. 페이지 하단에서 자산 저장을 선택합니다.
8. 각 데모 터빈 자산에 대해 단계 5~7을 반복합니다.
9. 데모 터빈(예: Demo Turbine Asset 1)을 선택합니다.
10. 측정을 선택합니다.
11. Wind Speed 속성 옆에 있는 복사 아이콘을 선택하여 알림 주제를 클립보드로 복사합니다. 이 자습서의 뒷부분에서 사용하도록 알림 주제를 저장합니다. 하나의 터빈에서만 알림 주제를 기록하면 됩니다.

Torque (KiloNewton Meter)	-	⊖ Disabled	-	2.128123
Wind Speed	-	✔ Enabled	\$aws/sitewise/asset-models/d8f8f...	26.49812

알림 주제는 다음 예시와 같은 형식이어야 합니다.

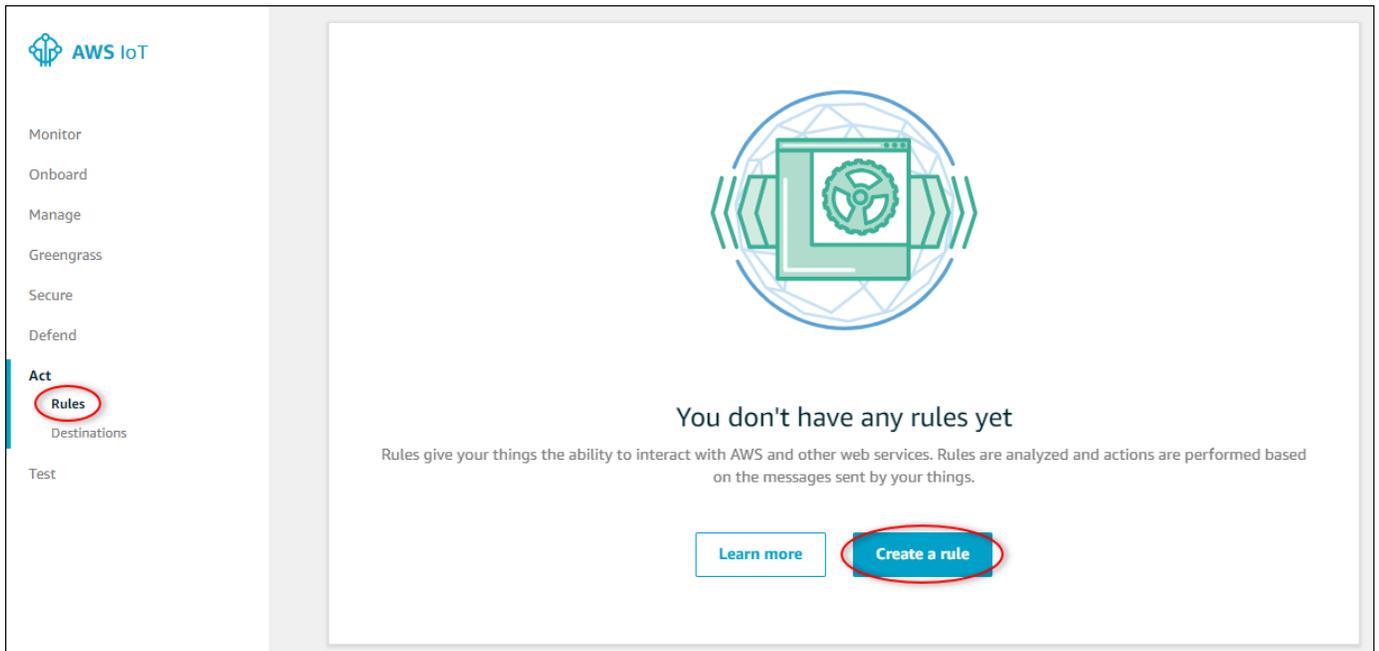
```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

2단계: AWS IoT Core에서 규칙 만들기

이 절차에서는 속성 값 알림 메시지를 파싱하고 Amazon DynamoDB 테이블에 데이터를 삽입하는 규칙을 AWS IoT Core에서 생성합니다. AWS IoT 핵심 규칙은 MQTT 메시지를 파싱하고 각 메시지의 내용 및 주제에 따라 작업을 수행합니다. 그런 다음 이 자습서의 일부로 생성한 DynamoDB 테이블에 데이터를 삽입하는 DynamoDB 작업이 포함된 규칙을 생성합니다.

DynamoDB 작업을 사용하여 규칙 생성

1. [AWS IoT 콘솔](#)로 이동합니다. 시작하기 버튼이 나타나면 선택합니다.
2. 왼쪽 탐색 창에서 작업을 선택한 다음 규칙을 선택합니다.



3. 아직 규칙이 없습니다 대화 상자가 나타나면 규칙 생성을 선택합니다. 그렇지 않은 경우 생성을 선택합니다.
4. 규칙에 대해 이름과 설명을 입력하십시오.

Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

WindSpeedRule

Description

A DynamoDBv2 rule that records wind data from wind turbine assets in AWS IoT SiteWise.

5. 이 자습서의 앞부분에서 저장한 알림 주제를 찾습니다.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

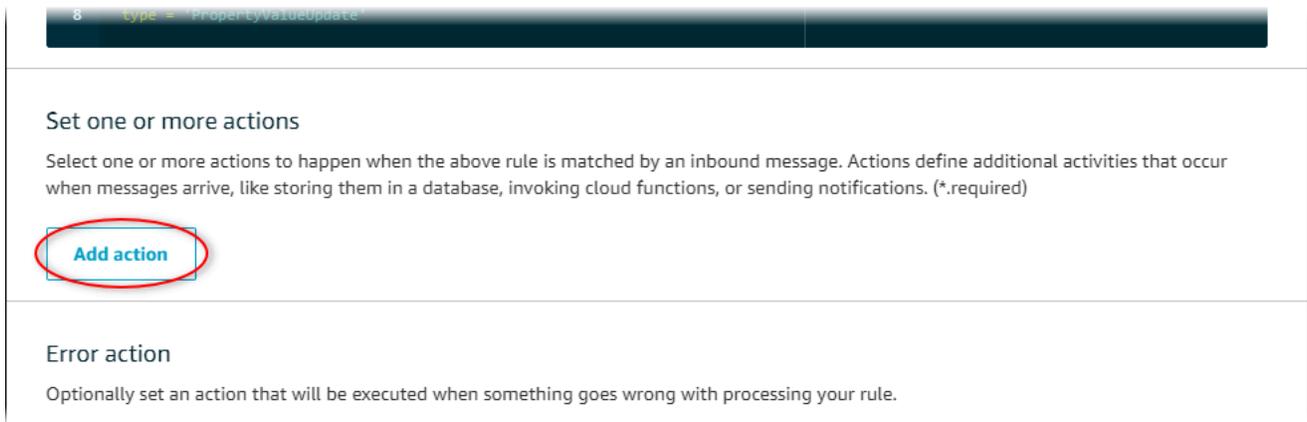
주제의 자산 ID (위의 IDassets/) 를 a로 바꾸십시오+. 그러면 모든 데모 풍력 터빈 에셋의 풍속 속성이 선택됩니다. + 주제 필터는 해당 주제에 포함된 단일 수준의 모든 노드를 수용합니다. 주제는 다음 예와 같아야 합니다.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

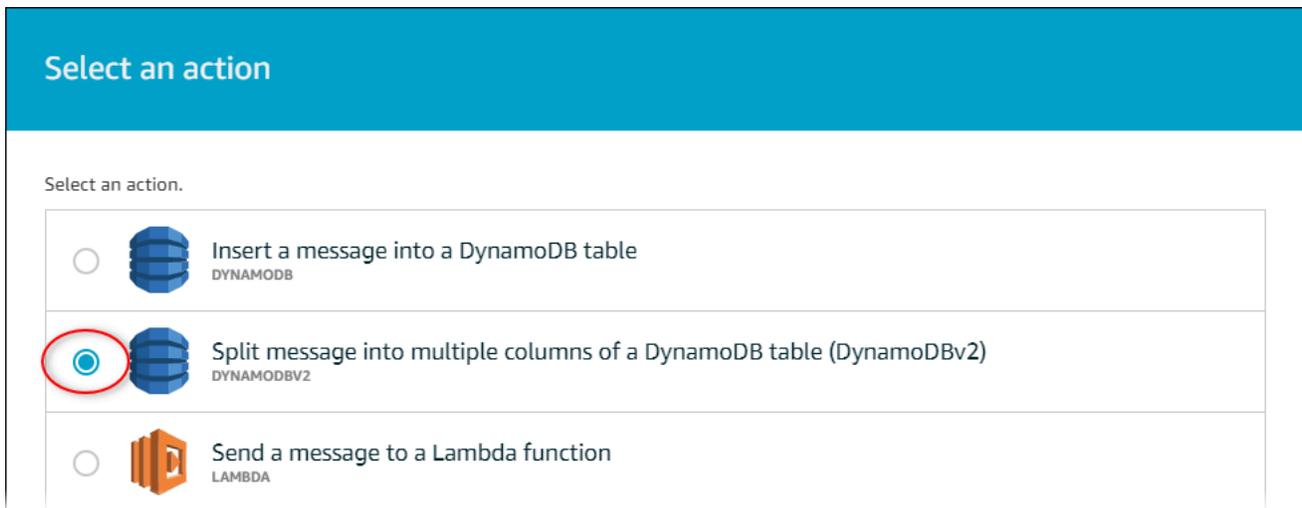
6. 다음 규칙 쿼리 문을 입력합니다. FROM 섹션의 주제를 해당 알림 주제로 바꿉니다.

```
SELECT
  payload.assetId AS asset,
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed,
  timestamp() AS timestamp
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

7. 하나 이상의 작업 설정에서 작업 추가를 선택합니다.



8. 작업 선택 페이지에서 메시지를 DynamoDB 테이블(DynamoDBv2)의 여러 열로 분할을 선택합니다.



9. 페이지 하단에서 작업 구성을 선택합니다.
10. 작업 구성 페이지에서 [새 리소스 생성]을 선택합니다.

DynamoDB 콘솔이 새 탭에서 열립니다. 다음 절차를 완료하는 동안 규칙 작업 탭을 열어 둡니다.

3단계: DynamoDB 테이블 생성

이 절차에서는 Amazon DynamoDB 테이블을 생성하여 규칙 작업으로부터 풍속 데이터를 수신합니다.

DynamoDB 테이블 생성

1. DynamoDB 콘솔 대시보드에서 테이블 생성을 선택합니다.
2. 테이블의 이름을 입력합니다.

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key

ⓘ

Add sort key

ⓘ

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

ⓘ You do not have the required role to enable Auto Scaling by default. Please refer to [documentation](#).

+ Add tags NEW!

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel
Create

3. 기본 키에서 다음 작업을 수행합니다.

- a. 파티션 키로 **timestamp**를 입력합니다.
- b. 번호 유형을 선택합니다.
- c. 정렬 키 추가 확인란을 선택합니다.
- d. 정렬 키로 **asset**을 입력하고 기본 정렬 키 유형인 문자열을 그대로 둡니다.

4. 생성을 선택하세요.

테이블 생성 중 알림이 사라지면 테이블이 준비된 것입니다.

5. 작업 구성 페이지가 있는 탭으로 돌아갑니다. 다음 절차를 완료하는 동안 DynamoDB 탭을 열어 둡니다.

4단계: DynamoDB 규칙 작업 구성

이 절차에서는 속성 값 업데이트의 데이터를 새 DynamoDB 테이블에 삽입하도록 Amazon DynamoDB 규칙 작업을 구성합니다.

DynamoDB 규칙 작업 구성

1. 작업 구성 페이지에서 테이블 이름 목록을 새로 고치고 새 DynamoDB 테이블을 선택합니다.

2. 역할 생성을 선택하여 규칙 작업을 수행할 수 있는 AWS IoT 코어 액세스 권한을 부여하는 IAM 역할을 생성합니다.
3. 역할 이름을 입력하고 역할 생성을 선택합니다.

4. 작업 추가를 선택합니다.
5. 페이지 하단에서 규칙 생성을 선택하여 규칙 생성을 완료합니다.

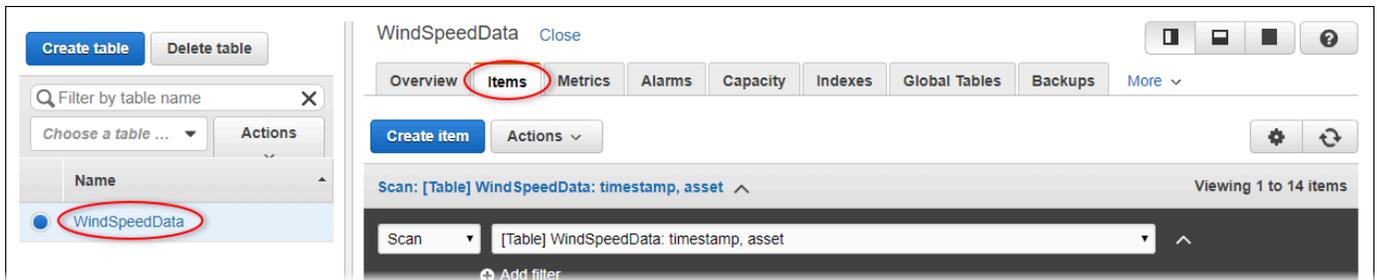
데모 자산 데이터가 DynamoDB 테이블에 나타나기 시작합니다.

5단계: DynamoDB에서 데이터 탐색

이 절차에서는 새 Amazon DynamoDB 테이블에서 데모 자산의 풍속 데이터를 살펴봅니다.

DynamoDB에서 자산 데이터 탐색

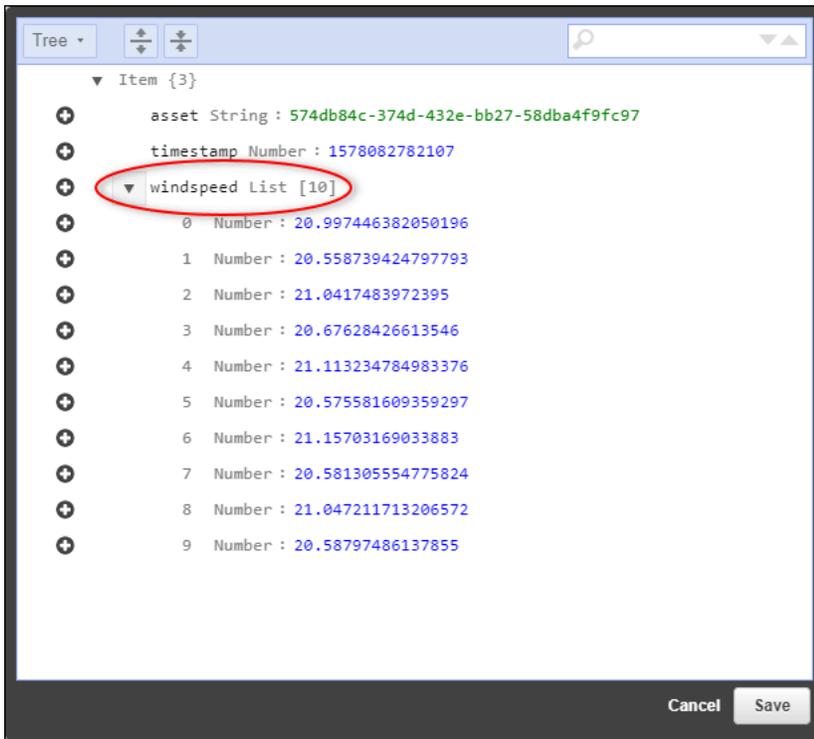
1. DynamoDB 테이블이 열려 있는 탭으로 돌아갑니다.
2. 앞에서 만든 테이블에서 항목 탭을 선택하여 테이블의 데이터를 봅니다. 테이블에 행이 표시되지 않으면 페이지를 새로 고칩니다. 몇 분 후에도 행이 나타나지 않는 경우 [규칙 문제 해결](#) 단원을 참조하십시오.



3. 테이블의 행에서 편집 아이콘을 선택하여 데이터를 확장합니다.

timestamp	asset	windspeed
1578093637414	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.18707553698584"}, {"N": "40.20834808480326"}, {"N": "40.21876939941959"}]
1578093637422	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.21081344172715"}, {"N": "40.218280888809424"}, {"N": "40.22691091326525"}]
1578093637451	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.218912043562895"}, {"N": "40.22691091326525"}, {"N": "40.22876939941959"}]
1578093637453	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.22876939941959"}, {"N": "40.21820505495924"}, {"N": "40.21876939941959"}]

4. windspeed 구조 옆의 화살표를 선택하여 풍속 데이터 포인트 목록을 확장합니다. 각 목록에는 풍력 발전 단지 데모에서 전송한 풍속 데이터 포인트 일괄이 반영되어 있습니다. AWS IoT SiteWise 고유의 용도의 규칙 작업을 설정하는 경우 다른 데이터 형식을 사용할 수 있습니다. 자세한 내용은 [자산 속성 알림 메시지 쿼리](#) 단원을 참조하십시오.



자습서를 완료했으니 규칙을 비활성화하거나 삭제하고 DynamoDB 테이블을 삭제하여 추가 요금이 발생하지 않도록 하십시오. 리소스를 정리하려면 [여기](#)를 참조하십시오. [6단계: 자습서 완료 후 리소스 정리](#)

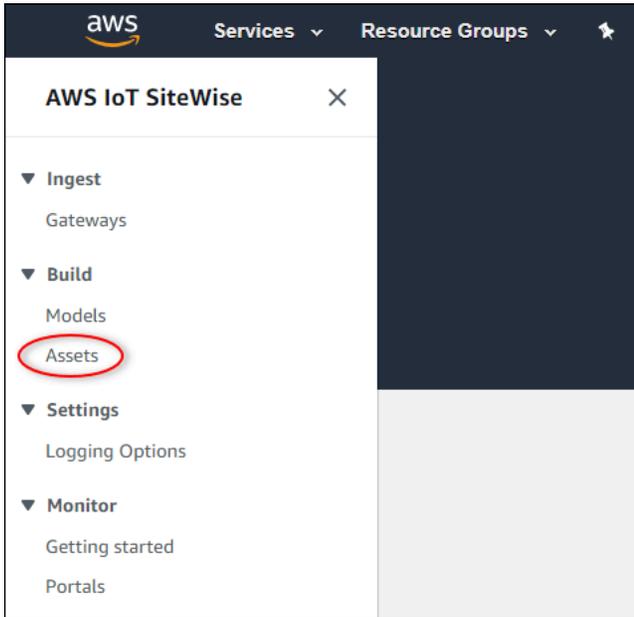
6단계: 자습서 완료 후 리소스 정리

자습서를 완료한 후 추가 요금이 발생하지 않도록 리소스를 정리합니다. 데모 풍력 발전 단지 자산은 데모를 생성할 때 선택한 기간이 끝나면 삭제됩니다. 데모를 수동으로 삭제할 수도 있습니다. 자세한 정보는 [데모 삭제 AWS IoT SiteWise](#)를 참조하세요.

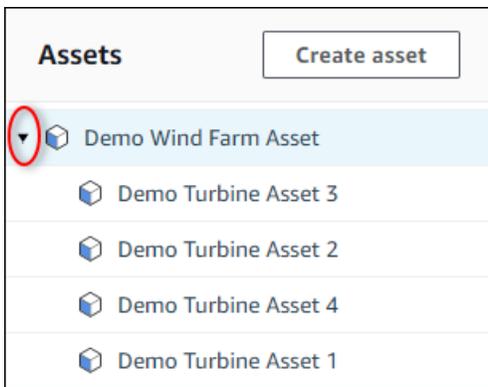
다음 절차를 사용하여 속성 값 업데이트 알림을 비활성화하고 (데모를 삭제하지 않은 경우), AWS IoT 규칙을 비활성화 또는 삭제하고, DynamoDB 테이블을 삭제합니다.

자산 속성에 대한 속성 값 업데이트 알림 비활성화

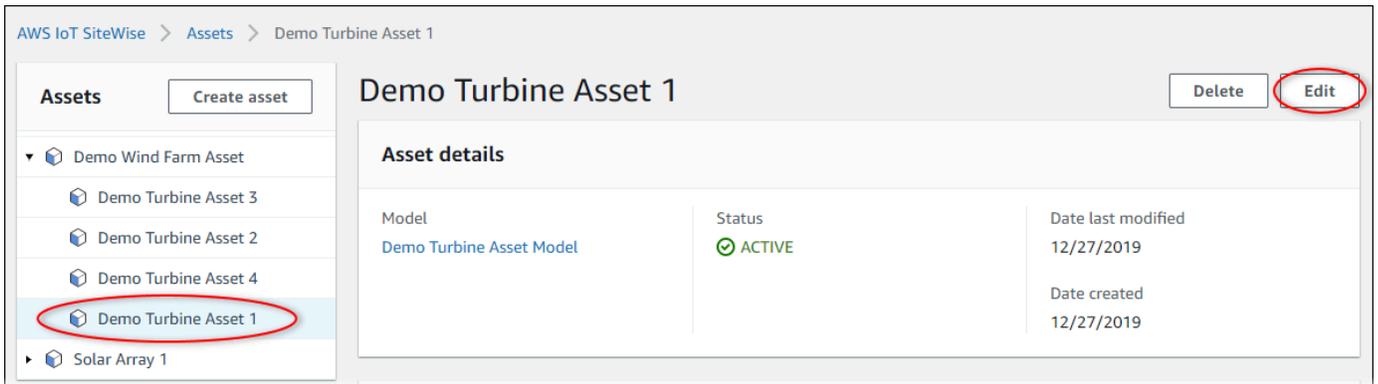
1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 자산을 선택합니다.



3. Demo Wind Farm Asset 옆에 있는 화살표를 선택하여 풍력 발전소 자산의 계층 구조를 확장합니다.



4. 데모 터빈을 선택하고 편집을 선택합니다.



5. Wind Speed 속성의 알림 상태를 비활성으로 업데이트합니다.

"Wind Speed"

Enter a property alias

Must be less than 2048 characters.

Notification status

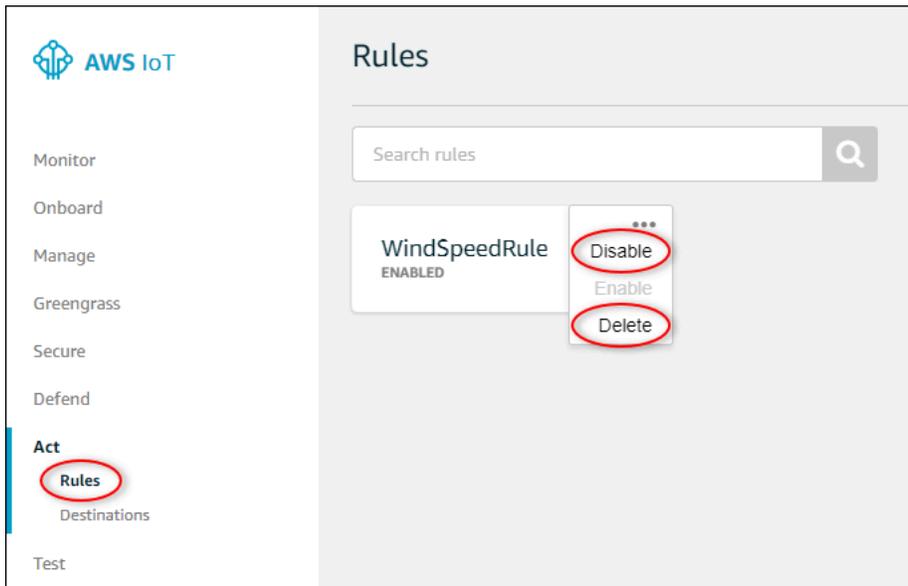
DISABLED

Notification will be published to topic \$aws/sitewise/asset-models/d8f8f20a-4d3a-491c-a9c5-352736979bdb/assets/db36f80f-ed03-44d9-84ef-817eb30d5497/properties/ca5b9e21-f19c-4ea1-8472-0e9400fc12bf

6. 페이지 하단에서 자산 저장을 선택합니다.
7. 각 데모 터빈 자산에 대해 4~6단계를 반복합니다.

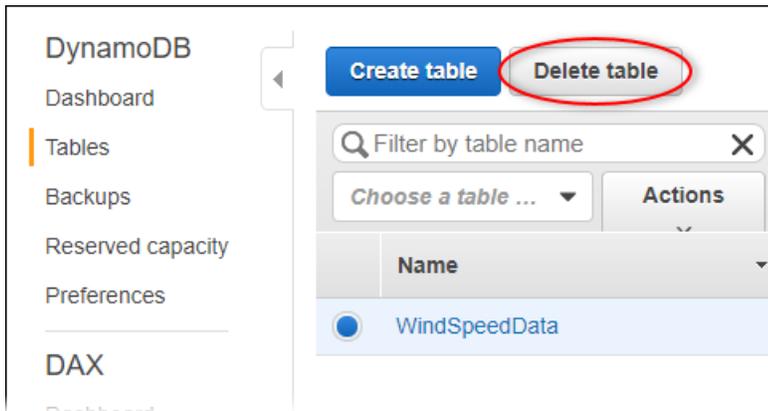
에서 규칙을 비활성화하거나 삭제하려면 AWS IoT Core

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 작업을 선택한 다음 규칙을 선택합니다.
3. 규칙에서 메뉴를 선택하고 비활성화 또는 삭제를 선택합니다.

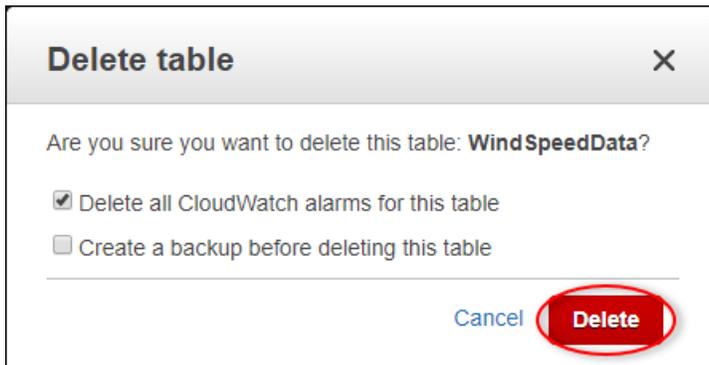


DynamoDB 테이블 삭제

1. [DynamoDB 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 테이블을 선택합니다.
3. 앞에서 생성한 테이블인 WindSpeedData를 선택합니다.
4. 테이블 삭제를 선택합니다.



5. 테이블 삭제 대화 상자에서 삭제를 선택합니다.



에 데이터 수집 AWS IoT SiteWise

AWS IoT SiteWise 산업 데이터를 효율적으로 수집하고 해당 자산과 상호 연관시키도록 설계되어 산업 운영의 다양한 측면을 나타냅니다. 이 문서는 데이터 수집의 실제 측면에 초점을 맞추어 다양한 산업 사용 사례에 AWS IoT SiteWise 맞는 다양한 방법을 제공합니다. 가상 산업 조직 구축에 대한 지침은 [산업 자산 모델링](#) 단원을 참조하십시오.

다음 옵션 중 하나를 AWS IoT SiteWise 사용하여 산업 데이터를 에 보낼 수 있습니다.

- AWS IoT SiteWise Edge - [SiteWise Edge 게이트웨이](#)를 데이터 서버 AWS IoT SiteWise 간의 중개자로 사용합니다. AWS IoT SiteWise SiteWise Edge 게이트웨이를 AWS IoT Greengrass 설정하기 위해 실행할 수 있는 모든 플랫폼에 배포할 수 있는 AWS IoT Greengrass 구성 요소를 제공합니다. 이 옵션은 [OPC-UA 서버 프로토콜과의 연결](#)을 지원합니다.
- AWS IoT SiteWise API - API를 [사용하여 다른 소스의 AWS IoT SiteWise 데이터를 업로드합니다](#). 스트리밍 [BatchPutAssetPropertyValue](#) API를 사용하여 몇 초 내에 수집하거나 배치 지향 [CreateBulkImportJob](#) API를 사용하여 대규모 일괄 처리를 비용 효율적으로 수행할 수 있습니다.
- AWS IoT 핵심 규칙 — 핵심 규칙을 사용하여 사물 또는 다른 [AWS IoT 서비스에서 게시한](#) MQTT 메시지의 데이터를 업로드할 수 있습니다. AWS IoT AWS
- AWS IoT Events 조치 - 특정 이벤트에 의해 트리거된 [AWS IoT Events 작업을](#) 에서 사용합니다. AWS IoT Events 이 방법은 데이터 업로드가 이벤트 발생과 관련된 시나리오에 적합합니다.
- AWS IoT Greengrass 스트림 관리자 - [AWS IoT Greengrass 스트림 관리자](#)를 사용하여 에지 장치를 사용하여 로컬 데이터 소스의 데이터를 업로드할 수 있습니다. 이 옵션은 데이터가 온프레미스 또는 엣지 로케이션에서 오는 상황에 적합합니다.

이러한 방법은 다양한 소스의 데이터를 관리하기 위한 다양한 솔루션을 제공합니다. 각 옵션의 세부 사항을 자세히 살펴보고 데이터 통합 기능이 제공하는 기능을 포괄적으로 이해하십시오. AWS IoT SiteWise

데이터 스트림 관리

자산 모델 및 자산 생성을 본격적으로 시작하기 전에 먼저 산업 장비에서 플랫폼으로 정보를 직접 전송하도록 데이터 소스를 설정하세요. AWS IoT SiteWise AWS IoT SiteWise 원시 데이터를 수집하는 데이터 스트림을 자동으로 생성하도록 설계되었습니다. 각 데이터 스트림은 고유한 별칭으로 식별되므로 각 데이터의 출처를 쉽게 추적할 수 있습니다.

예를 들어 AWS IoT SiteWise Edge 게이트웨이를 사용하여 OPC-UA 서버의 기온, 프로펠러 회전 속도 및 전력 출력 시계열 데이터를 전송하는 풍력 발전 단지를 예로 들어 보겠습니다. AWS IoT SiteWiseserver1-windfarm/3/turbine/7/temperature데이터 스트림 별칭은 풍력 발전소 #3 터빈의 #7 온도 값을 식별합니다. server1OPC-UA 데이터 소스의 이름입니다. server1접두사는 이 서버에서 들어오는 모든 데이터 스트림에 사용되므로 소스별로 데이터를 구성하는 데 도움이 됩니다.

자산 모델 및 자산을 만든 후에는 각 데이터 스트림을 특정 자산 속성과 연결하여 유입되는 데이터를 정리하십시오. 이러한 연결을 통해 데이터를 AWS IoT SiteWise 수집할 뿐만 아니라 자산 구조에 따라 데이터를 처리할 수도 있습니다. 필요한 경우 데이터 스트림과 자산 속성 간의 연결을 제거할 수도 있습니다.

현재는 데이터 스트림을 측정에만 연결할 수 있습니다. 측정값은 타임스탬프가 지정된 온도 값 또는 타임스탬프가 지정된 분당 회전 수(RPM) 값 같은, 디바이스의 원시 센서 데이터 스트림을 나타내는 자산 속성 유형입니다.

이러한 측정값이 지표나 변환을 정의하면 들어오는 데이터가 특정 계산을 트리거합니다. 자산 자산은 한 번에 하나의 데이터 스트림에만 연결될 수 있다는 점에 유의해야 합니다.

Note

자산 속성은 동시에 여러 데이터 스트림과 연결할 수 없습니다.

AWS IoT SiteWise Amazon 리소스 이름 (ARN) 리소스를 사용하여 TimeSeries 보관 요금을 결정합니다. 자세한 내용은 [AWS IoT SiteWise 요금](#)을 참조하십시오.

다음 섹션에서는 AWS IoT SiteWise 콘솔 또는 API를 사용하여 데이터 스트림을 관리하는 방법을 보여줍니다.

주제

- [데이터 스트림 관리](#)

데이터 스트림 관리

데이터 스트림 관리를 시작하려면 다음 작업을 완료합니다.

Note

2021년 11월 24일 AWS IoT SiteWise 이후에 처음 사용하는 경우 이 섹션을 건너뛰어도 됩니다. 이 날짜 AWS IoT SiteWise 이전에 사용을 시작한 고객은 자산 모델 및 자산 없이 데이터를 AWS IoT SiteWise 수집할 수 있도록 서비스 설정을 구성해야 합니다.

- IAM 역할에 다음 예와 같이 권한이 있는지 확인합니다.

Example IAM 사용자 정책

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PutAssetPropertyValuesAssetPropertyOnly",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*"
    },
    {
      "Sid": "PutAssetPropertyValuesPropertyAliasAllowed",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:time-series/*"
    }
  ]
}
```

Important

데이터 스트림으로 데이터를 수집하기 전에 다음을 수행하십시오.

- 데이터 스트림을 식별하기 위해 속성 별칭을 사용하는 경우 `time-series` 리소스에 권한이 있어야 합니다.
- 연결된 자산 속성이 포함된 자산을 식별하기 위해 자산 ID를 사용하는 경우 `asset` 리소스에 권한이 있어야 합니다.

IAM 정책을 구성하는 방법에 대한 자세한 내용은 IAM 사용 가이드의 [IAM 정책 관리](#) 섹션을 참조하세요.

- 자산 속성과 관련이 없는 데이터 스트림을 AWS IoT SiteWise 허용하도록 데이터 통합 설정을 구성하십시오.

주제

- [데이터 통합 설정 구성](#)
- [데이터 스트림 관리](#)

데이터 통합 설정 구성

Console

콘솔을 사용하여 자산 속성과 관련이 없는 데이터 스트림을 AWS IoT SiteWise 허용하도록 구성합니다. AWS IoT SiteWise

데이터 모으기 설정을 구성하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 패널의 설정 아래에서 데이터 모으기를 선택합니다.
3. 데이터 모으기 페이지에서 편집을 선택합니다.
4. 연결되지 않은 데이터 모으기 섹션에서 자산 속성과 연결되지 않은 데이터 스트림에 대한 데이터 모으기 활성화를 선택합니다.

Important

자산 속성과 연결되지 않은 데이터 스트림을 AWS IoT SiteWise 허용하도록 구성한 후에는 이 설정을 끌 수 없습니다.

5. 저장을 선택합니다.
6. 연결되지 않은 데이터 모으기 활성화에서 업데이트를 선택합니다. 연결이 끊긴 데이터 통합의 상태가 활성으로 바뀝니다. 이 프로세스는 완료하려면 몇 분 이상 걸릴 수 있습니다.

AWS CLI

API 작업을 사용하여 자산 속성과 연결되지 않은 데이터 스트림을 AWS IoT SiteWise 허용하도록 구성합니다. [PutStorageConfiguration](#) 다음 섹션에서는 AWS CLI를 사용합니다.

데이터 모으기를 구성하려면 AWS CLI 단원을 참조하십시오.

1. 자산 속성과 연결되지 않은 데이터 스트림을 AWS IoT SiteWise 수신하도록 구성하려면 다음 명령을 실행합니다.

Important

자산 속성과 연결되지 않은 데이터 스트림을 AWS IoT SiteWise 허용하도록 구성한 후에는 이 설정을 끌 수 없습니다.

```
aws iotsitewise put-storage-configuration \
    -\--storage-type SITEWISE_DEFAULT_STORAGE \
    -\--disassociated-data-storage ENABLED
```

MULTI_LAYER_STORAGE에 storageType를 구성할 수 있습니다. 자세한 정보는 [데이터 스트리지 관리](#)를 참조하세요.

Example 응답

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

이 프로세스는 완료에 몇 분 이상 걸릴 수 있습니다.

2. 스토리지 구성 정보를 검색하려면 다음 명령을 실행합니다.

```
aws iotsitewise describe-storage-configuration
```

Example 응답

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-11-16T15:54:14-07:00"
}
```

데이터 스트림 관리

AWS IoT SiteWise Console OR를 사용하여 데이터 스트림을 관리하십시오 AWS CLI.

Console

AWS IoT SiteWise 콘솔을 사용하여 데이터 스트림을 관리합니다.

데이터 스트림을 관리하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 패널에서 데이터 스트림을 선택합니다.
3. (선택 사항) 태그를 추가하거나 업데이트하려면 편집할 데이터 스트림을 선택한 다음 태그 관리를 선택합니다.

태그 편집 페이지에서 태그 추가를 선택합니다. 키 필드에 사용할 태그의 이름을 입력합니다.

저장을 선택합니다.

4. (선택 사항) 데이터 스트림 테이블에서 다음과 같은 방법으로 데이터 스트림을 필터링할 수 있습니다.
 - 첫 번째 드롭다운 메뉴에서 별칭 접두사 또는 자산 ID를 선택합니다.
 - Alias - 데이터 스트림의 별칭 접두사입니다. 대상 데이터 스트림에 별칭 접두사가 있는 경우 이 옵션을 선택할 수 있습니다.
 - 자산 ID - 자산 속성이 생성된 자산의 ID입니다. 대상 데이터 스트림이 자산 속성과 연결된 경우 이 옵션을 선택할 수 있습니다.

- 두 번째 드롭다운 메뉴에서 모든 데이터 스트림, 연결된 데이터 스트림 또는 연결되지 않은 데이터 스트림을 선택합니다.
 - 모든 데이터 스트림 - 자산 속성과 연결되거나 연결되지 않은 데이터 스트림.
 - 연결된 데이터 스트림 - 자산 속성과 연결된 데이터 스트림.
 - 연결되지 않은 데이터 스트림 - 자산 속성과 연결되지 않은 데이터 스트림.
5. 관리 중인 데이터 스트림을 선택합니다. AWS IoT SiteWise 선택한 데이터 스트림을 페이지 하단에 그래프로 표시합니다. 10개 이상을 선택하면 그래프에는 처음 10개만 표시됩니다.
 6. (선택 사항) 다음과 같은 방법으로 그래프를 구성합니다.
 - a. 집계 함수의 경우 다음 중 하나를 선택합니다.
 - 데이터 포인트 수 - 현재 시간 간격 동안 해당 변수에 대한 총 데이터 포인트 수를 반환합니다.
 - 평균 - 현재 시간 간격 동안 해당 변수 값의 평균을 반환합니다.
 - 합계 - 현재 시간 간격 동안 해당 변수 값의 합계를 반환합니다.
 - 최소값 - 현재 시간 간격 동안 해당 변수 값의 최소값을 반환합니다.
 - 최대값 - 현재 시간 간격 동안 해당 변수 값의 최대값을 반환합니다.

자세한 정보는 [공식 표현식에서 집계 함수 사용](#)을 참조하세요.
 - b. 시간 범위의 경우 다음 중 하나를 선택합니다.
 - 최근 1시간 - 그래프에는 지난 1시간 동안 집계된 데이터가 표시됩니다.
 - 지난 2시간 - 그래프에는 지난 2시간 동안의 집계된 데이터가 표시됩니다.
 - 지난 3시간 - 그래프에는 지난 3시간 동안의 집계된 데이터가 표시됩니다.
 - 지난 4시간 - 그래프에는 지난 4시간 동안의 집계된 데이터가 표시됩니다.
 - c. 시간 간격에서 다음 중 하나를 선택합니다.
 - 1분 - 지정된 시간 범위 동안 1분마다 데이터를 집계합니다.
 - 1시간 - 지정된 시간 범위 동안 1시간마다 데이터를 집계합니다.
 7. 데이터 스트림 관리를 선택합니다.
 8. 데이터 스트림 연결 업데이트 섹션의 측정값 이름 열에서 다음 중 하나를 수행합니다.
 - 데이터 스트림이 측정값과 연결된 경우 닫기 아이콘을 선택하여 연결을 삭제합니다.
 - 데이터 스트림이 측정값과 연결되어 있지 않은 경우 측정값 선택을 선택합니다.

9. 측정값 선택 테이블에서 대상 자산으로 이동한 다음 연결할 측정값을 선택합니다.
10. (선택 사항) 자산 속성 별칭 업데이트 섹션에서 각 측정값에 대한 고유한 별칭을 입력합니다.
11. 업데이트를 선택합니다.

상태 열에는 다음 값 중 하나를 표시할 수 있습니다.

- 대기 중 - 데이터 스트림 연결 또는 자산 속성 별칭을 업데이트하고 있습니다.
- 제출 - 연결 또는 자산 속성 별칭에 대한 변경 내용이 저장됩니다.
- 오류 AWS IoT SiteWise — 데이터 스트림 연결 또는 측정 별칭 업데이트 요청을 처리할 수 없습니다.
- 성공 - 데이터 스트림 연결 또는 측정값 별칭을 성공적으로 업데이트했습니다.

AWS CLI

다음 API 작업을 사용하여 데이터 스트림을 관리하세요. 코드 예제에서는 AWS CLI를 사용합니다.

- [AssociateTimeSeriesToAssetProperty](#)— 데이터 스트림 (시계열) 을 자산 속성과 연결합니다.
- [DisassociateTimeSeriesFromAssetProperty](#)— 자산 속성에서 데이터 스트림을 분리합니다.
- [DeleteTimeSeries](#)— 데이터 스트림을 삭제합니다.
- [DescribeTimeSeries](#)— 데이터 스트림에 대한 정보를 검색합니다.
- [ListTimeSeries](#)— 페이지별로 구분된 데이터 스트림 목록을 검색합니다.

AssociateTimeSeriesToAssetProperty

데이터 스트림을 자산 속성과 연결하려면 다음 명령을 실행합니다.

Important

지정된 자산 속성은 현재 어떤 데이터 스트림과도 연결되어 있지 않아야 합니다.

- 연결 중인 데이터 스트림의 *data-stream-alias* 별칭으로 바꾸십시오.
- *asset-ID*를 자산 속성이 생성된 자산의 ID로 바꿉니다.
- *property-ID*를 자산 속성의 ID로 바꿉니다.

```
aws iotsitewise associate-time-series-to-asset-property \
    --alias data-stream-alias \
    --assetId asset-ID \
    --propertyId property-ID
```

DisassociateTimeSeriesFromAssetProperty

데이터 스트림에서 자산 속성을 연결 해제하려면 다음 명령을 실행합니다.

- 연결 해제하려는 데이터 스트림의 *data-stream-alias* 별칭으로 바꾸십시오.
- *asset-ID*를 자산 속성이 생성된 자산의 ID로 바꿉니다.
- *property-ID*를 자산 속성의 ID로 바꿉니다.

```
aws iotsitewise disassociate-time-series-from-asset-property \
    --alias data-stream-alias \
    --assetId asset-ID \
    --propertyId property-ID
```

DeleteTimeSeries

데이터 스트림을 삭제하려면 다음 명령을 실행합니다.

삭제하려는 데이터 스트림의 *data-stream-alias* 별칭으로 바꾸십시오.

```
aws iotsitewise delete-time-series --alias data-stream-alias
```

데이터 스트림을 식별하려면 다음 중 하나를 수행합니다.

- 데이터 스트림이 자산 속성과 연결되어 있지 않은 경우 데이터 스트림의 *alias*를 지정합니다.
- 데이터 스트림이 자산 속성과 연결된 경우 다음 중 하나를 지정합니다.
 - 데이터 스트림의 *alias*입니다.
 - 자산 속성을 식별하는 *assetId* 및 *propertyId*.

DescribeTimeSeries

`DescribeTimeSeriesAPI` 작업을 사용하여 데이터 스트림을 성공적으로 연결 또는 연결 해제했는지 확인할 수 있습니다.

데이터 스트림에 대한 정보를 검색하려면 다음 명령을 실행합니다.

```
aws iotsitewise describe-time-series --alias data-stream-alias
```

데이터 스트림을 식별하려면 다음 중 하나를 수행합니다.

- 데이터 스트림이 자산 속성과 연결되어 있지 않은 경우 데이터 스트림의 `alias`를 지정합니다.
- 데이터 스트림이 자산 속성과 연결된 경우 다음 중 하나를 지정합니다.
 - 데이터 스트림의 `alias`입니다.
 - 자산 속성을 식별하는 `assetId` 및 `propertyId`.

ListTimeSeries

ListTimeSeriesAPI 작업을 사용하여 데이터 스트림을 성공적으로 삭제했는지 확인할 수 있습니다.

데이터 스트림의 페이지 목록을 검색하려면 다음 명령을 실행합니다.

```
aws iotsitewise list-time-series
```

API를 사용한 데이터 수집 AWS IoT SiteWise

AWS IoT SiteWise API를 사용하여 타임스탬프가 찍힌 산업 데이터를 자산의 속성 및 측정 속성으로 전송할 수 있습니다. API는 `timestamp-quality-value (TQV)` 구조가 포함된 페이로드를 받아들입니다.

[BatchPutAssetPropertyValue](#) 작업을 사용하여 데이터를 업로드할 수 있습니다. 이 작업을 통해 한 번에 여러 데이터 항목을 업로드하여 여러 장치에서 데이터를 수집한 다음 단일 요청으로 모두 전송할 수 있습니다.

Important

[BatchPutAssetPropertyValue](#) 작업에는 다음과 같은 할당량이 적용됩니다.

- 요청당 최대 10개 [항목](#).
- 항목당 최대 10개의 [속성 값](#)(TQV 데이터 포인트).
- AWS IoT SiteWise 는 타임스탬프가 과거 7일 이상 또는 미래 10분을 초과한 모든 데이터를 거부합니다.

이러한 할당량에 대한 자세한 내용은 API 참조를 참조하십시오
[BatchPutAssetPropertyValue](#).AWS IoT SiteWise

자산 자산을 식별하려면 다음 중 하나를 지정하십시오.

- 데이터가 전송되는 자산 `propertyId` 자산의 소유자. `assetId`
- 데이터 스트림 별칭(예: `/company/windfarm/3/turbine/7/temperature`)인 `propertyAlias`. 이 옵션을 사용하려면 먼저 자산 속성의 별칭을 설정해야 합니다. 속성 별칭을 설정하려면 [참조하십시오 산업 데이터 스트림을 자산 속성에 매핑](#).

다음 예제는 JSON 파일에 저장된 페이로드로부터 풍력 터빈의 온도와 분당 회전 수(RPM) 판독값을 전송하는 방법을 보여줍니다.

```
aws iotsitewise batch-put-asset-property-value --cli-input-json file://batch-put-payload.json
```

`batch-put-payload.json`의 예제 페이로드에는 다음 내용이 포함되어 있습니다.

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature",
      "propertyValues": [
        {
          "value": {
            "integerValue": 38
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
      "propertyValues": [
```

```

    {
      "value": {
        "doubleValue": 15.09
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      },
      "quality": "GOOD"
    }
  ]
}

```

페이로드의 각 항목은 고유한 문자열로 정의할 수 있는 entryId를 포함합니다. 요청 항목이 실패하면 각 오류에 해당 요청의 entryId가 포함되므로 다시 시도할 요청을 알 수 있습니다.

목록의 각 구조체는 a value, a timestamp, a 및 선택적으로 a를 포함하는 timestamp-quality-value (TQV) propertyValues 구조체입니다. quality

- value - 설정되는 속성 유형에 따라 다음 필드 중 하나를 포함하는 구조입니다.
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue
- timestamp - 현재 Unix Epoch 시간(초) timeInSeconds를 포함하는 구조입니다. 시간적으로 정확한 데이터가 있는 경우 timestamp 구조에서 offsetInNanos 키를 설정할 수도 있습니다. AWS IoT SiteWise 과거 7일보다 오래된 타임스탬프 또는 향후 10분 이후의 타임스탬프가 있는 모든 데이터 포인트를 거부합니다.
- quality - (선택 사항) 다음 품질 문자열 중 하나입니다.
 - GOOD - (기본값) 데이터가 어떤 문제의 영향도 받지 않습니다.
 - BAD - 데이터가 센서 고장과 같은 문제의 영향을 받습니다.
 - UNCERTAIN - 데이터가 센서 부정확과 같은 문제의 영향을 받습니다.

계산에서 데이터 품질을 AWS IoT SiteWise 처리하는 방법에 대한 자세한 내용은 수식 표현식의 [데이터 품질](#)을 참조하십시오.

규칙을 사용한 AWS IoT Core 데이터 수집

AWS IoT SiteWise 에서 규칙을 사용하여 AWS IoT 사물 및 기타 AWS 서비스에서 데이터를 전송합니다. AWS IoT Core 규칙은 MQTT 메시지를 변환하고 AWS 서비스와 상호 작용하는 작업을 수행합니다. AWS IoT SiteWise 규칙 작업은 API에서 메시지 데이터를 [BatchPutAssetPropertyValue](#) 작업에 전달합니다. AWS IoT SiteWise 자세한 내용은 AWS IoT 개발자 가이드의 [규칙](#) 및 [AWS IoT SiteWise 작업을 참조하십시오](#).

디바이스 새도우를 통해 데이터를 수집하는 규칙을 설정하는 데 필요한 단계를 안내하는 자습서를 따르려면 [여기](#)를 참조하십시오. [사물의 데이터 수집 AWS IoT](#)

다른 AWS 서비스로 데이터를 AWS IoT SiteWise 보낼 수도 있습니다. 자세한 정보는 [다른 AWS 서비스와의 상호작용](#)을 참조하세요.

주제

- [필요한 액세스 AWS IoT 권한 부여](#)
- [AWS IoT SiteWise 규칙 작업 구성](#)
- [Basic Ingest를 통한 비용 절감](#)

필요한 액세스 AWS IoT 권한 부여

IAM 역할을 사용하여 각 규칙이 액세스할 수 있는 AWS 리소스를 제어할 수 있습니다. 규칙을 생성하기 전에 규칙이 필요한 AWS 리소스에서 작업을 수행하도록 허용하는 정책을 포함하는 IAM 역할을 생성해야 합니다. AWS IoT 규칙을 실행할 때 이 역할을 말합니다.

AWS IoT 콘솔에서 규칙 작업을 생성하는 경우 루트 자산을 선택하여 선택한 자산 계층 구조에 액세스할 수 있는 역할을 생성할 수 있습니다. 규칙의 역할을 수동으로 정의하는 방법에 관한 자세한 내용은 AWS IoT 개발자 가이드의 [필수 액세스 권한 부여 AWS IoT](#) 및 [역할 권한 전달](#)을 참조하십시오.

AWS IoT SiteWise 규칙 조치의 경우 규칙이 데이터를 보내는 자산 속성에 `iotsitewise:BatchPutAssetPropertyValue` 대한 액세스를 허용하는 역할을 정의해야 합니다. 보안을 강화하기 위해 Condition 속성에 AWS IoT SiteWise 자산 계층 경로를 지정할 수 있습니다.

다음 예제 신뢰 정책은 특정 자산 및 해당 자산의 하위 자산에 대한 액세스를 허용합니다.

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

```

{
  "Effect": "Allow",
  "Action": "iotsitewise:BatchPutAssetPropertyValue",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "iotsitewise:assetHierarchyPath": [
        "/root node asset ID",
        "/root node asset ID/*"
      ]
    }
  }
}

```

Condition정책에서 삭제하여 모든 자산에 대한 액세스를 허용하십시오. 다음 예제 신뢰 정책은 현재 리전의 모든 자산에 대한 액세스를 허용합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*"
    }
  ]
}

```

AWS IoT SiteWise 규칙 작업 구성

AWS IoT SiteWise 규칙 작업은 규칙을 시작한 MQTT 메시지의 데이터를 의 자산 속성으로 보냅니다. AWS IoT SiteWise 여러 데이터 항목을 서로 다른 자산 속성에 동시에 업로드하여 장치의 모든 센서에 대한 업데이트를 메시지 하나로 전송할 수 있습니다. 각 데이터 입력에 대해 한 번에 여러 데이터 포인트를 업로드할 수도 있습니다.

Note

규칙 AWS IoT SiteWise 동작과 함께 데이터를 전송하는 경우 데이터는 BatchPutAssetPropertyValue 작업의 모든 요구 사항을 충족해야 합니다. 예를 들어, 데

이터에 현재 Unix Epoch 시간에서 7일 이전인 타임스탬프가 있을 수 없습니다. 자세한 내용은 [AWS IoT SiteWise API를 사용하여 데이터 수집](#)을 참조하십시오.

규칙 작업의 각 데이터 항목에 대해 자산 속성을 식별하고 해당 자산 속성에 대한 각 데이터 포인트의 타임스탬프, 품질 및 값을 지정합니다. 규칙 작업에는 모든 파라미터에 대한 문자열이 필요합니다.

항목에서 자산 속성을 식별하려면 다음 중 하나를 지정합니다.

- 데이터를 보낼 자산 속성의 자산 ID(assetId) 및 속성 ID(propertyId)입니다. 를 사용하여 자산 ID 및 속성 ID를 찾을 수 AWS IoT SiteWise Console 있습니다. 자산 ID를 알고 있는 경우 AWS CLI [DescribeAssetto](#) 호출을 사용하여 속성 ID를 찾을 수 있습니다.
- 데이터 스트림 별칭(예: /company/windfarm/3/turbine/7/temperature)인 속성 별칭(propertyAlias). 이 옵션을 사용하려면 먼저 자산 속성의 별칭을 설정해야 합니다. 속성 별칭을 설정하는 방법에 대한 자세한 내용은 [산업 데이터 스트림을 자산 속성에 매핑](#) 단원을 참조하십시오.

각 항목의 타임스탬프는 장비에서 보고한 타임스탬프 또는 에서 제공한 타임스탬프를 사용하십시오. AWS IoT Core타임스탬프에는 두 개의 매개변수가 있습니다.

- 초 단위 시간(timeInSeconds) – 센서 또는 장비가 데이터를 보고한 Unix Epoch 시간(초)입니다.
- 나노초 단위 오프셋(offsetInNanos) – (선택 사항) 초 단위 시간의 나노초 오프셋입니다.

Important

타임스탬프가 문자열이거나, 소수 부분이 있거나, AWS IoT SiteWise 에서 초 단위가 아닌 경우 요청을 거부합니다. 타임스탬프를 초 및 나노초 오프셋으로 변환해야 합니다. AWS IoT 규칙 엔진의 기능을 사용하여 타임스탬프를 변환할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- [정확한 시간을 보고하지 않는 디바이스의 타임스탬프 가져오기](#)
- [문자열 형식의 타임스탬프 변환](#)

작업의 여러 파라미터에 대한 대체 템플릿을 사용하여 계산하고 함수를 호출하며 메시지 페이로드에서 값을 가져올 수 있습니다. 자세한 내용은 AWS IoT 개발자 가이드의 [대체 템플릿](#)을 참조하십시오.

Note

대체 템플릿의 표현식은 SELECT 문과 별도로 평가되므로 대체 템플릿을 사용하면 AS 절을 사용하여 생성한 별칭은 참조할 수 없습니다. 지원되는 함수와 연산자 외에 원래 페이로드에 있는 정보만 참조할 수 있습니다.

주제

- [정확한 시간을 보고하지 않는 디바이스의 타임스탬프 가져오기](#)
- [문자열 형식의 타임스탬프 변환](#)
- [나노초 정밀도의 타임스탬프 문자열 변환](#)
- [규칙 구성의 예](#)
- [규칙 작업 문제 해결](#)

정확한 시간을 보고하지 않는 디바이스의 타임스탬프 가져오기

센서 또는 장비가 정확한 시간 데이터를 보고하지 않는 경우 [timestamp\(\)](#) 를 사용하여 AWS IoT 규칙 엔진에서 현재 유닉스 에포크 시간을 가져오세요. 이 함수는 시간을 밀리초 단위로 출력하므로 값을 초 단위 시간 및 나노초 단위 오프셋으로 변환해야 합니다. 이렇게 하려면 다음 변환을 사용합니다.

- 초 단위 시간(timeInSeconds)의 경우 $\{\text{floor}(\text{timestamp}() / 1E3)\}$ 를 사용하여 시간을 밀리초에서 초로 변환합니다.
- 나노초 단위 오프셋(offsetInNanos)의 경우 $\{(\text{timestamp}() \% 1E3) * 1E6\}$ 을 사용하여 타임스탬프의 나노초 오프셋을 계산합니다.

문자열 형식의 타임스탬프 변환

센서 또는 장비가 시간 데이터를 문자열 형식(예:) 으로 보고하는 경우 [time_to_epoch](#) (문자열, 문자열2020-03-03T14:57:14.699Z) 를 사용하십시오. 이 함수는 타임스탬프와 형식 패턴을 매개변수로 입력하고 시간을 밀리초 단위로 출력합니다. 그런 다음 시간을 초 단위의 시간으로, 나노초 단위의 오프셋으로 변환해야 합니다. 이렇게 하려면 다음 변환을 사용합니다.

- 초 단위 시간(timeInSeconds)의 경우 $\{\text{floor}(\text{time_to_epoch}("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") / 1E3)\}$ 타임스탬프 문자열을 밀리초로 변환한 다음 초로 변환하는 데 사용합니다.

- 나노초 단위 오프셋(offsetInNanos)의 경우

`${(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z')) % 1E3) * 1E6}`을 사용하여 타임스탬프 문자열의 나노초 오프셋을 계산합니다.

Note

이 `time_to_epoch` 함수는 최대 밀리초 정밀도의 타임스탬프 문자열을 지원합니다. 마이크로초 또는 나노초 정밀도의 문자열을 변환하려면 규칙이 타임스탬프를 숫자 값으로 변환하기 위해 호출하는 AWS Lambda 함수를 구성하십시오. 자세한 정보는 [나노초 정밀도의 타임스탬프 문자열 변환](#)을 참조하세요.

나노초 정밀도의 타임스탬프 문자열 변환

디바이스가 나노초 정밀도의 문자열 형식(예: 2020-03-03T14:57:14.699728491Z)으로 타임스탬프 정보를 전송하는 경우 다음 절차에 따라 규칙 작업을 구성합니다. 문자열의 타임스탬프를 초 단위의 시간 () 으로, 오프셋을 nanos (**timeInSeconds**) 단위로 변환하는 AWS Lambda 함수를 만들 수 있습니다. `offsetInNanos` 그런 다음 규칙 작업 파라미터에서 [aws_lambda \(FunctionARN, InputJSON\)](#) 를 사용하여 해당 Lambda 함수를 호출하고 규칙의 출력을 사용하십시오.

Note

이 단원에서는 다음 리소스를 만드는 방법을 잘 안다고 가정하고 고급 지침을 제공합니다.

- Lambda 함수. 자세한 내용은 AWS Lambda 개발자 가이드에서 [콘솔로 Lambda 함수 생성](#) 또는 [AWS CLI에서 Lambda 사용](#)을 참조하십시오.
- AWS IoT 규칙 작업이 포함된 규칙. AWS IoT SiteWise 자세한 정보는 [규칙을 사용한 AWS IoT Core 데이터 수집](#)을 참조하세요.

타임스탬프 문자열을 파싱하는 AWS IoT SiteWise 규칙 액션을 만들려면

1. 다음 속성을 사용하여 Lambda 함수를 만듭니다.

- 함수 이름 - 설명이 포함된 함수 이름(예: **ConvertNanosecondTimestampFromString**)을 사용합니다.
- 런타임 — Python 3.11 (**python3.11**) 과 같은 Python 3 런타임을 사용합니다.

- 권한 — 기본 Lambda 권한 () 을 사용하여 역할을 생성합니다. AWS LambdaBasicExecutionRole
- 계층 — Lambda 함수가 AWS사용할 SDK판다스-Python311 계층을 추가합니다. numpy
- 함수 코드 - timestamp라는 문자열 인수를 사용하고 해당 타임스탬프에 대해 timeInSeconds 및 offsetInNanos 값을 출력하는 다음 함수 코드를 사용합니다.

```
import json
import math
import numpy

# Converts a timestamp string into timeInSeconds and offsetInNanos in Unix epoch
time.
# The input timestamp string can have up to nanosecond precision.
def lambda_handler(event, context):
    timestamp_str = event['timestamp']
    # Parse the timestamp string as nanoseconds since Unix epoch.
    nanoseconds = numpy.datetime64(timestamp_str, 'ns').item()
    time_in_seconds = math.floor(nanoseconds / 1E9)
    # Slice to avoid precision issues.
    offset_in_nanos = int(str(nanoseconds)[-9:])
    return {
        'timeInSeconds': time_in_seconds,
        'offsetInNanos': offset_in_nanos
    }
```

이 Lambda 함수는 datetime64를 사용하여 ISO 8601 형식으로 타임스탬프 문자열을 입력합니다. NumPy

Note

타임스탬프 문자열이 ISO 8601 형식이 아닌 경우 타임스탬프 형식을 정의하는 pandas 으로 솔루션을 구현할 수 있습니다. 자세한 내용은 [pandas.to_datetime](#)을 참조하십시오.

2. 규칙에 맞게 AWS IoT SiteWise 작업을 구성할 때는 다음과 같은 시간 (초) 및 오프셋 (**timeInSecondsnanos**) 단위의 대체 템플릿을 사용하십시오. offsetInNanos 이러한 대체 템플릿은 메시지 페이로드에 타임스탬프 문자열(timestamp)이 포함되어 있다고 가정합니다. aws_lambda 함수는 두 번째 파라미터에 JSON 구조를 사용하므로 필요한 경우 아래 대체 템플릿을 수정할 수 있습니다.

- 초 단위 시간(timeInSeconds)에 다음 대체 템플릿을 사용합니다.

```
${aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).timeInSeconds}
```

- 나노초 단위 오프셋(offsetInNanos)에 다음 대체 템플릿을 사용합니다.

```
${aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).offsetInNanos}
```

각 매개변수에 대해 지역 및 계정 **ID# ## # ## ID#** 바꾸십시오. AWS Lambda 함수에 다른 이름을 사용한 경우 해당 이름도 변경합니다.

- AWS IoT 해당 권한으로 함수를 호출할 수 있는 권한을 부여하세요. `lambda:InvokeFunction` 자세한 내용은 [aws_lambda\(functionArn, inputJson\)](#)를 참조하십시오.
- 규칙을 테스트 (예: AWS IoT MQTT 테스트 클라이언트 사용) 하고 전송한 데이터가 AWS IoT SiteWise 수신되는지 확인합니다.

규칙이 예상대로 작동하지 않을 경우 [AWS IoT SiteWise 규칙 조치 문제 해결](#) 단원을 참조하십시오.

Note

이 솔루션은 각 타임스탬프 문자열에 대해 Lambda 함수를 두 번 호출합니다. 현재 규칙이 각 페이로드에서 동일한 타임스탬프를 갖는 여러 데이터 포인트를 처리하는 경우 Lambda 함수 호출 수를 줄이기 위해 다른 규칙을 만들 수 있습니다.

이렇게 하려면 재게시 작업을 사용하여 Lambda를 호출하고 `timeInSeconds` 및 `offsetInNanos`로 변환된 타임스탬프 문자열을 갖는 원래 페이로드를 게시하는 규칙을 만듭니다. 그런 다음 변환된 페이로드를 소비하는 AWS IoT SiteWise 규칙 작업이 포함된 규칙을 생성합니다. 이 접근 방식을 사용하면 규칙이 Lambda를 호출하는 횟수는 줄지만 규칙 작업 실행 횟수는 AWS IoT 늘릴 수 있습니다. 이 솔루션을 사용 사례에 적용하는 경우 각 서비스의 요금을 고려하십시오.

규칙 구성의 예

이 섹션에는 작업을 사용하여 규칙을 생성하기 위한 예제 규칙 구성이 포함되어 있습니다. AWS IoT SiteWise

Example 속성 별칭을 메시지 주제로 사용하는 규칙 작업의 예제

다음 예제에서는 주제 (topic ()) 를 통해 속성 별칭으로 사용하여 자산 속성을 식별하는 AWS IoT SiteWise 액션이 포함된 규칙을 만듭니다. 이 예제를 사용하여 모든 풍력 발전 단지의 모든 풍력 터빈에 이중 유형 데이터를 수집하는 규칙 하나를 정의하십시오. 이 예제에서는 모든 터빈 자산의 속성에 속성 별칭을 정의해야 합니다. 정수 유형 데이터를 수집하기 위한 유사한 두 번째 규칙을 정의해야 합니다.

```
aws iot create-topic-rule \
  --rule-name SiteWiseWindFarmRule \
  --topic-rule-payload file://sitewise-rule-payload.json
```

sitewise-rule-payload.json의 예제 페이로드에는 다음 내용이 포함되어 있습니다.

```
{
  "sql": "SELECT * FROM '/company/windfarm/+/turbine/+/+' WHERE type = 'double'",
  "description": "Sends data to the wind turbine asset property with the same alias as the topic",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "iotSiteWise": {
        "putAssetPropertyValueEntries": [
          {
            "propertyAlias": "${topic()}",
            "propertyValues": [
              {
                "timestamp": {
                  "timeInSeconds": "${timeInSeconds}"
                },
                "value": {
                  "doubleValue": "${value}"
                }
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

    ],
    "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
  }
}
]
}

```

이 규칙 작업을 수행하려면 풍력 터빈 속성 별칭 (예: /company/windfarm/3/turbine/7/temperature) 에 다음 메시지를 주제로 전송하여 데이터를 수집하십시오.

```

{
  "type": "double",
  "value": "38.3",
  "timeInSeconds": "1581368533"
}

```

Example timestamp()를 사용하여 시간을 결정하는 규칙 동작 예제

다음 예제에서는 ID로 자산 속성을 식별하고 [timestamp\(\)](#) 를 사용하여 현재 시간을 결정하는 AWS IoT SiteWise 액션이 포함된 규칙을 만듭니다.

```

aws iot create-topic-rule \
  --rule-name SiteWiseAssetPropertyRule \
  --topic-rule-payload file://sitewise-rule-payload.json

```

sitewise-rule-payload.json의 예제 페이로드에는 다음 내용이 포함되어 있습니다.

```

{
  "sql": "SELECT * FROM 'my/asset/property/topic'",
  "description": "Sends device data to an asset property",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "iotSiteWise": {
        "putAssetPropertyValueEntries": [
          {
            "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "propertyValues": [
              {

```

```

        "timestamp": {
            "timeInSeconds": "${floor(timestamp() / 1E3)}",
            "offsetInNanos": "${(timestamp() % 1E3) * 1E6}"
        },
        "value": {
            "doubleValue": "${value}"
        }
    }
]
},
"roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
}
}
]
}
}

```

이 규칙 액션을 사용하여 데이터를 수집하려면 에 다음 메시지를 보내십시오. `my/asset/property/topic`

```

{
  "type": "double",
  "value": "38.3"
}

```

규칙 작업 문제 해결

에서 AWS IoT SiteWise 규칙 조치의 AWS IoT Core 문제를 해결하려면 CloudWatch 로그를 구성하거나 규칙에 대한 오류 재게시 작업을 구성하십시오. 자세한 정보는 [AWS IoT SiteWise 규칙 조치 문제 해결](#)을 참조하세요.

Basic Ingest를 통한 비용 절감

AWS IoT Core [메시징 비용 AWS IoT Core 없이 데이터를 전송하는 데 사용할 수 있는 기본 AWS IoT 인제스트라는 기능을 제공합니다.](#) Basic Ingest는 수집 경로에서 게시/구독 메시지 브로커를 제거하여 대용량 데이터 수집 워크로드에 대한 데이터 흐름을 최적화합니다. 메시지를 라우팅해야 하는 규칙을 알고 있는 경우 Basic Ingest를 사용할 수 있습니다.

Basic Ingest를 사용하려면 특수 주제 `$aws/rules/rule-name`을 사용하여 특정 규칙으로 직접 메시지를 보냅니다. 예를 들어 SiteWiseWindFarmRule이라는 규칙에 메시지를 보내려면 주제 `$aws/rules/SiteWiseWindFarmRule`에 메시지를 보냅니다.

규칙 작업에서 [주제\(소수\)](#)가 포함된 대체 템플릿을 사용하는 경우 Basic Ingest 특수 주제(예: \$aws/rules/*rule-name/original-topic*)의 끝에서 원래 주제를 전달할 수 있습니다. 예를 들어 이전 섹션의 풍력 발전소 속성 별칭 예제와 함께 Basic Ingest를 사용하려면 다음 주제로 메시지를 보낼 수 있습니다.

```
$aws/rules/SiteWiseWindFarmRule//company/windfarm/3/turbine/7/temperature
```

Note

위 예제에는 규칙 작업에 표시되는 주제에서 기본 인제스트 접두사 (//) 가 AWS IoT 제거되므로 두 번째 슬래시 (\$aws/rules/*rule-name/*) 가 포함됩니다. 이 예제에서 규칙은 주제 / company/windfarm/3/turbine/7/temperature를 받습니다.

자세한 내용은 AWS IoT 개발자 가이드의 [Basic Ingest를 사용하여 메시징 비용 절감](#)을 참조하십시오.

에서 데이터 수집 AWS IoT Events

를 사용하면 AWS 클라우드에서 IoT 플릿을 위한 복잡한 이벤트 모니터링 애플리케이션을 구축할 수 있습니다. AWS IoT Events이벤트 발생 AWS IoT SiteWise 시 IoT SiteWise 작업을 사용하여 자산 속성에 데이터를 전송할 수 있습니다. AWS IoT Events

AWS IoT Events AWS 클라우드 내 IoT 장치 및 시스템에 대한 이벤트 모니터링 애플리케이션 개발을 간소화하도록 설계되었습니다. 를 사용하여 AWS IoT Events다음을 수행할 수 있습니다.

- IoT 플릿 전반의 변화, 이상 또는 특정 조건을 감지하고 이에 대응합니다.
- 운영 효율성을 높이고 IoT 에코시스템을 사전 예방적으로 관리할 수 있습니다.

AWS IoT SiteWise Action과 AWS IoT SiteWise 통합함으로써 특정 이벤트에 대응하여 자산 속성을 자동으로 업데이트할 수 있도록 기능을 AWS IoT Events 확장합니다. AWS IoT SiteWise 이러한 상호 작용을 통해 데이터 수집 및 관리를 단순화할 수 있습니다. 또한 실행 가능한 통찰력을 제공하여 역량을 강화할 수 있습니다.

자세한 내용은 AWS IoT Events 개발자 설명서에서 다음 주제를 참조하십시오.

- [이게 뭐죠? AWS IoT Events](#)
- [AWS IoT Events 작업](#)

- [IoT SiteWise 액션](#)

AWS IoT Greengrass 스트림 관리자 사용

AWS IoT Greengrass 스트림 관리자는 로컬 소스에서 AWS 클라우드로 데이터 스트림을 쉽게 전송할 수 있는 통합 기능입니다. 데이터 흐름을 관리하는 중간 계층 역할을 하므로, 엣지에서 작동하는 디바이스가 추가 분석 및 처리를 위해 데이터를 전송하기 AWS IoT SiteWise전에 데이터를 수집 및 저장할 수 있습니다.

AWS IoT SiteWise 콘솔에서 로컬 소스를 구성하여 데이터 대상을 추가합니다. 사용자 지정 AWS IoT Greengrass 솔루션에서 스트림 관리자를 사용하여 데이터를 수집할 AWS IoT SiteWise수도 있습니다.

Note

OPC-UA 소스에서 데이터를 수집하려면 에서 실행되는 AWS IoT SiteWise Edge 게이트웨이를 구성하십시오. AWS IoT Greengrass자세한 정보는 [SiteWise 에지 게이트웨이 사용](#)을 참조하세요.

로컬 소스 데이터의 대상을 구성하는 방법에 대한 자세한 내용은 을 참조하십시오. [데이터 소스 구성](#)

사용자 지정 AWS IoT Greengrass 솔루션에서 스트림 관리자를 사용하여 데이터를 수집하는 방법에 대한 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 다음 항목을 참조하십시오.

- [무엇입니까 AWS IoT Greengrass?](#)
- [AWS IoT Greengrass Core에서 데이터 스트림 관리](#)
- [AWS IoT SiteWise 자산 속성으로 데이터 내보내기](#)

API를 사용한 데이터 수집 CreateBulkImportJob

CreateBulkImportJobAPI를 사용하여 Amazon S3에서 대량의 데이터를 가져올 수 있습니다. Amazon S3에 데이터를 CSV 형식으로 저장해야 합니다. 데이터 파일에는 다음과 같은 열이 있을 수 있습니다.

Note

자산 속성을 식별하려면 다음 중 하나를 지정하십시오.

- 데이터를 보낼 자산 속성의 ASSET_ID 및 PROPERTY_ID.
- 데이터 스트림 별칭(예: /company/windfarm/3/turbine/7/temperature)인 ALIAS. 이 옵션을 사용하려면 먼저 자산 속성의 별칭을 설정해야 합니다. 속성 별칭을 설정하는 방법에 대한 자세한 내용은 [the section called “산업 데이터 스트림을 자산 속성에 매핑” 단원을 참조하십시오.](#)

- ALIAS – OPC-UA 서버 데이터 스트림 경로(예: /company/windfarm/3/turbine/7/temperature)와 같이 속성을 식별하는 속성 별칭입니다. 자세한 내용은 [산업 데이터 스트림을 자산 속성에 매핑](#) 단원을 참조하십시오
- ASSET_ID – 자산의 ID입니다.
- PROPERTY_ID – 자산 속성의 ID입니다.
- DATA_TYPE – 속성의 데이터 형식입니다. 다음 중 하나일 수 있습니다.
 - STRING – 최대 1,024바이트의 문자열입니다.
 - INTEGER – 범위가 [-2,147,483,648, 2,147,483,647]인 부호 있는 32비트 정수입니다.
 - DOUBLE – 범위가 [-10¹⁰⁰, 10¹⁰⁰]이고 IEEE 754 배정밀도를 갖는 부동 소수점 숫자입니다.
 - BOOLEAN - true 또는 false입니다.
- TIMESTAMP_SECONDS - Unix Epoch 시간으로 표시된 데이터 포인트의 타임스탬프입니다.
- TIMESTAMP_NANO_OFFSET - TIMESTAMP_SECONDS에서 변환된 나노초의 오프셋입니다.
- QUALITY - (선택 사항) 자산 속성 값의 품질입니다. 값은 다음 중 하나일 수 있습니다.
 - GOOD - (기본값) 데이터가 어떤 문제의 영향도 받지 않습니다.
 - BAD – 데이터가 센서 고장과 같은 문제의 영향을 받습니다.
 - UNCERTAIN – 데이터가 센서 부정확과 같은 문제의 영향을 받습니다.

계산에서 데이터 품질을 AWS IoT SiteWise 처리하는 방법에 대한 자세한 내용은 [수식 표현식의 데이터 품질을](#) 참조하십시오.

- VALUE - 자산 속성의 값입니다.

Example .csv 형식의 데이터 파일

```
asset_id,property_id,DOUBLE,1635201373,0,GOOD,1.0
asset_id,property_id,DOUBLE,1635201374,0,GOOD,2.0
asset_id,property_id,DOUBLE,1635201375,0,GOOD,3.0
```

```
unmodeled_alias1,DOUBLE,1635201373,0,GOOD,1.0
unmodeled_alias1,DOUBLE,1635201374,0,GOOD,2.0
unmodeled_alias1,DOUBLE,1635201375,0,GOOD,3.0
unmodeled_alias1,DOUBLE,1635201376,0,GOOD,4.0
unmodeled_alias1,DOUBLE,1635201377,0,GOOD,5.0
unmodeled_alias1,DOUBLE,1635201378,0,GOOD,6.0
unmodeled_alias1,DOUBLE,1635201379,0,GOOD,7.0
unmodeled_alias1,DOUBLE,1635201380,0,GOOD,8.0
unmodeled_alias1,DOUBLE,1635201381,0,GOOD,9.0
unmodeled_alias1,DOUBLE,1635201382,0,GOOD,10.0
```

AWS IoT SiteWise 대량 가져오기 작업을 생성하고 기존 작업에 대한 정보를 가져오기 위한 다음 API 작업을 제공합니다.

- [CreateBulkImportJob](#)— 새 대량 가져오기 작업을 생성합니다.
- [DescribeBulkImportJob](#)— 대량 가져오기 작업에 대한 정보를 검색합니다.
- [ListBulkImportJob](#)— 모든 대량 가져오기 작업의 요약 목록을 페이지별로 구분하여 검색합니다.

대량 가져오기 작업 생성(AWS CLI)

[CreateBulkImportJob](#) API 작업을 사용하여 Amazon S3에서 Amazon S3로 데이터를 AWS IoT SiteWise 전송합니다. [CreateBulkImportJob](#) API를 사용하여 비용 효율적인 방식으로 데이터를 소량 배치로 수집할 수 있습니다. 다음 예에는 AWS CLI가 사용됩니다.

Important

대량 가져오기 작업을 생성하기 전에 AWS IoT SiteWise 웹 티어 또는 AWS IoT SiteWise 콜드 티어를 활성화해야 합니다. 자세한 정보는 [스토리지 설정 구성](#)을 참조하세요.

대량 가져오기는 기록 데이터를 저장하도록 설계되었습니다. AWS IoT SiteWise. AWS IoT SiteWise 웹 티어 또는 AWS IoT SiteWise 콜드 티어에서는 계산이나 알림을 시작하지 않습니다.

다음 명령을 실행합니다. *file-name*을 대량 가져오기 작업 구성에 포함된 파일의 이름으로 바꿉니다.

```
aws iotsitewise create-bulk-import-job --cli-input-json file://file-name.json
```

Example 대량 가져오기 작업 구성

구성 설정의 예는 다음과 같습니다.

- *adaptive-ingestion-flag*을 true 또는 false로 바꿉니다.
 - 로 false 설정된 경우 대량 가져오기 작업은 기록 데이터를 로 AWS IoT SiteWise수집합니다.
 - 로 true 설정된 경우 대량 가져오기 작업은 다음 작업을 수행합니다.
 - 에 새 데이터를 수집합니다. AWS IoT SiteWise
 - 지표를 계산하고 변환을 수행하며, 7일 이내의 타임스탬프가 있는 데이터에 대한 알림을 지원합니다.
- 웹 티어 *delete-files-after-import##### ## # S3 ## ##### true ##### -flag#* 로 바꾸십시오 AWS IoT SiteWise .
- *error-bucket*을 이 대량 가져오기 작업과 관련된 오류가 전송되는 Amazon S3 버킷의 이름으로 바꿉니다.
- 이 대량 가져오기 *error-bucket-prefix*작업과 관련된 오류가 전송되는 Amazon S3 버킷의 접두사로 바꾸십시오.

Amazon S3에서는 접두사를 버킷의 폴더 이름으로 사용합니다. 각 Amazon S3 객체에는 버킷의 고유 식별자인 키가 있습니다. 버킷 내 각 객체는 정확히 하나의 키를 갖습니다. 접두사는 슬래시(/)로 끝나야 합니다. 자세한 내용을 알아보려면 Amazon Simple Storage Service 사용 설명서의 [접두어를 사용한 객체 구성](#)을 참조하십시오.

- *data-bucket*을 데이터를 가져오는 Amazon S3 버킷의 이름으로 바꿉니다.
- 데이터가 들어 있는 Amazon S3 객체의 *data-bucket-key*키로 대체합니다. 각 객체에는 고유 식별자인 키가 있습니다. 각 객체에는 정확히 하나의 키가 있습니다.
- 데이터를 포함하는 Amazon S3 객체의 특정 버전을 식별하려면 버전 *data-bucket-version-id*ID로 대체하십시오. 이 파라미터는 선택 사항입니다.
- *column-name*을 .csv 파일에 지정된 열 이름으로 대체하십시오.
- *job-name*을 대량 가져오기 작업을 식별하는 고유한 이름으로 대체하십시오.
- Amazon S3 데이터를 읽을 수 있는 IAM *job-role-arn* AWS IoT SiteWise 역할로 대체하십시오.

Note

역할에 다음 예에 표시된 것과 같은 권한이 있는지 확인합니다. *## ##* 데이터가 들어 있는 Amazon S3 버킷의 이름으로 교체합니다. 또한 *error-bucket#* 이 대량 가져오기 작업과 관련된 오류가 전송되는 Amazon S3 버킷의 이름으로 바꾸십시오.

```

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

```

```

{
  "adaptiveIngestion": adaptive-ingestion-flag,
  "deleteFilesAfterImport": delete-files-after-import-flag,
  "errorReportLocation": {
    "bucket": "error-bucket",
    "prefix": "error-bucket-prefix"
  },
  "files": [
    {
      "bucket": "data-bucket",
      "key": "data-bucket-key",

```

```

    "versionId": "data-bucket-version-id"
  }
],
"jobConfiguration": {
  "fileFormat": {
    "csv": {
      "columnNames": [ "column-name" ]
    }
  }
},
"jobName": "job-name",
"jobRoleArn": "job-role-arn"
}

```

Example 응답

```

{
  "jobId": "f8c031d0-01d1-4b94-90b1-afe8bb93b7e5",
  "jobStatus": "PENDING",
  "jobName": "myBulkImportJob"
}

```

대량 가져오기 작업 설명(AWS CLI)

[DescribeBulkImportJob](#) API 작업을 사용하여 대량 가져오기 작업에 대한 정보를 검색할 수 있습니다. 다음 예제에서는 `aws`를 사용하여 AWS CLI.

*job-ID*를 검색하려는 대량 가져오기 작업의 ID로 대체하세요.

```
aws iotsitewise describe-bulk-import-job --job-id job-ID
```

Example 응답

```

{
  "files": [
    {
      "bucket": "test-bucket",
      "key": "100Tags12Hours.csv"
    },
    {
      "bucket": "test-bucket",
      "key": "BulkImportData1MB.csv"
    }
  ]
}

```

```

    },
    {
      "bucket": "test-bucket",
      "key": "UnmodeledBulkImportData1MB.csv"
    }
  ],
  "errorReportLocation": {
    "prefix": "errors/",
    "bucket": "test-error-bucket"
  },
  "jobConfiguration": {
    "fileFormat": {
      "csv": {
        "columnNames": [
          "ALIAS",
          "DATA_TYPE",
          "TIMESTAMP_SECONDS",
          "TIMESTAMP_NANO_OFFSET",
          "QUALITY",
          "VALUE"
        ]
      }
    }
  },
  "jobCreationDate": 1645745176.498,
  "jobStatus": "COMPLETED",
  "jobName": "myBulkImportJob",
  "jobLastUpdateDate": 1645745279.968,
  "jobRoleArn": "arn:aws:iam::123456789012:role/DemoRole",
  "jobId": "f8c031d0-01d1-4b94-90b1-afe8bb93b7e5"
}

```

대량 가져오기 작업 목록(AWS CLI)

[ListBulkImportJobs](#) API 작업을 사용하여 모든 대량 가져오기 작업의 요약이 페이지별로 구분된 목록을 검색할 수 있습니다. 다음 예제에서는 `l` 를 사용합니다. AWS CLI

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED
```

Example 응답

```
{
```

```
"jobSummaries":[
  {
    "id":"bdbbfa52-d775-4952-b816-13ba1c7cb9da",
    "name":"myBulkImportJob",
    "status":"COMPLETED"
  },
  {
    "id":"15ffc641-dbd8-40c6-9983-5cb3b0bc3e6b",
    "name":"myBulkImportJob2",
    "status":"RUNNING"
  }
]
```

SiteWise 엣지 게이트웨이 사용

AWS IoT SiteWise 엣지 게이트웨이는 산업용 장비와 기기 사이의 중개자 역할을 합니다. AWS IoT SiteWise 엣지 게이트웨이는 온프레미스에서 데이터 수집 및 처리를 AWS IoT Greengrass V2 지원하는 게이트웨이에서 실행됩니다. AWS OpsHub for AWS IoT SiteWise 를 사용하여 SiteWise 엣지 게이트웨이를 관리하고 현장 운영을 모니터링할 수 있습니다.

로컬 장치의 SiteWise 모니터 포털을 사용하여 시설 내 데이터를 로컬로 모니터링할 수 있습니다. 자세한 내용은 [엣지에서 포털 활성화](#)을(를) 참조하세요.

주제

- [SiteWise 엣지 게이트웨이 요구 사항](#)
- [SiteWise 엣지 게이트웨이 생성](#)
- [로컬 장치에 SiteWise Edge 게이트웨이 소프트웨어 설치](#)
- [엣지 데이터 처리 활성화](#)
- [엣지에서의 데이터 처리](#)
- [AWS IoT SiteWise 퍼블리셔 구성 요소 구성](#)
- [데이터 소스 구성](#)
- [SiteWise 엣지 게이트웨이에 파트너 데이터 소스 추가](#)
- [팩 사용](#)
- [SiteWise 엣지 게이트웨이 관리](#)
- [지멘스 인더스트리얼 SiteWise 엣지의 러닝 엣지](#)
- [SiteWise 엣지 게이트웨이의 자산 필터링](#)
- [엣지에서의 AWS IoT SiteWise API 사용](#)
- [백업 및 복원 SiteWise 엣지 게이트웨이](#)
- [SiteWise 엣지 게이트웨이 설정 \(AWS IoT Greengrass Version 1\)](#)

SiteWise 엣지 게이트웨이 요구 사항

AWS IoT SiteWise 엣지 게이트웨이는 온프레미스에서 데이터 수집, 처리 및 게시를 지원하는 AWS IoT Greengrass 구성 요소 AWS IoT Greengrass V2 집합으로 실행됩니다. 에서 AWS IoT Greengrass V2 실행되는 SiteWise Edge 게이트웨이를 구성하려면 에서 게이트웨이를 만들고 SiteWise Edge 게이트웨이 소프트웨어를 실행하여 로컬 장치를 설정해야 합니다. AWS 클라우드

요구 사항

SiteWise Edge 게이트웨이 소프트웨어를 설치하고 실행하려면 로컬 장치가 다음 요구 사항을 충족해야 합니다.

- AWS IoT Greengrass V2 Core 소프트웨어 버전 [v2.3.0 이상](#)을 지원합니다. 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 [권한](#)을 참조하세요.
- 다음 지원 플랫폼 중 하나입니다.
 - 운영 체제: 우분투 20.04 이상
 - 아키텍처: x86_64 (AMD64) 또는 ARMv8 (Aarch64)
 - 운영체제: Red Hat Enterprise Linux (RHEL) 8
 - 아키텍처: x86_64 (AMD64) 또는 ARMv8 (Aarch64)
 - 운영체제: Amazon Linux 2
 - 아키텍처: x86_64 (AMD64) 또는 ARMv8 (Aarch64)
 - 운영체제: Debian 11
 - 아키텍처: x86_64 (AMD64) 또는 ARMv8 (Aarch64)
 - 운영체제: Windows Server 2019 이상
 - 아키텍처: x86_64 (AMD64)

Note

ARM 플랫폼은 데이터 수집 팩이 있는 SiteWise 에지 게이트웨이만 지원합니다. 데이터 프로세싱 팩은 지원되지 않습니다.

- 최소 4GB RAM
- SiteWise Edge 게이트웨이 소프트웨어에 사용할 수 있는 최소 10GB 디스크 공간.
- 엣지에서 데이터를 처리하려는 경우 로컬 장치가 다음 요구 사항도 충족해야 합니다. AWS IoT SiteWise
 - x86 64비트 쿼드코어 프로세서가 장착되어 있습니다.
 - 최소 16GB의 RAM이 있습니다.
 - Windows를 사용하는 경우 최소 32GB의 RAM이 있어야 합니다.
 - 최소 256GB의 디스크 여유 공간이 있습니다.

- 최소 디스크 공간 및 컴퓨팅 용량 요구 사항은 구현 및 사용 사례에 고유한 다양한 요인에 따라 달라집니다.
- 간헐적인 인터넷 연결을 위해 데이터 캐싱에 필요한 디스크 공간은 다음 요인에 따라 다릅니다.
 - 업로드되는 데이터 스트림의 개수
 - 초당 데이터 스트림당 데이터 포인트
 - 각 데이터 포인트의 크기
 - 통신 속도
 - 예기치 않은 네트워크 다운타임
- 데이터를 폴링하고 업로드하는 데 필요한 컴퓨팅 파워는 다음 요인에 따라 달라집니다.
 - 업로드되는 데이터 스트림의 개수
 - 초당 데이터 스트림당 데이터 포인트
- 다음 S3 버킷에 액세스하도록 로컬 디바이스를 구성하십시오 `iot-sitewise-gateway-<region>-748875242063`.
- 다음 포트에 액세스할 수 있도록 로컬 디바이스를 구성하십시오.
 - 로컬 장치는 포트 443에서 네트워크 인바운드 트래픽을 허용해야 합니다.
 - 로컬 장치는 포트 443과 8883에서 아웃바운드 트래픽을 허용해야 합니다.

필수 아웃바운드 서비스 엔드포인트의 전체 목록은 에지 게이트웨이의 [필수 서비스](#) 엔드포인트를 참조하십시오. AWS IoT SiteWise

- 다음 포트는 80, 443, 3001, 4569, 4572, 8000, 8081, 8082, 8084, 8085, 8445, 8086, 9000, 9500, 11080 및 50010에서 사용하도록 예약되어 있습니다. AWS IoT SiteWise예약된 포트를 트래픽용으로 사용하면 연결이 종료될 수 있습니다.

Note

스트림 관리자 구성 요소에는 자체 요구 AWS IoT Greengrass V2 사항이 있습니다. 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 [구성을](#) 참조하십시오.

- 자바 런타임 환경 (JRE) 버전 11 이상 디바이스의 PATH 환경 변수에서 Java를 사용할 수 있어야 합니다. Java를 사용하여 사용자 정의 구성 요소를 개발하려면 JDP(Java Development Kit)를 설치해야 합니다. [아마존 코레토 또는 OpenJDK를 사용하는 것이 좋습니다.](#)

에지 게이트웨이를 사용하려면 다음 권한이 있어야 합니다. SiteWise

Note

AWS IoT SiteWise 콘솔을 사용하여 SiteWise 에지 게이트웨이를 만드는 경우 이러한 권한이 자동으로 추가됩니다.

- 에지 게이트웨이의 IAM 역할은 AWS IoT Greengrass V2 디바이스의 SiteWise 에지 게이트웨이를 사용하여 자산 모델 데이터와 자산 데이터를 처리할 수 있도록 허용해야 합니다.

이 역할은 역할을 수입하기 위해 `credentials.iot.amazonaws.com` 서비스를 신뢰합니다.

권한 세부 정보

역할에는 다음과 같은 권한이 있어야 합니다.

- `iotsitewise` – 보안 주체가 엣지에서 자산 모델 데이터 및 자산 데이터를 검색할 수 있도록 합니다.
- `iot` – AWS IoT Greengrass V2 장치가 상호 작용할 수 있도록 허용합니다 AWS IoT.
- `logs` – AWS IoT Greengrass V2 디바이스에서 Amazon CloudWatch Logs로 로그를 전송할 수 있도록 허용합니다.
- `s3` – AWS IoT Greengrass V2 디바이스에서 Amazon S3에서 사용자 지정 구성 요소 아티팩트를 다운로드할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:List*",
        "iotsitewise:Describe*",
        "iotsitewise:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeCertificate",
        "logs:CreateLogGroup",

```

```

        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive",
        "iot:DescribeEndpoint"
    ],
    "Resource": "*"
}
]
}

```

SiteWise 에지 게이트웨이 생성

AWS IoT SiteWise 콘솔을 사용하여 SiteWise 에지 게이트웨이를 만들 수 있습니다. 이 절차에서는 자체 하드웨어에 설치할 자체 호스팅 SiteWise 에지 게이트웨이를 만드는 방법을 자세히 설명합니다. Siemens Industrial SiteWise Edge에서 실행되는 에지 게이트웨이를 만드는 방법에 대한 자세한 내용은 [이 링크를 참조하십시오. 지멘스 인더스트리얼 SiteWise 엣지의 러닝 엣지](#)

SiteWise 에지 게이트웨이 생성

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 에지 게이트웨이를 선택합니다.
3. 게이트웨이 생성을 선택합니다.
4. 배포 유형에서 셀프 호스팅 게이트웨이를 선택합니다.
5. SiteWise 에지 게이트웨이의 이름을 입력하거나 에서 생성한 AWS IoT SiteWise 이름을 사용합니다.
6. Greengrass 장치 OS에서 이 SiteWise Edge 게이트웨이를 설치할 장치의 운영 체제를 선택합니다.

Note

데이터 프로세싱 팩은 x86 플랫폼에서만 사용할 수 있습니다.

7. (선택 사항) 엣지에서 데이터를 처리하고 구성하려면 에지 기능에서 데이터 프로세싱 팩을 선택합니다.

 Note

회사 디렉터리의 사용자 그룹에 이 SiteWise Edge 게이트웨이에 대한 액세스 권한을 부여하려면 [을 참조하십시오. 엣지 기능 설정하기](#)

8. (선택 사항) 고급 구성에서 다음을 수행하십시오.
- Greengrass 코어 디바이스의 경우 다음 옵션 중 하나를 선택합니다.
 - 기본 설정 — 기본 설정을 사용하여 Greengrass 코어 디바이스를 AWS 자동으로 생성합니다. AWS IoT Greengrass V2
 1. Greengrass 코어 디바이스의 이름을 입력하거나 에서 생성한 이름을 사용합니다. AWS IoT SiteWise
 - 고급 설정 - 기존 Greengrass 코어 장치를 사용하거나 수동으로 생성하려는 경우 이 옵션을 선택합니다.
 1. Greengrass 코어 디바이스를 선택하거나 Greengrass 코어 디바이스 생성을 선택하여 AWS IoT Greengrass V2 콘솔에 Greengrass 코어 디바이스를 생성합니다. 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 AWS IoT Greengrass V2 [코어 장치 설정을](#) 참조하십시오.
9. 게이트웨이 생성을 선택합니다.
10. SiteWise Edge 게이트웨이 설치 프로그램 생성 대화 상자에서 생성 및 다운로드를 선택합니다. AWS IoT SiteWise 로컬 장치를 구성하는 데 사용할 수 있는 설치 프로그램을 자동으로 생성합니다.

 Important

설치 관리자 파일을 안전한 위치에 저장해야 합니다. 이 파일은 나중에 사용하게 됩니다.

이제 SiteWise Edge Gateway를 만들었으므로 [데이터 원본을](#) 추가하고 [게시자 구성 요소를](#) 구성한 다음 SiteWise Edge Gateway에서 데이터를 수신하여 클라우드로 전송하도록 합니다. AWS

로컬 장치에 SiteWise Edge 게이트웨이 소프트웨어 설치

SiteWise 에지 게이트웨이를 만든 후에는 로컬 장치에 SiteWise 에지 게이트웨이 소프트웨어를 설치해야 합니다. SiteWise 에지 게이트웨이 소프트웨어는 Linux 또는 Windows 서버 운영 체제가 설치된 로컬 장치에 설치할 수 있습니다.

Important

로컬 장치가 인터넷에 연결되어 있는지 확인하십시오.

Linux

다음 절차에서는 SSH를 사용하여 로컬 장치에 연결합니다. 또는 USB 플래시 드라이브나 기타 도구를 사용하여 설치 프로그램 파일을 로컬 장치로 전송할 수 있습니다. SSH를 사용하지 않으려면 아래의 2단계: SiteWise Edge 게이트웨이 소프트웨어 설치로 건너뛰십시오.

SSH 사전 요구 사항

SSH를 사용하여 디바이스에 연결하기 전에 다음 사전 요구 사항을 완료하세요.

- 디바이스의 IP 주소를 확인하세요.
- 사용자 이름을 가져와 장치에 연결하세요.
- 필요에 따라 로컬 컴퓨터에 SSH 클라이언트를 설치합니다.

로컬 컴퓨터에 기본적으로 SSH 클라이언트가 설치되어 있을 수 있습니다. 명령줄에 ssh를 입력하여 이를 확인할 수 있습니다. 컴퓨터가 명령을 인식하지 않는다면 SSH 클라이언트를 설치하면 됩니다.

- Linux 및 macOS - OpenSSH를 다운로드하여 설치합니다. 자세한 내용은 <https://www.openssh.com>을 참조하십시오

1단계: 설치 프로그램을 SiteWise Edge 게이트웨이 장치에 복사

다음 지침은 SSH 클라이언트를 사용하여 로컬 장치에 연결하는 방법을 설명합니다.

1. 디바이스에 연결하려면 컴퓨터의 터미널 창에서 다음 명령을 실행하고 **###** **###** IP를 권한 및 **IP ###** 높은 사용자 이름으로 대체합니다.

```
ssh username@IP
```

2. AWS IoT SiteWise 생성된 설치 프로그램 파일을 SiteWise Edge 게이트웨이 장치로 전송하려면 다음 명령을 실행합니다.

Note

- 설치 프로그램 파일을 저장하는 데 사용한 컴퓨터 상의 경로와 설치 프로그램 파일의 이름으로 바꿉니다 *path-to-saved-installer*.
- *IP ### # ##* IP 주소로 바꾸십시오.
- 설치 프로그램 파일을 받을 때 사용하는 로컬 장치의 *directory-to-receive-installer* 경로로 바꾸십시오.

```
scp path-to-saved-installer.sh user-name@IP-address:directory-to-receive-installer
```

2단계: SiteWise 에지 게이트웨이 소프트웨어 설치

다음 절차에서는 SiteWise Edge 게이트웨이 장치의 터미널 창에서 명령을 실행합니다.

1. 설치 파일에 실행 권한을 부여합니다.

```
chmod +x path-to-installer.sh
```

2. 설치 관리자를 실행합니다.

```
sudo ./path-to-installer.sh
```

Windows server

사전 조건

SiteWise Edge 게이트웨이 소프트웨어를 설치하려면 다음과 같은 사전 요구 사항이 있어야 합니다.

- 윈도우 서버 2019 이상 설치
- 관리자 권한
- PowerShell 버전 5.1 이상 설치됨

- SiteWise 에지 게이트웨이 설치 프로그램이 Windows 서버에 다운로드되어 프로비저닝됩니다.

1단계: 관리자 권한으로 실행 PowerShell

1. SiteWise 에지 게이트웨이를 설치하려는 Windows 서버에 관리자로 로그인합니다.
2. Windows 검색 PowerShell창에 입력합니다.
3. 검색 결과에서 Windows PowerShell 앱의 컨텍스트 (마우스 오른쪽 버튼 클릭) 메뉴를 엽니다. [Run as Administrator]를 선택합니다.

2단계: SiteWise 에지 게이트웨이 소프트웨어 설치

SiteWise Edge Gateway 장치의 터미널 창에서 다음 명령을 실행합니다.

1. SiteWise 에지 게이트웨이 설치 프로그램의 차단을 해제합니다.

```
unblock-file path-to-installer.ps1
```

2. 설치 관리자를 실행합니다.

```
./path-to-installer.ps1
```

Note

시스템에서 스크립트 실행이 비활성화된 경우 스크립트 실행 정책을 RemoteSigned으로 변경합니다.

```
Set-ExecutionPolicy RemoteSigned
```

엣지 데이터 처리 활성화

AWS IoT SiteWise Edge를 사용하여 장비 데이터를 로컬에서 수집, 저장, 구성 및 모니터링할 수 있습니다. SiteWise Edge를 사용하여 산업 데이터를 모델링하고, SiteWise Monitor를 사용하여 운영 직원이 로컬에서 데이터를 시각화할 수 있는 대시보드를 만들 수 있습니다. 데이터를 로컬에서 처리하여 로 전송하거나 API를 사용하여 온-프레미스로 처리할 수 있습니다. AWS 클라우드 AWS IoT SiteWise

AWS IoT SiteWise Edge를 사용하면 원시 데이터를 로컬에서 처리하고 집계된 데이터만 에 전송하도록 선택하여 대역폭 사용량과 클라우드 스토리지 비용을 최적화할 수 있습니다. AWS 클라우드

Note

- AWS IoT SiteWise 엣지 게이트웨이의 에지 데이터를 최대 SiteWise 30일까지 보존합니다. 데이터 보존 기간은 디바이스의 사용 가능한 디스크 공간에 따라 달라집니다.
- SiteWise Edge 게이트웨이와 30일 AWS 클라우드 동안 연결이 끊긴 경우 [데이터 프로세싱 팩은](#) 자동으로 비활성화됩니다.

엣지 기능 설정하기

AWS IoT SiteWise SiteWise Edge 게이트웨이가 데이터 수집 및 처리 방법을 결정하는 데 사용할 수 있는 다음 팩을 제공합니다. 팩을 선택하여 에지 게이트웨이에 SiteWise 엣지 기능을 활성화하십시오.

- 데이터 수집 팩을 사용하면 SiteWise 에지 게이트웨이가 여러 OPC-UA 서버에서 데이터를 수집한 다음 에지에서 서버로 데이터를 내보낼 수 있습니다. AWS 클라우드 SiteWise 에지 게이트웨이에 데이터 소스를 추가하면 활성화됩니다.
- 데이터 처리 팩을 사용하면 SiteWise 에지 게이트웨이가 엣지에서 장비 데이터를 처리할 수 있습니다. 예를 들어 자산 모델을 사용하여 지표와 변환을 계산할 수 있습니다. 자산 모델 및 자산에 대한 자세한 내용은 [산업 자산 모델링](#)을 참조하십시오.

Note

- 데이터 프로세싱 팩은 x86 플랫폼에서만 사용할 수 있습니다.
- 데이터 프로세싱 팩은 네트워크 프록시를 지원하지 않습니다.

에지 기능을 구성하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 에지 게이트웨이를 선택합니다.
3. SiteWise 에지 기능을 활성화하려는 에지 게이트웨이를 선택합니다.
4. 에지 기능 섹션에서 편집을 선택합니다.
5. Edge 기능 섹션에서 데이터 처리 팩 활성화 (추가 요금 발생) 를 선택합니다.

6. (선택 사항) Edge LDAP 연결 섹션에서 회사 디렉터리의 사용자 그룹에 이 Edge 게이트웨이에 대한 액세스 권한을 부여할 수 있습니다. SiteWise 사용자 그룹은 경량 디렉터리 액세스 프로토콜 (LDAP) 자격 증명을 사용하여 Edge 게이트웨이에 액세스할 수 있습니다. SiteWise 그런 다음 AWS IoT SiteWise 애플리케이션, AWS IoT SiteWise API 작업 또는 기타 도구를 사용하여 SiteWise 에지 게이트웨이를 관리할 수 있습니다. AWS OpsHub 자세한 정보는 [SiteWise 에지 게이트웨이 관리](#)을 참조하세요.

 Note

Linux 또는 Windows 자격 증명을 사용하여 SiteWise 에지 게이트웨이에 액세스할 수도 있습니다. 자세한 정보는 [Linux 운영 체제 자격 증명을 사용하여 SiteWise 에지 게이트웨이에 액세스](#)을 참조하세요.

- a. 활성화를 선택합니다.
- b. 공급자 이름에 LDAP 공급자의 이름을 입력합니다.
- c. 호스트 이름 또는 IP 주소에 LDAP 서버의 호스트 이름 또는 IP 주소를 입력합니다.
- d. 포트에 포트 번호를 입력합니다.
- e. 기본 고유 이름(DN)에 기본의 고유 이름(DN)을 입력합니다.

지원되는 속성 유형은 일반 이름 (CN), 지역 이름 (L), 이름 (ST), 조직 stateOrProvince 이름 (O), (OU), 국가 이름 (C), 주소 organizationalUnitName (도로), 도메인 구성 요소 (DC), 사용자 ID (UID) 입니다.

- f. 관리 그룹 DN의 경우 DN을 입력합니다.
- g. 사용자 그룹 DN의 경우 DN을 입력합니다.

7. 저장을 선택합니다.

이제 엣지 게이트웨이에서 엣지 기능을 활성화했으니 엣지에 대한 자산 모델을 구성해야 합니다. SiteWise 자산 모델 엣지 구성은 자산 속성이 계산되는 위치를 지정합니다. 엣지에서 모든 속성을 계산하거나 자산 모델 속성을 개별적으로 구성할 수 있습니다. 자산 모델 속성에는 [지표](#), [변환](#) 및 [측정](#)이 포함됩니다.

이러한 속성에 대한 자세한 내용은 [the section called “데이터 속성 정의”](#)을 참조하십시오.

자산 모델을 만든 후 엣지에 맞게 구성할 수 있습니다. 엣지에 대한 자산 모델 구성에 대한 자세한 내용은 [the section called “자산 모델 생성\(콘솔\)”](#)을 참조하십시오.

Note

자산 모델 및 대시보드는 10분마다 SiteWise Edge 게이트웨이와 Edge 게이트웨이 간에 자동으로 동기화됩니다. AWS 클라우드 로컬 SiteWise 에지 게이트웨이 애플리케이션에서 수동으로 동기화할 수도 있습니다.

엣지에서의 데이터 처리

엣지에서 에지 게이트웨이 데이터를 처리하려면 먼저 SiteWise 엣지에 대한 자산 모델을 구성해야 합니다. 자산 모델 엣지 구성은 자산 속성이 계산되는 위치를 지정합니다. 엣지에서 모든 속성을 계산하고 결과를 에지로 전송하도록 선택하거나 각 자산 속성을 계산할 위치를 개별적으로 사용자 지정할 수 있습니다. AWS 클라우드 자세한 설명은 [엣지 데이터 처리 활성화](#) 섹션을 참조하세요.

자산 속성에는 지표, 변환 및 측정이 포함됩니다.

- 지표는 지정된 기간에 걸친 자산의 집계된 데이터입니다. 기존 지표 데이터를 사용하여 새 지표를 계산할 수 있습니다. AWS IoT SiteWise 장기 보관을 위해 항상 지표를 AWS 클라우드로 전송합니다. AWS IoT SiteWise 기본적으로 AWS 클라우드에서 지표를 계산합니다. 에지에서 메트릭을 계산하도록 자산 모델을 구성할 수 있습니다. AWS IoT SiteWise 처리된 결과를 AWS 클라우드로 전송합니다.
- 변환은 자산 속성의 데이터 포인트를 한 양식에서 다른 양식으로 매핑하는 수학적 표현식입니다. 변환은 지표를 입력 데이터로 사용할 수 있으며 입력과 동일한 위치에서 계산 및 저장되어야 합니다. 에지에서 계산하도록 메트릭 입력을 구성하면 에지에서 관련 AWS IoT SiteWise 변환도 계산합니다.
- 측정값은 기본적으로 디바이스에서 수집하여 AWS 클라우드로 전송하는 원시 데이터 형식으로 지정됩니다. 이 데이터를 로컬 디바이스에 저장하도록 자산 모델을 구성할 수 있습니다.

이러한 속성에 대한 자세한 내용은 [the section called “데이터 속성 정의”](#)을 참조하십시오.

자산 모델을 만든 후 엣지에 맞게 구성할 수 있습니다. 엣지에 대한 자산 모델 구성에 대한 자세한 내용은 [the section called “자산 모델 생성\(콘솔\)”](#)을 참조하십시오.

Note

자산 모델 및 대시보드는 10분마다 AWS 클라우드와 SiteWise 에지 게이트웨이 간에 자동으로 동기화됩니다. 에서 수동으로 동기화할 수도 있습니다. [SiteWise 에지 게이트웨이 관리](#)

AWS IoT SiteWise REST API와 AWS Command Line Interface (AWS CLI) 를 사용하여 SiteWise 엣지에 있는 데이터를 에지 게이트웨이에 쿼리할 수 있습니다. SiteWise 엣지 게이트웨이에 엣지의 데이터를 쿼리하려면 먼저 다음 사전 요구 사항을 충족해야 합니다.

- REST API에 대한 자격 증명을 설정해야 합니다. 자격 증명 설정에 대한 자세한 내용은 [the section called “ SiteWise 에지 게이트웨이 관리”](#) 단원을 참조하십시오.
- SDK 엔드포인트는 엣지 게이트웨이의 IP 주소를 가리켜야 합니다. SiteWise 자세한 내용은 SDK 문서에서 확인할 수 있습니다. 예를 들어 AWS SDK for Java 2.x 개발자 가이드의 [사용자 지정 엔드포인트 지정](#)을 참조하십시오.
- SiteWise 에지 게이트웨이 인증서를 등록해야 합니다. SiteWise 에지 게이트웨이 인증서 등록에 대한 자세한 내용은 SDK 설명서에서 확인할 수 있습니다. 예를 들어 AWS SDK for Java 2.x 개발자 가이드의 [Node.js에 인증서 번들 등록](#)을 참조하십시오.

를 사용하여 AWS IoT SiteWise 데이터를 쿼리하는 방법에 대한 자세한 내용은 을 참조하십시오. [쿼리 데이터: AWS IoT SiteWise](#)

AWS IoT SiteWise 퍼블리셔 구성 요소 구성

AWS IoT SiteWise Edge Gateway를 만들고 소프트웨어를 설치한 후 Edge 게이트웨이가 데이터를 AWS 클라우드로 내보낼 수 있도록 Publisher 구성 요소를 설정합니다. SiteWise 자세한 내용은 AWS IoT Greengrass Version 2 개발자AWS IoT SiteWise [안내서의 게시자를](#) 참조하십시오.

Console

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 Edge 게이트웨이를 선택합니다.
3. 게시자를 구성하려는 SiteWise 에지 게이트웨이를 선택합니다.
4. 게시자 구성 섹션에서 편집을 선택합니다.
5. 게시 순서에서 다음 중 하나를 선택합니다.
 - 가장 오래된 데이터부터 게시 - SiteWise Edge 게이트웨이는 기본적으로 가장 오래된 데이터를 클라우드로 먼저 게시합니다.
 - 최신 데이터를 먼저 게시 — SiteWise 에지 게이트웨이는 최신 데이터를 클라우드로 먼저 게시합니다.
6. (선택 사항) SiteWise Edge 게이트웨이가 데이터를 압축하지 않도록 하려면 데이터 업로드 시 압축 활성화를 선택 취소하십시오.

7. (선택 사항) 이전 데이터를 게시하지 않으려면 만료된 데이터 제외를 선택하고 다음을 수행하십시오.
 - 마감 기간의 경우 값을 입력하고 단위를 선택합니다. 마감 기간은 5분에서 7일 사이여야 합니다. 예를 들어 마감 기간이 3일인 경우 3일이 지난 데이터는 클라우드에 게시되지 않습니다.
8. (선택 사항) 로컬 장치에서 데이터가 처리되는 방식에 대한 사용자 지정 설정을 지정하려면 로컬 저장소 설정을 선택하고 다음을 수행하십시오.
 - a. 보존 기간에 숫자를 입력하고 단위를 선택합니다. 보존 기간은 1분에서 30일 사이여야 하며, 교체 기간보다 길거나 같아야 합니다. 예를 들어 보존 기간이 14일인 경우 SiteWise Edge Gateway는 14일 동안 저장된 후 지정된 마감 기간보다 오래된 엣지의 모든 데이터를 삭제합니다.
 - b. 교체 기간의 경우 숫자를 입력하고 단위를 선택합니다. 순환 기간은 1분보다 크고 보존 기간과 같거나 짧아야 합니다. 예를 들어 순환 기간이 2일인 경우 SiteWise Edge 게이트웨이는 컷오프 기간보다 오래된 데이터를 일괄 처리하여 단일 파일에 저장합니다. SiteWise Edge 게이트웨이는 이들에 한 번씩 다음 로컬 디렉터리로 일괄 데이터를 전송합니다. /greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports
 - c. 스토리지 용량에는 1보다 크거나 같은 값을 입력합니다. 스토리지 용량이 2GB인 경우 로컬에 2GB 이상의 데이터가 저장되면 SiteWise Edge 게이트웨이가 데이터 삭제를 시작합니다.
9. 저장을 선택합니다.

AWS CLI

[UpdateGatewayCapabilityConfiguration](#) API를 사용하여 게시자를 구성할 수 있습니다. `capabilityNamespace` 파라미터를 `iotsitewise:publisher:2`로 설정합니다.

게시자는 사용자 정의할 수 있는 다음과 같은 구성 매개변수를 제공합니다.

SiteWisePublisherConfiguration

publishingOrder

데이터가 클라우드에 게시되는 순서. 이 매개 변수의 값은 다음 중 하나일 수 있습니다.

- `TIME_ORDER`(가장 오래된 데이터부터 게시) - 기본적으로 가장 오래된 데이터가 클라우드에 먼저 게시됩니다.
- `RECENT_DATA`(최신 데이터를 먼저 게시) — 최신 데이터가 클라우드에 먼저 게시됩니다.

dropPolicy

(선택 사항) 클라우드에 게시되는 데이터를 제어하는 정책입니다.

cutoffAge

마감 기간보다 이전의 데이터는 클라우드에 게시되지 않습니다. 마감 기간은 5분에서 7일 사이여야 합니다.

마감 연령을 지정할 때 m, h, d를 사용할 수 있습니다. m은 분, h는 시간, d는 일을 나타냅니다.

exportPolicy

(선택 사항) 엣지의 데이터 스토리지를 관리하는 정책입니다. 이 정책은 마감 기간 이전의 데이터에 적용됩니다.

retentionPeriod

SiteWise 엣지 게이트웨이는 지정된 보존 기간 동안 저장된 후 컷오프 기간보다 이른 엣지의 모든 데이터를 로컬 스토리지에서 삭제합니다. 보존 기간은 1분에서 30일 사이여야 하며, 교체 기간보다 길거나 같아야 합니다.

보존 기간을 지정할 때 m, h, d를 사용할 수 있습니다. m은 분, h는 시간, d는 일을 나타냅니다.

rotationPeriod

마감 기간보다 이른 데이터를 일괄 처리하여 단일 파일에 저장하는 시간 간격입니다. SiteWise Edge 게이트웨이는 각 순환 기간이 끝날 때 하나의 데이터 배치를 다음 로컬 디렉터리로 전송합니다. `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports` 교체 기간은 1분보다 크며 보존 기간과 같거나 작아야 합니다.

교체 기간을 지정할 때 m, h, d를 사용할 수 있습니다. m은 분, h는 시간, d는 일을 나타냅니다.

exportSizeLimitGB

로컬에 저장된 최대 허용 데이터 크기(GB)입니다. 이 할당량을 위반하면 SiteWise 엣지 게이트웨이는 로컬에 저장된 데이터 크기가 할당량 이하가 될 때까지 가장 빠른 데이터를 삭제하기 시작합니다. 이 파라미터의 값은 1보다 크거나 같아야 합니다.

Example 게시자 구성:

게시자 네임스페이스: `iotsitewise:publisher:2`

```
{
  "SiteWisePublisherConfiguration": {
    "publishingOrder": "TIME_ORDER",
    "dropPolicy": {
      "cutoffAge": "7d",
      "exportPolicy": {
        "retentionPeriod": "7d",
        "rotationPeriod": "6h",
        "exportLocation": "/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/
exports",
        "exportSizeLimitGB": 10
      }
    }
  }
}
```

데이터 소스 구성

에지 게이트웨이를 설정한 후에는 AWS IoT SiteWise 에지 게이트웨이가 로컬 산업 장비의 데이터를 수집할 수 있도록 데이터 소스를 구성할 수 있습니다. AWS IoT SiteWise 각 소스는 SiteWise Edge 게이트웨이가 산업 데이터 스트림을 연결하고 검색하는 OPC-UA 서버와 같은 로컬 서버를 나타냅니다. SiteWise 에지 게이트웨이 설정에 대한 자세한 내용은 [을 참조하십시오. AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 구성](#)

Note

AWS IoT SiteWise 소스를 추가하거나 편집할 때마다 SiteWise 에지 게이트웨이를 다시 시작합니다. SiteWise 에지 게이트웨이는 다시 시작하는 동안에는 데이터를 수집하지 않습니다. 에지 게이트웨이를 다시 시작하는 데 걸리는 시간은 SiteWise 에지 게이트웨이 소스의 태그 수에 따라 달라집니다. SiteWise 재시작 시간은 몇 초 (태그가 거의 없는 SiteWise 에지 게이트웨이의 경우) 에서 몇 분 (태그가 많은 SiteWise 에지 게이트웨이의 경우) 까지 걸릴 수 있습니다.

소스를 생성한 후에는 데이터 스트림을 자산 속성과 연결할 수 있습니다. 자산을 만들고 사용하는 방법에 대한 자세한 내용은 [산업 자산 모델링 및 산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.

CloudWatch 지표를 보고 데이터 소스가 연결되어 있는지 확인할 수 AWS IoT SiteWise 있습니다. 자세한 정보는 [AWS IoT Greengrass Version 2 게이트웨이 지표](#)를 참조하세요.

현재, AWS IoT SiteWise 지원되는 데이터 소스 프로토콜은 다음과 같습니다.

- [OPC-UA](#) — 산업 machine-to-machine 자동화를 위한 A (M2M) 통신 프로토콜.

Note

SiteWise AWS IoT Greengrass V2 현재 실행 중인 에지 게이트웨이는 Modbus TCP 및 이더넷 IP 소스를 지원하지 않습니다.

주제

- [OPC-UA 소스 설정](#)
- [데이터 소스 인증 구성](#)
- [소스 서버 데이터의 목적지 선택](#)

OPC-UA 소스 설정

AWS IoT SiteWise 콘솔 또는 SiteWise 에지 게이트웨이 기능을 사용하여 로컬 OPC-UA 서버를 나타내는 OPC-UA 소스를 정의하고 SiteWise 에지 게이트웨이에 추가할 수 있습니다.

주제

- [OPC-UA 소스 구성 \(콘솔\)](#)
- [OPC-UA 소스\(CLI\)를 구성합니다.](#)
- [OPC-UA 원본 서버가 에지 게이트웨이를 신뢰하도록 지원 SiteWise](#)
- [OPC-UA를 사용하여 데이터 모으기 범위 필터링](#)
- [OPC-UA 노드 필터 사용](#)

OPC-UA 소스 구성 (콘솔)

콘솔을 사용하여 OPC-UA 소스를 구성하려면 AWS IoT SiteWise

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.

2. 탐색 창에서 게이트웨이를 선택합니다.
3. SiteWise OPC-UA 소스를 추가하려면 에지 게이트웨이를 선택합니다.
4. 데이터 소스 추가를 선택합니다.
5. 소스 이름을 입력합니다.
6. 데이터 소스 서버의 로컬 엔드포인트를 입력합니다. 엔드포인트는 IP 주소 또는 호스트 이름일 수 있습니다. 로컬 엔드포인트에 포트 번호를 추가할 수도 있습니다. 예를 들어 로컬 엔드포인트는 다음과 같을 수 있습니다. **opc.tcp://203.0.113.0:49320**

Note

SiteWise 엣지 게이트웨이에 새로 나온 Siemens Industrial Edge 디바이스가 있고 엣지 애플리케이션과 동일한 Siemens 산업용 에지 디바이스에서 실행되는 Edge OPC UA 서버 애플리케이션에서 데이터를 수집하려면 다음을 입력하십시오. Deployment type AWS IoT SiteWise **opc.tcp://ie-opcua:48010**

7. (선택 사항) 선택할 노드 ID의 경우, 수집되는 데이터 스트림을 제한하는 노드 필터를 추가하십시오. AWS 클라우드기본적으로 SiteWise 에지 게이트웨이는 서버의 루트 노드를 사용하여 모든 데이터 스트림을 수집합니다. 노드 필터를 사용하면 모델링하는 데이터에 대한 경로만 포함함으로써 SiteWise 에지 게이트웨이의 시작 시간과 CPU 사용량을 줄일 수 있습니다. AWS IoT SiteWise 기본적으로 SiteWise Edge 게이트웨이는 로 시작하는 경로를 제외한 모든 OPC-UA 경로를 업로드합니다. /Server/ OPC-UA 노드 필터를 정의하려면 노드 경로와 * 및 ** 와일드카드 문자를 사용하면 됩니다. 자세한 정보는 [OPC-UA 노드 필터 사용](#)을 참조하세요.
8. 대상의 경우 소스 데이터의 대상을 선택합니다.
 - AWS IoT SiteWise 실시간 — 데이터를 AWS IoT SiteWise 스토리지로 직접 전송하려면 이 옵션을 선택합니다. 실시간으로 데이터를 수집 및 모니터링하고 엣지에서 데이터를 처리합니다.
 - AWS IoT SiteWise Amazon S3를 사용하여 버퍼링 — 데이터를 패킷 형식으로 Amazon S3로 전송한 다음 스토리지로 가져옵니다. AWS IoT SiteWise 데이터를 일괄적으로 수집하고 비용 효율적인 방식으로 기록 데이터를 저장하려면 이 옵션을 선택하십시오. 원하는 Amazon S3 버킷 위치와 Amazon S3에 데이터를 업로드할 빈도를 구성할 수 있습니다. 또한 데이터를 수집한 후 데이터를 어떻게 처리할지 선택할 수 있습니다. AWS IoT SiteWise Amazon S3에서 데이터를 모두 SiteWise 사용할 수 있도록 선택하거나 Amazon S3에서 자동으로 삭제하도록 선택할 수 있습니다.
 - Amazon S3 버킷은 스테이징 및 버퍼링 메커니즘이며 패킷 형식의 파일을 지원합니다.
 - AWS IoT SiteWise 스토리지로 데이터 가져오기 확인란을 선택하면 데이터가 먼저 Amazon S3에 업로드된 다음 AWS IoT SiteWise 스토리지로 업로드됩니다.

- Amazon S3에서 데이터 삭제 확인란을 선택하면 데이터를 SiteWise 스토리지로 가져온 후 Amazon S3에서 삭제됩니다.
- Amazon S3에서 데이터 삭제 확인란의 선택을 취소하면 데이터가 Amazon S3와 SiteWise 스토리지에 모두 저장됩니다.
- AWS IoT SiteWise 스토리지로 데이터 가져오기 확인란의 선택을 취소하면 데이터는 Amazon S3에만 저장됩니다. SiteWise 스토리지로 가져올 수는 없습니다.

AWS IoT SiteWise 제공되는 다양한 스토리지 옵션에 [데이터 스토리지 관리](#) 대한 자세한 내용은 을 참조하십시오. 가격 옵션에 대한 자세한 내용은 [AWS IoT SiteWise 가격을](#) 참조하십시오.

- AWS IoT Greengrass 스트림 관리자 — AWS IoT Greengrass 스트림 관리자를 사용하여 채널 입력, Amazon Kinesis Data Streams의 스트림 AWS IoT Analytics, 자산 속성 또는 Amazon Simple Storage Service (Amazon S3) AWS IoT SiteWise의 객체 등 데이터를 전송할 수 있습니다. AWS 클라우드 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 AWS IoT Greengrass [Core에서 데이터 스트림 관리를](#) 참조하십시오.

AWS IoT Greengrass 스트림 이름을 입력합니다.

데이터 소스를 구성하는 동안 선택을 위한 노드 ID는 데이터 흐름의 대상을 결정하는 데 사용됩니다.

- Amazon S3를 사용하여 동일한 데이터를 AWS IoT SiteWise 실시간 및AWS IoT SiteWise 버퍼링에 게시하는 경우 두 대상에 모두 게시하는 두 개의 데이터 소스를 추가해야 합니다.
- 일부는 AWS IoT SiteWise 실시간으로 게시되고 나머지 부분은 Amazon S3를 사용하여AWS IoT SiteWise Buffered에 게시되도록 데이터를 분할하려면 다음 데이터 별칭을 필터링해야 합니다.

```
/Alias01/Data1
/Alias02/Data1
/Alias03/Data1
/Alias03/Data2
```

예를 들어, Amazon S3를 사용하여 /**/Data1 노드 필터, AWS IoT SiteWise 실시간을 가리키는 데이터 소스와 /**/Data2 AWS IoT SiteWise 버퍼링을 가리키는 다른 데이터 소스를 추가할 수 있습니다.

9. 고급 구성 창에서 다음을 수행할 수 있습니다.

- a. 원본 서버와 SiteWise Edge 게이트웨이 간에 전송되는 연결 및 데이터에 대한 메시지 보안 모드를 선택합니다. 이 필드는 OPC-UA 보안 정책과 메시지 보안 모드의 조합입니다. OPC-UA 서버에 지정한 것과 동일한 보안 정책 및 메시지 보안 모드를 선택합니다.
- b. 소스에 인증이 필요한 경우 인증 구성 목록에서 AWS Secrets Manager 암호를 선택합니다. SiteWise Edge 게이트웨이는 이 데이터 원본에 연결할 때 이 암호의 인증 자격 증명을 사용합니다. 암호를 데이터 원본 인증에 사용하려면 SiteWise Edge Gateway AWS IoT Greengrass 구성 요소에 암호를 연결해야 합니다. 자세한 정보는 [the section called “데이터 소스 인증 구성”](#)을 참조하세요.

 Tip

데이터 서버에는 [Allow anonymous login(익명 로그인 허용)]이라는 옵션이 있을 수 있습니다. 이 옵션이 [Yes(예)]인 경우 원본에 인증이 필요하지 않습니다.

- c. (선택 사항) 데이터 스트림 접두사를 입력합니다. SiteWise 에지 게이트웨이는 이 소스의 모든 데이터 스트림에 이 접두사를 추가합니다. 데이터 스트림 접두사를 사용하여 서로 다른 소스에서 이름이 같은 데이터 스트림을 구분합니다. 각 데이터 스트림은 계정 내에서 고유한 이름을 가져야 합니다.
- d. (선택 사항) 속성 그룹의 경우 새 그룹 추가를 선택합니다.
 - i. 속성 그룹의 이름을 입력합니다.
 - ii. 속성의 경우:
 1. 노드 경로의 경우 OPC-UA 노드 필터를 추가하여 업로드되는 OPC-UA 경로를 제한하십시오. AWS IoT SiteWise 형식은 노드 ID 선택 형식과 비슷합니다.
 - iii. 그룹 설정에서 다음을 수행합니다.
 1. 데이터 품질 설정에서 Collector가 AWS IoT SiteWise 수집할 데이터 품질 유형을 선택합니다.
 2. 스캔 모드 설정의 경우 다음과 같은 표준 구독 속성을 구성하십시오.
 - 스캔 모드의 경우 다음 중 하나를 선택합니다. 스캔 모드에 대한 자세한 내용은 [the section called “OPC-UA를 사용하여 데이터 모으기 범위 필터링하기”](#).
 - 모든 데이터 포인트를 전송하려면 구독을 선택하고 다음을 설정합니다.
 - [데이터 변경 트리거](#) — 데이터 변경 알림을 시작하는 조건입니다.

- [구독 대기열 크기](#) - 모니터링되는 항목에 대한 알림이 대기열에 있는 특정 지표에 대한 OPC—UA 서버의 대기열 깊이입니다.
- [구독 게시 간격](#) — 구독이 생성될 때 지정된 게시 주기 간격 (밀리초) 입니다.
- 스냅샷 간격 - AWS IoT SiteWise Edge가 꾸준한 데이터 스트림을 수집하도록 하기 위한 스냅샷 빈도 제한 시간 설정입니다.
- 스캔 속도 - SiteWise Edge 게이트웨이가 레지스터를 읽기를 원하는 속도입니다. AWS IoT SiteWise Edge 게이트웨이에 허용되는 최소 스캔 속도를 자동으로 계산합니다. SiteWise
- 특정 간격으로 데이터 포인트를 전송하려면 Poll을 선택하고 스캔 속도를 입력합니다.

3. Subscribe의 스캔 모드를 선택한 경우 소스의 데드밴드 유형 및 관련 설정을 구성하십시오. 이를 통해 소스가 어떤 데이터를 전송하고 어떤 데이터를 AWS IoT SiteWise 삭제할지 제어할 수 있습니다. 데드밴드 설정에 대한 자세한 내용은 [the section called "OPC-UA를 사용하여 데이터 모으기 범위 필터링하기"](#)을 참조하세요.

10. 저장을 선택합니다.

OPC-UA 소스(CLI)를 구성합니다.

를 사용하여 에지 게이트웨이의 OPC-UA 데이터 소스를 정의할 수 있습니다 SiteWise . AWS CLI이렇게 하려면 OPC-UA 기능 구성 JSON 파일을 만들고 [update-gateway-capability-configuration](#) 명령을 사용하여 에지 게이트웨이 구성을 업데이트하십시오. SiteWise 단일 기능 구성에서 모든 OPC-UA 소스를 정의해야 합니다.

이 기능에는 다음과 같은 네임스페이스가 있습니다.

- `iotsitewise:opcuacollector:2`

요청 구문

```
{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",

```

```

    "certificateChain": "string",
  },
  "endpointUri": "string",
  "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256" |
"AES128_SHA256_RSAOAE" | "AES256_SHA256_RSAPSS",
  "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
  "identityProvider": {
    "type": "Anonymous" | "Username",
    "usernameSecretArn": "string"
  },
  "nodeFilterRules": [
    {
      "action": "INCLUDE",
      "definition": {
        "type": "OpcUaRootPath",
        "rootPath": "string"
      }
    }
  ]
},
"measurementDataStreamPrefix": "string"
"destination": {
  "type": "StreamManager",
  "streamName": "string",
  "streamBufferSize": integer
},
"propertyGroups": [
  {
    "name": "string",
    "nodeFilterRuleDefinitions": [
      {
        "type": "OpcUaRootPath",
        "rootPath": "string"
      }
    ]
  },
  "deadband": {
    "type": "PERCENT" | "ABSOLUTE",
    "value": double,
    "eguMin": double,
    "eguMax": double,
    "timeoutMilliseconds": integer
  },
  "scanMode": {
    "type": "EXCEPTION" | "POLL",

```

```

    "rate": integer
  },
  "dataQuality": {
    "allowGoodQuality": true | false,
    "allowBadQuality": true | false,
    "allowUncertainQuality": true | false
  },
  "subscription": {
    "dataChangeTrigger": "STATUS" | "STATUS_VALUE" | "STATUS_VALUE_TIMESTAMP",
    "queueSize": integer,
    "publishingIntervalMilliseconds": integer,
    "snapshotFrequencyMilliseconds": integer
  }
}
]
}
]
}

```

요청 본문

소스

다음 정보를 포함하는 각 OPC-UA 원본 정의 구조의 목록입니다.

이름

고유하고 친숙한 원본 이름입니다.

endpoint

다음 정보가 포함된 엔드포인트 구조입니다.

인증서 트러스트

다음 정보가 포함된 인증서 신뢰 정책 구조:

type

원본에 대한 인증서 신뢰 모드입니다. 다음 중 하나를 선택합니다.

- **TrustAny**— SiteWise 엣지 게이트웨이는 OPC-UA 소스에 연결할 때 모든 인증서를 신뢰합니다.
- **X509**— SiteWise 에지 게이트웨이는 OPC-UA 소스에 연결할 때 X.509 인증서를 신뢰합니다. 이 옵션을 선택하는 경우, `certificateTrust`에 `certificateBody`을 정

의해야 합니다. `certificateTrust`에 `certificateChain`을 정의할 수도 있습니다.

인증서 기관

(선택 사항) X.509 인증서의 본문입니다.

`certificateTrust`에서 `type`에 대해 X509을 선택하는 경우, 이 필드는 필수입니다.

인증서 체인

(선택 사항) X.509 인증서에 대한 신뢰 체인입니다.

이 필드는 `certificateTrust`에서 `type`에 대해 X509를 선택할 때만 사용됩니다.

엔드포인트 URI

OPC-UA 원본의 로컬 엔드포인트입니다. 로컬 엔드포인트를 예를 들면 `opc.tcp://203.0.113.0:49320`과 같습니다.

보안 정책

OPC-UA 원본에서 읽은 메시지를 보호할 수 있도록 사용할 보안 정책입니다. 다음 중 하나를 선택합니다.

- NONE— SiteWise 에지 게이트웨이는 OPC-UA 소스의 메시지를 보호하지 않습니다. 다른 보안 정책을 선택하는 것이 좋습니다. 이 옵션을 선택하는 경우, `messageSecurityMode`에 대해 NONE을 선택해야 합니다
- BASIC256_SHA256— Basic256Sha256 보안 정책.
- AES128_SHA256_RSA0AEP— Aes128_Sha256_Rsa0aep 보안 정책.
- AES256_SHA256_RSAPSS— Aes256_Sha256_RsaPss 보안 정책.
- BASIC128_RSA15— (더 이상 사용되지 않음) Basic128Rsa15 보안 정책은 더 이상 안전한 것으로 간주되지 않으므로 OPC-UA 사양에서 더 이상 사용되지 않습니다. 다른 보안 정책을 선택하는 것이 좋습니다. 자세한 내용은 [Basic128Rsa15](#)를 참조하세요.
- BASIC256— (더 이상 사용되지 않음) Basic256 보안 정책은 더 이상 안전한 것으로 간주되지 않으므로 OPC-UA 사양에서 더 이상 사용되지 않습니다. 다른 보안 정책을 선택하는 것이 좋습니다. 자세한 내용은 [Basic256](#)를 참조하세요.

Important

이외의 NONE 보안 정책을 선택하는 경우 SIGN 또는 SIGN_AND_ENCRYPT 정책을 선택해야 합니다 `messageSecurityMode`. 또한 SiteWise Edge 게이트웨이를 신뢰

하도록 원본 서버를 구성해야 합니다. 자세한 정보는 [OPC-UA 원본 서버가 에지 게이트웨이를 신뢰하도록 지원 SiteWise](#) 을 참조하세요.

메시지 SecurityMode

OPC-UA 원본에 대한 연결을 보호하는 데 사용할 메시지 보안 모드입니다. 다음 중 하나를 선택합니다.

- NONE— SiteWise 에지 게이트웨이는 OPC-UA 소스에 대한 연결을 보호하지 않습니다. 다른 메시지 보안 모드를 선택하는 것이 좋습니다. 이 옵션을 선택하는 경우, securityPolicy에 대해 NONE을 선택해야 합니다
- SIGN— SiteWise 에지 게이트웨이와 OPC-UA 소스 간에 전송되는 데이터는 서명되지만 암호화되지는 않습니다.
- SIGN_AND_ENCRYPT – 게이트웨이와 OPC-UA 원본 간 전송 중 데이터는 서명되고 암호화됩니다.

Important

NONE 이외의 메시지 보안 모드를 선택하는 경우 NONE 이외의 securityPolicy를 선택해야 합니다. 또한 원본 서버가 SiteWise Edge 게이트웨이를 신뢰하도록 구성해야 합니다. 자세한 정보는 [OPC-UA 원본 서버가 에지 게이트웨이를 신뢰하도록 지원 SiteWise](#) 을 참조하세요.

ID 제공자

다음 정보가 포함된 자격 증명 공급자 구조입니다.

type

원본에 필요한 인증 보안 인증의 유형입니다. 다음 중 하나를 선택합니다.

- Anonymous – 원본에 연결하려면 인증이 필요하지 않습니다.
- Username – 원본에 연결하려면 사용자 이름과 암호가 필요합니다. 이 옵션을 선택하는 경우, identityProvider에 usernameSecretArn을 정의해야 합니다.

사용자 이름 SecretArn

(선택 사항) 시크릿의 ARN. AWS Secrets Manager SiteWise 에지 게이트웨이는 이 소스에 연결할 때 이 암호의 인증 자격 증명을 사용합니다. 암호를 소스 인증에 사용하려면

SiteWise 에지 게이트웨이의 IoT SiteWise 커넥터에 암호를 연결해야 합니다. 자세한 정보는 [데이터 소스 인증 구성](#)을 참조하세요.

identityProvider에서 type에 대해 Username을 선택하는 경우, 이 필드는 필수입니다.

노드 FilterRules

클라우드로 전송할 OPC-UA 데이터 스트림 경로를 정의하는 노드 필터 규칙 구조 목록입니다. AWS 노드 필터를 사용하면 모델링하는 데이터에 대한 경로만 포함함으로써 SiteWise Edge 게이트웨이의 시작 시간과 CPU 사용량을 줄일 수 있습니다. AWS IoT SiteWise기본적으로 SiteWise Edge 게이트웨이는 로 시작하는 경로를 제외한 모든 OPC-UA 경로를 업로드합니다. /Server/ OPC-UA 노드 필터를 정의하려면 노드 경로와 * 및 ** 와일드카드 문자를 사용하면 됩니다. 자세한 내용은 [OPC-UA 노드 필터 사용](#) 섹션을 참조하세요.

각 목록 구조에는 다음 정보가 포함되어야 합니다.

action

이 노드 필터 규칙에 대한 작업입니다. 다음과 같은 옵션을 선택할 수 있습니다.

- INCLUDE— SiteWise 에지 게이트웨이에는 이 규칙과 일치하는 데이터 스트림만 포함됩니다.

정의

다음 정보가 포함된 노드 필터 규칙 구조:

type

이 규칙의 노드 필터 경로 유형입니다. 다음과 같은 옵션을 선택할 수 있습니다.

- OpcUaRootPath— SiteWise 에지 게이트웨이는 OPC-UA 경로 계층 구조의 루트를 기준으로 이 노드 필터 경로를 평가합니다.

루트패스

OPC-UA 경로 계층의 루트에 대해 평가할 노드 필터 경로입니다. 이 경로는 /로 시작되어야 합니다.

측정 접두사 DataStream

원본의 모든 데이터 스트림 앞에 추가할 문자열입니다. SiteWise Edge 게이트웨이는 이 소스의 모든 데이터 스트림에 이 접두사를 추가합니다. 데이터 스트림 접두사를 사용하여 서로 다른 소스에서 이름이 같은 데이터 스트림을 구분합니다. 각 데이터 스트림은 계정 내에서 고유한 이름을 가져야 합니다.

프로퍼티/그룹

(선택 사항) 프로토콜에서 요청한 deadband 및 scanMode을 정의하는 속성 그룹 목록입니다.

이름

속성 그룹의 이름입니다. 고유 식별자여야 합니다.

데드밴드

deadband 구조에는 다음 정보가 포함되어 있습니다.

type

지원되는 데드밴드 유형입니다. 허용되는 값은 ABSOLUTE 및 PERCENT입니다.

USD 상당

데드밴드의 값입니다. type가 ABSOLUTE인 경우 이 값은 단위가 없는 이중 값입니다. type가 PERCENT인 경우 이 값은 1과 100 사이의 두 배입니다.

구민

(선택 사항) PERCENT 데드밴드를 사용할 때의 엔지니어링 단위 최소값입니다. OPC-UA 서버에 엔지니어링 유닛이 구성되어 있지 않은 경우 이를 설정합니다.

예구맥스

(선택 사항) PERCENT 데드밴드를 사용할 때의 엔지니어링 단위 최대값입니다. OPC-UA 서버에 엔지니어링 유닛이 구성되어 있지 않은 경우 이를 설정합니다.

타임아웃 (밀리초)

타임아웃까지 남은 시간 (밀리초). 최소값은 100입니다.

스캔 모드

scanMode 구조에는 다음 정보가 포함되어 있습니다.

type

scanMode의 지원되는 유형입니다. 허용되는 값은 POLL 및 EXCEPTION입니다.

속도

스캔 모드의 샘플링 간격입니다.

노드 FilterRule 정의

(선택 사항) 속성 그룹에 포함할 노드 경로 목록입니다. 속성 그룹은 겹칠 수 없습니다. 이 필드에 값을 지정하지 않으면 그룹에 루트 아래의 모든 경로가 포함되므로 추가 속성 그룹을

생성할 수 없습니다. `nodeFilterRuleDefinitions` 구조에는 다음 정보가 포함되어 있습니다.

type

지원되는 유형은 `OpcUaRootPath`뿐입니다. 이것은 `rootPath` 값이 OPC-UA 브라우저 공간의 루트에 상대적인 경로임을 지정합니다.

루트 경로

속성 그룹에 포함할 경로(루트에 대한 경로)를 지정하는 쉼표로 구분된 목록입니다.

기능 구성 예제

다음 예제는 JSON 파일에 저장된 페이로드에서 OPC-UA SiteWise 에지 게이트웨이 기능 구성을 정의합니다.

```
aws iotsitewise update-gateway-capability-configuration \
--capability-namespace "iotsitewise:opcuacollector:2" \
--capability-configuration file://opc-ua-configuration.json
```

Example : OPC-UA 소스 구성

다음 `opc-ua-configuration.json` 파일은 안전하지 않은 기본 OPC-UA 원본 구성을 정의합니다.

```
{
  "sources": [
    {
      "name": "Wind Farm #1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.0:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
          "type": "Anonymous"
        },
        "nodeFilterRules": []
      },
      "measurementDataStreamPrefix": ""
    }
  ]
}
```

```

]
}

```

Example : 정의된 속성 그룹이 있는 OPC-UA 소스 구성

다음 `opc-ua-configuration.json` 파일은 정의된 속성 그룹이 있는 안전하지 않은 기본 OPC-UA 소스 구성을 정의합니다.

```

{
  "sources": [
    {
      "name": "source1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://10.0.0.9:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
          "type": "Anonymous"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank"
            }
          }
        ]
      },
      "measurementDataStreamPrefix": "propertyGroups",
      "propertyGroups": [
        {
          "name": "Deadband_Abs_5",
          "nodeFilterRuleDefinitions": [
            {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank/Temperature/TT-001"
            },
            {
              "type": "OpcUaRootPath",

```

```

        "rootPath": "/Utilities/Tank/Temperature/TT-002"
      }
    ],
    "deadband": {
      "type": "ABSOLUTE",
      "value": 5.0,
      "timeoutMilliseconds": 120000
    }
  },
  {
    "name": "Polling_10s",
    "nodeFilterRuleDefinitions": [
      {
        "type": "OpcUaRootPath",
        "rootPath": "/Utilities/Tank/Pressure/PT-001"
      }
    ],
    "scanMode": {
      "type": "POLL",
      "rate": 10000
    }
  },
  {
    "name": "Percent_Deadband_Timeout_90s",
    "nodeFilterRuleDefinitions": [
      {
        "type": "OpcUaRootPath",
        "rootPath": "/Utilities/Tank/Flow/FT-*"
      }
    ],
    "deadband": {
      "type": "PERCENT",
      "value": 5.0,
      "eguMin": -100,
      "eguMax": 100,
      "timeoutMilliseconds": 90000
    }
  }
]
}

```

Example : 속성을 포함한 OPC-UA 소스 구성

다음 `opc-ua-configuration.json`의 JSON 예제는 다음 속성을 가진 OPC-UA 원본 구성을 정의합니다.

- 모든 인증서를 신뢰합니다.
- BASIC256 보안 정책을 사용하여 메시지를 보호합니다.
- SIGN_AND_ENCRYPT 모드를 사용하여 연결을 보호합니다.
- Secrets Manager 암호에 저장된 인증 보안 인증을 사용합니다.
- 경로가 `/WindFarm/2/WindTurbine/`로 시작하는 데이터 스트림을 제외한 데이터 스트림을 필터링합니다.
- 모든 데이터 스트림 경로의 첫 부분에 `/Washington`을 추가하여 다른 영역에서 이 “Wind Farm #2”와 “Wind Farm #2”를 구분합니다.

```
{
  "sources": [
    {
      "name": "Wind Farm #2",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.1:49320",
        "securityPolicy": "BASIC256",
        "messageSecurityMode": "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Username",
          "usernameSecretArn":
"arn:aws:secretsmanager:region:123456789012:secret:greenrass-windfarm2-auth-1ABCDE"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "/WindFarm/2/WindTurbine/"
            }
          }
        ]
      }
    }
  ],
}
```

```

        "measurementDataStreamPrefix": "/Washington"
    }
]
}

```

Example : 인증서 신뢰를 사용하는 OPC-UA 소스 구성

다음 `opc-ua-configuration.json`의 JSON 예제는 다음 속성을 가진 OPC-UA 원본 구성을 정의합니다.

- 지정된 X.509 인증서를 신뢰합니다.
- BASIC256 보안 정책을 사용하여 메시지를 보호합니다.
- SIGN_AND_ENCRYPT 모드를 사용하여 연결을 보호합니다.

```

{
  "sources": [
    {
      "name": "Wind Farm #3",
      "endpoint": {
        "certificateTrust": {
          "type": "X509",
          "certificateBody": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAKGA1UEBhMVCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xZDASBgNVBAwTC01BTSBDb25zb2x1MRIw
EAYDVQQDEw1UZXR0Q21sYW1mZG9w0BCQEWEG5vb251QGftYXpvbi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAKGA1UEBh
MVCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xZDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYW1mZG
9wHhAdBgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
          "certificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAKGA1UEBhMVCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZ

```

```

WF0dGx1MQ8wDQYDVQKKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIw
EAYDVQQDEw1UZXN0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxMzA0BjBBAgTALDlBMRAwDgYDZDQHEwDZWF0dGx1MQ8wDQYDVQKKEwZBb
WF6b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWVx
HmAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEiBb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVvXyUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
    },
    "endpointUri": "opc.tcp://203.0.113.2:49320",
    "securityPolicy": "BASIC256",
    "messageSecurityMode": "SIGN_AND_ENCRYPT",
    "identityProvider": {
      "type": "Anonymous"
    },
    "nodeFilterRules": []
  },
  "measurementDataStreamPrefix": ""
}
]
}

```

OPC-UA 원본 서버가 에지 게이트웨이를 신뢰하도록 지원 SiteWise

OPC-UA 원본을 구성할 때 **없음 messageSecurityMode** 이외의 것을 선택하는 경우 원본 서버가 Edge 게이트웨이를 신뢰할 수 있도록 설정해야 합니다. AWS IoT SiteWise SiteWise 에지 게이트웨이는 원본 서버에 필요할 수 있는 인증서를 생성합니다. 프로세스는 원본 서버에 따라 다릅니다. 자세한 내용은 서버 설명서를 참조하십시오.

다음 절차는 기본 단계를 간략하게 설명합니다.

OPC-UA 서버가 에지 게이트웨이를 신뢰할 수 있도록 하려면 SiteWise

1. OPC-UA 서버 구성을 위한 인터페이스를 엽니다.
2. OPC-UA 서버 관리자의 사용자 이름과 암호를 입력합니다.

3. 인터페이스에서 신뢰할 수 있는 클라이언트를 찾은 다음, AWS IoT SiteWise 게이트웨이 클라이언트를 선택합니다.
4. 신뢰를 선택하십시오.

OPC-UA 클라이언트 인증서 내보내기

일부 OPC-UA 서버는 Edge 게이트웨이를 신뢰하기 위해 OPC-UA 클라이언트 인증서 파일에 대한 액세스 권한이 필요합니다. SiteWise OPC-UA 서버에 적용되는 경우 다음 절차를 사용하여 에지 게이트웨이에서 OPC-UA 클라이언트 인증서를 내보낼 수 있습니다. SiteWise 그런 다음 OPC-UA 서버에서 인증서를 가져올 수 있습니다.

원본에 대한 OPC-UA 클라이언트 인증서 파일을 내보내려면

1. 다음 명령을 실행하여 인증서 파일이 있는 디렉터리로 변경합니다. `#### ## aws.iot# ## # ## # ## #####. SiteWiseEdgeCollectorOpcua Greengrass` 작업 폴더에서 `## ## # ## # ## #####` 바꿉니다.

```
##### ##### ## ### /greengrass/v2/work/aws.iot###.
SiteWiseEdgeCollector리눅스 및 C용 오픈쿠아: /#####/v2/work/aws.iot.
SiteWiseEdgeCollector##### ###.
```

```
cd /sitewise-work/source-name/opcua-certificate-store
```

2. 이 소스에 대한 SiteWise 에지 게이트웨이의 OPC-UA 클라이언트 인증서가 파일에 있습니다. `aws-iot-opcua-client.pfx`

다음 명령을 실행하여 인증서를 `aws-iot-opcua-client-certificate.pem`라는 .pem 파일로 내보냅니다

```
keytool -exportcert -v -alias aws-iot-opcua-client -keystore aws-iot-opcua-client.pfx -storepass amazon -storetype PKCS12 -rfc > aws-iot-opcua-client-certificate.pem
```

3. SiteWise 에지 게이트웨이에서 OPC-UA 서버로 인증서 파일을 전송합니다. `aws-iot-opcua-client-certificate.pem`

그러려면 SSH 프로토콜을 사용하여 파일을 전송하는 `scp` 프로그램과 같은 일반적인 소프트웨어를 사용하면 됩니다. 자세한 내용은 Wikipedia에서 [비밀번호 복사](#)를 참조하세요.

Note

SiteWise 엣지 게이트웨이가 Amazon Elastic Compute Cloud (Amazon EC2) 에서 실행 중이고 처음으로 연결하는 경우 연결하기 위한 사전 요구 사항을 구성해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [Linux 인스턴스에 연결을](#) 참조하십시오.

4. Edge 게이트웨이를 신뢰할 수 있도록 OPC-UA 서버에 인증서 파일 () 을 가져오십시오. `aws-iot-opcua-client-certificate.pem` SiteWise 사용하는 원본 서버에 따라 단계가 다를 수 있습니다. 서버 설명서를 참조하세요.

OPC-UA를 사용하여 데이터 모으기 범위 필터링

스캔 모드 및 데드밴드 범위를 사용하여 OPC-UA 소스로 데이터를 수집하는 방식을 제어할 수 있습니다. 이러한 기능을 통해 수집할 데이터의 종류와 서버와 SiteWise 엣지 게이트웨이가 이 정보를 교환하는 방법과 시기를 제어할 수 있습니다.

품질을 기준으로 데이터를 수집하거나 필터링합니다.

데이터 품질 설정을 구성하여 OPC-UA 소스에서 수집되는 데이터를 제어할 수 있습니다. 데이터 소스는 전송 시 품질 등급을 메타데이터로 포함합니다. 다음 옵션 중 하나 또는 모두를 선택할 수 있습니다.

- Good
- Bad
- Uncertain

스캔 모드로 데이터 수집 빈도 제어

OPC-UA 스캔 모드를 구성하여 OPC-UA 소스에서 데이터를 수집하는 방식을 제어할 수 있습니다. 구독 또는 폴링 모드를 선택할 수 있습니다.

- 구독 모드 - OPC-UA 소스는 스캔 속도에 정의된 주파수로 SiteWise Edge 게이트웨이로 전송할 데이터를 수집합니다. 서버는 값이 변경된 경우에만 데이터를 전송하므로 이 빈도는 SiteWise 엣지 게이트웨이가 데이터를 수신하는 최대 빈도입니다.
- 폴링 모드 - SiteWise Edge 게이트웨이는 스캔 속도에 정의된 설정된 빈도로 OPC-UA 소스를 폴링합니다. 서버는 값 변경 여부에 관계없이 데이터를 전송하므로 SiteWise 엣지 게이트웨이는 항상 이 간격으로 데이터를 수신합니다.

Note

폴링 모드 옵션은 이 소스의 데드밴드 설정보다 우선 적용됩니다.

데드밴드 범위를 사용하여 OPC-UA 데이터 모으기 필터링

OPC-UA 소스 속성 그룹에 데드밴드를 적용하여 특정 데이터를 클라우드로 보내는 대신 필터링하여 삭제할 수 있습니다. AWS 데드밴드는 OPC-UA 소스에서 들어오는 데이터 값의 예상 변동 기간을 지정합니다. 값이 이 창에 해당하는 경우 OPC-UA 서버는 해당 값을 클라우드로 전송하지 않습니다. AWS 데드밴드 필터링을 사용하여 처리하고 클라우드로 보내는 데이터의 양을 줄일 수 있습니다. AWS SiteWise 에지 게이트웨이의 OPC-UA 소스를 설정하는 방법을 알아보려면 [the section called “데이터 소스 구성”](#)을 참조하십시오.

Note

서버는 데드밴드에서 지정한 기간 내에 있는 모든 데이터를 삭제합니다. 이러한 폐기된 데이터는 복구할 수 없습니다.

데드밴드 유형

OPC-UA 서버 속성 그룹에 두 가지 유형의 데드밴드를 지정할 수 있습니다. 이를 통해 AWS 클라우드로 전송할 데이터의 양과 폐기할 데이터의 양을 선택할 수 있습니다.

- **백분율** — 측정값의 예상 변동 비율을 사용하여 기간을 지정합니다. 서버는 이 백분율로 정확한 기간을 계산하고 범위를 벗어나는 데이터를 AWS 클라우드로 전송합니다. 예를 들어 화씨 -100도에서 화씨 +100도 범위의 센서에 2% 데드밴드 값을 지정하면 값이 화씨 4도 이상 변경될 때 서버에 데이터를 클라우드로 보내도록 지시합니다. AWS

Note

소스 서버에서 엔지니어링 단위를 정의하지 않는 경우 이 기간의 최소 및 최대 데드밴드 값을 선택적으로 지정할 수 있습니다. 엔지니어링 단위 범위가 제공되지 않은 경우 OPC-UA 서버는 측정 데이터 유형의 전체 범위를 기본값으로 설정합니다.

- **절대** — 정확한 단위를 사용하여 기간을 지정합니다. 예를 들어 센서의 데드밴드 값을 2로 지정하면 값이 2단위 이상 변경될 때 서버가 데이터를 AWS 클라우드로 보내도록 지시합니다. 정상 작동 중에 변동이 정기적으로 예상되는 동적 환경에는 절대 데드밴딩을 사용할 수 있습니다.

데드밴드 타임아웃

데드밴드 타임아웃 설정을 선택적으로 구성할 수 있습니다. 이 타임아웃이 지나면 OPC-UA 서버는 예상 데드밴드 변동 범위 내에 있더라도 현재 측정값을 전송합니다. 타임아웃 설정을 사용하면 값이 정의된 데드밴드 범위를 초과하지 않는 경우에도 항상 꾸준한 AWS IoT SiteWise 데이터 스트림을 수집하도록 할 수 있습니다.

OPC-UA 노드 필터 사용

SiteWise 에지 게이트웨이의 OPC-UA 데이터 소스를 정의할 때 노드 필터를 정의할 수 있습니다. 노드 필터를 사용하면 SiteWise 에지 게이트웨이가 클라우드로 보내는 데이터 스트림 경로를 제한할 수 있습니다. 노드 필터를 사용하면 모델링하는 데이터에 대한 경로만 포함함으로써 SiteWise Edge 게이트웨이의 시작 시간과 CPU 사용량을 줄일 수 있습니다. 기본적으로 SiteWise Edge 게이트웨이는 로 시작하는 경로를 제외한 모든 OPC-UA 경로를 업로드합니다. /Server/ 노드 필터에 * 및 ** 와일드카드 문자를 사용하여 하나의 필터에 여러 데이터 스트림 경로를 포함할 수 있습니다. 에지 게이트웨이의 OPC-UA 소스를 설정하는 방법을 알아보려면 [참조하십시오. SiteWise 데이터 소스 구성](#)

Note

AWS IoT SiteWise 소스를 추가하거나 편집할 때마다 SiteWise 에지 게이트웨이를 다시 시작합니다. SiteWise 에지 게이트웨이는 다시 시작하는 동안에는 데이터를 수집하지 않습니다. 에지 게이트웨이를 다시 시작하는 데 걸리는 시간은 SiteWise 에지 게이트웨이 소스의 태그 수에 따라 달라집니다. SiteWise 재시작 시간은 몇 초 (태그가 거의 없는 SiteWise 에지 게이트웨이의 경우) 에서 몇 분 (태그가 많은 SiteWise 에지 게이트웨이의 경우) 까지 걸릴 수 있습니다.

다음 표에는 OPC-UA 데이터 소스를 필터링하는 데 사용할 수 있는 와일드카드가 나열되어 있습니다.

OPC-UA 노드 필터 와일드카드

와일드카드	설명
*	데이터 스트림 경로의 단일 레벨과 일치합니다.

와일드카드	설명
**	데이터 스트림 경로의 여러 레벨을 일치합니다.

Note

광범위한 필터로 원본을 구성한 다음 나중에 더 제한적인 필터를 사용하도록 원본을 변경하면 새 필터와 일치하지 않는 데이터의 저장이 AWS IoT SiteWise 중지됩니다.

Example 노드 필터를 사용한 예제 시나리오

다음의 가상 데이터 스트림을 고려하십시오.

- /WA/Factory 1/Line 1/PLC1
- /WA/Factory 1/Line 1/PLC2
- /WA/Factory 1/Line 2/Counter1
- /WA/Factory 1/Line 2/PLC1
- /OR/Factory 1/Line 1/PLC1
- /OR/Factory 1/Line 2/Counter2

이전 데이터 스트림을 사용해 노드 필터를 정의하여 OPC-UA 소스에서 포함할 데이터를 제한할 수 있습니다.

- 이 예제에서 모든 노드를 선택하려면 / 또는 /**/를 사용합니다. ** 와일드카드 문자가 있는 여러 디렉터리 또는 폴더를 포함할 수 있습니다.
- 모든 PLC 데이터 스트림을 선택하려면 /***/PLC* 또는 /**/PLC*을 사용합니다.
- 이 예제에서 모든 카운터를 선택하려면 /**/Counter* 또는 /***/Counter*를 사용합니다.
- Line 2의 모든 카운터를 선택하려면 /**/Line 2/Counter*를 사용합니다.

데이터 소스 인증 구성

OPC-UA 서버 연결에 인증 자격 증명이 필요한 경우를 AWS Secrets Manager 사용하여 암호를 만들어 SiteWise 에지 게이트웨이에 배포할 수 있습니다. AWS Secrets Manager 디바이스의 비밀을 암호화하여 사용하기 전까지 사용자 이름과 비밀번호를 안전하게 유지합니다. AWS IoT Greengrass

Secret Manager 구성 요소에 대한 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 [Secret Manager](#)를 참조하십시오.

Secrets Manager 비밀에 대한 액세스 관리에 대한 자세한 내용은 다음을 참조하십시오.

- 내 [AWS Secrets Manager 비밀에 대한 권한을 가진 사람은 누구입니까?](#)
- [계정 내에서 요청이 허용되는지 거부되는지 판별합니다.](#)

1단계: 소스 인증 암호 만들기

를 AWS Secrets Manager 사용하여 데이터 원본에 대한 인증 암호를 만들 수 있습니다. 암호에서 데이터 소스에 대한 인증 세부 정보를 포함하는 **username** 및 **password** 키-값 쌍을 정의합니다.

보안 암호(콘솔) 생성

1. [AWS Secrets Manager 콘솔](#)로 이동합니다.
2. 새 암호 저장을 선택합니다.
3. 암호 유형에서 다른 암호 유형을 선택합니다.
4. 키/값 쌍에서 다음을 수행하십시오.
 1. 첫 번째 입력 상자에 사용자 이름을 **username** 입력하고 두 번째 입력 상자에 사용자 이름을 입력합니다.
 2. Add row(행 추가)를 선택합니다.
 3. 첫 번째 입력 상자에 암호를 **password** 입력하고 두 번째 입력 상자에 암호를 입력합니다.
5. 암호화 키에 대해 AWS/secretsmanager를 선택하고 다음을 선택합니다.
6. 새 암호 저장 페이지에서 암호 이름을 입력합니다.
7. (선택 사항) 이 암호를 식별하는 데 도움이 되는 설명을 입력하고 다음을 선택합니다.
8. (선택 사항) 새 암호 저장 페이지에서 자동 교체를 켤 수 있습니다. 자세한 내용은 AWS Secrets Manager 사용 설명서의 [비밀번호 교체](#)를 참조하세요.
9. 교체 일정을 지정합니다.
10. 이 암호를 교체할 수 있는 Lambda 함수를 선택한 후 다음을 선택합니다.
11. 보안 구성을 검토한 다음 저장을 선택합니다.

에지 게이트웨이의 상호 AWS Secrets Manager작용을 승인하려면 SiteWise 에지 게이트웨이의 IAM 역할이 작업을 허용해야 합니다 SiteWise . secretsmanager:GetSecretValue Greengrass 코어

디바이스를 사용하여 IAM 정책을 검색할 수 있습니다. IAM 정책 업데이트에 대한 자세한 내용은 사용 설명서의 [IAM 정책 편집](#)을 참조하십시오. AWS Identity and Access Management

Example 정책

*secret-arn*을 이전 단계에서 생성한 암호의 Amazon 리소스 이름(ARN)으로 변경합니다. 암호의 ARN을 가져오는 방법에 대한 자세한 내용은 AWS Secrets Manager 사용 설명서의 [AWS Secrets Manager에서 보안 암호 가져오기](#)를 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": [
        "secret-arn"
      ]
    }
  ]
}
```

2단계: SiteWise 엣지 게이트웨이 디바이스에 시크릿 배포

AWS IoT SiteWise 콘솔을 사용하여 SiteWise 엣지 게이트웨이에 암호를 배포할 수 있습니다.

암호를 업데이트하려면 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 게이트웨이를 선택합니다.
3. 게이트웨이 목록에서 대상 SiteWise 엣지 게이트웨이를 선택합니다.
4. 게이트웨이 구성 섹션에서 Greengrass 코어 디바이스 링크를 선택하여 SiteWise Edge 게이트웨이와 연결된 AWS IoT Greengrass 코어를 엽니다.
5. 탐색 창에서 배포를 선택합니다.
6. 대상 배포를 선택한 다음 수정을 선택합니다.
7. 대상 지정 페이지에서 다음을 선택합니다.
8. 구성 요소 선택 페이지의 공용 구성 요소 섹션에서 선택한 구성 요소만 표시를 끕니다.

9. `aws.greengrass`를 검색하여 선택합니다. SecretManager구성 요소를 선택하고 다음을 선택합니다.
10. 선택한 구성 요소 목록에서 `aws.greengrass`를 선택합니다. SecretManager구성 요소를 선택한 다음 구성 요소 구성을 선택합니다.
11. 병합할 구성 필드에 다음 JSON 객체를 추가합니다.

 Note

`secret-arn`을 이전 단계에서 생성한 암호의 ARN으로 변경합니다. 암호의 ARN을 가져오는 방법에 대한 자세한 내용은 AWS Secrets Manager 사용 설명서의 [AWS Secrets Manager에서 보안 암호 가져오기](#)를 참조하세요.

```
{
  "cloudSecrets":[
    {
      "arn":"secret-arn"
    }
  ]
}
```

12. 확인을 선택합니다.
13. 다음을 선택합니다.
14. 구성 고급 설정 페이지에서 다음을 선택합니다.
15. 배포 구성을 검토한 다음 배포를 선택합니다.

3단계: 인증 구성 추가

AWS IoT SiteWise 콘솔을 사용하여 SiteWise 에지 게이트웨이에 인증 구성을 추가할 수 있습니다.

인증 구성을 추가하려면 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 게이트웨이 목록에서 대상 SiteWise 에지 게이트웨이를 선택합니다.
3. 데이터 소스 목록에서 대상 데이터 소스를 선택한 다음 편집을 선택합니다.
4. 데이터 소스 추가 페이지에서 고급 구성을 선택합니다.

5. 인증 구성의 경우 이전 단계에서 배포한 암호를 선택합니다.
6. 저장을 선택합니다.

소스 서버 데이터의 목적지 선택

데이터는 AWS IoT SiteWise 엣지에서 실시간으로 또는 Amazon S3를 사용하여 일괄적으로 내보냅니다. 또한 스트림을 사용하여 다른 구성 요소로 스트림을 전송할 수 있습니다 AWS IoT Greengrass .

- AWS IoT SiteWise 실시간 — 데이터를 AWS IoT SiteWise 스토리지로 직접 전송하려면 이 옵션을 선택합니다. 실시간으로 데이터를 수집 및 모니터링하고 엣지에서 데이터를 처리합니다.
- AWS IoT SiteWise Amazon S3를 사용하여 버퍼링 — 데이터를 파켓 형식으로 Amazon S3로 전송한 다음 스토리지로 가져옵니다. AWS IoT SiteWise 데이터를 일괄적으로 수집하고 비용 효율적인 방식으로 기록 데이터를 저장하려면 이 옵션을 선택하십시오. 원하는 Amazon S3 버킷 위치와 Amazon S3에 데이터를 업로드할 빈도를 구성할 수 있습니다. 또한 데이터를 수집한 후 데이터를 어떻게 처리할지 선택할 수 있습니다. AWS IoT SiteWise Amazon S3에서 데이터를 모두 SiteWise 사용할 수 있도록 선택하거나 Amazon S3에서 자동으로 삭제하도록 선택할 수 있습니다.
- Amazon S3 버킷은 스테이징 및 버퍼링 메커니즘이며 파켓 형식의 파일을 지원합니다.
- AWS IoT SiteWise 스토리지로 데이터 가져오기 확인란을 선택하면 데이터가 먼저 Amazon S3에 업로드된 다음 AWS IoT SiteWise 스토리지로 업로드됩니다.
 - Amazon S3에서 데이터 삭제 확인란을 선택하면 데이터를 SiteWise 스토리지로 가져온 후 Amazon S3에서 삭제됩니다.
 - Amazon S3에서 데이터 삭제 확인란의 선택을 취소하면 데이터가 Amazon S3와 SiteWise 스토리지에 모두 저장됩니다.
- AWS IoT SiteWise 스토리지로 데이터 가져오기 확인란의 선택을 취소하면 데이터는 Amazon S3에만 저장됩니다. SiteWise 스토리지로 가져올 수는 없습니다.

AWS IoT SiteWise 제공되는 다양한 스토리지 옵션에 [데이터 스토리지 관리](#) 대한 자세한 내용은 을 참조하십시오. 가격 옵션에 대한 자세한 내용은 [AWS IoT SiteWise 가격을](#) 참조하십시오.

- AWS IoT Greengrass 스트림 관리자 — AWS IoT Greengrass 스트림 관리자를 사용하여 채널 입력, Amazon Kinesis Data Streams의 스트림 AWS IoT Analytics, 자산 속성 또는 Amazon Simple Storage Service (Amazon S3) AWS IoT SiteWise의 객체 등 데이터를 전송할 수 있습니다. AWS 클라우드 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 AWS IoT Greengrass [Core에서 데이터 스트림 관리를](#) 참조하십시오.

다음 예제는 필수 데이터 스트림 메시지 구조를 보여줍니다. 모든 필드를 입력해야 합니다.

```

{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
      "timestamp": {
        "offsetInNanos": number,
        "timeInSeconds": number
      },
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      }
    }
  ]
}

```

Note

데이터 스트림 메시지는 구조에 (assetId및propertyId) 또는 propertyAlias 둘 중 하나를 포함해야 합니다.

assetId

(선택 사항) 업데이트할 자산의 ID.

propertyAlias

(선택 사항) 속성을 식별하는 별칭 (예: OPC-UA 서버 데이터 스트림 경로). 예:

```
/company/windfarm/3/turbine/7/temperature
```

자세한 내용은 사용 설명서의 [산업 데이터 스트림을 자산 자산에 매핑](#)을 참조하십시오.AWS IoT SiteWise

propertyId

(선택 사항) 이 항목의 자산 자산 ID.

propertyValues

(필수) 업로드할 속성 값 목록. propertyValues배열 요소를 최대 10개까지 지정할 수 있습니다.

quality

(선택 사항) 자산 속성 값의 품질입니다.

timestamp

(필수) 자산 속성 값의 타임스탬프입니다.

offsetInNanos

(선택 사항) 의 나노초 오프셋. timeInSeconds

timeInSeconds

(필수) Unix epoch 형식의 타임스탬프 날짜 (초 단위). offsetInNanos에서 나노초 단위의 데이터를 제공합니다.

value

(필수) 자산 속성의 값입니다.

Note

value필드에는 다음 값 중 하나만 존재할 수 있습니다.

booleanValue

(선택 사항) Boolean (true또는false) 유형의 자산 속성 데이터.

doubleValue

(선택 사항) double 유형의 자산 속성 데이터 (부동 소수점 숫자).

integerValue

(선택 사항) 정수 유형의 자산 속성 데이터 (정수).

stringValue

(선택 사항) 문자열 유형의 자산 속성 데이터 (문자 시퀀스).

SiteWise 에지 게이트웨이에 파트너 데이터 소스 추가

AWS IoT SiteWise 에지 게이트웨이를 사용하는 경우 파트너 데이터 소스를 에지 게이트웨이에 연결하고 SiteWise 에지 게이트웨이 및 AWS 클라우드의 SiteWise 파트너로부터 데이터를 받을 수 있습니다. 이러한 파트너 데이터 소스는 AWS와 파트너 간의 파트너십을 통해 개발된 AWS IoT Greengrass 구성 요소입니다. 파트너 데이터 소스를 추가하면 이 구성 요소가 생성되어 SiteWise 에지 게이트웨이에 배포됩니다. AWS IoT SiteWise

파트너 데이터 소스를 추가하려면 다음을 수행합니다.

- [파트너 데이터 소스 추가](#)
- 파트너의 웹 포털로 이동하여 파트너 데이터 원본이 SiteWise Edge 게이트웨이에 연결되도록 구성하십시오.

주제

- [보안](#)
- [파트너 데이터 소스 추가](#)
- [에지 게이트웨이에 docker 설정 SiteWise](#)
- [SiteWise 엣지 게이트웨이 파트너 데이터 소스](#)

보안

AWS, 고객, 파트너 간의 [공동 책임 모델의](#) 일환으로 보안의 다양한 측면을 책임지는 사람이 누구인지 다음과 같이 설명합니다.

고객의 책임

- 파트너를 심사합니다.
- 파트너에게 부여된 네트워크 액세스를 구성합니다.

AWS 책임

- 파트너에게 필요한 클라우드 리소스를 제외한 고객 AWS 클라우드 리소스로부터 파트너를 격리합니다. 이 경우에는 AWS IoT SiteWise 수집입니다.
- 파트너 솔루션을 SiteWise Edge 게이트웨이 시스템 리소스 (CPU, 메모리, 파일 시스템) 의 적절한 사용량으로 제한합니다.

파트너의 책임

- 보안 기본값을 사용합니다.

- 패치 및 기타 적절한 업데이트를 통해 시간이 지나도 솔루션을 안전하게 유지합니다.
- 고객 데이터의 기밀을 유지합니다.

파트너 데이터 소스 추가

파트너 데이터 원본을 SiteWise 에지 게이트웨이에 연결하려면 파트너 데이터 원본을 데이터 원본으로 추가하세요. 데이터 원본으로 추가하면 개인 AWS IoT Greengrass 구성 요소가 SiteWise 에지 게이트웨이에 배포됩니다. AWS IoT SiteWise

사전 조건

파트너 데이터 소스를 추가하려면 다음을 수행해야 합니다.

- 파트너와 함께 계정을 생성합니다.
- 계정을 바인딩합니다.

파트너 데이터 소스를 사용하여 SiteWise 에지 게이트웨이를 만들려면

새 SiteWise 에지 게이트웨이를 만들려면 의 단계를 완료하세요 [SiteWise 에지 게이트웨이 생성](#). SiteWise 에지 게이트웨이를 만든 후에는 다음 단계에 따라 파트너 데이터 원본을 추가하세요. [기존 SiteWise 에지 게이트웨이에 파트너 데이터 원본을 추가하려면](#)

기존 SiteWise 에지 게이트웨이에 파트너 데이터 원본을 추가하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 게이트웨이를 선택합니다.
3. 파트너 데이터 소스를 연결하려는 SiteWise 에지 게이트웨이를 선택합니다.
4. 데이터 소스에서 데이터 소스 추가를 선택합니다.
5. 소스 유형에서 SiteWise 에지 게이트웨이를 연결할 파트너를 선택합니다.

Note

현재 사용 가능한 파트너 데이터 EasyEdge 소스는 이 (가) 유일합니다. EasyEdge 데이터 소스를 처음 추가할 때는 [EasyEdge 계정을](#) 만들어야 합니다.

6. 소스 이름을 입력합니다.
7. 파트너에게 데이터 소스에 대한 액세스 권한을 부여하려면 승인을 선택합니다.

8. AWS IoT SiteWise가 AWS IoT SiteWise 게시자 구성 요소를 업데이트하고, 데이터 처리 팩이 활성화된 경우 AWS IoT SiteWise 프로세서 구성 요소를 업데이트하려면 구성 요소 업데이트를 선택합니다.
9. 저장을 선택합니다.

에지 게이트웨이에 docker 설정 SiteWise

파트너 데이터 소스를 추가하려면 [Docker Engine](#) 1.9.1 이상이 로컬 장치에 설치되어 있어야 합니다.

Note

버전 20.10은 Edge 게이트웨이 소프트웨어와 호환이 확인된 최신 버전입니다. SiteWise

Docker가 설치되었는지 확인하려면

Docker가 설치되었는지 확인하려면 SiteWise Edge 게이트웨이에 연결된 터미널에서 다음 명령을 실행합니다.

```
docker info
```

명령이 `docker is not recognized` 결과를 반환하거나 이전 버전의 Docker가 설치된 경우 계속하기 전에 [Docker 엔진을 설치](#)하십시오.

Docker를 설정하려면

Docker 컨테이너 구성 요소를 실행하는 시스템 사용자에게 루트 또는 관리자 권한이 있거나 루트 또는 관리자가 아닌 사용자로 실행하도록 Docker를 구성해야 합니다.

Linux 디바이스에서는 `sudo` 없이 Docker 명령을 호출하려면 `ggc_user` 사용자를 `docker` 그룹에 추가해야 합니다.

Docker 컨테이너 구성 요소를 실행하는 데 사용하는 `ggc_user` 또는 루트 사용자가 아닌 대상을 `docker` 그룹에 추가하려면 다음 명령을 실행합니다.

```
sudo usermod -aG docker ggc_user
```

자세한 내용은 [Docker 엔진의 Linux 사후 설치 단계](#)를 참조하십시오.

SiteWise 엣지 게이트웨이 파트너 데이터 소스

아래 정보를 사용하여 파트너 데이터 소스를 구성하십시오.

EasyEdge

포털:

<https://studio.easyedge.io/>

EasyEdge 문서:

[EasyEdge 에 대한 AWS](#)

[EasyEdge 요구 사항](#) - 방화벽 구성에 필요한 엔드포인트 및 포트를 비롯한 EasyEdge 요구 사항에 대한 정보입니다. 참고: 이 설명서에 액세스하려면 EasyEdge 계정이 필요합니다.

팩 사용

AWS IoT SiteWise 에지 게이트웨이는 다양한 팩을 사용하여 데이터 수집 및 처리 방법을 결정합니다.

현재 사용 가능한 팩은 다음과 같습니다.

- 데이터 수집 팩 — 이 팩을 사용하여 산업 데이터를 수집하고 AWS 클라우드 대상으로 라우팅할 수 있습니다. 기본적으로 이 팩은 SiteWise Edge 게이트웨이에서 자동으로 활성화됩니다.
- 데이터 처리 팩 - 이 팩을 사용하면 SiteWise 에지 구성 자산 모델 및 자산과 에지 게이트웨이 통신을 활성화할 수 있습니다. 엣지 구성을 사용하여 현장에서 계산하고 처리할 자산 데이터를 제어할 수 있습니다. 그런 다음 데이터를 AWS IoT SiteWise 또는 다른 AWS 서비스로 보낼 수 있습니다. 데이터 처리 팩에 대한 자세한 내용은 [the section called “엣지 데이터 처리 활성화”](#)을 참조하십시오.

팩 업그레이드하기

Important

데이터 처리 팩 버전을 2.0.x 이전(및 포함)에서 버전 2.1.x로 업그레이드하면 로컬에 저장된 측정값의 데이터가 손실됩니다.

SiteWise 에지 게이트웨이는 다양한 팩을 사용하여 데이터 수집 및 처리 방법을 결정합니다. AWS IoT SiteWise 콘솔을 사용하여 팩을 업그레이드할 수 있습니다.

팩을 업그레이드하려면 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 게이트웨이를 선택합니다.
3. 게이트웨이 목록에서 업그레이드하려는 팩이 포함된 SiteWise Edge 게이트웨이를 선택합니다.
4. 게이트웨이 구성 섹션에서 사용 가능한 소프트웨어 업데이트를 선택합니다.
5. 소프트웨어 버전 편집 페이지의 게이트웨이 구성 요소 업데이트 섹션에서 다음을 수행하십시오.
 - OPC-UA 컬렉터를 업데이트하려면 버전을 선택한 다음 배포를 선택합니다.
 - 퍼블리셔를 업데이트하려면 버전을 선택한 다음 배포를 선택합니다.
 - 데이터 처리 팩을 업데이트하려면 버전을 선택한 다음 배포를 선택합니다.
6. 새 버전 배포를 완료하면 [완료] 를 선택합니다.

팩을 업그레이드하는 데 문제가 있는 경우 [SiteWise 에지 게이트웨이에 팩을 배포할 수 없습니다.](#)을 참조하십시오.

SiteWise 에지 게이트웨이 관리

AWS IoT SiteWise 콘솔 및 API 작업을 사용하여 AWS IoT SiteWise 에지 게이트웨이를 관리할 수 있습니다. 또한 [AWS OpsHub AWS IoT SiteWise Windows용](#) 애플리케이션을 사용하여 로컬 장치에서 SiteWise 에지 게이트웨이의 일부 측면을 관리할 수 있습니다.

로컬 장치의 디스크 사용량을 모니터링하려면 AWS OpsHub for AWS IoT SiteWise 애플리케이션을 사용하는 것이 좋습니다. Gateway.AvailableDiskSpace 및 Gateway.UsedPercentageDiskSpace Amazon CloudWatch 지표를 모니터링하고 경보를 생성하여 디스크 공간이 부족해지면 알림을 받을 수도 있습니다. Amazon CloudWatch 경보에 대한 자세한 내용은 [정적 임계값 기반 CloudWatch 경보 생성](#)을 참조하십시오.

디바이스에 향후 데이터를 저장할 공간이 충분한지 확인합니다. 로컬 디바이스의 공간이 거의 부족해지면 서비스는 예정된 데이터를 위한 공간을 확보하기 위해 가장 오래된 타임스탬프가 있는 소량의 데이터를 자동으로 삭제합니다.

서비스가 데이터를 삭제했는지 확인하려면 다음을 수행합니다.

1. for 애플리케이션에 로그인합니다. AWS OpsHub AWS IoT SiteWise
2. 설정을 선택합니다.
3. 로그의 경우 시간 범위를 지정한 다음 다운로드를 선택합니다.

4. 로그 파일 압축을 끕니다.
5. `## ### ## ##### ## ## ##### ##### #####. SiteWise Edge ##### ##### ## #
#####.`

콘솔을 AWS IoT SiteWise 통한 SiteWise 에지 게이트웨이 관리

AWS IoT SiteWise 콘솔을 사용하여 AWS 계정의 모든 SiteWise 에지 게이트웨이를 구성, 업데이트 및 모니터링할 수 있습니다.

[콘솔에서 SiteWise 에지 게이트웨이 페이지로 이동하여 에지 게이트웨이를 볼 수 있습니다.](#) [AWS IoT SiteWise](#) 특정 게이트웨이의 에지 게이트웨이 세부 정보 페이지에 액세스하려면 에지 게이트웨이의 이름을 선택합니다.

에지 게이트웨이 세부 정보 페이지의 개요 탭에서 다음을 수행할 수 있습니다.

- 데이터 소스 섹션에서 데이터 원본 구성을 업데이트하고 추가 데이터 원본을 구성합니다.
- CloudWatch 지표 콘솔에서 데이터 소스별로 수집된 데이터 포인트 수를 보려면 공개 CloudWatch 지표를 선택합니다.
- Edge 기능 섹션에서 편집을 클릭하여 SiteWise Edge 게이트웨이에 데이터 팩을 추가합니다.
- 게이트웨이 구성 섹션에서 SiteWise 에지 게이트웨이의 연결 상태를 확인합니다.
- 게시자 구성 섹션에서 게시자 구성 요소의 SiteWise Edge 게이트웨이 동기화 상태 및 구성을 확인합니다. AWS IoT SiteWise

에지 게이트웨이 세부 정보 페이지의 업데이트 탭에서는 에지 게이트웨이에 배포된 현재 구성 요소 및 팩 버전을 볼 수 있습니다. 또한 새 버전이 출시되면 이를 배포하는 곳이기도 합니다.

다음은 사용하여 SiteWise AWS OpsHub 에지 게이트웨이를 관리합니다.

AWS IoT SiteWise

AWS OpsHub for AWS IoT SiteWise 애플리케이션을 사용하여 SiteWise 에지 게이트웨이를 관리하고 모니터링합니다. 이 애플리케이션은 다음과 같은 모니터링 및 관리 옵션을 제공합니다.

- 개요에서 다음을 수행할 수 있습니다.
 - SiteWise 에지 게이트웨이 장치 데이터에 대한 통찰력을 얻고, 문제를 식별하고, SiteWise 에지 게이트웨이의 성능을 개선하는 데 도움이 되는 SiteWise 에지 게이트웨이 세부 정보를 확인하십시오.

- 엣지에 있는 로컬 서버 및 장비의 데이터를 모니터링하는 SiteWise 모니터 포털을 확인하세요. 자세한 내용은 AWS IoT SiteWise Monitor 사용 설명서에서 [AWS IoT SiteWise Monitor란 무엇입니까?](#)를 참조하십시오.
- Health에는 SiteWise Edge 게이트웨이의 데이터를 표시하는 대시보드가 있습니다. 프로세스 엔지니어와 같은 도메인 전문가는 대시보드를 사용하여 SiteWise 에지 게이트웨이 동작의 개요를 볼 수 있습니다.
- 자산에서 로컬 장치에 배포된 자산과 자산 속성에 대해 수집 또는 계산된 마지막 값을 볼 수 있습니다.
- 설정에서 다음을 수행합니다.
 - 데이터 프로세싱 팩이 설치된 경우 SiteWise 에지 게이트웨이 구성 정보를 보고 리소스를 AWS 클라우드와 동기화할 수 있습니다.
 - 다른 도구를 사용하여 SiteWise Edge 게이트웨이에 액세스하는 데 사용할 수 있는 인증 파일을 다운로드합니다.
 - SiteWise Edge 게이트웨이 문제를 해결하는 데 사용할 수 있는 로그를 다운로드하십시오.
 - SiteWise 에지 게이트웨이에 배포된 AWS IoT SiteWise 구성 요소를 볼 수 있습니다.

Important

사용하려면 AWS OpsHub 다음이 필요합니다 AWS IoT SiteWise.

- 로컬 장치와 AWS OpsHub for AWS IoT SiteWise 애플리케이션이 동일한 네트워크에 연결되어 있어야 합니다.
- 데이터 프로세싱 팩을 활성화해야 합니다.

를 사용하여 SiteWise 에지 게이트웨이를 관리하려면 AWS OpsHub

1. [AWS OpsHubAWS IoT SiteWise Windows용](#) 애플리케이션을 다운로드하고 설치합니다.
2. 애플리케이션을 엽니다.
3. 게이트웨이에 로컬 자격 증명을 설정하지 않은 경우 아래 단계에 [로컬 운영 체제 자격 SiteWise 증명을 사용하여 에지 게이트웨이에 액세스](#) 따라 설정하십시오.
4. Linux 또는 LDAP (경량 디렉터리 액세스 프로토콜) 자격 증명을 사용하여 SiteWise 에지 게이트웨이에 로그인할 수 있습니다. SiteWise 에지 게이트웨이에 로그인하려면 다음 중 하나를 수행하십시오.

Linux

1. 호스트 이름 또는 IP 주소에 로컬 장치의 호스트 이름 또는 IP 주소를 입력합니다.
2. 인증에서 Linux를 선택합니다.
3. 사용자 이름에는 Linux 운영 체제의 사용자 이름을 입력합니다.
4. 암호에는 Linux 운영 체제의 비밀번호를 입력합니다.
5. 로그인을 선택합니다.

LDAP

1. 호스트 이름 또는 IP 주소에 로컬 장치의 호스트 이름 또는 IP 주소를 입력합니다.
2. 인증에서 LDAP를 선택합니다.
3. 사용자 이름에 LDAP 사용자 이름을 입력합니다.
4. 비밀번호에 LDAP 비밀번호를 입력합니다.
5. 로그인을 선택합니다.

로컬 운영 체제 자격 SiteWise 증명을 사용하여 에지 게이트웨이에 액세스

경량 디렉터리 액세스 프로토콜 (LDAP) 외에도 Linux 또는 Windows 자격 증명을 사용하여 SiteWise 에지 게이트웨이에 액세스할 수 있습니다.

Important

Linux 자격 증명으로 SiteWise 에지 게이트웨이에 액세스하려면 SiteWise 에지 게이트웨이의 데이터 처리 팩을 활성화해야 합니다.

Linux 운영 체제 자격 증명을 사용하여 SiteWise 에지 게이트웨이에 액세스

다음 단계에서는 Ubuntu가 설치된 디바이스를 사용한다고 가정합니다. 다른 Linux 배포판을 사용하는 경우 해당 디바이스의 관련 설명서를 참조하십시오.

Linux 사용자 풀을 생성하려면

1. 다음 명령을 실행하여 관리자 그룹을 생성합니다.

```
sudo groupadd --system SWE_ADMIN_GROUP
```

SWE_ADMIN_GROUP 그룹의 사용자는 SiteWise 에지 게이트웨이에 대한 관리자 액세스를 허용할 수 있습니다.

2. 다음 명령을 실행하여 사용자 그룹을 생성합니다.

```
sudo groupadd --system SWE_USER_GROUP
```

SWE_USER_GROUP 그룹의 사용자는 SiteWise Edge 게이트웨이에 대한 읽기 전용 액세스를 허용할 수 있습니다.

3. 관리자 그룹에 사용자를 추가하려면 다음 명령을 실행합니다. **### ##**과 **####**를 추가하려는 사용자 이름 및 비밀번호로 변경합니다.

```
sudo useradd -p $(openssl passwd -1 password) user-name
```

4. SWE_ADMIN_GROUP 또는 SWE_USER_GROUP에 사용자를 추가하려면 **### ##**을 이전 단계에서 추가한 사용자 이름으로 바꾸십시오.

```
sudo usermod -a -G SWE_ADMIN_GROUP user-name
```

이제 AWS OpsHub for AWS IoT SiteWise 애플리케이션에서 사용자 이름과 암호를 사용하여 SiteWise Edge 게이트웨이에 로그인할 수 있습니다.

Windows 자격 증명을 사용하여 SiteWise 에지 게이트웨이에 액세스

다음 단계에서는 Windows가 설치된 디바이스를 사용한다고 가정합니다.

Important

보안은 사용자와 사용자 AWS 간의 공동 책임입니다. 대문자, 소문자, 숫자 및 기호 조합을 사용하여 12자 이상의 강력한 비밀번호 정책을 생성합니다. 또한 포트 443에서 들어오는 트래픽은 허용하고 다른 모든 포트에서 들어오는 트래픽은 차단하도록 Windows 방화벽 규칙을 설정합니다.

Windows Server 사용자 풀을 만들려면

1. 관리자 PowerShell 권한으로 실행하세요.
 - a. SiteWise 에지 게이트웨이를 설치하려는 Windows 서버에 관리자로 로그인합니다.
 - b. Windows 검색 PowerShell창에 입력합니다.
 - c. 검색 결과에서 Windows PowerShell 앱을 마우스 오른쪽 버튼으로 클릭합니다. [Run as Administrator]를 선택합니다.
2. 다음 명령을 실행하여 관리자 그룹을 생성합니다.

```
net localgroup SWE_ADMIN_GROUP /add
```

SWE_ADMIN_GROUP 그룹의 사용자만 SiteWise Edge 게이트웨이에 대한 관리자 액세스를 허용할 수 있습니다.

3. 다음 명령을 실행하여 사용자 그룹을 생성합니다.

```
net localgroup SWE_USER_GROUP /add
```

SiteWise 에지 게이트웨이에 대한 읽기 전용 액세스를 허용하려면 SWE_USER_GROUP 그룹의 사용자여야 합니다.

4. 사용자를 추가하려면 다음 명령을 실행합니다. `### ##`과 `####`를 생성하려는 사용자 이름 및 비밀번호로 바꾸십시오.

```
net user user-name password /add
```

5. 관리자 그룹에 사용자를 추가하려면 다음 명령을 실행합니다. `### ##`을 추가하려는 사용자 이름으로 바꿉니다.

```
net localgroup SWE_ADMIN_GROUP user-name /add
```

이제 AWS OpsHub for AWS IoT SiteWise 애플리케이션에서 사용자 이름과 암호를 사용하여 SiteWise Edge 게이트웨이에 로그인할 수 있습니다.

SiteWise 에지 게이트웨이 인증서 관리

에지 게이트웨이 디바이스에서 SiteWise 모니터 및 Grafana와 같은 타사 애플리케이션을 사용할 수 SiteWise 있습니다. 이러한 애플리케이션을 사용하려면 서비스에 대한 TLS 연결이 필요합니다.

SiteWise 에지 게이트웨이는 현재 자체 서명된 인증서를 사용합니다. SiteWise 모니터 포털과 같은 브라우저를 사용하여 애플리케이션을 여는 경우 신뢰할 수 없는 인증서에 대한 경고 메시지가 표시될 수 있습니다.

다음은 AWS OpsHub for AWS IoT SiteWise 애플리케이션에서 신뢰할 수 있는 인증서를 다운로드하는 방법을 보여줍니다.

1. 애플리케이션에 로그인합니다.
2. 설정을 선택합니다.
3. 인증에서 인증서 다운로드를 선택합니다.

다음은 Google Chrome 또는 FireFox 를 사용한다고 가정합니다. 다른 브라우저를 사용하는 경우 해당 브라우저의 관련 설명서를 참조하십시오. 이전 단계에서 다운로드한 인증서를 브라우저에 추가하려면 다음 중 하나를 수행합니다.

- Google 크롬을 사용하는 경우 Google 크롬 엔터프라이즈 도움말 문서의 [인증서 설정](#)을 따르십시오.
- Firefox를 사용하는 경우 Oracle 설명서의 [Mozilla 또는 Firefox 브라우저에 인증서를 로드하는 방법](#)을 따르십시오.

SiteWise Edge 게이트웨이 구성 요소 팩의 버전 변경

AWS IoT SiteWise 콘솔을 사용하여 SiteWise 에지 게이트웨이의 구성 요소 팩 버전을 변경할 수 있습니다.

SiteWise Edge 게이트웨이 구성 요소 팩의 버전을 변경하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 게이트웨이를 선택합니다.
3. 팩 버전을 변경하려는 SiteWise Edge 게이트웨이를 선택합니다.
4. 게이트웨이 구성에서 소프트웨어 버전 보기를 선택합니다.
5. 소프트웨어 버전 편집 페이지에서 버전을 업데이트하려는 팩의 경우 배포할 버전을 선택하고 배포를 선택합니다.
6. 완료를 선택합니다.

지멘스 인더스트리얼 SiteWise 엣지의 러닝 엣지

디바이스에서 에지 게이트웨이를 AWS 계정 실행하여 Siemens Industrial Edge 디바이스의 데이터를 사용자 디바이스로 SiteWise 수집할 수 있습니다. 이를 위해서는 Siemens Industrial SiteWise Edge 디바이스 (신규) 를 배포 대상으로 하는 엣지 게이트웨이 리소스를 생성하고 구성 파일을 다운로드한 후 Siemens IEM (Industrial Edge Management) 포털을 통해 Siemens 앱에 업로드합니다. [필수 Siemens 리소스를 설정하는 방법을 포함하여 Siemens Industrial Edge에서 Edge를 실행하는 AWS IoT SiteWise 방법에 대한 자세한 내용은 산업용 엣지란? 을 참조하십시오.](#) 지멘스 설명서에서

Note

지멘스는 Edge의 공급업체 또는 공급업체가 아닙니다. AWS IoT SiteWise 지멘스 인더스트리얼 엣지 마켓플레이스는 독립 마켓플레이스입니다.

주제

- [필수 조건](#)
- [보안](#)
- [구성 파일 생성](#)
- [문제 해결](#)
- [문의처](#)

필수 조건

지멘스 인더스트리얼 엣지에서 Edge를 AWS IoT SiteWise 실행하려면 다음이 필요합니다.

- [지멘스 디지털 익스체인지 플랫폼 계정](#)
- 지멘스 인더스트리얼 엣지 허브 (iehub) 계정
- 지멘스 산업용 엣지 관리 (IEM) 인스턴스
- 지멘스 산업용 에지 디바이스 (IED) 또는 지멘스 산업용 에지 가상 디바이스 (IeVD)
- 지멘스 인더스트리얼 엣지 디바이스 배포 대상에 대한 액세스 권한을 얻으려면 [AWS IoT SiteWise 콘솔로](#) 이동하여 액세스 요청을 선택합니다.

보안

고객과 파트너 간의 [AWS공동 책임 모델의](#) 일환으로 다음은 보안의 다양한 측면을 담당하는 사람을 설명합니다.

고객의 책임

- 파트너를 심사합니다.
- 파트너에게 부여된 네트워크 액세스를 구성합니다.
- AWS IoT SiteWise Edge를 실행하는 장치를 물리적으로 보호합니다.

AWS 책임

- 파트너를 고객 AWS 클라우드 리소스에서 격리합니다.

파트너의 책임

- 보안 기본값을 사용합니다.
- 패치 및 기타 적절한 업데이트를 통해 시간이 지나도 솔루션을 안전하게 유지합니다.
- 고객 데이터의 기밀을 유지합니다.
- 파트너 마켓플레이스에서 사용할 수 있는 다른 애플리케이션을 검토합니다.

이 기능의 미리 보기 단계에서는 파트너 장치에 AWS IoT SiteWise 캐시된 고객 데이터와 파트너 마켓플레이스를 통해 설치된 기타 애플리케이션이 파트너 장치에 캐시된 고객 데이터에 액세스할 수 있습니다.

구성 파일 생성

적절한 Siemens 계정과 IEM 인스턴스가 있으면 배포 유형의 Siemens Industrial SiteWise Edge 디바이스의 에지 게이트웨이를 생성할 수 있습니다.

구성 파일을 만들려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 Edge 게이트웨이를 선택합니다.
3. 게이트웨이 생성을 선택합니다.
4. 배포 유형으로는 Siemens 산업용 에지 디바이스 - 신규를 선택합니다.
5. SiteWise 에지 게이트웨이의 이름을 입력하거나 에서 생성한 AWS IoT SiteWise 이름을 사용하십시오.
6. (선택 사항) 고급 구성에서 다음을 수행합니다.

- 사물의 이름을 AWS IoT Core 입력하거나 에서 생성한 이름을 사용하십시오 AWS IoT SiteWise.
7. 게이트웨이 생성을 선택합니다.
 8. SiteWise Edge 게이트웨이 구성 파일 생성 대화 상자에서 생성 및 다운로드를 선택합니다. AWS IoT SiteWise AWS IoT SiteWise Edge 애플리케이션을 구성하는 데 사용할 구성 파일을 자동으로 생성합니다.

Important

구성 파일을 안전한 위치에 저장해야 합니다. 이 파일은 나중에 사용하게 됩니다.

이제 SiteWise 에지 게이트웨이를 만들었으니 다음을 수행하여 SiteWise 에지 게이트웨이 설정을 완료하십시오.

1. [데이터 소스 추가](#)
2. [퍼블리셔 구성 요소 구성](#)

구성 파일이 준비되고 SiteWise Edge 게이트웨이가 구성되면 Siemens Industrial AWS IoT SiteWise Edge Marketplace에서 Edge 애플리케이션을 다운로드하고 Siemens IEM (산업용 에지 관리) 포털을 사용하여 설치합니다. 그런 다음 지멘스 산업용 에지 관리 (IEM) 포털을 통해 Siemens 산업용 에지 디바이스에 액세스하고 에지 게이트웨이를 설치하려는 디바이스에 구성 파일을 업로드하십시오. SiteWise

문제 해결

Siemens 산업용 SiteWise 에지 디바이스의 에지 게이트웨이 문제를 해결하려면 지멘스 산업용 에지 관리 (IEM) 또는 지멘스 산업용 에지 디바이스 (IED) 포털을 통해 애플리케이션 로그에 액세스할 수 있습니다. 자세한 내용은 Siemens 설명서의 로그 [다운로드](#)를 참조하십시오.

'SESSION_TAKEN_OVER' 또는 'com.aws.greengrass.mqttclient'가 보입니다.

MqttClient: 스포일러를 통해 메시지를 게시하지 못했습니다. 다시 시도하겠습니다.' 로그에서

로그에 포함된 경고 SESSION_TAKEN_OVER 또는 오류가 표시되는 경우 여러 장치의 여러 SiteWise Edge 게이트웨이에 동일한 구성 파일을 사용하려고 시도한 것일 수 있습니다.

com.aws.greengrass.mqttclient.MqttClient: Failed to publish the message via Spooler and will retry. /greengrass/v2/logs/greengrass.log 각 SiteWise 에지 게이트웨이에 연결하려면 고유한 구성 파일이 필요합니다. AWS 계정

com.aws.greengrass.deploy가 보이네요. IotJobsHelper: 배포 작업을 찾을 수 없습니다. ' 또는 '배포 결과가 이미 보고되었습니다.' 로그에서

Deployment result already reported.로그에

com.aws.greengrass.deployment.IotJobsHelper: No deployment job found. 또는 이 보이면 동일한 구성 파일을 재사용하려고 하는 것일 수 있습니다. /greengrass/v2/logs/greengrass.log

해결 방법은 여러 가지가 있습니다.

- 구성 파일을 재사용하려면 다음과 같이 하십시오.
 1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
 2. 탐색 창에서 게이트웨이를 선택합니다.
 3. 재사용할 SiteWise 에지 게이트웨이를 선택합니다.
 4. 업데이트 탭을 선택합니다.
 5. 다른 퍼블리셔 버전을 선택하고 배포를 선택합니다.
- 의 단계에 따라 새 구성 [구성 파일 생성](#) 파일을 생성합니다.

로그에 '구성 파일에 AWS_REGION이 누락되었습니다'가 표시됩니다.

Siemens Config file missing AWS_REGION 로그에 보면 구성 파일의 JSON이 손상된 것입니다. 새 구성 파일을 생성해야 합니다. 의 [구성 파일 생성](#) 단계에 따라 새 구성 파일을 생성합니다.

문의처

- 애플리케이션에 대한 액세스를 요청하려면 [AWS IoT SiteWise 콘솔](#)로 이동하여 액세스 요청을 선택합니다.
- 애플리케이션 문제 해결에 도움이 필요하면 [AWS IoT SiteWise 콘솔](#)로 이동하여 SiteWise Edge Gateway의 세부 정보 페이지로 이동한 다음 지원 받기를 선택합니다.

SiteWise 에지 게이트웨이의 자산 필터링

에지 필터링을 사용하면 데이터 처리에 사용하기 위해 자산의 하위 집합만 특정 SiteWise 에지 게이트웨이로 전송하여 자산을 보다 효율적으로 관리할 수 있습니다. 자산이 트리 또는 상위-하위 구조로 정렬된 경우 SiteWise Edge 게이트웨이의 IAM 역할에 연결된 IAM 정책을 설정하여 트리의 루트, 즉 부모 및 해당 하위 항목만 특정 Edge 게이트웨이로 전송하도록 허용할 수 있습니다. SiteWise

Note

기존 자산을 트리 구조로 정렬하는 경우 구조를 만든 후 구조에 추가한 각 기존 자산으로 이동하여 [편집] 을 선택한 다음 [저장] 을 선택하여 새 구조가 AWS IoT SiteWise 인식되는지 확인합니다.

에지 필터링 설정

에지 게이트웨이의 IAM 역할에 다음 IAM 정책을 추가하고 `< root-asset-id ># SiteWise` 에지 게이트웨이로 전송하려는 루트 자산의 ID로 대체하여 에지 게이트웨이에 에지 필터링을 설정합니다. SiteWise SiteWise

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise>ListAssociatedAssets"
      ],
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringNotLike": {
          "iotsitewise:assetHierarchyPath": "/<root-asset-id>*"
        }
      }
    }
  ]
}
```

현재 SiteWise 에지 게이트웨이에 제거하려는 자산이 있는 경우 에지 게이트웨이에 로그인하고 다음 명령을 실행하여 캐시를 AWS IoT SiteWise 삭제하여 SiteWise Edge 게이트웨이를 강제로 동기화하십시오 SiteWise .

```
sudo rm /greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/sync-app/sync_resource_bundles/edge.json
```

엣지에서의 AWS IoT SiteWise API 사용

사용 가능한 AWS IoT SiteWise API의 하위 집합을 엣지별 API와 함께 사용하여 엣지의 자산 모델 및 해당 자산과 상호 작용할 수 있습니다. 에셋 모델은 엣지에서 실행되도록 구성되어야 합니다. 자세한 설명은 [엣지에서의 데이터 처리](#) 섹션을 참조하세요.

이러한 API를 사용하여 자산 모델 및 자산에 대한 데이터를 수집하고, 배포된 포털 및 대시보드 지표를 모니터링하고, 엣지에서 자산 데이터를 수집할 수 있습니다. 이는 웹 API 직접 호출 없이 AWS IoT SiteWise와 상호 작용할 수 있도록 네트워크의 중앙 호스트를 제공합니다.

주제

- [AWS IoT SiteWise 엣지 디바이스에 사용할 수 있는 모든 API](#)
- [AWS IoT SiteWise 엣지 디바이스와 함께 사용하기 위한 엣지 전용 API](#)
- [자습서: SiteWise Edge 게이트웨이에서 자산 모델 목록 가져오기](#)

AWS IoT SiteWise 엣지 디바이스에 사용할 수 있는 모든 API

엣지에서 디바이스로 작업할 때 다양한 API를 사용하여 AWS IoT SiteWise와 상호 작용하고 디바이스에서 로컬로 작업을 완료할 수 있습니다.

사용 가능한 AWS IoT SiteWise API

엣지 디바이스에서 사용할 수 있는 AWS IoT SiteWise API는 다음과 같습니다.

- [ListAssetModels](#)
- [DescribeAssetModel](#)
- [ListAssets](#)
- [DescribeAsset](#)
- [DescribeAssetProperty](#)

- [ListAssociatedAssets](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjectAssets](#)
- [ListProjects](#)
- [DescribeDashboard](#)
- [DescribePortal](#)
- [DescribeProject](#)

사용 가능한 엣지 전용 API

다음 API는 엣지에 있는 디바이스에서 로컬로 사용됩니다.

- [인증](#) - 이 API를 사용하여 API 직접 호출에 사용할 SigV4 임시 보안 인증을 가져옵니다.

AWS IoT SiteWise 엣지 디바이스와 함께 사용하기 위한 엣지 전용 API

엣지에서 사용할 수 있는 AWS IoT SiteWise API 외에도 엣지별 API도 있습니다. 이러한 엣지별 API가 아래에 설명되어 있습니다.

인증

SiteWise Edge 게이트웨이에서 자격 증명을 가져옵니다. 로컬 사용자를 추가하거나 LDAP 또는 Linux 사용자 풀을 사용하여 시스템에 연결해야 합니다. 사용자 추가에 대한 자세한 내용은 [LDAP](#) 또는 [Linux 사용자 풀](#)을 참조하십시오.

요청 구문

```
POST /authenticate HTTP/1.1
Content-type: application/json
{
  "username": "string",
  "password": "string",
  "authMechanism": "string"
```

```
}

```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

사용자 이름

요청 호출을 검증하는 데 사용되는 사용자 이름입니다.

유형: 문자열

필수 항목 여부: 예

비밀번호

보안 인증을 요청하는 사용자의 비밀번호입니다.

유형: 문자열

필수 항목 여부: 예

authMechanism

호스트에서 이 사용자의 유효성을 검사하기 위한 인증 방법입니다.

유형: 문자열

유효한 값: ldap, linux, winnt

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json
{
  "accessKeyId": "string",
  "secretAccessKey": "string",
  "sessionToken": "string",
  "region": "edge"
}
```

```
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 JSON 형식으로 반환됩니다.

accessKeyId

임시 보안 인증 정보를 식별하는 액세스 키 ID입니다.

길이 제약: 최소 길이는 16입니다. 최대 길이 128.

패턴: `[\w]*`

secretAccessKey

요청에 서명하는 데 사용되는 비밀 액세스 키입니다.

유형: 문자열

sessionToken

사용자가 임시 보안 인증을 사용하기 위해 서비스 API에 전달해야 하는 토큰입니다.

유형: 문자열

region

API 직접 호출 대상 지역입니다.

유형: CONSTANT - edge

Errors

IllegalArgumentException

제공된 본문 문서의 형식이 잘못되어 요청이 거부되었습니다. 오류 메시지에 특정 오류가 설명되어 있습니다.

HTTP 상태 코드: 400

AccessDeniedException

사용자는 현재 ID 제공자에 기반한 유효한 보안 인증을 가지고 있지 않습니다. 오류 메시지에 인증 메커니즘이 설명되어 있습니다.

HTTP 상태 코드: 403

TooManyRequestsException

요청이 인증 시도 한도에 도달했습니다. 오류 메시지는 인증을 새로 시도할 때까지 기다려야 하는 시간이 포함됩니다.

HTTP Status Code: 429

자습서: SiteWise Edge 게이트웨이에서 자산 모델 목록 가져오기

사용 가능한 AWS IoT SiteWise API의 하위 집합을 엣지별 API와 함께 사용하여 엣지의 자산 모델 및 해당 자산과 상호 작용할 수 있습니다. 이 자습서에서는 에지 게이트웨이에 대한 임시 자격 증명을 얻고 AWS IoT SiteWise 에지 게이트웨이의 자산 모델 목록을 가져오는 방법을 안내합니다 SiteWise .

사전 조건

이 자습서의 단계에서는 다양한 도구를 사용할 수 있습니다. 이러한 도구를 사용하려면 해당 필수 조건이 설치되어 있어야 합니다.

이 튜토리얼을 완료하려면 다음이 필요합니다.

- 배포 및 실행 중인 [SiteWise 에지 게이트웨이 요구 사항](#)
- 포트 443을 통해 동일한 네트워크의 SiteWise 에지 게이트웨이에 액세스할 수 있습니다.
- [OpenSSL](#) 설치됨
- (AWS OpsHub용AWS IoT SiteWise) [AWS OpsHub애플리케이션용 AWS IoT SiteWise](#)
- (curl) [curl](#) 설치됨
- (Python) [urllib3](#) 설치됨
- (Python) [Python3](#) 설치됨
- (Python) [Boto3](#) 설치됨
- (Python) [BotoCore](#)설치

1단계: SiteWise 에지 게이트웨이 서비스 서명 인증서 받기

SiteWise 에지 게이트웨이에서 사용할 수 있는 API에 TLS 연결을 설정하려면 신뢰할 수 있는 인증서가 필요합니다. OpenSSL 또는 AWS OpsHub for를 사용하여 이 인증서를 생성할 수 있습니다. AWS IoT SiteWise

OpenSSL

Note

이 명령을 실행하려면 [OpenSSL](#)이 설치되어 있어야 합니다.

터미널을 열고 다음 명령을 실행하여 SiteWise Edge 게이트웨이에서 서명된 인증서를 가져옵니다. SiteWise 에지 게이트웨이의 <sitewise_gateway_ip> IP로 바꾸십시오.

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | openssl
x509 -outform PEM > GatewayCert.pem
```

AWS OpsHub for AWS IoT SiteWise

AWS IoT SiteWise에 대해 AWS OpsHub를 사용할 수 있습니다. 자세한 설명은 [SiteWise 에지 게이트웨이 관리](#) 섹션을 참조하세요.

이 자습서에서는 다운로드한 SiteWise Edge 게이트웨이 인증서의 절대 경로를 사용합니다. 다음 명령을 실행하여 인증서의 전체 경로를 내보내고 <absolute_path_to_certificate>를 인증서 경로로 대체합니다.

```
export PATH_TO_CERTIFICATE='<absolute_path_to_certificate>'
```

2단계: SiteWise 에지 게이트웨이 호스트 이름 가져오기

Note

이 명령을 실행하려면 [OpenSSL](#)이 설치되어 있어야 합니다.

자습서를 완료하려면 SiteWise 에지 게이트웨이의 호스트 이름이 필요합니다. 에지 게이트웨이의 호스트 이름을 가져오려면 SiteWise 에지 게이트웨이의 <sitewise_gateway_ip> IP로 대체하여 다음을 실행합니다. SiteWise

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | grep -Po
'CN = \K.*' | head -1
```

다음 명령을 실행하여 나중에 사용할 수 있도록 호스트 이름을 내보내고 에지 게이트웨이의 호스트 이름으로 <your_edge_gateway_hostname> 대체합니다 SiteWise .

```
export GATEWAY_HOSTNAME='<your_edge_gateway_hostname>'
```

3단계: 에지 게이트웨이의 임시 자격 SiteWise 증명 가져오기

이제 서명된 인증서와 SiteWise 에지 게이트웨이의 호스트 이름을 얻었으므로 게이트웨이에서 API를 실행할 수 있도록 임시 자격 증명을 가져와야 합니다. API를 사용하여 에지 게이트웨이를 통해 AWS OpsHub AWS IoT SiteWise 또는 SiteWise Edge 게이트웨이에서 직접 이러한 자격 증명을 가져올 수 있습니다.

Important

자격 증명은 4시간마다 만료되므로 에지 게이트웨이에서 API를 사용하기 직전에 자격 증명을 받아야 합니다 SiteWise . 보안 인증을 4시간 이상 캐시하지 마십시오.

를 사용하여 임시 자격 증명을 받으십시오. AWS OpsHubAWS IoT SiteWise

Note

[AWS IoT SiteWise 애플리케이션용 AWS OpsHub](#)을 설치해야 합니다.

AWS IoT SiteWise 애플리케이션용 AWS OpsHub을 사용하여 임시 보안 인증을 받으려면 다음과 같이 하십시오.

1. 애플리케이션에 로그인합니다.
2. 설정을 선택합니다.
3. 인증에서 보안 인증 복사를 선택합니다.
4. 환경에 맞는 옵션을 확장하고 복사를 선택합니다.
5. 나중에 사용할 수 있도록 보안 인증을 저장합니다.

SiteWise 에지 게이트웨이 API를 사용하여 임시 자격 증명을 받으세요.

SiteWise Edge gateway API를 사용하여 임시 자격 증명을 가져오려면 Python 스크립트 또는 curl을 사용할 수 있습니다. 먼저 SiteWise Edge 게이트웨이의 사용자 이름과 암호가 있어야 합니다. SiteWise

에지 게이트웨이는 SigV4 인증 및 권한 부여를 사용합니다. 사용자 추가에 대한 자세한 내용은 [LDAP](#) 또는 [Linux 사용자 풀](#)을 참조하십시오. 이러한 자격 증명은 다음 단계에서 API를 사용하는 데 필요한 SiteWise 에지 게이트웨이의 로컬 자격 증명을 가져오는 데 사용됩니다. AWS IoT SiteWise

Python

Note

[urllib3](#)와 [Python3](#)이 설치되어 있어야 합니다.

Python을 사용하여 보안 인증을 가져오려면

1. `get_credentials.py`라는 파일을 생성하고 다음 코드를 그 안에 복사합니다.

```

...
The following demonstrates how to get the credentials from the SiteWise Edge
gateway. You will need to add local users or connect your system to LDAP/AD
https://docs.aws.amazon.com/iot-sitewise/latest/userguide/manage-gateways-
ggv2.html#create-user-pool

Example usage:
    python3 get_credentials.py -e https://<gateway_hostname> -c
    <path_to_certificate> -u '<gateway_username>' -p '<gateway_password>' -m
    '<method>'
...
import urllib3
import json
import urllib.parse
import sys
import os
import getopt

"""
This function retrieves the AWS IoT SiteWise Edge gateway credentials.
"""
def get_credentials(endpoint, certificatePath, user, password, method):
    http = urllib3.PoolManager(cert_reqs='CERT_REQUIRED', ca_certs=
    certificatePath)
    encoded_body = json.dumps({
        "username": user,
        "password": password,

```

```
        "authMechanism": method,
    })

    url = urllib.parse.urljoin(endpoint, "/authenticate")

    response = http.request('POST', url,
        headers={'Content-Type': 'application/json'},
        body=encoded_body)

    if response.status != 200:
        raise Exception(f'Failed to authenticate! Response status
{response.status}')

    auth_data = json.loads(response.data.decode('utf-8'))

    accessKeyId = auth_data["accessKeyId"]
    secretAccessKey = auth_data["secretAccessKey"]
    sessionToken = auth_data["sessionToken"]
    region = "edge"

    return accessKeyId, secretAccessKey, sessionToken, region

def print_help():
    print('Usage:')
    print(f'{os.path.basename(__file__)} -e <endpoint> -c <path/to/certificate>
-u <user> -p <password> -m <method> -a <alias>')
    print('')
    print('-e, --endpoint    edge gateway endpoint. Usually the Edge gateway
hostname.')
    print('-c, --cert_path path to downloaded gateway certificate')
    print('-u, --user        Edge user')
    print('-p, --password    Edge password')
    print('-m, --method      (Optional) Authentication method (linux, winnt,
ldap), default is linux')
    sys.exit()

def parse_args(argv):
    endpoint = ""
    certificatePath = None
    user = None
    password = None
    method = "linux"
```

```
try:
    opts, args = getopt.getopt(argv, "he:c:u:p:m:",
["endpoint=", "cert_path=", "user=", "password=", "method="])
except getopt.GetoptError:
    print_help()

for opt, arg in opts:
    if opt == '-h':
        print_help()
    elif opt in ("-e", "--endpoint"):
        endpoint = arg
    elif opt in ("-u", "--user"):
        user = arg
    elif opt in ("-p", "--password"):
        password = arg
    elif opt in ("-m", "--method"):
        method = arg.lower()
    elif opt in ("-c", "--cert_path"):
        certificatePath = arg

if method not in ['ldap', 'linux', 'winnt']:
    print("not valid method parameter, required are ldap, linux, winnt")
    print_help()

if (user == None or password == None):
    print("To authenticate against edge user, password have to be passed
together, and the region has to be set to 'edge'")
    print_help()

if(endpoint == ""):
    print("You must provide a valid and reachable gateway hostname")
    print_help()

return endpoint,certificatePath, user, password, method

def main(argv):
    # get the command line args
    endpoint, certificatePath, user, password, method = parse_args(argv)

    accessKeyId, secretAccessKey, sessionToken, region=get_credentials(endpoint,
certificatePath, user, password, method)
```

```

print("Copy and paste the following credentials into the shell, they are
valid for 4 hours:")
print(f"export AWS_ACCESS_KEY_ID={accessKeyId}")
print(f"export AWS_SECRET_ACCESS_KEY={secretAccessKey}")
print(f"export AWS_SESSION_TOKEN={sessionToken}")
print(f"export AWS_REGION={region}")
print()

```

```

if __name__ == "__main__":
    main(sys.argv[1:])

```

2. 터미널에서 `get_credentials.py`를 실행하여 `<gateway_username>` 및 `<gateway_password>`를 생성한 보안 인증으로 교체합니다.

```

python3 get_credentials.py -e https://$GATEWAY_HOSTNAME -c $PATH_TO_CERTIFICATE
-u '<gateway_username>' -p '<gateway_password>' -m 'linux'

```

curl

Note

[curl](#)이 설치되어 있어야 합니다.

curl을 사용하여 보안 인증을 가져오려면

1. `<gateway_username>` 및 `<gateway_password>`를 생성한 자격 증명으로 교체하여 터미널에서 다음 명령을 실행합니다.

```

curl --cacert $PATH_TO_CERTIFICATE --location \
-X POST https://$GATEWAY_HOSTNAME:443/authenticate \
--header 'Content-Type: application/json' \
--data-raw '{
    "username": "<gateway_username>",
    "password": "<gateway_password>",
    "authMechanism": "linux"
}'

```

응답은 다음과 같아야 합니다.

```
{
  "username": "sweuser",
  "accessKeyId": "<accessKeyId>",
  "secretAccessKey": "<secretAccessKey>",
  "sessionToken": "<sessionToken>",
  "sessionExpiryTime": "2022-11-17T04:51:40.927095Z",
  "authMechanism": "linux",
  "role": "edge-user"
}
```

2. 터미널에서 다음 명령을 실행합니다.

```
export AWS_ACCESS_KEY_ID=<accessKeyId>
export AWS_SECRET_ACCESS_KEY=<secretAccessKey>
export AWS_SESSION_TOKEN=<sessionToken>
export AWS_REGION=edge
```

4단계: SiteWise 에지 게이트웨이의 자산 모델 목록 가져오기

이제 서명된 인증서, SiteWise 에지 게이트웨이 호스트 이름, 에지 게이트웨이의 임시 자격 증명에 준비되었으므로 ListAssetModels API를 사용하여 SiteWise 에지 게이트웨이의 자산 모델 목록을 가져올 수 있습니다.

Python

Note

[Python3](#), [Boto3](#)가 설치되어 있어야 합니다. [BotoCore](#)

Python을 사용하여 자산 모델 목록을 가져오려면

1. list_asset_model.py라는 파일을 만들고 그 안에 다음 코드를 복사합니다.

```
import json
import boto3
import botocore
import os
```

```
# create the client using the credentials
client = boto3.client("iotsitewise",
    endpoint_url= "https://" + os.getenv("GATEWAY_HOSTNAME"),
    region_name=os.getenv("AWS_REGION"),
    aws_access_key_id=os.getenv("AWS_ACCESS_KEY_ID"),
    aws_secret_access_key=os.getenv("AWS_SECRET_ACCESS_KEY"),
    aws_session_token=os.getenv("AWS_SESSION_TOKEN"),
    verify=os.getenv("PATH_TO_CERTIFICATE"),
    config=botocore.config.Config(inject_host_prefix=False))

# call the api using local credentials
response = client.list_asset_models()
print(response)
```

2. 터미널에서 `list_asset_model.py`를 실행합니다.

```
python3 list_asset_model.py
```

curl

Note

[curl](#)이 설치되어 있어야 합니다.

curl을 사용하여 자산 모델 목록을 가져오려면

터미널에서 다음 명령을 실행합니다.

```
curl \
  --request GET https://$GATEWAY_HOSTNAME:443/asset-models \
  --cacert $PATH_TO_CERTIFICATE \
  --aws-sigv4 "aws:amz:edge:iotsitewise" \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  -H "x-amz-security-token:$AWS_SESSION_TOKEN"
```

응답은 다음과 같아야 합니다.

```
{
```

```

    "assetModelSummaries": [
      {
        "arn": "arn:aws:iotsitewise:{region}:{account-id}:asset-model/{asset-
model-id}",
        "creationDate": 1.669245291E9,
        "description": "This is a small example asset model",
        "id": "{asset-model-id}",
        "lastUpdateDate": 1.669249038E9,
        "name": "Some Metrics Model",
        "status": {
          "error": null,
          "state": "ACTIVE"
        }
      },
      .
      .
      .
    ],
    "nextToken": null
  }

```

백업 및 복원 SiteWise 에지 게이트웨이

이 항목에서는 SiteWise 에지 게이트웨이를 복원하고 메트릭 데이터를 백업하는 방법을 다룹니다. 동일한 시스템에서 손상된 SiteWise 에지 게이트웨이와 관련된 문제가 발생하여 문제를 해결해야 하는 경우 [SiteWise 에지 게이트웨이 문제 해결 AWS IoT SiteWise](#) 설명서를 참조하십시오.

Note

이 항목에서 다루는 지침은 AWS IoT Greengrass V2 버전 2.1.0 이상에 설치된 SiteWise 에지 게이트웨이를 대상으로 합니다.

지표 데이터의 일일 백업

새 머신에서 데이터를 전송하거나 복원하려는 경우 백업을 만드는 것이 중요합니다. 데이터를 백업하면 전송 또는 복원 프로세스 중에 운영 데이터가 손실될 위험이 크게 줄어듭니다.

influxdb 폴더 경로는 다음과 같습니다.

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/influxdb
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeProcessor\influxdb
```

폴더 전체와 그 아래의 모든 항목을 백업하는 것이 좋습니다.

1.0 SiteWise Edge에서 외장 하드 드라이브나 AWS 클라우드에 메트릭 데이터를 주기적으로 백업하는 것이 좋습니다.

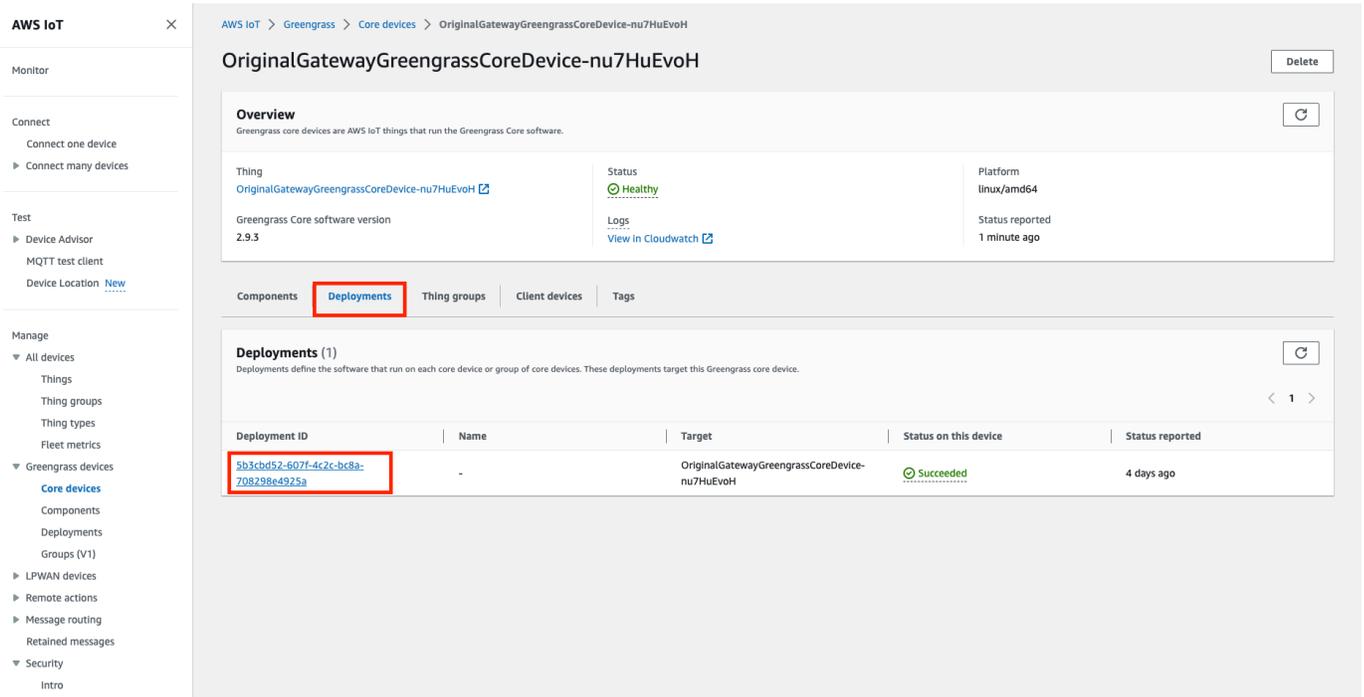
SiteWise 에지 게이트웨이 복원

다음 절차를 사용하여 SiteWise 에지 게이트웨이를 복원하십시오.

1. SiteWise 에지 게이트웨이를 만들 때 다운로드한 설치 스크립트를 사용하여 새 시스템에서 SiteWise 에지 게이트웨이를 복원합니다. [SiteWise Edge 게이트웨이를 설정하려면 로컬 장치에 SiteWise Edge 게이트웨이 소프트웨어 설치 절차를 읽어보십시오.](#)

설치 스크립트를 분실했거나 찾을 수 없는 경우 [AWS 고객 지원](#)에 문의하십시오.

2. SiteWise 에지 게이트웨이가 설치되면 [AWS IoT Greengrass 콘솔](#)에 로그인합니다.
3. 구성 요소를 재배포하려면 관리로 이동한 다음 AWS IoT Greengrass 디바이스에서 Core 디바이스를 선택합니다.
4. AWS IoT Greengrass 코어 디바이스 테이블에서 SiteWise 에지 게이트웨이에 해당하는 코어 디바이스를 선택합니다.
5. 디바이스 페이지에서 배포 탭을 열고 배포 ID를 선택하면 선택한 ID로 배포 페이지가 열립니다.



6. 배포 페이지로 이동한 후 오른쪽 상단의 작업 버튼을 누르고 수정 옵션을 선택하여 새 배포를 시작합니다. 배포를 구성합니다. 배포를 그대로 유지하려면 검토 및 배포로 건너뛰십시오.
7. 배포 상태가 Completed가 될 때까지 기다리십시오.

Note

또한 SiteWise Edge의 모든 구성 요소가 완전히 설정되고 실행되는 데 몇 분 정도 걸립니다.

AWS IoT SiteWise 데이터 복원

새 시스템에 데이터를 복원하려면 다음 절차를 사용하십시오.

1. influxdb 폴더를 새 컴퓨터에 복사합니다.
2. 터미널에서 다음 명령을 실행하여 SiteWise EdgeProcessor 구성 요소를 중지합니다.

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component stop -n aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component stop -n
aws.iot.SiteWiseEdgeProcesso
```

3. 데이터를 백업한 경로를 찾고 다음 명령을 실행합니다.

Linux

```
sudo yes | sudo cp -rf <influxdb_backup_path> /greengrass/v2/work/
aws.iot.SiteWiseEdgeProcessor/influxdb
```

PowerShell

```
Copy-Item -Recurse -Force <influxdb_backup_path>\* C:\greengrass
\v2\work\aws.iot.SiteWiseEdgeProcessor\
```

Windows

```
robocopy <influxdb_backup_path> C:\greengrass\v2\work
\aws.iot.SiteWiseEdgeProcessor\ /E
```

4. SiteWiseEdgeProcessor 구성 요소를 다시 시작합니다.

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component restart -n
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component restart -n
aws.iot.SiteWiseEdgeProcessor
```

백업 및 복원의 성공 여부 확인

이 절차를 사용하여 백업된 데이터 및 에지 게이트웨이 복원의 유효성을 검사하십시오. SiteWise

Note

이 절차를 수행하려면 를 설치해야 합니다. AWS OpsHub AWS IoT SiteWise 자세한 내용은 for 를 [사용한 SiteWise AWS OpsHub 에지 게이트웨이 관리를](#) 참조하십시오. AWS IoT SiteWise

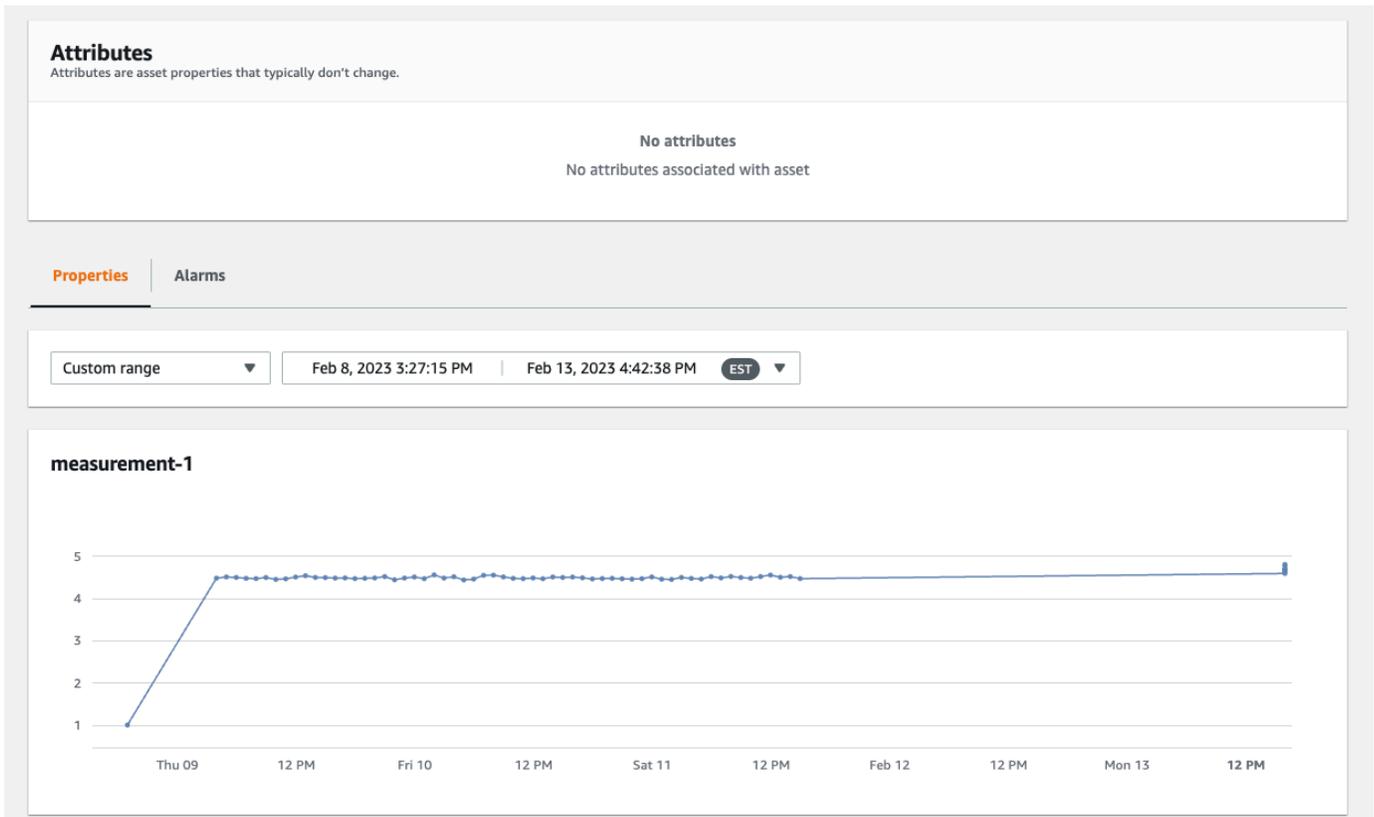
1. 공개 AWS OpsHub 대상. AWS IoT SiteWise
2. SiteWise Edge Gateway 설정 페이지에서 구성 요소 테이블에 나열된 각 구성 요소의 상태를 확인합니다. 상태 색상이 녹색이고 판독값에 RUNNING이 표시되는지 확인합니다.

The screenshot displays the 'Gateway' settings page in the AWS IoT SiteWise console. At the top, there is a green notification bar that says 'Connection successful.' Below this, the page is divided into several sections:

- Gateway configuration:** Shows the hostname/IP address (54.202.67.122), data collection pack (2.2.0, Status: Enabled), and data processing pack (2.1.29, Status: Enabled). It also shows the last sync time (2/13/2023 4:44 PM) and last sync status (Successful). A 'Sync' button is present.
- Authentication:** Provides options to download a server certificate or copy signature version 4 credentials.
- Logs:** Includes a filter for 'Last 1 hour' and a 'Download' button.
- Components:** A table listing the gateway components and their status. All three components are shown as 'RUNNING'.

Name	Status
aws.iot.SiteWiseEdgeProcessor	RUNNING
aws.iot.SiteWiseEdgeCollectorOpcua	RUNNING
aws.iot.SiteWiseEdgePublisher	RUNNING

3. 포털 대시보드에서 과거 데이터를 검증하여 이전 데이터와 새 데이터가 모두 제대로 설정되었는지 확인하십시오. 과거 데이터와 새 데이터 사이에는 다운타임이 있을 수 있습니다. 단, 데이터 포인트가 수집되지 않는 기간은 예외입니다.



에지 게이트웨이 백업 또는 복원에 문제가 발생하는 경우 다음 문제 [해결 항목인 SiteWise AWS IoT SiteWise 에지 게이트웨이](#) 문제 해결을 참조하십시오.

SiteWise 에지 게이트웨이 설정 (AWS IoT Greengrass Version 1)

Note

SiteWise 에서 AWS IoT Greengrass V1 실행 중인 에지 게이트웨이는 2021년 7월 29일 이전에 이 기능을 사용하기 시작한 경우에만 사용할 수 있습니다. 그렇지 않으면 [SiteWise Edge 게이트웨이가 에서 실행되도록 설정합니다](#). AWS IoT Greengrass V2

SiteWise Edge 게이트웨이를 AWS IoT SiteWise 사용하여 산업용 데이터를 전송하여 산업용 장비에서 데이터를 업로드할 수 있습니다. SiteWise 에지 게이트웨이는 데이터 산업 장비 AWS IoT SiteWise 간의 중개자 역할을 합니다. AWS IoT SiteWise SiteWise Edge 게이트웨이를 AWS IoT Greengrass 설정하기 위해 실행할 수 있는 모든 장치에 배포할 수 있는 AWS IoT Greengrass 구성 요소를 제공합니다. AWS IoT SiteWise [OPC-UA 서버 프로토콜과의 연결](#)을 지원합니다.

에서 AWS IoT Greengrass V1 실행되는 AWS IoT SiteWise 에지 게이트웨이가 있는 경우 SiteWise 에지 게이트웨이를 로 업그레이드할 수 있습니다. AWS IoT Greengrass V2 자세한 내용은 [SiteWise 에지 게이트웨이를 에서 로 업그레이드하기 위한 지침](#)을 참조하십시오. AWS IoT Greengrass V1 AWS IoT Greengrass V2

주제

- [AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 장치 선택](#)
- [AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 구성](#)
- [AWS IoT Greengrass V1 SiteWise 에지 게이트웨이에서 데이터 소스 구성](#)

AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 장치 선택

산업 운영에 가장 적합한 로컬 장치를 선택하십시오. 실행 가능한 모든 장치에서 SiteWise Edge 게이트웨이를 구성할 수 있습니다 AWS IoT Greengrass. 모든 로컬 장치는 다음 요구 사항을 충족해야 합니다.

- AWS IoT Greengrass Core 소프트웨어 v1.10.2 이상을 지원합니다. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 [지원되는 플랫폼 및 요구 조건](#)을 참조하십시오.
- 최소 4GB의 RAM이 있어야 합니다.
- 최소 10GB의 디스크 여유 공간이 있습니다.
- Java 8 가상 머신(JVM)을 지원합니다.

엣지에서 데이터를 처리하려는 경우 로컬 장치가 다음 요구 사항도 충족해야 합니다. AWS IoT SiteWise

- x86 64비트 쿼드코어 프로세서가 장착되어 있습니다.
- 최소 16GB의 RAM이 있습니다.
- Windows를 사용하는 경우 최소 32GB의 RAM이 있어야 합니다.
- 최소 256GB의 디스크 여유 공간이 있습니다.

간헐적인 인터넷 연결을 위해 데이터 캐싱에 필요한 디스크 공간은 다음 요인에 따라 다릅니다.

- 업로드되는 데이터 스트림의 개수
- 초당 데이터 스트림당 데이터 포인트
- 각 데이터 포인트의 크기

- 통신 속도
- 예기치 않은 네트워크 다운타임

데이터를 폴링하고 업로드하는 데 필요한 컴퓨팅 파워는 다음 요인에 따라 달라집니다.

- 업로드되는 데이터 스트림의 개수
- 초당 데이터 스트림당 데이터 포인트

AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 구성

AWS IoT SiteWise 에지 게이트웨이는 산업용 장비와 AWS IoT SiteWise 기기 사이의 중개자 역할을 합니다. 실행 AWS IoT Greengrass가 가능한 모든 장치에 SiteWise Edge 게이트웨이 소프트웨어를 배포할 수 있습니다. 자세한 정보는 [AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 장치 선택](#)을 참조하세요.

에지 게이트웨이의 데이터 처리 팩을 사용하여 에지 디바이스에서 로컬로 데이터를 처리할 수 SiteWise 있습니다. AWS IoT SiteWise SiteWise 에지 게이트웨이를 추가할 때 이 작업을 수행합니다 AWS IoT SiteWise. 엣지에서의 데이터 처리에 대한 자세한 정보는 [the section called “엣지 데이터 처리 활성화”](#)을 참조하십시오.

Note

로컬 및 회사 네트워크에 대한 IT 관리자 액세스 권한을 가진 사람과 함께 다음 단계를 완료하는 것이 좋습니다. 이러한 단계를 수행하려면 산업용 장비에 대해 잘 알고 있고 방화벽 설정을 구성할 권한이 있는 사람이 필요할 수 있습니다.

주제

- [SiteWise 에지 게이트웨이 환경 설정](#)
- [IAM 정책 및 역할 생성](#)
- [그룹 구성 AWS IoT Greengrass](#)
- [AWS IoT SiteWise 커넥터 구성](#)
- [SiteWise 에지 게이트웨이 추가 AWS IoT SiteWise](#)

SiteWise 에지 게이트웨이 환경 설정

이 절차에서는 함께 사용할 SiteWise 에지 게이트웨이를 AWS IoT Greengrass 설치하고 구성합니다
AWS IoT SiteWise.

Note

이 섹션에서는 apt 명령을 사용하여 패키지를 설치하는 방법이 포함되어 있습니다. Ubuntu 등의 시스템을 실행하는 시스템에 적용할 수 있습니다. 유사한 시스템을 사용하지 않는 경우, 배포 설명서를 참조하고 권장 패키지 설치 프로그램을 사용하십시오.

SiteWise 에지 게이트웨이를 설정하려면

1. 필요에 따라 SiteWise 에지 게이트웨이의 [BIOS](#) 설정을 다음과 같이 수정합니다.
 - a. 잠재적 전원 장애 발생 후 (해당하는 경우) SiteWise 에지 게이트웨이가 자동으로 다시 시작되도록 하십시오.
 - b. 해당하는 경우 SiteWise 에지 게이트웨이가 최대 절전 모드나 절전 모드로 전환되지 않도록 하십시오.
2. SiteWise 에지 게이트웨이가 인터넷에 연결되어 있는지 확인하십시오.
3. (선택 사항) 마우스, 키보드 및 모니터 없이 SiteWise Edge 게이트웨이를 사용하려면 다음 단계를 수행하여 SiteWise Edge 게이트웨이에 설정하십시오 ssh.
 - a. SSH 패키지를 아직 설치하지 않은 경우, 다음 명령을 실행합니다.

```
sudo apt install ssh
```

- b. 다음 명령을 실행합니다.

```
service ssh status
```

- c. SSH 서버가 실행 중인지 확인하려면 출력에서 Active: active (running)를 검색하십시오.
- d. 나가려면 Q를 누릅니다.

다음 명령을 실행하여 SSH를 사용하여 다른 컴퓨터에서 SiteWise Edge 게이트웨이에 연결합니다. **### ### ###** 로그인으로, **IP#** SiteWise Edge 게이트웨이의 IP 주소로 대체합니다.

```
ssh username@IP
```

-p *port-number* 인수를 사용하여 기본 포트 22 이외의 포트에 연결할 수 있습니다.

4. AWS IoT Greengrass Core 소프트웨어 v1.10.2 이상을 다운로드하여 설치하고 에지 게이트웨이를 위한 AWS IoT Greengrass 그룹을 생성하십시오 SiteWise . 이렇게 하려면 AWS IoT Greengrass 개발자 안내서의 [AWS IoT Greengrass 시작하기](#)의 지침을 참조하십시오.

빨리 시작하려면 [AWS IoT Greengrass 디바이스 설정](#) 스크립트를 실행하는 것이 좋습니다. AWS IoT Greengrass 요구 사항 및 프로세스를 더 면밀히 검토하려면 [모듈 1과 모듈 2의](#) 단계를 따라 설정하면 됩니다. AWS IoT Greengrass

Important

지원되는 [AWS 지역](#)을 AWS IoT SiteWise 검토하세요. 대상 AWS IoT Greengrass 지역을 선택할 때는 해당 지역도 지원하는지 확인하세요 AWS IoT SiteWise. 그렇지 않으면 SiteWise 에지 게이트웨이를 연결할 수 없습니다 AWS IoT SiteWise.

다음 단계를 계속하기 전에 SiteWise Edge 게이트웨이에 AWS IoT Greengrass Core 소프트웨어를 설치해야 합니다.

5. 다음 명령을 실행하여 Java 8을 설치합니다.

```
sudo apt update
sudo apt install openjdk-8-jre
```

이 안내서의 뒷부분에서 설치하는 SiteWise 에지 게이트웨이 소프트웨어는 Java 8 런타임을 사용합니다.

6. 다음 명령을 실행하여 Java가 성공적으로 설치되었는지 확인합니다.

```
java -version
```

7. AWS IoT Greengrass Core 소프트웨어는 java8 디렉토리를 가정합니다. 다음 명령을 실행하여 Java 설치를 해당 java8 디렉터리에 연결합니다.

```
sudo ln -s /usr/bin/java /usr/bin/java8
```

8. 다음 명령을 실행하여 `/var/sitewise` 데이터 디렉토리를 만들고 해당 디렉토리에 대한 `ggc_user` 권한을 부여합니다. AWS IoT SiteWise 이 디렉토리에 데이터를 저장합니다. 는 이 절차의 AWS IoT Greengrass 앞부분에서 설정할 `ggc_user` 때 만들었습니다.

```
sudo mkdir /var/sitewise
sudo chown ggc_user /var/sitewise
sudo chmod 700 /var/sitewise
```

가 AWS IoT SiteWise 사용하는 기본 디렉토리입니다. `/var/sitewise` 디렉토리 경로를 사용자 지정 (예: 바꾸기 `/var/sitewise/var/custom/path/`) 할 수 있지만 이렇게 하려면 SiteWise Edge 게이트웨이를 만든 후 추가 단계가 필요합니다. 자세한 내용은 [AWS IoT SiteWise 커넥터 구성](#)의 6단계를 참조하십시오.

9. 필요한 경우 IT 관리자에게 다음 엔드포인트와 포트를 로컬 네트워크 허용 목록에 추가하도록 요청하십시오.

- 443, 8443, 8883 포트.

⚠ Important

모든 네트워크 통신에 포트 443만 사용하도록 AWS IoT Greengrass Core를 구성할 수 있습니다. 자세한 내용은 AWS IoT Greengrass 개발자 안내서의 [포트 443 또는 네트워크 프록시를 통해 연결](#)을 참조하십시오.

- SiteWise 에지 게이트웨이의 IP 주소 (포트 443). IP 주소를 얻으려면 `ip address` 또는 `ifconfig` 명령을 실행하고 `inet` 값(예: `203.0.113.0`)을 기록하십시오.
- AWS IoT SiteWise 데이터 엔드포인트: `data.iotsitewise.region.amazonaws.com` (포트 443).
- SiteWise 에지 게이트웨이가 AWS 사용하는 엔드포인트는 다음과 같습니다. 이 `/greengrass-root/config/config.json` 파일은 다음에서 찾으실 수 있습니다. `greengrass-root`를 AWS IoT Greengrass 설치 루트로 바꿉니다.
 - `ggHost: greengrass-ats.iot.region.amazonaws.com`(포트 443, 8443 및 8883)
 - `iotHost: prefix-ats.iot.region.amazonaws.com`(포트 443, 8443 및 8883)

자세한 내용은 [AWS IoT Greengrass 엔드포인트 및 할당량](#)을 참조하십시오.

10. AWS IoT Greengrass Core 소프트웨어가 아직 실행되고 있지 않은 경우 다음 명령을 실행하여 AWS IoT Greengrass Core 소프트웨어를 시작합니다. `greengrass-root# ## ###` 교체하십시오. AWS IoT Greengrass 기본 `greengrass-root`는 `/greengrass`입니다.

```
cd /greengrass-root/ggc/core
sudo ./greengrassd start
```

Greengrass successfully started with PID: *some-PID-number* 메시지가 표시되어야 합니다.

11. SiteWise Edge 게이트웨이가 켜질 때 AWS IoT Greengrass Core 소프트웨어가 자동으로 시작되도록 구성하십시오. SiteWise 에지 게이트웨이 운영 체제의 설명서를 참조하십시오.

IAM 정책 및 역할 생성

SiteWise Edge 게이트웨이가 사용자 대신 액세스할 AWS IoT SiteWise 수 있도록 하려면 AWS Identity and Access Management (IAM) 정책 및 역할을 생성해야 합니다.

IAM 정책 및 역할을 생성하려면

1. [IAM 콘솔](#)로 이동합니다.
2. 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.

	Policy name	Type	Used as
<input type="radio"/>	AdministratorAccess	Job function	Permissions poli
<input type="radio"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="radio"/>	AlexaForBusinessFullAccess	AWS managed	None
<input type="radio"/>	AlexaForBusinessGatewayExecution	AWS managed	None
<input type="radio"/>	AlexaForBusinessReadOnlyAccess	AWS managed	None
<input type="radio"/>	AmazonAPIGatewayAdministrator	AWS managed	None
<input type="radio"/>	AmazonAPIGatewayInvokeFullAccess	AWS managed	None
<input type="radio"/>	AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	None

3. JSON 탭에서 정책 필드의 현재 내용을 삭제한 후, 다음 정책을 필드에 붙여 넣으십시오.

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": "iotsitewise:BatchPutAssetPropertyValue",
        "Resource": "*"
      }
    ]
  }

```

Note

보안을 강화하기 위해 Condition 속성에 AWS IoT SiteWise 자산 계층 경로를 지정할 수 있습니다. 다음 예제는 자산 계층 구조 경로를 지정하는 신뢰 정책입니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/root node asset ID",
            "/root node asset ID/*"
          ]
        }
      }
    }
  ]
}

```

4. 정책 검토를 선택합니다.
5. 정책의 이름과 설명을 입력한 다음 정책 생성을 선택하십시오.
6. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.

Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

Create role Delete role

Search

Role name	Description
<input type="checkbox"/> Admin	
<input type="checkbox"/> AwsSecurityAudit	

7. 신뢰할 수 있는 엔티티 유형 선택 아래에서 AWS 서비스를 선택합니다. 이 역할을 사용할 서비스 선택에서 이 역할을 사용할 서비스로 Greengrass를 선택한 후, 다음: 권한을 선택하십시오.

Create role

- 1
- 2
- 3
- 4

Select type of trusted entity



AWS service
EC2, Lambda and others



Another AWS account
Belonging to you or 3rd party



Web identity
Cognito or any OpenID provider



SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

API Gateway	CodeBuild	EC2 - Fleet	Inspector	Redshift
AWS Support	CodeDeploy	EKS	IoT	Rekognition
AppSync	Config	EMR	Kinesis	S3
Application Auto Scaling	Connect	ElasticCache	Lambda	SMS
Application Discovery Service	DMS	Elastic Beanstalk	Lex	SNS
Auto Scaling	Data Lifecycle Manager	Elastic Container Service	Machine Learning	SWF
Batch	Data Pipeline	Elastic Transcoder	Macie	SageMaker
CloudFormation	DeepLens	ElasticLoadBalancing	MediaConvert	Service Catalog
CloudHSM	Directory Service	Glue	OpsWorks	Step Functions
CloudTrail	DynamoDB	Greengrass	RAM	Storage Gateway
CloudWatch Events	EC2	GuardDuty	RDS	Trusted Advisor

Select your use case

* Required

Cancel

Next: Permissions

8. 생성한 정책을 검색하고 확인란을 선택한 후 다음: 태그를 선택하십시오.

Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy ↻

Filter policies Showing 1 result

	Policy name ▼	Used as	Description
<input checked="" type="checkbox"/>	SiteWiseDemo	None	Policy for the SiteWise demo.

▶ Set permissions boundary

* Required

Cancel

Previous

Next: Tags

9. (선택 사항) 역할에 태그를 추가한 후 다음: 검토를 선택하십시오.
10. 역할 이름 및 설명을 입력한 후 역할 생성을 선택합니다.

Create role



Review

Provide the required information below and review this role before you create it.

Role name* Use alphanumeric and '+=,@-_' characters. Maximum 64 characters.

Role description Maximum 1000 characters. Use alphanumeric and '+=,@-_' characters.

Trusted entities AWS service: greengrass.amazonaws.com

Policies [SiteWiseDemo](#)

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

[Cancel](#)

[Previous](#)

[Create role](#)

11. 녹색 배너에서 새 역할에 대한 링크를 선택하십시오. 검색 필드를 사용하여 역할을 찾을 수도 있습니다.

✔ The role **SiteWiseDemo** has been created.

Create role
Delete role

Role name ▾	Description
<input type="checkbox"/> Admin	
<input type="checkbox"/> AwsSecurityAudit	
<input type="checkbox"/> AwsSecurityNacundaAudit	
<input type="checkbox"/> AWSServiceRoleForIscangardControllerRoleInternal	This role will allow Iscangard to manage a

12. 신뢰 관계 탭을 선택한 후 신뢰 관계 편집을 선택합니다.

[Roles](#) > [SiteWiseDemo](#)

Summary

Role ARN	arn:aws:iam::[redacted]:role/SiteWiseDemo 🔗
Role description	Allows Greengrass to call AWS services on your behalf. Edit
Instance Profile ARNs	🔗
Path	/
Creation time	2018-11-21 13:56 PST
Maximum CLI/API session duration	1 hour Edit

Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

Edit trust relationship

Trusted entities

The following trusted entities can assume this role.

Conditions

The following conditio

13. 정책 필드의 현재 내용을 다음으로 바꾸고 신뢰 정책 업데이트를 선택하십시오.

```

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

```

```

    "Service": "greengrass.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
]
}

```

그룹 구성 AWS IoT Greengrass

그룹에 IAM 역할을 연결하고 스트림 관리자를 활성화하려면

1. [AWS IoT Greengrass 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창의 Greengrass에서 그룹을 선택한 후 [SiteWise 에지 게이트웨이 환경 설정](#)에서 생성한 그룹을 선택합니다.

The screenshot shows the AWS IoT Greengrass console interface. On the left, the 'Groups' menu item is circled in red. The main content area displays 'Greengrass groups (1)' with a search bar and a table of groups. The table has columns for Name, ID, and Created. One group is listed with the name 'SiteWiseDemo', which is circled in red. The ID is 'a1b2c3d4-5678-90ab-cdef-11111EXAMPLE' and it was created '9 months ago'.

<input type="checkbox"/>	Name	ID	Created
<input type="checkbox"/>	SiteWiseDemo	a1b2c3d4-5678-90ab-cdef-11111EXAMPLE	9 months ago

3. 왼쪽 탐색 창에서 설정을 선택합니다. 그룹 역할 섹션에서 역할 추가를 선택합니다.

GREENGRASS GROUP
SiteWiseDemo
Not deployed Actions ▾

Deployments **Group Role** Add Role

Subscriptions No role has been attached to the SiteWiseDemo Group

Cores

Devices **Group ID**

Lambdas 1ff7b6c9-06d9-46f5-9f3e-88894dc19b37

Resources

Connectors Certification authority (CA) and local connection configuration

Tags

Settings

Device certificate lifetime period
By changing this setting you control the period during which a Device can establish a communication with its Core. The next new period will be 7 days.

4. [IAM 정책 및 역할 생성](#)에서 생성한 역할을 선택한 다음 저장을 선택하십시오.

Your Group's IAM Role

Adding an IAM Role to your Group establishes a trust relationship between your trusting account and the Core.

Select an IAM Role with a Greengrass Role Type

Search Role name

SiteWiseDemo

Cancel Back Save

5. 설정 페이지의 스트림 관리자 섹션에서 편집을 선택합니다.

스트림 관리자는 AWS IoT Greengrass Core에서 데이터를 AWS 클라우드로 스트리밍할 수 있게 AWS IoT Greengrass 해주는 기능입니다. SiteWise 에지 게이트웨이를 사용하려면 스트림 관리자를 활성화해야 합니다. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 [AWS IoT Greengrass Core에서 데이터 스트림 관리](#)를 참조하십시오.

[Update default Lambda execution configuration](#)

Stream manager

[Edit](#)

Stream manager enables the Core to ingest and process data streams and export them to cloud targets. [Learn more](#)

Status

Disabled

CloudWatch logs configuration

[Edit](#)

6. 활성화됨을 선택한 다음, 저장을 선택합니다.
7. 왼쪽 상단 모서리에서 서비스를 선택하여 다음 절차를 준비하십시오.

AWS IoT SiteWise 커넥터 구성

이 절차에서는 Greengrass 그룹의 AWS IoT SiteWise 커넥터를 구성합니다. 구성 요소는 일반 에지 시나리오의 개발 라이프사이클을 가속화하는 사전 빌드된 모듈입니다. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 [AWS IoT Greengrass 커넥터](#)를 참조하십시오.

커넥터를 AWS IoT SiteWise 구성하려면

1. [AWS IoT Greengrass 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창의 Greengrass에서 그룹을 선택한 후 [SiteWise 에지 게이트웨이 환경 설정](#)에서 생성한 그룹을 선택합니다.

The screenshot shows the AWS IoT Greengrass console interface. On the left, the navigation menu includes 'Monitor', 'Onboard', 'Manage', and 'Greengrass'. Under 'Greengrass', the 'Groups' option is circled in red. The main content area displays 'Greengrass groups (1)' with a search bar and a table of groups. The table has columns for 'Name', 'ID', and 'Created'. One group is listed with the name 'SiteWiseDemo', which is circled in red. The ID is 'a1b2c3d4-5678-90ab-cdef-11111EXAMPLE' and it was created '9 months ago'. Buttons for 'Delete' and 'Create group' are visible at the top right.

3. 왼쪽 탐색 페이지에서 커넥터를 선택합니다. 커넥터 페이지에서 커넥터 추가를 선택합니다.

GREENGRASS GROUP

SiteWiseDemo

Not deployed Actions ▾

- Deployments
- Subscriptions
- Cores
- Devices
- Lambdas
- Resources
- Connectors**
- Tags
- Settings

Connectors

Connectors are modules that provide built-in integration with services, protocols, or infrastructure. [Learn more](#)



Accelerate your development

Connectors make it easier to develop applications by providing built-in integration with services, protocols, or infrastructure. [Learn more](#)

Add a connector

4. SiteWise목록에서 IoT를 선택하고 다음을 선택합니다.

ADD A CONNECTOR TO YOUR GREENGRASS GROUP

Select a connector

STEP 1/2

Select a connector to add to this group. Connectors that are already in the group are disabled in the list. [Learn more](#)

<input type="radio"/> CloudWatch Metrics	Version: 2	Learn more
<input type="radio"/> Device Defender	Version: 2	Learn more
<input type="radio"/> Docker Application Deployment	Version: 1	Learn more
<input checked="" type="radio"/> IoT SiteWise	Version: 2	Learn more
<input type="radio"/> Iot Analytics	Version: 2	Learn more
<input type="radio"/> Kinesis Firehose	Version: 3	Learn more
<input type="radio"/> ML Feedback	Version: 1	Learn more
<input type="radio"/> ML Image Classification ARMv7	Version: 2	Learn more
<input type="radio"/> ML Image Classification Aarch64 JTX2	Version: 2	Learn more
<input type="radio"/> ML Image Classification x86_64	Version: 2	Learn more

Cancel
Next

5. 서버에 인증이 필요한 경우 서버의 사용자 이름과 AWS Secrets Manager 암호로 암호를 만들 수 있습니다. 그런 다음 각 암호를 Greengrass 그룹에 연결하고 사용자 이름/비밀번호 암호에 대한 ARN 목록에서 선택할 수 있습니다. 암호 생성 및 구성에 대한 자세한 내용은 [원본 인증 구성](#)을 참조하십시오. 나중에 커넥터에 암호를 추가할 수도 있습니다.

List of ARNs for OPC-UA username/password secrets (optional)

List of AWS Secret ARNs

2 secrets selected		Create ↗	Refresh	Clear	Close
Search					
<input checked="" type="checkbox"/>	greengrass-factory1-auth				
<input checked="" type="checkbox"/>	greengrass-factory2-auth				

- 다른 `/var/sitewise` 경로로 SiteWise 에지 게이트웨이를 설정하는 경우 로컬 스토리지 경로에 해당 경로를 입력합니다.
- (선택 사항) 커넥터의 최대 디스크 버퍼 크기를 입력합니다. AWS IoT Greengrass 코어와 AWS 클라우드 연결이 끊어지면 커넥터는 성공적으로 연결될 때까지 데이터를 캐시합니다. 캐시 크기가 최대 디스크 버퍼 크기를 초과하면 커넥터는 대기열에서 가장 오래된 데이터를 폐기합니다.
- 추가를 선택합니다.
- 오른쪽 상단 모서리의 작업 메뉴에서 배포를 선택하십시오.
- Automatic detection(자동 검색)을 선택하여 배포를 시작합니다.

배포가 실패하면 배포를 다시 선택하십시오. 배포가 계속 실패하면 [AWS IoT Greengrass 배포 문제 해결](#)을 참조하십시오.

SiteWise 에지 게이트웨이 추가 AWS IoT SiteWise

이 절차에서는 SiteWise 에지 게이트웨이의 Greengrass 그룹을 에 추가합니다. AWS IoT SiteWise SiteWise 에지 게이트웨이를 등록하면 서비스가 데이터 소스 구성을 SiteWise 에지 게이트웨이에 배포할 수 있습니다. AWS IoT SiteWise

SiteWise 에지 게이트웨이를 추가하려면 AWS IoT SiteWise

- [AWS IoT SiteWise 콘솔](#)로 이동합니다.
- 게이트웨이 추가를 선택합니다.
- SiteWise 게이트웨이 추가 페이지에서 다음을 수행하십시오.

- a. SiteWise 에지 게이트웨이의 이름을 입력합니다. 쉽게 식별할 수 있도록 이름에 SiteWise 에지 게이트웨이의 위치를 포함하는 것을 고려해 보십시오.
- b. Greengrass 그룹 ID의 경우 앞서 생성한 Greengrass 그룹을 선택합니다.

Example

AWS IoT SiteWise > Gateways > Add SiteWise gateway

Add SiteWise gateway

Select a connected gateway

SiteWise utilizes an on-premises gateway that collects data from local data servers and uploads the selected data. Once you or your IT Administrator have installed the software, registered it to AWS IoT Greengrass and connected it to your local network you can add it to the SiteWise service.
[Learn more about this process and ordering hardware](#)

Gateway name
 Using the deployment location as a name makes identifying your gateway easier.

Greengrass group ID
 SiteWise gateway appliances must be connected to via AWS IoT Greengrass.

- c. (선택 사항) 엷지 기능의 경우 데이터 처리 팩을 선택합니다. 이를 통해 SiteWise 에지 게이트웨이와 엷지에 구성된 모든 자산 모델 및 자산 간의 통신이 가능합니다. 자세한 정보는 [the section called “엷지 데이터 처리 활성화”](#)을 참조하세요.

Important

SiteWise 에지 게이트웨이에 데이터 처리 팩을 추가하는 경우 AWS IoT Greengrass 그룹에 SiteWise 에지 커넥터를 구성하고 배포해야 합니다. 다음 단계를 따르십시오.

- d. 게이트웨이 추가를 선택합니다.
4. SiteWise 에지 게이트웨이에 데이터 처리 팩을 추가하는 경우 AWS IoT Greengrass 그룹에 AWS IoT SiteWise 데이터 프로세서 커넥터를 구성하고 배포하십시오. 다음 단계에 따라 AWS IoT SiteWise 데이터 프로세서 [the section called “AWS IoT SiteWise 커넥터 구성”](#) 커넥터를 구성하십시오.

- a. AWS IoT Greengrass 콘솔의 커넥터 선택에서 AWS IoT SiteWise 데이터 프로세서를 선택합니다.
- b. 로컬 스토리지 경로에는 SiteWise 에지 게이트웨이의 경로를 입력합니다.
- c. 추가를 선택합니다.
- d. 오른쪽 상단의 작업 메뉴에서 배포를 선택한 다음 자동 감지를 선택하여 배포를 시작합니다.

SiteWise 에지 게이트웨이를 배포한 후에는 SiteWise 에지 게이트웨이에서 데이터를 수집하는 데 사용할 각 산업 장비의 소스를 추가할 수 있습니다. 자세한 정보는 [데이터 소스 구성](#)을 참조하세요.

Amazon CloudWatch 지표를 보고 SiteWise Edge 게이트웨이가 연결되었는지 확인할 수 AWS IoT SiteWise 있습니다. 자세한 정보는 [AWS IoT Greengrass Version 1 게이트웨이 메트릭](#)을 참조하세요.

AWS IoT Greengrass V1 SiteWise 에지 게이트웨이에서 데이터 소스 구성

에지 게이트웨이를 설정한 후에는 AWS IoT SiteWise 에지 게이트웨이가 로컬 산업 장비의 데이터를 수집할 수 SiteWise 있도록 데이터 소스를 구성할 수 있습니다. AWS IoT SiteWise 각 소스는 SiteWise Edge 게이트웨이가 산업 데이터 스트림을 연결하고 검색하는 OPC-UA 서버와 같은 로컬 서버를 나타냅니다. SiteWise 에지 게이트웨이 설정에 대한 자세한 내용은 [AWS IoT Greengrass V1 SiteWise 에지 게이트웨이 구성](#)을 참조하십시오.

Note

AWS IoT SiteWise 소스를 추가하거나 편집할 때마다 SiteWise 에지 게이트웨이를 다시 시작합니다. SiteWise 에지 게이트웨이는 다시 시작하는 동안에는 데이터를 수집하지 않습니다. 에지 게이트웨이를 다시 시작하는 데 걸리는 시간은 SiteWise 에지 게이트웨이 소스의 태그 수에 따라 달라집니다. SiteWise 재시작 시간은 몇 초 (태그가 거의 없는 SiteWise 에지 게이트웨이의 경우) 에서 몇 분 (태그가 많은 SiteWise 에지 게이트웨이의 경우) 까지 걸릴 수 있습니다.

소스를 생성한 후에는 데이터 스트림을 자산 속성과 연결할 수 있습니다. 자산을 만들고 사용하는 방법에 대한 자세한 내용은 [산업 자산 모델링 및 산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하십시오.

CloudWatch 지표를 보고 데이터 소스가 연결되어 있는지 확인할 수 AWS IoT SiteWise 있습니다. 자세한 정보는 [AWS IoT Greengrass Version 1 게이트웨이 메트릭](#)을 참조하세요.

현재, AWS IoT SiteWise 지원되는 데이터 소스 프로토콜은 다음과 같습니다.

- [OPC-UA](#) — 산업 machine-to-machine 자동화를 위한 A (M2M) 통신 프로토콜.

- [Modbus TCP](#) — 프로그래밍 가능 로직 컨트롤러(PLC)와 인터페이스하는 데 사용되는 데이터 통신 프로토콜입니다.
- [이더넷/IP\(EIP\)](#) — CIP(Common Industrial Protocol)를 표준 이더넷에 적용시키는 산업 네트워크 프로토콜입니다.

Note

SiteWise AWS IoT Greengrass V2 현재 실행 중인 에지 게이트웨이는 Modbus TCP 및 이더넷 IP 소스를 지원하지 않습니다.

주제

- [Modbus TCP 소스 설정](#)
- [이더넷/IP\(EIP\) 소스를 구성합니다.](#)
- [원본 인증 구성](#)
- [커넥터 업그레이드](#)

Modbus TCP 소스 설정

AWS IoT SiteWise 콘솔 또는 AWS IoT SiteWise 에지 게이트웨이 기능을 사용하여 Modbus TCP 소스를 정의하고 에지 게이트웨이에 추가할 수 있습니다. SiteWise 이 소스는 로컬 Modbus TCP 서버를 나타냅니다.

Note

- SiteWise AWS IoT Greengrass V2 현재 실행 중인 에지 게이트웨이는 Modbus TCP 소스를 지원하지 않습니다.
- Modbus TCP 소스를 사용하려면 AWS IoT SiteWise 커넥터를 설치해야 합니다.

Modbus TCP 소스를 사용하여 에지 게이트웨이에서 수신된 소스의 데이터 유형을 다른 데이터 유형으로 변환할 수 있습니다. SiteWise 소스 데이터 유형에 따라 대상 데이터에 대해 선택할 수 있는 데이터 유형이 결정됩니다. Modbus TCP 소스를 사용하여 바이트를 스왑하도록 선택할 수도 있습니다. 다음 표는 호환되는 소스 데이터 유형, 대상 데이터 유형 및 스왑 모드에 대한 자세한 정보를 제공합니다.

스왑 모드에 대한 자세한 내용은 Modbus 메시지 인코딩에 대한 [Modbus RTU 메시지에서 실제\(부동소수점\) 및 32비트 데이터를 인코딩하는 방법](#) 문서를 참조하십시오.

소스 데이터 유형	호환되는 대상 데이터 형식	호환되는 스왑 모드	호환되는 커넥터 버전
ASCII	String	noSwap	2
UTF8	String	noSwap	2
ISO8859	String	noSwap	2
Int16	Integer, Double, String	noSwap	1 및 2
Int32	Integer, Double, String	노스왑, 바이트스왑, byteWordSwap 워드스왑	1 및 2
Float	Double, String	노스왑 byteWordSwap, 바이트스왑, 워드스왑	1 및 2
불	불	noSwap	1 및 2
Hex-dump	String	noSwap	1 및 2

주제

- [모드버스 TCP 소스 구성 \(콘솔\)](#)
- [Modbus TCP 소스 \(CLI\) 설정](#)

모드버스 TCP 소스 구성 (콘솔)

모드버스 TCP 소스를 구성하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 게이트웨이를 선택합니다.

3. 원본을 만들려는 SiteWise Edge 게이트웨이에서 [Manage] 를 선택한 다음 [세부 정보 보기] 를 선택합니다.
4. 오른쪽 상단 모서리에서 New source(새 소스)를 선택합니다.
5. 프로토콜 옵션의 경우 Modbus TCP를 선택합니다.
6. Modbus TCP 소스 구성의 경우 소스 이름을 입력합니다.
7. IP 주소에 데이터 소스 서버 IP 주소를 입력합니다.
8. (선택 사항) 소스 서버의 포트 및 장치 ID를 입력합니다.
9. (선택 사항) 최소 요청 간 기간에는 서버로 전송되는 후속 요청 사이의 시간 간격을 입력합니다. SiteWise 에지 게이트웨이는 디바이스와 보유한 레지스터 수를 기반으로 최소 허용 간격을 자동으로 계산합니다.
10. 속성 그룹의 경우 이름을 입력합니다.
11. 속성의 경우:
 - a. 태그에 레지스터 세트의 속성 별칭을 입력합니다. 예를 들어 **TT-001**입니다.
 - b. 레지스터 주소에 레지스터 세트를 시작하는 레지스터 주소를 입력합니다.
 - c. 소스 데이터 유형에서 데이터를 변환하려는 Modbus TCP 데이터 유형을 선택합니다. 기본값은 Hex 덤프입니다.

 Note

선택한 소스 데이터 유형에 따라 선택할 수 있는 데이터 크기, 대상 데이터 유형 및 스왑 모드가 결정됩니다. 자세한 정보는 [the section called “Modbus TCP 소스 설정”](#)을 참조하세요.

- d. 데이터 크기에는 레지스터 주소에서 시작할 때 읽을 레지스터 수를 입력합니다. 이는 이 소스에 대해 선택한 소스 데이터 유형에 따라 결정됩니다.
 - e. 대상 데이터 유형에서 AWS IoT SiteWise 데이터를 변환할 데이터 유형을 선택합니다. 기본값은 String입니다. 대상 유형은 이 원본에 대해 선택한 원본 데이터 유형과 호환되어야 합니다. 자세한 정보는 [the section called “Modbus TCP 소스 설정”](#)을 참조하세요.
 - f. 스왑 모드의 경우 레지스터 세트에서 데이터를 읽는 데 사용할 데이터 스왑 모드를 선택합니다. 스왑 모드는 이 소스에 대해 선택한 소스 데이터 유형과 호환되어야 합니다. 자세한 정보는 [the section called “Modbus TCP 소스 설정”](#)을 참조하세요.
12. 스캔 속도의 경우 SiteWise Edge 게이트웨이가 레지스터를 읽을 속도를 업데이트하십시오. AWS IoT SiteWise Edge 게이트웨이에 허용되는 최소 스캔 속도를 자동으로 계산합니다. SiteWise

13. (선택 사항) 목적지에서 소스 데이터를 보낼 위치를 선택합니다. 기본적으로 소스는 데이터를 로 보냅니다 AWS IoT SiteWise. 대신 AWS IoT Greengrass 스트림을 사용하여 데이터를 로컬 대상 또는 클라우드로 내보낼 수 있습니다. AWS

Note

엣지에서 이 소스의 데이터를 처리하려면 소스 데이터의 대상으로 선택해야 AWS IoT SiteWise 합니다. AWS IoT SiteWise 엣지에서의 데이터 처리에 대한 자세한 정보는 [the section called “엣지 데이터 처리 활성화”](#)을 참조하십시오.

데이터를 다른 목적지로 보내려면

- a. 목적지 옵션에서 기타 목적지를 선택합니다.
- b. Greengrass 스트림 이름에 스트림의 정확한 이름을 입력합니다. AWS IoT Greengrass

Note

이미 만든 스트림을 사용하거나 새 AWS IoT Greengrass 스트림을 만들어 데이터를 내보낼 수 있습니다. 기존 스트림을 사용하려면 스트림의 정확한 이름을 입력해야 합니다. 그렇지 않으면 새 스트림이 만들어집니다. AWS IoT Greengrass 스트림 작업에 대한 자세한 내용은 AWS IoT Greengrass 개발자 가이드의 [데이터 스트림 관리](#)를 참조하십시오.

14. Add source(소스 추가)를 선택합니다.

AWS IoT SiteWise SiteWise 엣지 게이트웨이 구성을 AWS IoT Greengrass 코어에 배포합니다. 배포를 수동으로 시작할 필요는 없습니다.

Modbus TCP 소스 (CLI) 설정

SiteWise 엣지 게이트웨이 기능에서 Modbus TCP 데이터 소스를 정의할 수 있습니다. 단일 기능 구성에서 모든 Modbus TCP 소스를 정의해야 합니다.

Note

Modbus TCP 소스를 사용하려면 AWS IoT SiteWise 커넥터를 설치해야 합니다.

이 기능에는 다음과 같은 버전이 있습니다.

버전	네임스페이스
1	iotsitewise:modbuscollector:1

Modbus TCP 기능 구성 파라미터

기능 구성에서 Modbus TCP 소스를 정의하는 경우 capabilityConfiguration JSON 문서에 다음 정보를 지정해야 합니다.

소스

각각 다음 정보를 포함하는 Modbus-TCP 소스 정의 구조 목록입니다.

이름

고유하고 친숙한 원본 이름입니다.

measurementDataStream접두사

(선택 사항) 원본의 모든 데이터 스트림 앞에 추가할 문자열입니다. SiteWise Edge 게이트웨이는 이 소스의 모든 데이터 스트림에 이 접두사를 추가합니다. 데이터 스트림 접두사를 사용하여 서로 다른 소스에서 이름이 같은 데이터 스트림을 구분합니다. 각 데이터 스트림은 계정 내에서 고유한 이름을 가져야 합니다.

대상

다음 정보가 포함된 목적지 구조입니다.

type

목적지의 유형입니다.

스트림 이름

AWS IoT Greengrass 스트림의 이름.

streamBufferSize

스트림 버퍼의 크기입니다.

endpoint

다음 정보가 포함된 엔드포인트 구조입니다.

IP주소

Modbus TCP 소스의 IP 주소입니다.

port

(선택 사항) Modbus TCP 소스의 포트입니다.

유닛 ID

(선택 사항) unitId입니다. 기본값은 1입니다.

minimumInterRequest소요 시간

각 요청 사이의 최소 기간(밀리초)입니다.

프로퍼티 그룹

프로토콜에서 요청한 태그 정의를 정의하는 속성 그룹 목록입니다.

이름

속성 그룹의 이름입니다. 고유 식별자여야 합니다.

tagPathDefinitions

소스 내의 측정 위치입니다. 예를 들어 바이트 및 단어 순서, 주소, 변환 유형 등이 있습니다. 각 MeasurementPathDefinition의 구조는 커넥터에 의해 정의됩니다.

스캔 모드

소스의 스캔 모드 동작과 구성 가능한 파라미터를 정의합니다.

이더넷/IP(EIP) 소스를 구성합니다.

AWS IoT SiteWise 콘솔 또는 SiteWise 에지 게이트웨이 기능을 사용하여 이더넷 IP 소스를 정의하고 SiteWise 에지 게이트웨이에 추가할 수 있습니다. 이 소스는 로컬 이더넷 IP 서버를 나타냅니다.

Note

- SiteWise AWS IoT Greengrass V2 현재 실행 중인 에지 게이트웨이는 이더넷 IP 소스를 지원하지 않습니다.
- 이더넷 IP 소스를 사용하려면 AWS IoT SiteWise 커넥터를 설치해야 합니다.

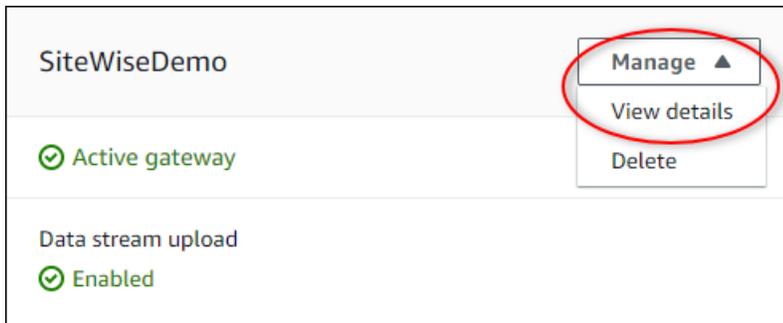
주제

- [이더넷/IP 소스 구성 \(콘솔\)](#)
- [이더넷/IP 소스 구성 \(CLI\)](#)

이더넷/IP 소스 구성 (콘솔)

이더넷/IP 소스를 구성하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 게이트웨이를 선택합니다.
3. 소스를 만들려는 SiteWise 에지 게이트웨이에서 관리를 선택한 다음 세부 정보 보기를 선택합니다.



4. 오른쪽 상단 모서리에서 New source(새 소스)를 선택합니다.
5. 프로토콜 옵션에서 이더넷/IP(EIP)를 선택합니다.
6. EtherNet/IP 소스 구성의 경우 소스 이름을 입력합니다.
7. IP 주소에 데이터 소스 서버 IP 주소를 입력합니다.
8. (선택 사항) 소스 서버의 포트를 입력합니다.
9. 최소 요청 간 기간에는 서버로 전송되는 후속 요청 사이의 시간 간격을 입력합니다. SiteWise 에지 게이트웨이는 디바이스와 보유한 레지스터 수를 기반으로 최소 허용 간격을 자동으로 계산합니다.
10. 속성 그룹의 경우 이름을 입력합니다.
11. 속성의 경우:
 - a. 태그에 레지스터 세트의 속성 별칭을 입력합니다. 예를 들어 **boiler.inlet.temperature.value**입니다.
 - b. 대상 데이터 유형에서 AWS IoT SiteWise 데이터를 변환할 데이터 유형을 선택합니다. 기본값은 String입니다.

12. 스캔 속도의 경우 SiteWise Edge 게이트웨이가 레지스터를 읽을 속도를 업데이트하십시오. AWS IoT SiteWise Edge 게이트웨이에 허용되는 최소 스캔 속도를 자동으로 계산합니다. SiteWise
13. (선택 사항) 목적지에서 소스 데이터를 보낼 위치를 선택합니다. 기본적으로 소스는 데이터를 로 보냅니다 AWS IoT SiteWise. 대신 AWS IoT Greengrass 스트림을 사용하여 데이터를 로컬 대상 또는 클라우드로 내보낼 수 있습니다. AWS

Note

엣지에서 이 소스의 데이터를 처리하려면 소스 데이터의 대상으로 선택해야 AWS IoT SiteWise 합니다. AWS IoT SiteWise 엣지에서의 데이터 처리에 대한 자세한 정보는 [the section called “엣지 데이터 처리 활성화”](#)을 참조하십시오.

데이터를 다른 목적지로 보내려면

- a. 목적지 옵션에서 기타 목적지를 선택합니다.
- b. Greengrass 스트림 이름에 스트림의 정확한 이름을 입력합니다. AWS IoT Greengrass

Note

이미 만든 스트림을 사용하거나 새 AWS IoT Greengrass 스트림을 만들어 데이터를 내보낼 수 있습니다. 기존 스트림을 사용하려면 스트림의 정확한 이름을 입력해야 합니다. 그렇지 않으면 새 스트림이 만들어집니다. AWS IoT Greengrass 스트림 작업에 대한 자세한 내용은 AWS IoT Greengrass 개발자 가이드의 [데이터 스트림 관리](#)를 참조하십시오.

14. Add source(소스 추가)를 선택합니다.

AWS IoT SiteWise SiteWise 에지 게이트웨이 구성을 AWS IoT Greengrass 코어에 배포합니다. 배포를 수동으로 시작할 필요는 없습니다.

이더넷/IP 소스 구성 (CLI)

에지 게이트웨이 기능에서 EIP 데이터 소스를 정의할 수 있습니다 SiteWise . 단일 기능 구성에서 모든 EIP 소스를 정의해야 합니다.

Note

이더넷 IP 소스를 사용하려면 AWS IoT SiteWise 커넥터를 설치해야 합니다.

이 기능에는 다음과 같은 버전이 있습니다.

버전	네임스페이스
1	iotsitewise:eipcollector:1

EIP 기능 구성 파라미터

기능 구성에서 EIP 소스를 정의하는 경우 capabilityConfiguration JSON 문서에 다음 정보를 지정해야 합니다.

소스

각각 다음 정보를 포함하는 EIP 소스 정의 구조의 목록입니다.

이름

고유하고 친숙한 원본 이름입니다. 이 이름은 최대 256자일 수 있습니다.

destinationPathPrefix

(선택 사항) 원본의 모든 데이터 스트림 앞에 추가할 문자열입니다. SiteWise Edge 게이트웨이는 이 소스의 모든 데이터 스트림에 이 접두사를 추가합니다. 데이터 스트림 접두사를 사용하여 서로 다른 소스에서 이름이 같은 데이터 스트림을 구분합니다. 각 데이터 스트림은 계정 내에서 고유한 이름을 가져야 합니다.

대상

다음 정보가 포함된 목적지 구조입니다.

type

목적지의 유형입니다.

스트림 이름

AWS IoT Greengrass 스트림의 이름.

streamBufferSize

스트림 버퍼의 크기입니다.

endpoint

다음 정보가 포함된 엔드포인트 구조입니다.

IP주소

EIP 소스의 IP 주소입니다.

port

(선택 사항) EIP 소스의 포트입니다. 허용되는 값은 1과 65535 사이의 숫자입니다.

minimumInterRequest지속 시간

(선택 사항) 각 요청 사이의 최소 기간(밀리초)입니다.

프로퍼티 그룹

프로토콜에서 요청한 태그 정의를 정의하는 속성 그룹 목록입니다. 각 소스에는 하나의 속성 그룹이 있을 수 있습니다.

이름

속성 그룹의 이름입니다. 최대 길이 256자의 고유 식별자여야 합니다.

tagPathDefinitions

이더넷/IP 디바이스에서 수집할 데이터와 출력을 위해 데이터를 변환하는 방법을 지정하는 구조 목록입니다.

type

tagPathDefinition의 유형입니다. 예를 들어 EIPTagPath입니다.

경로

tagPathDefinition의 경로입니다. 경로의 각 태그는 최대 40자일 수 있으며 문자나 밑줄로 시작할 수 있습니다. 태그는 연속된 밑줄이나 뒤에 밑줄을 포함할 수 없습니다. 경로 앞에는 의 모든 값이 접두사로 붙습니다. destinationPathPrefix

dstDataType

태그 데이터를 출력할 데이터 유형입니다. 허용되는 값은 integer, double, string, boolean입니다.

스캔 모드

소스의 스캔 모드 동작과 구성 가능한 파라미터를 정의합니다.

type

스캔 모드 동작의 유형입니다. 사용 가능한 값은 POLL입니다.

속도

이더넷/IP 소스에서 커넥터가 태그를 읽어야 하는 속도(밀리초)입니다.

원본 인증 구성

OPC-UA 서버에 연결하기 위해 인증 보안 인증 정보가 필요한 경우, AWS Secrets Manager의 각 소스에 대한 암호로 사용할 사용자 이름과 암호를 정의할 수 있습니다. 그런 다음 Greengrass 그룹 및 IoT SiteWise 커넥터에 암호를 추가하여 SiteWise Edge 게이트웨이에 암호를 제공합니다. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 AWS IoT Greengrass [코어에 암호 배포를](#) 참조하십시오.

SiteWise 에지 게이트웨이에 암호가 제공되면 소스를 구성할 때 암호를 선택할 수 있습니다. 그러면 SiteWise 에지 게이트웨이는 원본에 연결할 때 암호의 인증 자격 증명을 사용합니다. 자세한 정보는 [데이터 소스 구성](#)을 참조하세요.

주제

- [원본 인증 암호 만들기](#)
- [Greengrass 그룹에 시크릿 추가하기](#)
- [IoT SiteWise 커넥터에 시크릿 추가](#)

원본 인증 암호 만들기

이 절차에서는 Secrets Manager에서 원본에 대한 인증 암호를 만듭니다. 암호에서 원본에 대한 인증 세부 정보를 포함하는 **username** 및 **password** 키-값 쌍을 정의합니다.

원본 인증 암호를 만들려면

1. [Secrets Manager 콘솔](#)로 이동합니다.
2. 새 암호 저장을 선택합니다.
3. 암호 유형 선택에서 다른 암호 유형을 선택합니다.

4. OPC-UA 서버의 인증 값에 대한 **username** 및 **password** 키값 쌍을 입력하고 다음을 선택합니다.

The screenshot shows the 'Select secret type' section with four radio button options: 'Credentials for RDS database', 'Credentials for Redshift cluster', 'Credentials for DocumentDB database', and 'Other type of secrets (e.g. API key)'. The 'Other type of secrets' option is selected and highlighted with a red box. Below this is the 'Specify the key/value pairs to be stored in this secret' section, which has a 'Secret key/value' tab selected. Two rows of key-value pairs are added: 'username' and 'password', each with a 'Remove' button. Below the rows is a '+ Add row' link. At the bottom of the form is the 'Select the encryption key' section, which has a dropdown menu set to 'DefaultEncryptionKey' and a refresh button. At the bottom right of the form are 'Cancel' and 'Next' buttons, with 'Next' highlighted in orange.

5. greengrass-로 시작하는 암호 이름(예: **greengrass-factory1-auth**)을 입력합니다.

⚠ Important

암호에 액세스하려면 기본 AWS IoT Greengrass 서비스 역할에 greengrass- 접두사를 사용해야 합니다. 이 접두사 없이 비밀의 이름을 지정하려면 비밀에 액세스할 수 있는 AWS IoT Greengrass 사용자 지정 권한을 부여해야 합니다. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 [암호 값 가져오기 AWS IoT Greengrass 허용을 참조](#)하십시오.

Store a new secret

Secret name and description Info

Secret name

Give the secret a name that enables you to find and manage it easily.

greengrass-factory1-auth

Secret name must contain only alphanumeric characters and the characters /_+=@-

6. 설명을 입력하고 다음을 선택합니다.
7. (선택 사항) 자동 교체 구성 페이지에서 암호에 대한 자동 교체를 구성합니다. 자동 교체를 구성하는 경우 암호가 교체될 때마다 Greengrass 그룹을 다시 배포해야 합니다.
8. 자동 교체 구성 페이지에서 다음을 선택합니다.
9. 새 암호를 검토하고 저장을 선택합니다.

Greengrass 그룹에 시크릿 추가하기

이 절차에서는 소스 인증 암호를 AWS IoT Greengrass 그룹에 추가하여 IoT SiteWise 커넥터에서 사용할 수 있도록 합니다.

Greengrass 그룹에 암호를 추가하려면

1. [AWS IoT Greengrass 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창의 Greengrass에서 그룹을 선택한 다음 그룹을 선택합니다.

The screenshot shows the AWS IoT Greengrass console interface. On the left, the navigation menu includes 'Monitor', 'Onboard', 'Manage', and 'Greengrass'. Under 'Greengrass', 'Groups' is highlighted with a red circle. The main content area displays 'Greengrass groups (1)' with a search bar and a table of groups. The table has columns for 'Name', 'ID', and 'Created'. One group is listed with the name 'SiteWiseDemo', ID 'a1b2c3d4-5678-90ab-cdef-11111EXAMPLE', and 'Created' '9 months ago'. The 'Name' 'SiteWiseDemo' is circled in red.

<input type="checkbox"/>	Name	ID	Created
<input type="checkbox"/>	SiteWiseDemo	a1b2c3d4-5678-90ab-cdef-11111EXAMPLE	9 months ago

3. 탐색 페이지에서 리소스를 선택합니다.
4. 리소스 페이지에서 암호 탭을 선택한 다음 암호 리소스 추가를 선택합니다.

GREENGRASS GROUP

SiteWiseDemo

Not deployed Actions ▾

Deployments

Subscriptions

Cores

Devices

Lambdas

Resources

Connectors

Tags

Settings

Resources

Local Machine Learning **Secret**



Allow Lambda functions and connectors to securely access secret resources

Secret resources reference passwords, API keys, OAuth tokens, or other credentials stored in AWS Secrets Manager. At runtime, Lambda functions and connectors can use secret resources to access third-party services. [Learn more](#)

Add a secret resource

5. 선택을 선택하고 목록에서 암호를 선택합니다.
6. 다음을 선택합니다.
7. 암호 리소스 이름에 암호 리소스의 이름을 입력하고 저장을 선택합니다.

ADD A RESOURCE TO YOUR GREENGRASS GROUP

Name your secret resource

STEP 3/3

Your secret resource will be added to the group. Give it a unique name so you can easily identify it. [Learn more](#)

Secret resource name

The name can contain alphanumeric characters, colons, underscores, and dashes.

Secret name
greengrass-factory1-auth

Labels
AWSCURRENT

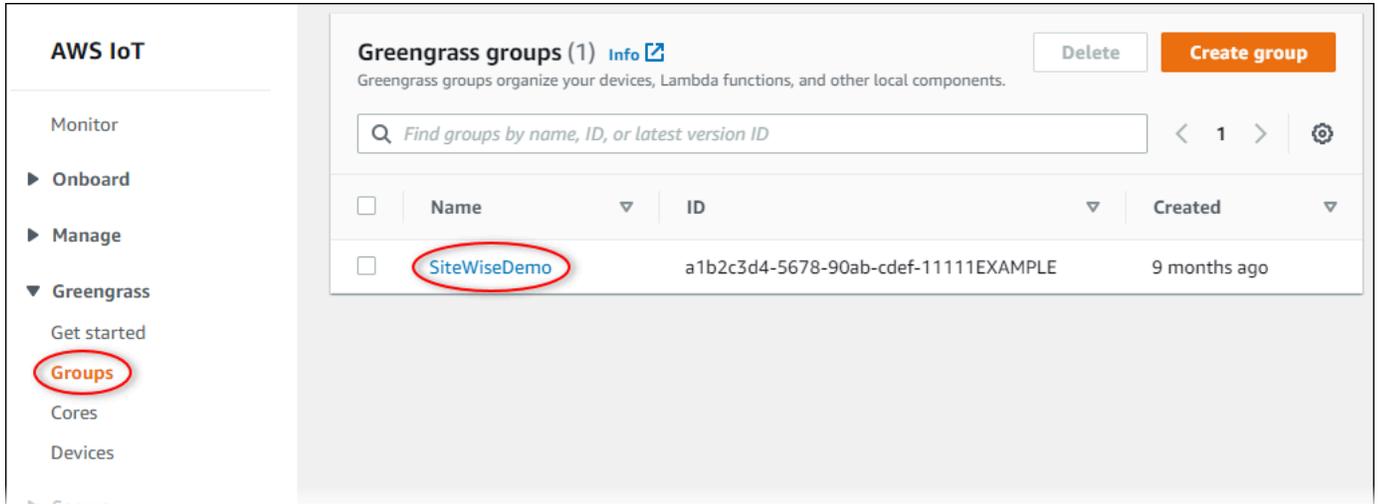
[Cancel](#) [Back](#) **Save**

IoT SiteWise 커넥터에 시크릿 추가

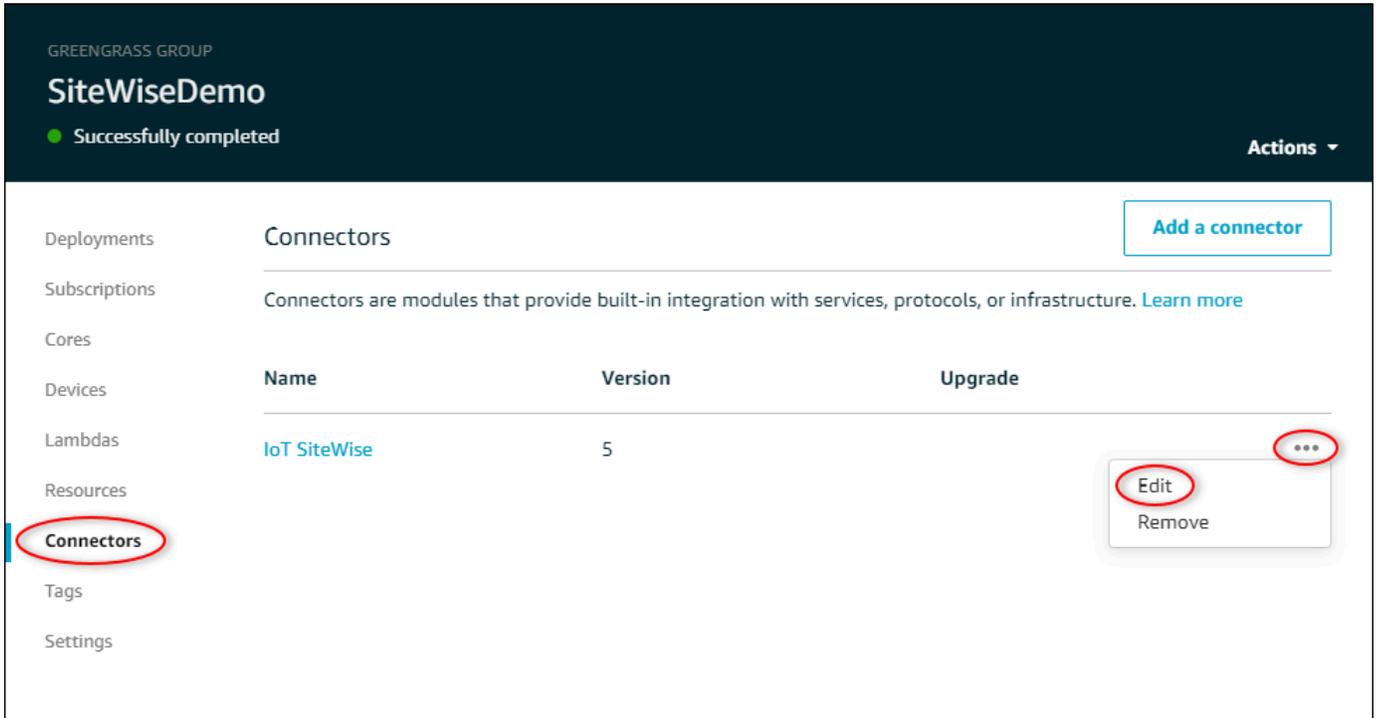
이 절차에서는 소스 인증 암호를 IoT SiteWise 커넥터에 추가하여 SiteWise Edge 게이트웨이에서 사용할 수 있도록 AWS IoT SiteWise 합니다.

IoT SiteWise 커넥터에 시크릿을 추가하려면

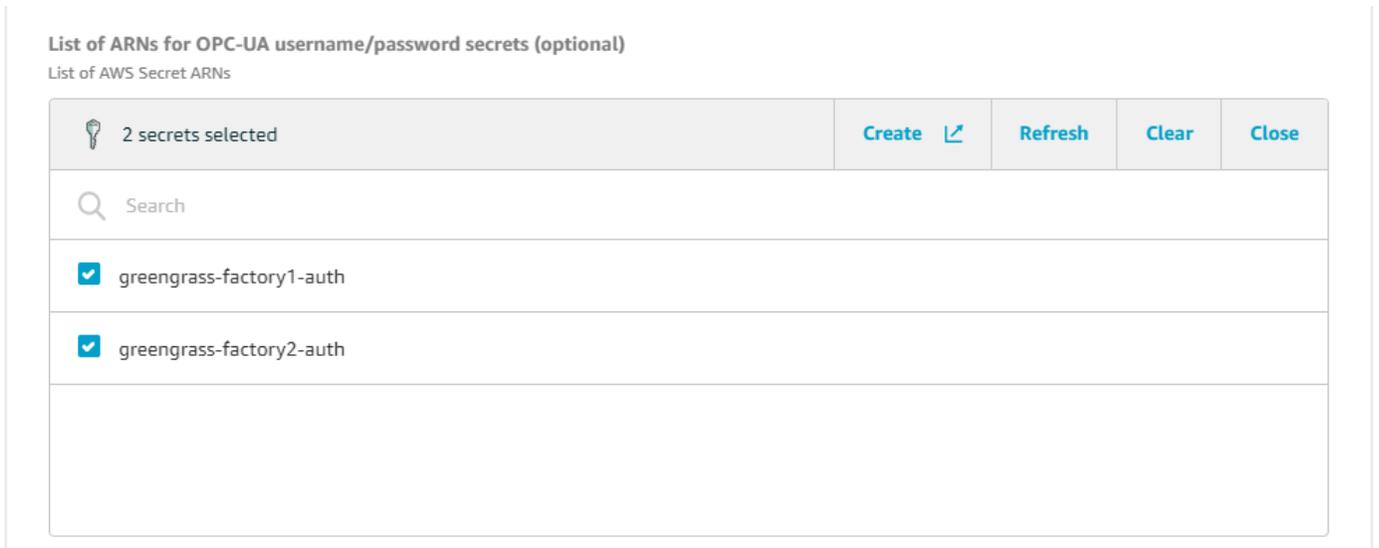
1. [AWS IoT Greengrass 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창의 Greengrass에서 그룹을 선택한 다음 그룹을 선택합니다.



3. 탐색 페이지에서 커넥터를 선택합니다.
4. IoT SiteWise 커넥터의 줄임표 아이콘을 선택하여 옵션 메뉴를 연 다음 편집을 선택합니다.



5. OPC-UA 사용자 이름/암호 암호에 대한 ARN 목록에서 선택을 선택한 다음 이 Edge 게이트웨이에 추가할 각 암호를 선택합니다. SiteWise 암호를 만들어야 하는 경우, [원본 인증 암호 만들기](#)을 참조하십시오.



암호가 표시되지 않으면 새로 고침을 선택합니다. 그래도 암호가 표시되지 않으면, [Greengrass 그룹에 암호를 추가했는지 확인](#)합니다.

6. 저장을 선택합니다.
7. 오른쪽 상단 모서리의 작업 메뉴에서 배포를 선택하십시오.
8. Automatic detection(자동 검색)을 선택하여 배포를 시작합니다.

배포가 실패하면 배포를 다시 선택하십시오. 배포가 계속 실패하면 [AWS IoT Greengrass 배포 문제 해결](#)을 참조하십시오.

그룹이 배포된 후, 새 암호를 사용하도록 원본을 구성할 수 있습니다. 자세한 정보는 [데이터 소스 구성](#)을 참조하세요.

커넥터 업그레이드

Important

IoT SiteWise 커넥터 버전 6에는 [AWS IoT Greengrass 코어 소프트웨어 v1.10.0 및 스트림 관리자라는 새로운 요구 사항이 도입되었습니다](#). 커넥터를 업그레이드하기 전에 SiteWise 에지 게이트웨이가 이러한 요구 사항을 충족하는지 확인하세요. 그렇지 않으면 에지 게이트웨이를 배포할 수 없습니다. SiteWise

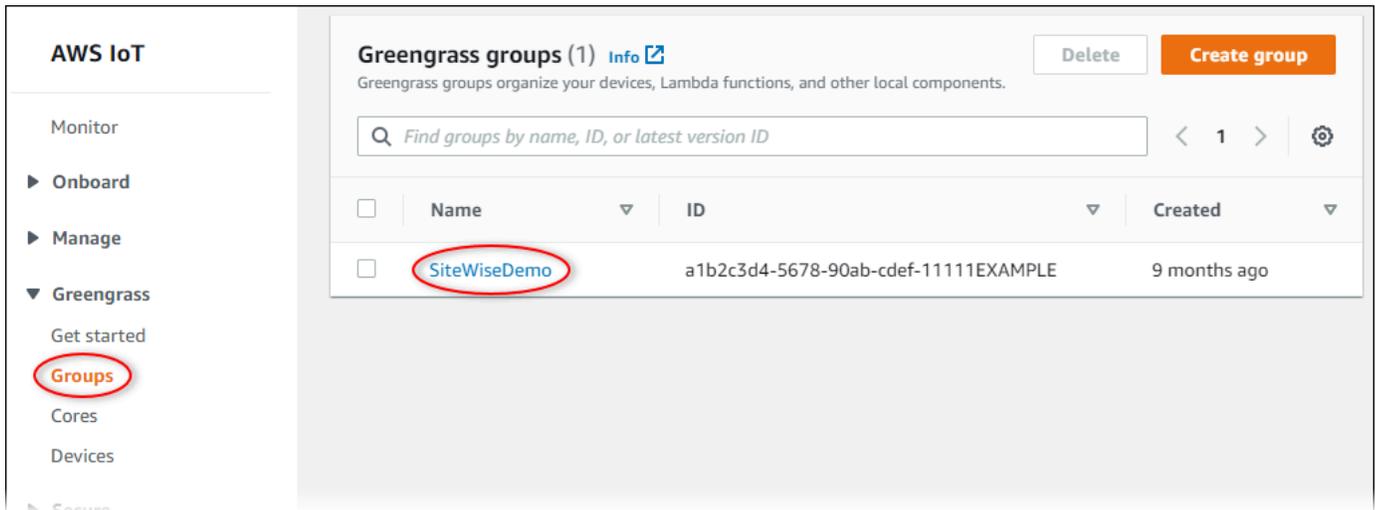
새 IoT 커넥터 버전이 출시된 후 SiteWise Edge 게이트웨이의 SiteWise 커넥터를 쉽게 업그레이드할 수 있습니다.

Note

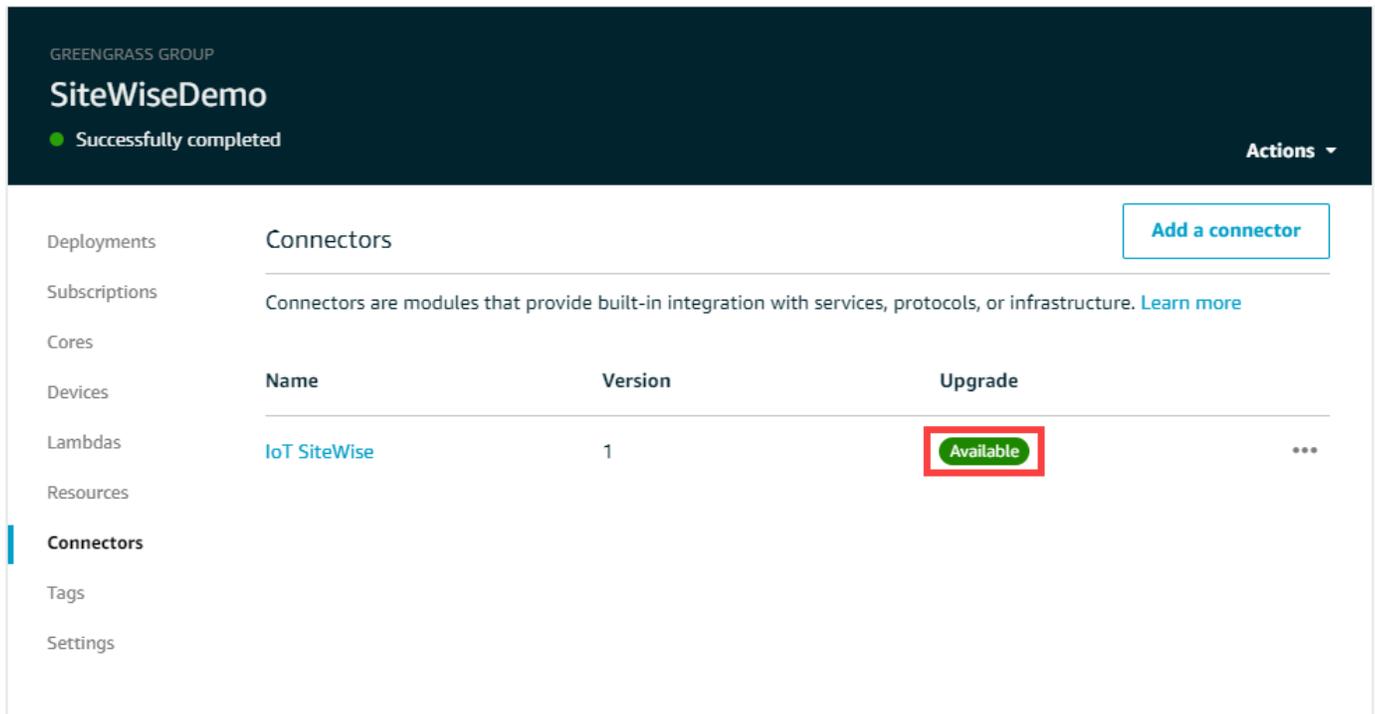
이 절차에서는 Greengrass 그룹을 재배포하고 Edge 게이트웨이를 다시 시작합니다 SiteWise . SiteWise 에지 게이트웨이는 다시 시작하는 동안에는 데이터를 수집하지 않습니다. 에지 게이트웨이를 다시 시작하는 데 걸리는 시간은 SiteWise 에지 게이트웨이 소스의 태그 수에 따라 달라집니다. SiteWise 재시작 시간은 몇 초 (태그가 거의 없는 SiteWise 에지 게이트웨이의 경우) 에서 몇 분 (태그가 많은 SiteWise 에지 게이트웨이의 경우) 까지 걸릴 수 있습니다.

IoT SiteWise 커넥터를 업그레이드하려면

1. [AWS IoT Greengrass 콘솔](#)로 이동합니다.
2. 탐색 창의 Greengrass에서 그룹을 선택한 다음 SiteWise Edge 게이트웨이를 설정할 때 만든 그룹을 선택합니다.



3. 탐색 창에서 커넥터를 선택합니다.
4. 커넥터 페이지에서 IoT SiteWise 커넥터 옆의 사용 가능을 선택합니다.



사용 가능 요소가 표시되지 않으면 커넥터가 이미 최신 버전입니다.

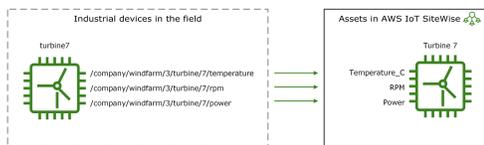
5. 커넥터 업그레이드 페이지에서 커넥터의 파라미터를 입력한 다음 업그레이드를 선택합니다.
6. 오른쪽 상단 모서리의 작업 메뉴에서 배포를 선택하십시오.
7. Automatic detection(자동 검색)을 선택하여 배포를 시작합니다.

배포가 실패하면 배포를 다시 선택하십시오. 배포가 계속 실패하면 [AWS IoT Greengrass 배포 문제 해결](#)을 참조하십시오.

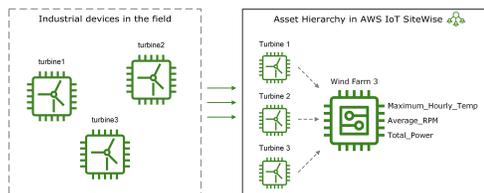
산업 자산 모델링

AWS IoT SiteWise 자산을 사용하여 산업 운영을 가상으로 표현할 수 있습니다. 자산은 하나 이상의 데이터 스트림을 업로드하는 장치, 장비 또는 프로세스를 나타냅니다. AWS 클라우드에 들어 자산 디바이스는 공기 온도, 프로펠러 회전 속도 및 전력 출력 시계열 측정을 AWS IoT SiteWise의 자산 속성으로 전송하는 풍력 터빈일 수 있습니다.

각 데이터 스트림은 고유한 속성 별칭에 해당합니다. 예를 들어 별칭 `/company/windfarm/3/turbine/7/temperature`는 풍력 발전소 #3의 터빈 #7에서 나오는 온도 데이터 스트림을 고유하게 식별합니다. 온도 데이터를 섭씨에서 화씨로 변환하는 것과 같이 수학적 표현식을 사용하여 수신되는 측정 데이터를 변환하도록 AWS IoT SiteWise 자산을 구성할 수 있습니다.



자산은 전체 풍력 단지와 같은 디바이스의 논리적 그룹화를 나타낼 수도 있습니다. 자산을 다른 자산에 연결하여 복잡한 산업 조직을 나타내는 자산 계층 구조를 생성할 수 있습니다. 자산은 관련 하위 자산 내의 데이터에 액세스할 수 있습니다. 이렇게 하면 AWS IoT SiteWise 표현식을 사용하여 풍력 발전 단지의 순 전력 출력과 같은 집계 지표를 계산할 수 있습니다.



모든 자산은 자산 모델에서 생성해야 합니다. 자산 모델은 자산의 형식을 표준화하는 선언형 구조입니다. 자산 모델은 기기 그룹을 나타내는 자산의 데이터를 처리할 수 있도록 동일한 유형의 여러 자산에 대해 일관된 정보를 적용합니다. 앞의 다이어그램에서는 모든 터빈이 공통된 속성 집합을 공유하므로 세 터빈 모두에 동일한 자산 모델을 사용합니다.

구성 요소 모델을 만들 수도 있습니다. 구성 요소 모델은 자산 모델이나 다른 구성 요소 모델에 포함할 수 있는 특수한 유형의 자산 모델입니다. 컴포넌트 모델을 사용하여 센서, 모터 등과 같은 재사용 가능한 공통 하위 어셈블리를 정의하여 여러 자산 모델에서 공유할 수 있습니다.

자산 모델을 정의한 후 산업 자산을 생성할 수 있습니다. 자산을 생성하려면 ACTIVE 자산 모델을 선택하여 해당 모델의 자산을 생성합니다. 그런 다음, 데이터 스트림 별칭, 속성 같은 자산별 정보를 채울 수

있습니다. 앞의 다이어그램에서는 하나의 자산 모델에서 3개의 터빈 자산을 생성한 다음 각 터빈에 `/company/windfarm/3/turbine/7/temperature` 같은 데이터 스트림 별칭을 연결합니다.

기존 자산, 자산 모델 및 구성 요소 모델을 업데이트하고 삭제할 수도 있습니다. 자산 모델을 업데이트 하면 해당 자산 모델을 기반으로 하는 모든 자산에 기본 모델의 변경 사항이 반영됩니다. 구성 요소 모델을 업데이트하면 해당 구성 요소 모델을 참조하는 모든 자산 모델을 기반으로 하는 모든 자산에 적용됩니다.

자산 모델은 매우 복잡할 수 있습니다. 예를 들어 하위 구성 요소가 많은 복잡한 장비를 모델링할 때 그렇습니다. 이러한 자산 모델을 체계적으로 구성하고 유지 관리할 수 있도록 사용자 지정 복합 모델을 사용하여 관련 속성을 그룹화하거나 공유 구성 요소를 재사용할 수 있습니다. 자세한 정보는 [사용자 지정 복합 모델 \(구성 요소\)](#)을 참조하세요.

주제

- [자산 및 모델 상태](#)
- [사용자 지정 복합 모델 \(구성 요소\)](#)
- [개체 ID로 작업하기](#)
- [자산 모델 및 구성 요소 모델 생성](#)
- [자산 만들기](#)
- [자산 검색](#)
- [산업 데이터 스트림을 자산 속성에 매핑](#)
- [속성 값 업데이트](#)
- [자산 연결 및 연결 해제](#)
- [자산 및 모델 업데이트](#)
- [자산 및 모델 삭제](#)
- [자산 및 모델을 사용한 대량 작업](#)

자산 및 모델 상태

자산, 자산 모델 또는 구성 요소 모델을 생성, 업데이트 또는 삭제할 때 변경 사항이 적용되는 데 시간이 걸립니다. AWS IoT SiteWise 이러한 작업을 비동기적으로 해결하고 각 리소스의 상태를 업데이트합니다. 각 자산, 자산 모델 및 구성 요소 모델에는 리소스 상태와 오류 메시지 (해당하는 경우)가 포함된 상태 필드가 있습니다. 상태는 다음 값 중 하나일 수 있습니다.

- ACTIVE— 리소스가 활성 상태입니다. 이 상태에서만 자산, 자산 모델 및 구성 요소 모델을 쿼리하고 상호 작용할 수 있습니다.
- CREATING— 리소스가 생성되고 있습니다.
- UPDATING— 리소스가 업데이트되고 있습니다.
- DELETING— 리소스가 삭제되고 있습니다.
- PROPAGATING— (자산 모델 및 구성 요소 모델만 해당) 변경 내용이 모든 종속 리소스 (자산 모델에서 자산으로 또는 구성 요소 모델에서 자산 모델로) 에 전파되고 있습니다.
- FAILED— 생성 또는 업데이트 작업 중에 리소스를 검증하지 못했습니다. 표현식의 순환 참조 때문일 수 있습니다. FAILED상태에 있는 리소스는 삭제할 수 있습니다.

일부 생성, 업데이트 및 삭제 작업은 자산, 자산 모델 또는 구성 요소 모델을 작업이 해결되는 시점이 아닌 ACTIVE 다른 상태로 AWS IoT SiteWise 배치합니다. 이러한 작업 중 하나를 수행한 후 리소스를 쿼리하거나 리소스와 상호 작용하려면 상태가 로 ACTIVE 변경될 때까지 기다려야 합니다. 그렇지 않으면 요청이 실패합니다.

주제

- [자산의 상태 확인](#)
- [자산 모델 또는 구성 요소 모델의 상태 확인](#)

자산의 상태 확인

AWS IoT SiteWise 콘솔 또는 API를 사용하여 자산 상태를 확인할 수 있습니다.

주제

- [자산의 상태 확인\(콘솔\)](#)
- [자산 상태 확인 \(AWS CLI\)](#)

자산의 상태 확인(콘솔)

AWS IoT SiteWise 콘솔에서 자산의 상태를 확인하려면 다음 절차를 따르십시오.

자산의 상태를 확인하려면(콘솔)

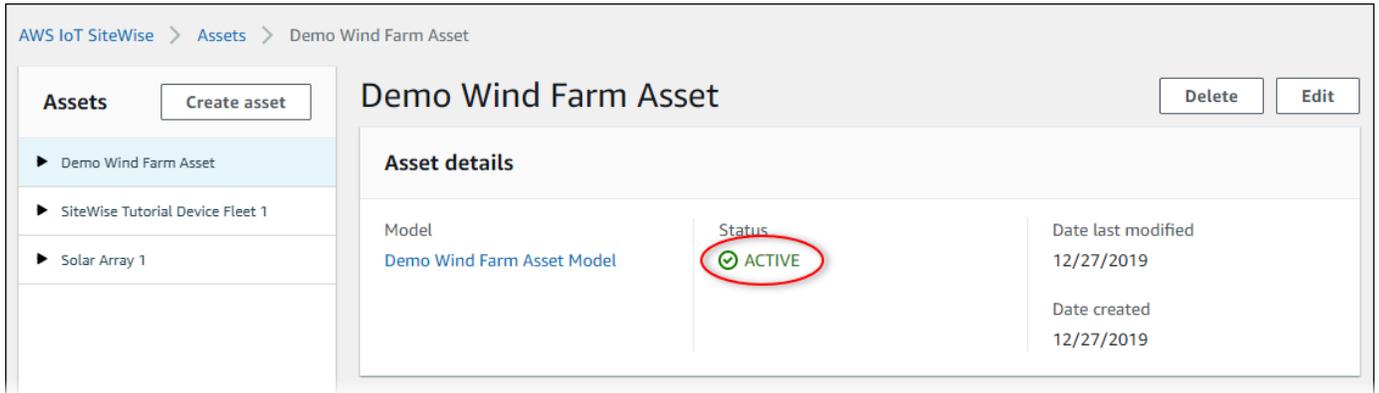
1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.

3. 확인할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 자산 세부 정보 패널에서 상태를 찾습니다.



자산 상태 확인 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 상태를 확인할 수 있습니다.

자산 상태를 확인하려면 `assetId` 매개변수와 함께 [DescribeAsset](#) 작업을 사용하십시오.

자산 상태를 확인하려면 (AWS CLI)

- 다음 명령을 실행하여 자산을 설명합니다. `asset-id# ### ID` 또는 외부 ID로 바꾸십시오. 외부 ID는 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

```
aws iotsitewise describe-asset --asset-id asset-id
```

이 작업은 자산의 세부 정보가 포함된 응답을 반환합니다. 응답에는 다음과 같은 구조의 `assetStatus` 개체가 포함됩니다.

```
{
  ...
  "assetStatus": {
    "state": "String",
    "error": {
```

```
    "code": "String",  
    "message": "String"  
  }  
}  
}
```

자산의 상태는 JSON 객체의 `assetStatus.state`에 있습니다.

자산 모델 또는 구성 요소 모델의 상태 확인

AWS IoT SiteWise 콘솔 또는 API를 사용하여 자산 모델 또는 구성 요소 모델의 상태를 확인할 수 있습니다.

주제

- [자산 모델 또는 구성 요소 모델의 상태 확인 \(콘솔\)](#)
- [자산 모델 또는 구성 요소 모델의 상태 확인 \(AWS CLI\)](#)

자산 모델 또는 구성 요소 모델의 상태 확인 (콘솔)

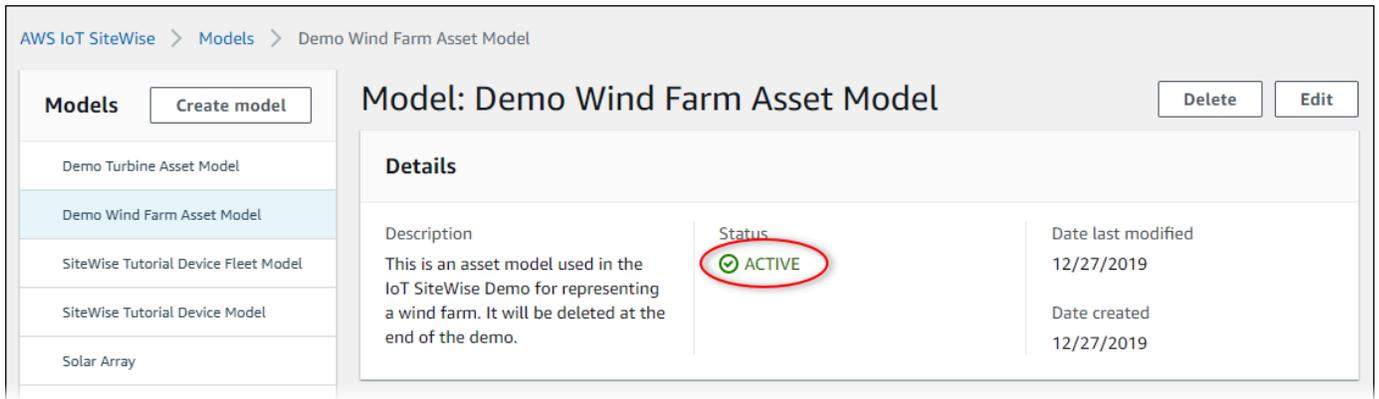
다음 절차를 사용하여 AWS IoT SiteWise 콘솔에서 자산 모델 또는 구성 요소 모델의 상태를 확인합니다.

Tip

에셋 모델과 구성 요소 모델은 모두 탐색 창의 모델 아래에 나열됩니다. 선택한 자산 모델 또는 구성 요소 모델의 세부 정보 패널에는 해당 유형이 표시됩니다.

에셋 모델 또는 컴포넌트 모델의 상태 확인하기 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 확인할 모델을 선택합니다.
4. 세부 정보 패널에서 상태를 찾습니다.



자산 모델 또는 구성 요소 모델의 상태 확인 (AWS CLI)

를 AWS CLI 사용하여 자산 모델 또는 구성 요소 모델의 상태를 확인할 수 있습니다.

자산 모델 또는 구성 요소 모델의 상태를 확인하려면 `assetModelId` 매개변수와 함께 [DescribeAsset 모델](#) 작업을 사용합니다.

Tip

AWS CLI에서는 구성 요소 모델을 자산 모델 유형으로 정의합니다. 따라서 두 모델 유형에 모두 동일한 [DescribeAsset 모델](#) 작업을 사용합니다. 응답의 `assetModelType` 필드는 해당 필드가 a인지 ASSET_MODEL a인지를 나타냅니다 COMPONENT_MODEL.

자산 모델 또는 구성 요소 모델의 상태를 확인하려면 (AWS CLI)

- 다음 명령을 실행하여 모델을 설명합니다. `asset-model-id#` 자산 모델 또는 구성 요소 모델의 ID 또는 외부 ID로 대체합니다. 외부 ID는 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

이 작업은 모델의 세부 정보가 포함된 응답을 반환합니다. 응답에는 다음과 같은 구조의 `assetModelStatus` 객체가 포함됩니다.

```
{
  ...
  "assetModelStatus": {
```

```

    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}

```

모델 상태는 JSON 객체에 있습니다. `assetModelStatus.state`

사용자 지정 복합 모델 (구성 요소)

부품이 많고 복잡한 기계 부품과 같이 특히 복잡한 산업 자산을 모델링할 때는 자산 모델을 체계적으로 정리하고 유지 관리할 수 있게 유지하는 것이 어려울 수 있습니다.

이러한 경우 사용자 지정 복합 모델 또는 구성 요소 (콘솔을 사용하는 경우) 를 기존 자산 모델 및 구성 요소 모델에 추가할 수 있습니다. 이를 통해 관련 속성을 그룹화하고 하위 구성 요소 정의를 재사용하여 체계적으로 정리할 수 있습니다.

사용자 지정 복합 모델에는 두 가지 유형이 있습니다.

- 인라인 사용자 지정 복합 모델은 사용자 지정 복합 모델이 속하는 자산 모델 또는 구성 요소 모델에 적용되는 그룹화된 속성 집합을 정의합니다. 이를 사용하여 관련 속성을 그룹화할 수 있습니다. 이름, 설명 및 자산 모델 속성 세트에 구성됩니다. 재사용할 수 없습니다.
- 구성 요소 모델 기반 사용자 지정 복합 모델은 자산 모델 또는 구성 요소 모델에 포함하려는 구성 요소 모델을 참조합니다. 이를 사용하여 모델에 표준 서버어셈블리를 포함할 수 있습니다. 이는 참조하는 컴포넌트 모델의 이름, 설명 및 ID로 구성됩니다. 구성 요소 모델에는 고유한 속성이 없습니다. 참조된 구성 요소 모델은 생성된 모든 자산에 관련 속성을 제공합니다.

다음 섹션에서는 설계에서 사용자 지정 복합 모델을 사용하는 방법을 설명합니다.

주제

- [인라인 사용자 지정 복합 모델](#)
- [Component-model-based 사용자 지정 복합 모델](#)
- [경로를 사용하여 사용자 지정 복합 모델 속성 참조](#)

인라인 사용자 지정 복합 모델

인라인 사용자 지정 복합 모델은 관련 속성을 그룹화하여 자산 모델을 구성하는 방법을 제공합니다.

예를 들어 로봇 자산을 모델링하고 싶다고 가정해 보겠습니다. 로봇에는 서보 모터, 전원 공급 장치 및 배터리가 포함됩니다. 각 구성 부품에는 모델에 포함하려는 고유한 특성이 있습니다. 다음과 같은 속성을 `robot_model` 가진 자산 모델을 정의할 수 있습니다.

- `robot_model`
 - `servo_status` (정수)
 - `servo_position` (더블)
 - `powersupply_status` (정수)
 - `powersupply_temperature` (더블)
 - `battery_status` (정수)
 - `battery_charge` (더블)

하지만 일부 경우에는 서브어셈블리가 많을 수도 있고 서브어셈블리 자체에 많은 속성이 있을 수도 있습니다. 이러한 경우에는 이전 예제에서처럼 속성이 너무 많아서 모델 루트의 단일 단순 목록에서 참조하고 유지 관리하기가 번거로울 수 있습니다.

이러한 상황을 해결하기 위해 인라인 사용자 지정 복합 모델을 사용하여 속성을 그룹화할 수 있습니다. 인라인 사용자 지정 복합 모델은 자체 속성을 정의하는 사용자 지정 복합 모델입니다. 예를 들어 다음과 같이 로봇을 모델링할 수 있습니다.

- `robot_model`
 - `servo`
 - `status`(정수)
 - `position`(더블)
 - `powersupply`
 - `status`(정수)
 - `temperature` (더블)
 - `battery`
 - `status`(정수)

- charge(더블)

위 예제에서 servopowersupply, 및 battery 는 robot_model 자산 모델 내에 정의된 인라인 사용자 지정 복합 모델의 이름입니다. 그런 다음 각 복합 모델은 고유한 속성을 정의합니다.

Note

이 경우 각 사용자 지정 복합 모델은 자체 속성을 정의하므로 모든 속성이 자산 모델 자체 (robot_model이 경우) 의 일부가 됩니다. 이러한 속성은 다른 자산 모델이나 구성 요소 모델과 공유되지 않습니다. 예를 들어 이라는 servo 인라인 사용자 지정 복합 모델도 포함하는 다른 자산 모델을 만든 경우 servo 내부로 robot_model 변경해도 다른 자산 모델의 servo 정의에는 영향을 주지 않습니다.

이러한 공유를 구현하려면 (예: 모든 자산 모델이 공유할 수 있는 서보에 대한 정의를 하나만 갖는 경우) 대신 해당 정의를 위한 구성 요소 모델을 만든 다음 이를 참조하는 구성 요소 모델 기반 복합 모델을 만들면 됩니다. 자세한 내용은 다음 섹션을 참조하십시오.

인라인 사용자 지정 복합 모델을 만드는 방법에 대한 자세한 내용은 [참조하십시오 사용자 지정 복합 모델 생성 \(구성 요소\)](#).

Component-model-based 사용자 지정 복합 모델

에서 부품 모델을 AWS IoT SiteWise 생성하여 재사용 가능한 표준 서버어셈블리를 정의할 수 있습니다. 컴포넌트 모델을 생성한 후에는 다른 에셋 모델 및 컴포넌트 모델에 해당 참조를 추가할 수 있습니다. 컴포넌트를 참조하려는 모델에 component-model-based 사용자 지정 복합 모델을 추가하면 됩니다. 여러 모델의 컴포넌트에 대한 참조를 추가하거나 동일한 모델 내에 여러 번 추가할 수 있습니다.

이렇게 하면 모델 간에 동일한 정의가 중복되는 것을 방지할 수 있습니다. 또한 구성 요소 모델에 대한 모든 변경 사항이 해당 구성 요소 모델을 사용하는 모든 자산 모델에 반영되므로 모델 유지 관리가 단순해집니다.

예를 들어, 산업용 설비에 모두 같은 종류의 서보 모터를 사용하는 여러 유형의 장비가 있다고 가정해 보겠습니다. 이들 중 일부는 단일 장비에 여러 서보 모터를 탑재하고 있습니다. 각 장비 유형에 대한 자산 모델을 생성하지만 servo 매번 그 정의를 복제하고 싶지는 않을 것입니다. 한 번만 모델링하여 다양한 자산 모델에 사용하고 싶을 것입니다. 나중에 정의를 servo 변경하면 모든 모델 및 자산에서 해당 정의가 업데이트됩니다.

이러한 방식으로 이전 예제에서 로봇을 모델링하려면 다음과 같이 서보 모터, 전원 공급 장치 및 배터리를 구성 요소 모델로 정의할 수 있습니다.

- `servo_component_model`
 - `status`(정수)
 - `position`(더블)

- `powersupply_component_model`
 - `status`(정수)
 - `temperature` (더블)

- `battery__component_model`
 - `status`(정수)
 - `charge`(더블)

그런 다음 이러한 구성 요소를 참조하는 자산 모델 (예 `robot_model`;) 을 정의할 수 있습니다. 여러 자산 모델이 동일한 구성 요소 모델을 참조할 수 있습니다. 로봇에 서보 모터가 여러 개 있는 경우처럼 하나의 자산 모델에서 동일한 구성 요소 모델을 여러 번 참조할 수도 있습니다.

- `robot_model`
 - `servo1`(참조:) `servo_component_model`
 - `servo2`(참조:`servo_component_model`)
 - `servo3`(참조:`servo_component_model`)
 - `powersupply` (참조:`powersupply_component_model`)
 - `battery`(참조:`battery_component_model`)

구성 요소 모델을 만드는 방법에 대한 자세한 내용은 [을 참조하십시오](#) [컴포넌트 모델 생성](#).

다른 모델에서 구성 요소 모델을 참조하는 방법에 대한 자세한 내용은 [을 참조하십시오](#) [사용자 지정 복합 모델 생성 \(구성 요소\)](#).

경로를 사용하여 사용자 지정 복합 모델 속성 참조

자산 모델, 구성 요소 모델 또는 사용자 지정 복합 모델에서 속성을 생성하는 경우 해당 값을 사용하는 다른 속성 (예: [변환](#) 및 [지표](#)) 에서 속성을 참조할 수 있습니다.

AWS IoT SiteWise 속성을 참조할 수 있는 다양한 방법을 제공합니다. 가장 간단한 방법은 대개 해당 속성 ID를 사용하는 것입니다. 그러나 참조하려는 속성이 사용자 지정 복합 모델에 있는 경우 경로로 참조하는 대신 경로를 사용하여 참조하는 것이 더 유용할 수 있습니다.

경로는 자산 모델 및 복합 모델 내의 중첩된 복합 모델 간의 위치를 기준으로 속성을 지정하는 경로 세그먼트의 순서가 지정된 시퀀스입니다.

속성 경로 확보

[속성 path 필드에서 속성의 경로를 가져올 수 있습니다. AssetModel](#)

예를 들어 속성이 position 있는 사용자 지정 복합 모델을 포함하는 자산 모델이 robot_model servo 있다고 가정해 보겠습니다. [DescribeAssetModelCompositeModel](#) on을 servo 호출하면 position 속성에 다음과 같은 path 필드가 나열됩니다.

```
"path": [
  {
    "id": "asset model ID",
    "name": "robot_model"
  },
  {
    "id": "composite model ID",
    "name": "servo"
  },
  {
    "id": "property ID",
    "name": "position"
  }
]
```

속성 경로 사용

변환이나 메트릭과 같은 다른 속성을 참조하는 속성을 정의할 때 속성 경로를 사용할 수 있습니다.

속성은 변수를 사용하여 다른 속성을 참조합니다. 변수 작업에 대한 자세한 내용은 [공식 표현식에서 변수 사용](#).

속성을 참조하는 변수를 정의할 때는 속성의 ID 또는 경로를 사용할 수 있습니다.

참조된 속성의 경로를 사용하는 변수를 정의하려면 속성 값의 `propertyPath` 필드를 지정하십시오.

예를 들어 경로를 사용하여 속성을 참조하는 지표가 있는 자산 모델을 정의하려면 `CreateAsset` 다음과 같은 페이로드를 `Model`에 전달할 수 있습니다.

```
{
  ...
  "assetModelProperties": [
    {
      ...
      "type": {
        "metric": {
          ...
          "variables": [
            {
              "name": "variable name",
              "value": {
                "propertyPath": [
                  path segments
                ]
              }
            }
          ]
        }
      },
      ...
    },
    ...
  ],
  ...
}
```

개체 ID로 작업하기

AWS IoT SiteWise 자산, 자산 모델, 속성 및 계층과 같은 다양한 유형의 영구 개체를 정의합니다. 이러한 모든 객체에는 객체를 검색, 업데이트 및 삭제하는 데 사용할 수 있는 고유한 식별자가 있습니다.

AWS IoT SiteWise 고객을 위한 다양한 ID 생성 옵션이 있습니다. AWS IoT SiteWise 객체 생성 시 기본적으로 하나를 생성합니다. 또한 사용자는 개체에 자신의 ID를 제공할 수 있습니다.

주제

- [개체 UUID로 작업하기](#)
- [외부 ID 사용](#)

개체 UUID로 작업하기

에 있는 모든 영구 AWS IoT SiteWise 개체에는 해당 개체를 식별하는 [UUID가](#) 있습니다. 예를 들어 자산 모델에는 자산 모델 ID가 있고 자산에는 자산 ID가 있는 식입니다. 이 ID는 개체를 만들 때 할당되며 개체의 수명 기간 동안 변경되지 않습니다.

새 개체를 만들면 기본적으로 고유한 ID가 AWS IoT SiteWise 생성됩니다. 생성 시 UUID 형식으로 자체 ID를 제공할 수도 있습니다.

Note

UUID는 생성된 AWS 지역 내에서, 그리고 동일한 객체 유형에 대해 전체적으로 고유해야 합니다. AWS IoT SiteWise 자동으로 생성되는 ID는 항상 고유합니다. 고유한 ID를 선택하는 경우 고유한 ID인지 확인하세요.

예를 들어 Model을 호출하여 새 자산 [CreateAsset모델](#)을 만드는 경우 요청의 선택적 `assetModelId` 필드에 고유한 UUID를 제공할 수 있습니다.

반대로 `assetModelId` 요청에서 생략하면 새 자산 모델에 대한 UUID가 AWS IoT SiteWise 생성됩니다.

외부 ID 사용

UUID가 아닌 다른 형식으로 자체 ID를 정의하려면 외부 ID를 할당할 수 있습니다. 예를 들어, 사용하지 않는 AWS 시스템에서 사용 중인 ID를 재사용하거나 사람이 더 쉽게 읽을 수 있도록 하려면 이 작업을 수행할 수 있습니다. 외부 ID는 형식이 더 유연합니다. 다른 방법으로는 UUID를 사용하는 AWS IoT SiteWise API 작업에서 객체를 참조하는 데 이를 사용할 수 있습니다.

UUID와 마찬가지로 각 외부 ID는 컨텍스트 내에서 고유해야 합니다. 예를 들어, 동일한 외부 ID를 가진 두 개의 자산 모델을 가질 수 없습니다. 또한 UUID와 마찬가지로 개체는 수명 주기 동안 외부 ID를 하나만 가질 수 있으며 이는 변경할 수 없습니다.

외부 ID와 UUID 간의 차이점

외부 ID는 다음과 같은 점에서 UUID와 다릅니다.

- 모든 개체에는 UUID가 있지만 외부 ID는 선택 사항입니다.
- AWS IoT SiteWise 외부 ID를 생성하지 않습니다. 이러한 정보는 사용자가 직접 제공합니다.
- 개체에 아직 외부 ID가 없는 경우 언제든지 외부 ID를 할당할 수 있습니다.

외부 ID의 형식

유효한 외부 ID에는 다음과 같은 속성이 있습니다.

- 길이는 2자에서 128자 사이입니다.
- 첫 글자와 마지막 문자는 영숫자 (A-Z, a-z, 0-9) 여야 합니다.
- 첫 번째와 마지막 이외의 문자는 영숫자이거나 다음 중 하나여야 합니다. _ - . :

예를 들어 외부 ID는 다음 정규 표현식을 준수해야 합니다.

```
[a-zA-Z0-9][a-zA-Z0-9_\-\.:\-]*[a-zA-Z0-9]+
```

외부 ID를 사용하여 개체 참조

UUID를 사용하여 객체를 참조할 수 있는 곳에서는 외부 ID가 있는 경우 외부 ID를 대신 사용할 수 있습니다. 이렇게 하려면 문자열에 외부 ID를 추가합니다. `externalId`:

예를 들어 UUID (자산 모델 ID) 가 `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE` 있고 외부 ID도 있는 자산 모델이 있다고 가정해 보겠습니다. `myExternalId` [DescribeAsset모델에 대한 세부 정보를 얻으려면 모델을](#) 호출하십시오. 다음 중 하나를 값으로 사용할 수 있습니다 `assetModelId`.

- 자산 모델 ID (UUID) 자체 사용: `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`
- 외부 ID 사용 시: `externalId:myExternalId`

```
aws iotsitewise describe-asset-model --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
aws iotsitewise describe-asset-model --asset-model-id externalId:myExternalId
```

Note

`externalId`: 접두사 자체는 외부 ID의 일부가 아닙니다. UUID 또는 외부 ID를 허용하는 API 작업에 외부 ID를 제공할 때만 접두사를 제공하면 됩니다. 예를 들어 기존 객체를 쿼리하거나 업데이트할 때 접두사를 제공하십시오.

자산 모델을 만들 때와 같이 객체의 외부 ID를 정의할 때는 접두사를 포함하지 마십시오.

이러한 방식으로 UUID 대신 외부 ID를 사용할 수 있지만 전부는 아닌 많은 API 작업에 사용할 수 있습니다. AWS IoT SiteWise 예를 [GetAssetPropertyValue](#) 들어는 UUID를 사용해야 하며 외부 ID 사용을 지원하지 않습니다.

특정 API 작업에서 이 사용을 지원하는지 확인하려면 [API](#) 참조를 참조하십시오.

자산 모델 및 구성 요소 모델 생성

AWS IoT SiteWise 자산 모델 및 구성 요소 모델은 산업 데이터의 표준화를 촉진합니다. 자산 모델 또는 부품 모델에는 이름, 설명, 자산 속성 및 속성을 함께 그룹화하거나 서버어셈블리의 부품 모델을 참조하는 사용자 지정 복합 모델 (선택 사항) 이 포함됩니다.

- 자산 모델을 사용하여 자산을 생성합니다. 위에 나열된 기능 외에도 자산 모델에는 자산 간의 관계를 정의하는 계층 정의가 포함될 수 있습니다.
- 구성 요소 모델은 자산 모델 또는 다른 구성 요소 모델 내의 하위 어셈블리를 나타냅니다. 컴포넌트 모델을 생성할 때 에셋 모델 및 다른 컴포넌트 모델에 참조를 추가할 수 있습니다. 하지만 구성 요소 모델에서 직접 에셋을 생성할 수는 없습니다.

자산 모델 또는 구성 요소 모델을 생성한 후 사용자 지정 복합 모델을 생성하여 속성을 함께 그룹화하거나 기존 구성 요소 모델을 참조할 수 있습니다.

자산 모델 및 구성 요소 모델을 만드는 방법에 대한 자세한 내용은 다음 섹션을 참조하십시오.

주제

- [자산 모델 생성](#)
- [컴포넌트 모델 생성](#)
- [데이터 속성 정의](#)
- [사용자 지정 복합 모델 생성 \(구성 요소\)](#)

자산 모델 생성

AWS IoT SiteWise 자산 모델은 산업 데이터의 표준화를 촉진합니다. 자산 모델에는 이름, 설명, 자산 속성 및 자산 계층 구조 정의가 포함되어 있습니다. 예를 들어 온도, 분당 회전 수(RPM) 및 전력 속성을 사용하여 풍력 터빈 모델을 정의할 수 있습니다. 그런 다음, 순 전력 출력 속성 및 풍력 터빈 계층 구조 정의를 사용하여 풍력 발전소 모델을 정의할 수 있습니다.

Note

- 가장 낮은 수준의 노드부터 시작해서 조직을 모델링하는 것이 좋습니다. 예를 들어 풍력 발전소 모델을 생성하기 전에 풍력 터빈 모델을 생성합니다. 자산 계층 정의에는 기존 자산 모델에 대한 참조가 포함되어 있습니다. 이 방법을 사용하면 모델을 생성할 때 자산 계층 구조를 정의할 수 있습니다.
- 자산 모델에는 다른 자산 모델이 포함될 수 없습니다. 다른 모델 내에서 서브어셈블리로 참조할 수 있는 모델을 정의해야 하는 경우에는 컴포넌트--> 모델을 대신 생성해야 합니다. 자세한 정보는 [컴포넌트 모델 생성](#)을 참조하세요.

다음 섹션에서는 AWS IoT SiteWise 콘솔 또는 API를 사용하여 자산 모델을 생성하는 방법을 설명합니다. 다음 단원에서는 모델을 생성하는 데 사용할 수 있는 다양한 유형의 자산 속성과 자산 계층 구조에 대해서도 설명합니다.

주제

- [자산 모델 생성\(콘솔\)](#)
- [자산 모델 생성 \(AWS CLI\)](#)
- [자산 모델 예](#)
- [자산 모델 계층 구조 정의](#)

자산 모델 생성(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산 모델을 만들 수 있습니다. AWS IoT SiteWise 콘솔은 유효한 자산 모델을 정의하는 데 도움이 되는 수식 자동 완성과 같은 다양한 기능을 제공합니다.

자산 모델을 생성하려면(콘솔)

- [AWS IoT SiteWise 콘솔](#)로 이동합니다.

2. 탐색 창에서 모델을 선택합니다.
3. Create model(모델 생성)을 선택합니다.
4. 모델 생성 페이지에서 다음을 수행합니다.
 - a. 자산 모델의 이름을 입력합니다(예: **Wind Turbine** 또는 **Wind Turbine Model**). 이 이름은 이 리전의 계정에 있는 모든 모델에서 고유해야 합니다.
 - b. (선택 사항) 모델의 외부 ID를 추가합니다. 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
 - c. (선택 사항) 모델의 측정 정의를 추가합니다. 측정값은 장비의 데이터 스트림을 나타냅니다. 자세한 정보는 [장비의 데이터 스트림 정의\(측정값\)](#)을 참조하세요.
 - d. (선택 사항) 모델의 변환 정의를 추가합니다. 변환은 데이터를 한 양식에서 다른 양식으로 매핑하는 공식입니다. 자세한 정보는 [데이터 변환\(변환\)](#)을 참조하세요.
 - e. (선택 사항) 모델의 지표 정의를 추가합니다. 지표는 일정 간격으로 데이터를 집계하는 공식입니다. 지표는 연결된 자산에서 데이터를 입력할 수 있으므로 작업 또는 작업의 하위 집합을 나타내는 값을 계산할 수 있습니다. 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)을 참조하세요.
 - f. (선택 사항) 모델의 계층 구조 정의를 추가합니다. 계층은 자산 간의 관계입니다. 자세한 정보는 [자산 모델 계층 구조 정의](#)을 참조하세요.
 - g. (선택 사항) 자산 모델에 대한 태그를 추가합니다. 자세한 정보는 [리소스에 태그 지정하기 AWS IoT SiteWise](#)을 참조하세요.
 - h. Create model(모델 생성)을 선택합니다.

자산 모델을 생성하면 AWS IoT SiteWise 콘솔이 새 모델의 페이지로 이동합니다. 이 페이지에서 모델의 상태를 볼 수 있습니다. 상태는 처음에 CREATING(생성 중)입니다. 이 페이지는 자동으로 업데이트되므로 모델 상태가 업데이트될 때까지 기다릴 수 있습니다.

Note

복잡한 모델의 경우, 자산 모델 생성 프로세스가 몇 분 정도 소요될 수 있습니다. 자산 모델 상태가 활성이면 자산 모델을 사용하여 자산을 생성할 수 있습니다. 자세한 정보는 [자산 및 모델 상태](#)를 참조하세요.

5. (선택 사항) 자산 모델을 만든 후 엣지에 맞게 자산 모델을 구성할 수 있습니다. SiteWise Edge에 대한 자세한 내용은 [엣지 데이터 처리 활성화](#)를 참조하십시오.

- a. 모델 페이지에서 엣지용 구성을 선택합니다.
- b. 모델 구성 페이지에서 모델의 엣지 구성을 선택합니다. 이는 이 자산 모델과 관련된 속성을 계산하고 저장할 AWS IoT SiteWise 수 있는 위치를 제어합니다. 엣지용 모델 구성에 대한 자세한 내용은 [the section called “엣지 기능 설정하기”](#) 단원을 참조하세요.
- c. 사용자 지정 엣지 구성의 경우 각 자산 모델 속성을 계산하고 저장할 위치를 선택합니다.

AWS IoT SiteWise

Note

연결된 변환 및 지표는 동일한 위치에 구성되어야 합니다. 엣지용 모델 구성에 대한 자세한 내용은 [the section called “엣지 기능 설정하기”](#) 단원을 참조하세요.

- d. 저장을 선택합니다. 이제 모델 페이지에서 엣지 구성이 구성됨이어야 합니다.

자산 모델 생성 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 모델을 만들 수 있습니다.

[CreateAsset모델](#) 작업을 사용하여 속성과 계층이 포함된 자산 모델을 만들 수 있습니다. 이 작업에는 다음과 같은 구조의 페이로드가 필요합니다.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition
}
```

자산 모델을 만들려면 ()AWS CLI

1. `asset-model-payload.json`이라는 파일을 만들고 다음 JSON 객체를 파일에 복사합니다.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [
```

```

    ],
    "assetModelHierarchies": [

    ],
    "assetModelCompositeModels": [

    ]
  }

```

2. 선호하는 JSON 텍스트 편집기를 사용하여 다음에 대한 `asset-model-payload.json` 파일을 편집합니다.
 - a. 자산 모델의 이름(`assetModelName`)을 입력합니다(예: **Wind Turbine** 또는 **Wind Turbine Model**). 이 이름은 계정의 모든 자산 모델 및 구성 요소 모델에서 고유해야 AWS 리전합니다.
 - b. (선택 사항) 자산 모델의 외부 ID (`assetModelExternalId`) 를 입력합니다. 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
 - c. (선택 사항) 자산 모델에 대한 설명(`assetModelDescription`)을 입력하거나 `assetModelDescription` 키-값 페어를 제거합니다.
 - d. (선택 사항) 모델의 자산 속성(`assetModelProperties`)을 정의합니다. 자세한 정보는 [데이터 속성 정의](#)을 참조하십시오.
 - e. (선택 사항) 모델의 자산 계층 구조(`assetModelHierarchies`)를 정의합니다. 자세한 정보는 [자산 모델 계층 구조 정의](#)을 참조하십시오.
 - f. (선택 사항) 모델의 경보를 정의합니다. 경보는 다른 속성을 모니터링하므로 장비나 프로세스에 주의가 필요한 시점을 식별할 수 있습니다. 각 경고 정의는 경고가 사용하는 속성 집합을 표준화하는 복합 모델(`assetModelCompositeModels`)입니다. 자세한 내용은 [경보로 데이터 모니터링 및 자산 모델에서 경고 정의](#) 섹션을 참조하십시오.
 - g. (선택 사항) 자산 모델에 대한 태그(`tags`)를 추가합니다. 자세한 정보는 [리소스에 태그 지정하기 AWS IoT SiteWise](#)을 참조하십시오.
3. 다음 명령을 실행하여 JSON 파일의 정의에서 자산 모델을 생성합니다.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

이 작업은 자산을 생성할 때 참조하는 `assetModelId`가 포함된 응답을 반환합니다. 응답에는 모델의 상태(`assetModelStatus.state`)도 포함됩니다. 상태는 처음에 CREATING입니다. 자산 모델의 상태는 변경 사항이 전파될 때까지 CREATING입니다.

Note

복잡한 모델의 경우, 자산 모델 생성 프로세스가 몇 분 정도 소요될 수 있습니다. 자산 모델의 현재 상태를 확인하려면 `를 지정하여 DescribeAsset모델 작업을 사용합니다. assetModelId 자산 모델 상태가 ACTIVE이면 자산 모델을 사용하여 자산을 생성할 수 있습니다. 자세한 정보는 자산 및 모델 상태을 참조하세요.`

4. (선택 사항) 자산 모델을 위한 사용자 지정 복합 모델을 생성합니다. 사용자 지정 복합 모델을 사용하면 모델 내의 속성을 그룹화하거나 컴포넌트 모델을 참조하여 서버어셈블리를 포함할 수 있습니다. 자세한 정보는 [사용자 지정 복합 모델 생성 \(구성 요소\)](#)을 참조하세요.

자산 모델 예

이 섹션에는 `를 사용하여 자산 모델을 만드는 데 사용할 수 있는 예제 자산 모델 정의가 포함되어 있습니다. AWS CLI AWS IoT SiteWise 이러한 자산 모델은 풍력 터빈과 풍력 발전 단지를 나타냅니다. 풍력 터빈 자산은 원시 센서 데이터를 수집하고 전력 및 평균 풍속과 같은 값을 계산합니다. 풍력 발전 단지 자산은 풍력 발전 단지 내 모든 풍력 터빈의 총 출력과 같은 값을 계산합니다.`

주제

- [풍력 터빈 자산 모델](#)
- [풍력 발전소 자산 모델](#)

풍력 터빈 자산 모델

다음 자산 모델은 풍력 발전소의 터빈을 나타냅니다. 풍력 터빈은 센서 데이터를 수집하여 전력 및 평균 풍속과 같은 값을 계산합니다.

Note

이 예제 모델은 데모의 풍력 터빈 모델과 비슷합니다. AWS IoT SiteWise 자세한 정보는 [AWS IoT SiteWise 데모 사용](#)을 참조하세요.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "Wind Turbine Asset Model",
  "assetModelDescription": "Represents a turbine in a wind farm.",
```

```
"assetModelProperties": [  
  {  
    "name": "Location",  
    "dataType": "STRING",  
    "type": {  
      "attribute": {  
        "defaultValue": "Renton"  
      }  
    }  
  },  
  {  
    "name": "Make",  
    "dataType": "STRING",  
    "type": {  
      "attribute": {  
        "defaultValue": "Amazon"  
      }  
    }  
  },  
  {  
    "name": "Model",  
    "dataType": "INTEGER",  
    "type": {  
      "attribute": {  
        "defaultValue": "500"  
      }  
    }  
  },  
  {  
    "name": "Torque (KiloNewton Meter)",  
    "dataType": "DOUBLE",  
    "unit": "kNm",  
    "type": {  
      "measurement": {}  
    }  
  },  
  {  
    "name": "Wind Direction",  
    "dataType": "DOUBLE",  
    "unit": "Degrees",  
    "type": {  
      "measurement": {}  
    }  
  }  
],
```

```
{
  "name": "RotationsPerMinute",
  "dataType": "DOUBLE",
  "unit": "RPM",
  "type": {
    "measurement": {}
  }
},
{
  "name": "Wind Speed",
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
},
{
  "name": "RotationsPerSecond",
  "dataType": "DOUBLE",
  "unit": "RPS",
  "type": {
    "transform": {
      "expression": "rpm / 60",
      "variables": [
        {
          "name": "rpm",
          "value": {
            "propertyId": "RotationsPerMinute"
          }
        }
      ]
    }
  }
},
{
  "name": "Overdrive State",
  "dataType": "DOUBLE",
  "type": {
    "transform": {
      "expression": "gte(torque, 3)",
      "variables": [
        {
          "name": "torque",
          "value": {
```

```

        "propertyId": "Torque (KiloNewton Meter)"
    }
}
],
}
},
{
    "name": "Average Power",
    "dataType": "DOUBLE",
    "unit": "Watts",
    "type": {
        "metric": {
            "expression": "avg(torque) * avg(rps) * 2 * 3.14",
            "variables": [
                {
                    "name": "torque",
                    "value": {
                        "propertyId": "Torque (Newton Meter)"
                    }
                },
                {
                    "name": "rps",
                    "value": {
                        "propertyId": "RotationsPerSecond"
                    }
                }
            ],
            "window": {
                "tumbling": {
                    "interval": "5m"
                }
            }
        }
    }
},
{
    "name": "Average Wind Speed",
    "dataType": "DOUBLE",
    "unit": "m/s",
    "type": {
        "metric": {
            "expression": "avg(windspeed)",
            "variables": [

```

```
        {
          "name": "windspeed",
          "value": {
            "propertyId": "Wind Speed"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "5m"
        }
      }
    }
  },
  {
    "name": "Torque (Newton Meter)",
    "dataType": "DOUBLE",
    "unit": "Nm",
    "type": {
      "transform": {
        "expression": "knm * 1000",
        "variables": [
          {
            "name": "knm",
            "value": {
              "propertyId": "Torque (KiloNewton Meter)"
            }
          }
        ]
      }
    }
  },
  {
    "name": "Overdrive State Time",
    "dataType": "DOUBLE",
    "unit": "Seconds",
    "type": {
      "metric": {
        "expression": "statetime(overdrive_state)",
        "variables": [
          {
            "name": "overdrive_state",
            "value": {
```

```

        "propertyId": "Overdrive State"
      }
    }
  ],
  "window": {
    "tumbling": {
      "interval": "5m"
    }
  }
}
],
"assetModelHierarchies": []
}

```

풍력 발전소 자산 모델

다음 자산 모델은 여러 풍력 터빈으로 구성된 풍력 발전소를 나타냅니다. 이 자산 모델은 풍력 터빈 모델의 [계층 구조](#)를 정의합니다. 이를 통해 풍력 발전 단지는 풍력 발전 단지의 모든 풍력 터빈에 대한 데이터에서 값 (예: 평균 전력)을 계산할 수 있습니다.

Note

이 예제 모델은 데모의 풍력 발전 단지 모델과 유사합니다. AWS IoT SiteWise 자세한 정보는 [AWS IoT SiteWise 데모 사용](#)을 참조하세요.

이 자산 모델은 [풍력 터빈 자산 모델](#)에 따라 다릅니다. `propertyId` 및 `childAssetModelId` 값을 기존 풍력 터빈 자산 모델의 값으로 바꿉니다.

```

{
  "assetModelName": "Wind Farm Asset Model",
  "assetModelDescription": "Represents a wind farm.",
  "assetModelProperties": [
    {
      "name": "Code",
      "dataType": "INTEGER",
      "type": {
        "attribute": {
          "defaultValue": "300"
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "name": "Location",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Renton"
      }
    }
  },
  {
    "name": "Reliability Manager",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Mary Major"
      }
    }
  },
  {
    "name": "Total Overdrive State Time",
    "dataType": "DOUBLE",
    "unit": "seconds",
    "type": {
      "metric": {
        "expression": "sum(overdrive_state_time)",
        "variables": [
          {
            "name": "overdrive_state_time",
            "value": {
              "propertyId": "ID of Overdrive State Time property in Wind Turbine Asset Model",
              "hierarchyId": "Turbine Asset Model"
            }
          }
        ]
      },
      "window": {
        "tumbling": {
          "interval": "5m"
        }
      }
    }
  }
}

```

```

    },
    {
      "name": "Total Average Power",
      "dataType": "DOUBLE",
      "unit": "Watts",
      "type": {
        "metric": {
          "expression": "sum(turbine_avg_power)",
          "variables": [
            {
              "name": "turbine_avg_power",
              "value": {
                "propertyId": "ID of Average Power property in Wind Turbine Asset Model",
                "hierarchyId": "Turbine Asset Model"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "5m"
            }
          }
        }
      }
    }
  ],
  "assetModelHierarchies": [
    {
      "name": "Turbine Asset Model",
      "childAssetModelId": "ID of Wind Turbine Asset Model"
    }
  ]
}

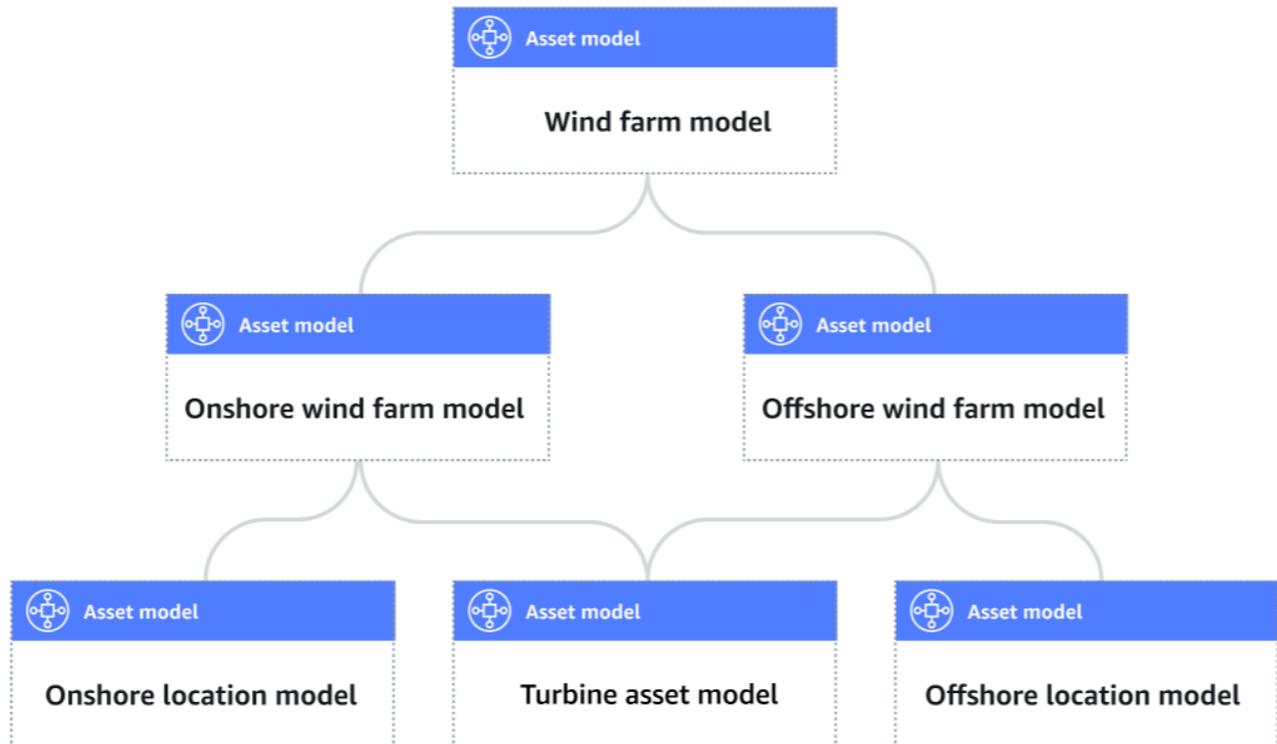
```

자산 모델 계층 구조 정의

자산 모델 계층 구조를 정의하여 산업 운영의 자산 모델 간에 논리적 연관성을 만들 수 있습니다. 예를 들어, 육상 및 해상 풍력 단지로 구성된 풍력 발전 단지를 정의할 수 있습니다. 육상 풍력 발전 단지에는 터빈과 육상 위치가 포함됩니다. 해상 풍력 발전 단지에는 터빈과 해상 위치가 있습니다.



Asset model hierarchy



계층 구조를 통해 하위 자산 모델을 상위 자산 모델에 연결하면 상위 자산 모델의 지표에서 하위 자산 모델 지표의 데이터를 입력할 수 있습니다. 자산 모델 계층 구조 및 지표를 사용하여 작업 또는 작업의 하위 집합에 대한 통찰을 제공하는 통계를 계산할 수 있습니다. 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)을 참조하세요.

각 계층 구조는 상위 자산 모델과 하위 자산 모델 간의 관계를 정의합니다. 상위 자산 모델에서는 동일한 하위 자산 모델에 여러 계층 구조를 정의할 수 있습니다. 예를 들어 풍력 발전 단지에 두 가지 유형의 풍력 터빈이 있고 모든 풍력 터빈이 동일한 자산 모델로 표시되는 경우 각 유형에 대한 계층 구조를 정의할 수 있습니다. 그런 다음 풍력 발전 단지 모델에서 지표를 정의하여 각 풍력 터빈 유형에 대한 독립된 통계와 결합된 통계를 계산할 수 있습니다.

상위 자산 모델을 여러 하위 자산 모델과 연결할 수 있습니다. 예를 들어 두 개의 서로 다른 자산 모델로 표현되는 육상 풍력 발전 단지와 해상 풍력 발전 단지가 있는 경우 이러한 자산 모델을 동일한 상위 풍력 발전 단지 자산 모델과 연결할 수 있습니다.

하위 자산 모델을 여러 상위 자산 모델과 연결할 수도 있습니다. 예를 들어, 모든 풍력 터빈이 동일한 자산 모델로 표현되는 두 가지 유형의 풍력 발전 단지가 있는 경우 풍력 터빈 자산 모델을 서로 다른 풍력 발전 단지 자산 모델과 연결할 수 있습니다.

Note

자산 모델 계층 구조를 정의할 때 하위 자산 모델은 ACTIVE이거나 이전 ACTIVE 버전이어야 합니다. 자세한 정보는 [자산 및 모델 상태](#)를 참조하세요.

계층적 자산 모델을 정의하고 자산을 생성한 후에는 자산을 연결하여 상위-하위 관계를 작성할 수 있습니다. 자세한 내용은 [자산 만들기](#) 및 [자산 연결 및 연결 해제](#) 섹션을 참조하세요.

주제

- [자산 모델 계층 구조 정의\(콘솔\)](#)
- [자산 계층 구조 정의 \(\)AWS CLI](#)

자산 모델 계층 구조 정의(콘솔)

AWS IoT SiteWise 콘솔에서 자산 모델의 계층 구조를 정의할 때 다음 매개변수를 지정합니다.

- 계층 구조 이름 - 계층 구조의 이름입니다(예: **Wind Turbines**).
- 계층 구조 모델 - 하위 자산 모델입니다.
- 계층 외부 ID (선택 사항) — 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

자세한 정보는 [자산 모델 생성\(콘솔\)](#)을 참조하세요.

자산 계층 구조 정의 ()AWS CLI

AWS IoT SiteWise API를 사용하여 자산 모델의 계층 구조를 정의할 때 다음 매개변수를 지정합니다.

- name - 계층 구조의 이름입니다(예: **Wind Turbines**).
- childAssetModelId— 계층 구조에 대한 하위 자산 모델의 ID 또는 외부 ID. [ListAsset모델](#) 작업을 사용하여 기존 자산 모델의 ID를 찾을 수 있습니다.

Example 계층 구조 정의 예

다음 예제에서는 풍력 터빈에 대한 풍력 발전소의 관계를 나타내는 자산 모델 계층 구조를 보여 줍니다. 이 개체는 [AssetModel계층](#) 구조의 예입니다. 자세한 정보는 [자산 모델 생성 \(AWS CLI\)](#)을 참조하세요.

```
{
  ...
  "assetModelHierarchies": [
    {
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    },
  ],
}
```

컴포넌트 모델 생성

AWS IoT SiteWise 컴포넌트 모델을 사용하여 에셋 모델이나 다른 컴포넌트 모델에서 참조할 수 있는 서브어셈블리를 정의합니다. 이렇게 하면 컴포넌트의 정의를 다른 여러 모델에서 재사용하거나 동일한 모델 내에서 여러 번 재사용할 수 있습니다.

구성 요소 모델을 정의하는 프로세스는 자산 모델을 정의하는 것과 매우 유사합니다. 자산 모델과 마찬가지로 구성 요소 모델에도 이름, 설명 및 자산 속성이 있습니다. 하지만 구성 요소 모델에는 자산 계층 정의를 포함할 수 없습니다. 구성 요소 모델 자체는 자산을 직접 생성하는 데 사용할 수 없기 때문입니다. 또한 구성 요소 모델은 경보를 정의할 수 없습니다.

예를 들어 모터 온도, 엔코더 온도 및 절연 저항 속성을 사용하여 서보 모터의 부품을 정의할 수 있습니다. 그런 다음 CNC 기계와 같이 서보 모터가 포함된 장비의 자산 모델을 정의할 수 있습니다.

Note

- 가장 낮은 수준의 노드부터 시작해서 조직을 모델링하는 것이 좋습니다. 예를 들어, CNC 기계의 자산 모델을 생성하기 전에 서보 모터 컴포넌트를 생성하십시오. 자산 모델에는 기존 부품 모델에 대한 참조가 포함됩니다.
- 구성 요소 모델에서 직접 에셋을 만들 수는 없습니다. 구성 요소를 사용하는 자산을 만들려면 자산에 대한 자산 모델을 만들어야 합니다. 그런 다음 구성 요소를 참조하는 사용자 지정 복합 모델을 만듭니다. 자산 모델을 만드는 방법에 대한 자세한 [자산 모델 생성](#) 내용은 사용

자 지정 복합 모델 만들기에 대한 자세한 내용은 [참조하십시오](#) [사용자 지정 복합 모델 생성 \(구성 요소\)](#).

다음 섹션에서는 AWS IoT SiteWise API를 사용하여 구성 요소 모델을 만드는 방법을 설명합니다.

주제

- [구성 요소 모델 생성 \(AWS CLI\)](#)
- [예제 컴포넌트 모델](#)

구성 요소 모델 생성 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 구성 요소 모델을 만들 수 있습니다.

[CreateAsset모델](#) 작업을 사용하여 속성이 있는 구성 요소 모델을 만들 수 있습니다. 이 작업에는 다음과 같은 구조의 페이로드가 필요합니다.

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
}
```

구성 요소 모델을 만들려면 ()AWS CLI

1. 라는 component-model-payload.json 파일을 만든 다음 다음 JSON 객체를 파일에 복사합니다.

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [

  ]
}
```

2. 선호하는 JSON 텍스트 편집기를 사용하여 다음에 대한 component-model-payload.json 파일을 편집합니다.

- a. 구성 요소 모델의 이름 (assetModelName) 을 입력합니다 (예: **Servo Motor** 또는 **Servo Motor Model**). 이 이름은 계정의 모든 자산 모델 및 구성 요소 모델에서 고유해야 AWS 리전합니다.
 - b. (선택 사항) 구성 요소 모델의 외부 ID (assetModelExternalId) 를 입력합니다. 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
 - c. (선택 사항) 자산 모델에 대한 설명(assetModelDescription)을 입력하거나 assetModelDescription 키값 페어를 제거합니다.
 - d. (선택 사항) 구성 요소 모델의 자산 속성 (assetModelProperties) 을 정의합니다. 자세한 정보는 [데이터 속성 정의](#)을 참조하십시오.
 - e. (선택 사항) 자산 모델에 대한 태그(tags)를 추가합니다. 자세한 정보는 [리소스에 태그 지정하기 AWS IoT SiteWise](#)을 참조하십시오.
3. 다음 명령을 실행하여 JSON 파일의 정의에서 구성 요소 모델을 생성합니다.

```
aws iotsitewise create-asset-model --cli-input-json file://component-model-payload.json
```

이 작업은 자산 모델이나 다른 구성 요소 모델의 구성 요소 모델에 대한 참조를 추가할 때 참조하는 내용이 포함된 응답을 반환합니다. assetModelId 응답에는 모델의 상태 (assetModelStatus.state)도 포함됩니다. 상태는 처음에 CREATING입니다. 구성 요소 모델의 상태는 변경 내용이 CREATING 전파될 때까지입니다.

Note

복잡한 모델의 경우 구성 요소 모델 생성 프로세스에 최대 몇 분이 걸릴 수 있습니다. 컴포넌트 모델의 현재 상태를 확인하려면 를 지정하여 [DescribeAsset모델](#) 작업을 사용하십시오 assetModelId. 구성 요소 모델 상태가 되면 ACTIVE 자산 모델이나 다른 구성 요소 모델에서 구성 요소 모델에 대한 참조를 추가할 수 있습니다. 자세한 정보는 [자산 및 모델 상태](#)을 참조하십시오.

4. (선택 사항) 구성 요소 모델을 위한 사용자 지정 복합 모델을 생성합니다. 사용자 지정 복합 모델을 사용하면 모델 내에서 속성을 그룹화하거나 다른 부품 모델을 참조하여 서버어셈블리를 포함시킬 수 있습니다. 자세한 정보는 [사용자 지정 복합 모델 생성 \(구성 요소\)](#)을 참조하십시오.

예제 컴포넌트 모델

이 섹션에는 AWS CLI 및 AWS IoT SiteWise SDK를 사용하여 구성 요소 모델을 만드는 데 사용할 수 있는 예제 구성 요소 모델 정의가 포함되어 있습니다. 이 컴포넌트 모델은 CNC 기계와 같은 다른 장비에 사용할 수 있는 서보 모터를 나타냅니다.

주제

- [서보 모터 컴포넌트 모델](#)

서보 모터 컴포넌트 모델

다음 부품 모델은 CNC 기계와 같은 장비 내에서 사용할 수 있는 서보 모터를 나타냅니다. 서보 모터는 온도 및 전기 저항과 같은 다양한 측정을 제공합니다. 이러한 측정값은 서보 모터 부품 모델을 참조하는 자산 모델에서 생성된 자산의 속성으로 사용할 수 있습니다.

```
{
  "assetModelName": "ServoMotor",
  "assetModelType": "COMPONENT_MODEL",
  "assetModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

데이터 속성 정의

자산 속성은 자산 데이터를 포함하는 각 자산 내부의 구조입니다. 자산 속성은 다음 유형 중 하나일 수 있습니다.

- 속성 - 디바이스 제조업체 또는 지리적 위치 같은, 일반적으로 정적인 자산 속성입니다. 자세한 정보는 [정적 데이터\(속성\) 정의](#)를 참조하세요.
- 측정 - 타임스탬프가 찍힌 회전 속도 값 또는 타임스탬프가 찍힌 섭씨 온도 값과 같은 자산의 원시 기기 센서 데이터 스트림입니다. 측정은 데이터 스트림 별칭에 의해 정의됩니다. 자세한 정보는 [장비의 데이터 스트림 정의\(측정값\)](#)를 참조하세요.
- 변형 - 타임스탬프가 찍힌 화씨 온도 값 같은, 자산의 변형된 시계열 값입니다. 변형은 표현식 및 해당 표현식과 함께 사용할 변수로 정의됩니다. 자세한 정보는 [데이터 변환\(변환\)](#)을 참조하세요.
- 지표 - 시간 단위의 평균 온도 같은, 지정된 시간 간격 동안 집계된 자산의 데이터입니다. 지표는 시간 간격, 표현식 및 해당 표현식과 함께 사용할 변수로 정의됩니다. 지표 표현식은 관련 자산의 지표 속성을 입력할 수 있으므로 작업 또는 작업의 하위 집합을 나타내는 지표를 계산할 수 있습니다. 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)을 참조하세요.

자세한 정보는 [자산 모델 생성](#)을 참조하세요.

측정, 변형 및 지표를 사용하여 OEE(전반적인 장비 효율성)를 계산하는 방법의 예는 [OEE 계산 기준 AWS IoT SiteWise](#) 단원을 참고하십시오.

주제

- [정적 데이터\(속성\) 정의](#)
- [장비의 데이터 스트림 정의\(측정값\)](#)
- [데이터 변환\(변환\)](#)
- [속성과 기타 자산의 데이터 집계\(지표\)](#)
- [공식 표현식 사용](#)

정적 데이터(속성) 정의

자산 속성은 디바이스 제조업체 또는 지리적 위치 같은, 일반적으로 정적인 정보를 나타냅니다. 자산 템플릿에서 생성한 각 자산은 해당 모델의 속성을 포함합니다.

주제

- [속성 정의\(콘솔\)](#)
- [속성 정의 \(\)AWS CLI](#)

속성 정의(콘솔)

AWS IoT SiteWise 콘솔에서 자산 모델의 속성을 정의할 때 다음 매개변수를 지정합니다.

- 이름 - 속성의 이름입니다.
- 기본값 - (선택 사항) 이 속성의 기본값입니다. 모델에서 생성된 자산은 속성에 대해 이 값을 가집니다. 모델에서 생성된 자산의 기본값을 재정의하는 방법에 대한 자세한 내용은 [속성 값 업데이트](#) 단원을 참조하세요.
- 데이터 형식 - 속성의 데이터 형식으로 다음 중 하나입니다.
 - 문자열 - 최대 1,024바이트의 문자열입니다.
 - 정수 - 범위가 [-2,147,483,648, 2,147,483,647]인 부호 있는 32비트 정수입니다.
 - 실수 - 범위가 [-10¹⁰⁰, 10¹⁰⁰]이고 IEEE 754 배정밀도를 갖는 부동 소수점 숫자입니다.
 - 부울 - true 또는 false입니다.
- 외부 ID - (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

자세한 정보는 [자산 모델 생성\(콘솔\)](#)을 참조하세요.

속성 정의 ()AWS CLI

AWS IoT SiteWise API를 사용하여 자산 모델의 속성을 정의할 때 다음 매개변수를 지정합니다.

- name - 속성의 이름입니다.
- defaultValue - (선택 사항) 이 속성의 기본값입니다. 모델에서 생성된 자산은 속성에 대해 이 값을 가집니다. 모델에서 생성된 자산의 기본값을 재정의하는 방법에 대한 자세한 내용은 [속성 값 업데이트](#) 단원을 참조하세요.
- dataType - 속성의 데이터 형식으로 다음 중 하나입니다.
 - STRING - 최대 1,024바이트의 문자열입니다.
 - INTEGER - 범위가 [-2,147,483,648, 2,147,483,647]인 부호 있는 32비트 정수입니다.
 - DOUBLE - 범위가 [-10¹⁰⁰, 10¹⁰⁰]이고 IEEE 754 배정밀도를 갖는 부동 소수점 숫자입니다.
 - BOOLEAN - true 또는 false입니다.

- `externalId`— (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 [AWS IoT SiteWise 사용 설명서의 외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

Example 속성 정의 예

다음 예제에서는 자산의 모델 번호를 나타내는 속성을 기본값과 함께 보여 줍니다. 이 개체는 [AssetModel속성이](#) 포함된 속성의 예입니다. 이 객체를 [CreateAsset모델](#) 요청 페이로드의 일부로 지정하여 속성 속성을 생성할 수 있습니다. 자세한 정보는 [자산 모델 생성 \(AWS CLI\)](#)을 참조하세요.

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Model number",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "BLT123"
        }
      }
    }
  ],
  ...
}
```

장비의 데이터 스트림 정의(측정값)

측정은 타임스탬프가 지정된 온도 값 또는 타임스탬프가 지정된 분당 회전 수(RPM) 값 같은, 디바이스의 원시 센서 데이터 스트림을 나타냅니다.

주제

- [측정 정의\(콘솔\)](#)
- [측정값 정의 \(\)AWS CLI](#)

측정 정의(콘솔)

AWS IoT SiteWise 콘솔에서 자산 모델의 측정을 정의할 때 다음 매개변수를 지정합니다.

- 이름 – 속성의 이름입니다.
- 단위 – (선택 사항) 속성에 대한 과학적 단위(예: mm 또는 섭씨)입니다.

- 데이터 형식 - 속성의 데이터 형식으로 하나입니다.
 - 문자열 - 최대 1,024바이트의 문자열입니다.
 - 정수 - 범위가 [-2,147,483,648, 2,147,483,647]인 부호 있는 32비트 정수입니다.
 - 실수 - 범위가 [-10¹⁰⁰, 10¹⁰⁰]이고 IEEE 754 배정밀도를 갖는 부동 소수점 숫자입니다.
 - 부울 - true 또는 false입니다.
- 외부 ID - (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

자세한 정보는 [자산 모델 생성\(콘솔\)](#)을 참조하세요.

측정값 정의 (AWS CLI)

AWS IoT SiteWise API를 사용하여 자산 모델의 측정을 정의할 때 다음 매개변수를 지정합니다.

- name - 속성의 이름입니다.
- dataType - 속성의 데이터 형식으로 하나입니다.
 - STRING - 최대 1,024바이트의 문자열입니다.
 - INTEGER - 범위가 [-2,147,483,648, 2,147,483,647]인 부호 있는 32비트 정수입니다.
 - DOUBLE - 범위가 [-10¹⁰⁰, 10¹⁰⁰]이고 IEEE 754 배정밀도를 갖는 부동 소수점 숫자입니다.
 - BOOLEAN - true 또는 false입니다.
- unit - (선택 사항) 속성에 대한 과학적 단위(예: mm 또는 섭씨)입니다.
- externalId - (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

Example 측정 정의 예

다음 예제에서는 자산의 온도 센서 판독값을 나타내는 측정을 보여줍니다. 이 개체는 [측정값](#)이 포함된 [AssetModel속성의](#) 예입니다. 이 객체를 [CreateAsset모델](#) 요청 페이로드의 일부로 지정하여 측정 속성을 생성할 수 있습니다. 자세한 정보는 [자산 모델 생성 \(AWS CLI\)](#)을 참조하세요.

나중에 고유한 디바이스 데이터 스트림을 사용하도록 각 자산을 구성해야 하므로 자산 모델을 정의할 때 [측정](#) 구조는 빈 구조입니다. 자산의 측정 속성을 디바이스의 센서 데이터 스트림에 연결하는 방법에 대한 자세한 내용은 [산업 데이터 스트림을 자산 속성에 매핑](#) 단원을 참조하세요.

```
{
  ...
}
```

```

    "assetModelProperties": [
      {
        "name": "Temperature C",
        "dataType": "DOUBLE",
        "type": {
          "measurement": {}
        },
        "unit": "Celsius"
      }
    ],
    ...
  }

```

데이터 변환(변환)

변환은 자산 속성의 데이터 포인트를 한 양식에서 다른 양식으로 매핑하는 수학적 표현식입니다. 변환 표현식은 자산 속성 변수, 리터럴, 연산자 및 함수로 구성됩니다. 변환된 데이터 포인트는 입력 데이터 포인트와 one-to-one 관계를 유지합니다. AWS IoT SiteWise 입력 속성이 새 데이터 포인트를 받을 때마다 변환된 새 데이터 포인트를 계산합니다.

예를 들어, 자산에 섭씨 단위의 Temperature_C라는 온도 측정 스트림이 있는 경우 $Temperature_F = 9/5 * Temperature_C + 32$ 공식을 사용하여 각 데이터 포인트를 화씨로 변환할 수 있습니다. Temperature_C 측정 스트림에서 데이터 포인트를 AWS IoT SiteWise 수신할 때마다 해당 Temperature_F 값이 몇 초 내에 계산되어 Temperature_F 속성으로 사용할 수 있습니다.

변환에 둘 이상의 변수가 포함된 경우 이전에 도착한 데이터 포인트가 계산을 즉시 시작합니다. 부품 제조업체가 변환을 사용하여 제품 품질을 모니터링하는 예를 들어 보겠습니다. 제조업체는 부품 유형에 따라 다른 표준을 사용하여 다음 측정값을 사용하여 공정을 나타냅니다.

- Part_Number - 부품 유형을 식별하는 문자열.
- Good_Count - 부품이 표준을 충족할 경우 1씩 증가하는 정수.
- Bad_Count - 부품이 표준을 충족하지 않을 경우 1씩 증가하는 정수.

제조업체에서는 `if(eq(Part_Number, "BLT123") and (Bad_Count / (Good_Count + Bad_Count) > 0.1), "Caution", "Normal")`와 같은 Quality_Monitor 변환도 생성합니다.

이 변환은 특정 부품 유형에서 생산된 불량 부품의 비율을 모니터링합니다. 부품 번호가 BLT123 이고 불량 부품의 비율이 10%(0.1)를 초과하면 변환이 "Caution"으로 반환됩니다. 그렇지 않으면 변환이 "Normal"로 반환됩니다.

Note

- Part_Number에서 다른 측정값보다 먼저 새 데이터 포인트를 받으면 Quality_Monitor 변환은 새 Part_Number 값과 최신 Good_Count 및 Bad_Count 값을 사용합니다. 오류를 방지하려면 다음 제조 실행 전에 Good_Count 및 Bad_Count를 다시 설정하세요.
- 모든 변수에 새 데이터 포인트가 입력된 후에만 표현식을 평가하려면 [지표](#)를 사용하세요.

주제

- [변환 정의\(콘솔\)](#)
- [변환 정의 \(\)AWS CLI](#)

변환 정의(콘솔)

AWS IoT SiteWise 콘솔에서 자산 모델에 대한 변환을 정의할 때 다음 매개변수를 지정합니다.

- 이름 - 속성의 이름입니다.
- 단위 - (선택 사항) 속성에 대한 과학적 단위(예: mm 또는 섭씨)입니다.
- 데이터 유형 - 변환의 데이터 유형으로, 실수 또는 문자열일 수 있습니다.
- 외부 ID — (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
- 공식 - 변환 표현식입니다. 변환 표현식에서는 집계 함수나 시간 함수를 사용할 수 없습니다. 자동 완성 기능을 열려면 입력을 시작하거나 아래쪽 화살표 키를 누릅니다. 자세한 정보는 [공식 표현식 사용](#)을 참조하세요.

Important

변환은 정수, 실수, 부울 또는 문자열 유형인 속성을 입력할 수 있습니다. 부울은 0(거짓)과 1(참)로 변환됩니다.

변환은 속성이 아닌 하나 이상의 특성과 원하는 수의 속성인 특성을 입력해야 합니다. AWS IoT SiteWise 는 속성이 아닌 입력 특성이 새 데이터 포인트를 수신할 때마다 새로 변환된 데이터 포인트를 계산합니다. 새 속성 값은 변환 업데이트를 시작하지 않습니다. 자산 속성 데이터 API 작업에 대한 동일한 요청 비율이 변환 계산 결과에 적용됩니다.

공식 표현식은 실수 또는 문자열 값만 출력할 수 있습니다. 중첩 표현식은 문자열과 같은 다른 데이터 유형을 출력할 수 있지만 공식 전체는 숫자나 문자열로 평가되어야 합니다. [jp 함](#)

수를 사용하여 문자열을 숫자로 변환할 수 있습니다. 부울 값은 1(참) 또는 0(거짓)이어야 합니다. 자세한 정보는 [정의되지 않은 값, 무한 값 및 오버플로우 값](#)을 참조하세요.

자세한 정보는 [자산 모델 생성\(콘솔\)](#)을 참조하세요.

변환 정의 (AWS CLI)

AWS IoT SiteWise API를 사용하여 자산 모델의 변환을 정의할 때 다음 파라미터를 지정합니다.

- name - 속성의 이름입니다.
- unit - (선택 사항) 속성에 대한 과학적 단위(예: mm 또는 섭씨)입니다.
- dataType - 변환의 데이터 유형으로 DOUBLE 또는 STRING이어야 합니다.
- externalId - (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
- expression - 변환 표현식입니다. 변환 표현식에서는 집계 함수나 시간 함수를 사용할 수 없습니다. 자세한 정보는 [공식 표현식 사용](#)을 참조하세요.
- variables - 표현식에서 사용할 자산의 기타 특성을 정의하는 변수 목록입니다. 각 변수 구조에는 표현식에 사용할 간단한 이름과 해당 변수에 연결할 속성을 식별하는 value 구조가 포함되어 있습니다. value 구조에는 다음 정보가 포함되어 있습니다.
 - propertyId - 값을 입력할 속성의 ID입니다. ID 대신 속성 이름을 사용할 수 있습니다.

⚠ Important

변환은 정수, 실수, 부울 또는 문자열 유형의 속성을 입력할 수 있습니다. 부울은 0(거짓)와 1(참)로 변환됩니다.

변환은 속성이 아닌 하나 이상의 특성과 원하는 수의 속성인 특성을 입력해야 합니다. AWS IoT SiteWise 는 속성이 아닌 입력 특성이 새 데이터 포인트를 수신할 때마다 새로 변환된 데이터 포인트를 계산합니다. 새 속성 값은 변환 업데이트를 시작하지 않습니다. 자산 속성 데이터 API 작업에 대한 동일한 요청 비율이 변환 계산 결과에 적용됩니다.

공식 표현식은 실수 또는 문자열 값만 출력할 수 있습니다. 중첩 표현식은 문자열과 같은 다른 데이터 유형을 출력할 수 있지만 공식 전체는 숫자나 문자열로 평가되어야 합니다. [jp 함수](#)를 사용하여 문자열을 숫자로 변환할 수 있습니다. 부울 값은 1(참) 또는 0(거짓)이어야 합니다. 자세한 정보는 [정의되지 않은 값, 무한 값 및 오버플로우 값](#)을 참조하세요.

Example 변환 정의

다음 예는 자산의 온도 측정 데이터를 섭씨에서 화씨로 변환하는 변형 속성을 보여줍니다. 이 객체는 [트랜스폼이 포함된 AssetModel프로퍼티](#)의 예입니다. 이 객체를 [CreateAssetModel](#) 요청 페이로드의 일부로 지정하여 변환 속성을 생성할 수 있습니다. 자세한 정보는 [자산 모델 생성 \(AWS CLI\)](#)을 참조하세요.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "type": {
        "transform": {
          "expression": "9/5 * temp_c + 32",
          "variables": [
            {
              "name": "temp_c",
              "value": {
                "propertyId": "Temperature C"
              }
            }
          ]
        }
      },
      "unit": "Fahrenheit"
    }
  ],
  ...
}
```

Example 세 개의 변수를 포함하는 변환 정의

다음 예제는 BLT123 부품 중 10% 이상이 표준을 충족하지 않는 경우 경고 메시지("Caution")를 반환하는 변환 속성을 보여줍니다. 그렇지 않으면 정보 메시지("Normal")를 반환합니다.

```
{
  ...
  "assetModelProperties": [
    ...
```

```

{
  "name": "Quality_Monitor",
  "dataType": "STRING",
  "type": {
    "transform": {
      "expression": "if(eq(Part_Number,\"BLT123\") and (Bad_Count / (Good_Count +
Bad_Count) > 0.1), \"Caution\", \"Normal\")",
      "variables": [
        {
          "name": "Part_Number",
          "value": {
            "propertyId": "Part Number"
          }
        },
        {
          "name": "Good_Count",
          "value": {
            "propertyId": "Good Count"
          }
        },
        {
          "name": "Bad_Count",
          "value": {
            "propertyId": "Bad Count"
          }
        }
      ]
    }
  }
}
...
}

```

속성과 기타 자산의 데이터 집계(지표)

지표는 집계 함수를 사용하여 모든 입력 데이터 포인트를 처리하고 지정된 시간 간격당 단일 데이터 포인트를 출력하는 수학적 표현식입니다. 예를 들어, 어떤 지표는 온도 데이터 스트림의 시간별 평균 온도를 계산할 수 있습니다.

지표는 연결된 자산의 지표에서 데이터를 입력할 수 있으므로 작업 또는 작업의 하위 집합에 대한 통찰을 제공하는 통계를 계산할 수 있습니다. 예를 들어, 지표는 풍력 발전소의 모든 풍력 터빈에 대한 시간당 평균 온도를 계산할 수 있습니다. 자산 간의 연결을 정의하는 방법에 대한 자세한 내용은 [자산 모델 계층 구조 정의](#) 단원을 참조하세요.

또한 지표는 각 시간 간격의 데이터를 집계하지 않고도 다른 속성의 데이터를 입력할 수 있습니다. 공식에서 [속성](#)을 지정하는 경우 공식을 계산할 때 AWS IoT SiteWise 는 해당 속성의 [최신](#) 값을 사용합니다. 공식에서 지표를 지정하는 경우 수식을 계산하는 시간 간격의 [마지막](#) 값을 AWS IoT SiteWise 사용합니다. 즉, $OEE = Availability * Quality * Performance$ 와 같은 지표를 정의할 수 있습니다. 여기서 Availability, Quality, Performance는 모두 동일한 자산 모델의 다른 지표입니다.

AWS IoT SiteWise 또한 모든 자산 속성에 대한 기본 집계 지표 세트를 자동으로 계산합니다. 계산 비용을 줄이기 위해, 기본 계산에 대한 사용자 지정 지표를 정의하는 대신 이러한 집계를 사용할 수 있습니다. 자세한 정보는 [자산 속성 집계 쿼리](#)을 참조하세요.

주제

- [지표 정의\(콘솔\)](#)
- [지표 정의 \(\)AWS CLI](#)

지표 정의(콘솔)

AWS IoT SiteWise 콘솔에서 자산 모델의 지표를 정의할 때 다음 매개변수를 지정합니다.

- 이름 - 속성의 이름입니다.
- 데이터 유형 - 변환의 데이터 유형으로, 실수 또는 문자열일 수 있습니다.
- 외부 ID - (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
- 공식 - 지표 표현식입니다. 지표 표현식은 [집계 함수](#)를 사용하여 계층 구조의 모든 관련 자산에 대한 속성 데이터를 입력할 수 있습니다. 입력을 시작하거나 아래쪽 화살표 키를 눌러 자동 완성 기능을 엽니다. 자세한 정보는 [공식 표현식 사용](#)을 참조하세요.

Important

지표는 정수, 실수, 부울 또는 문자열 유형의 속성만 입력할 수 있습니다. 부울은 0(거짓)과 1(참)로 변환됩니다.

지표의 표현식에서 지표 입력 변수를 정의하는 경우, 이러한 입력의 시간 간격은 출력 지표와 동일해야 합니다.

공식 표현식은 실수 또는 문자열 값만 출력할 수 있습니다. 중첩 표현식은 문자열과 같은 다른 데이터 유형을 출력할 수 있지만 공식 전체는 숫자나 문자열로 평가되어야 합니다. [jp 함수](#)를 사용하여 문자열을 숫자로 변환할 수 있습니다. 부울 값은 1(참) 또는 0(거짓)이어야 합니다. 자세한 정보는 [정의되지 않은 값, 무한 값 및 오버플로우 값](#)을 참조하세요.

- 시간 간격 - 시간 간격 지표입니다. AWS IoT SiteWise 는 다음 연속 창 시간 간격을 지원하며, 여기서 각 간격은 이전 간격이 끝날 때 시작됩니다.
 - 1분 - 1분입니다. 1분이 끝날 때마다 계산됩니다(오전 12:00:00, 오전 12:01:00, 오전 12:02:00 등).
 - 5분 - 5분입니다. 정각에 시작하여 5분이 끝날 때마다 계산됩니다(오전 12:00:00, 오전 12:05:00, 오전 12:10:00 등).
 - 15분 - 15분입니다. 정각에 시작하여 15분이 끝날 때마다 계산됩니다(오전 12:00:00, 오전 12:15:00, 오전 12:30:00 등).
 - 1시간 - 1시간(60분)입니다. UTC를 기준으로 한 시간이 끝날 때마다 계산됩니다(오전 12:00:00, 오전 01:00:00, 오전 02:00:00 등).
 - 1일 - 1일(24시간)입니다. UTC를 기준으로 하루가 끝날 때마다 계산됩니다(월요일 오전 12:00:00, 화요일 오전 12:00:00 등).
 - 1주 - 1주(7일)입니다. UTC를 기준으로 일요일이 끝날 때마다 계산됩니다(매주 월요일 오전 12:00:00).
- 사용자 지정 간격 - 1분에서 1주 사이의 원하는 시간 간격을 입력할 수 있습니다.
- 오프셋 날짜 - (선택 사항) 데이터를 집계할 기준 날짜입니다.
- 오프셋 시간 - (선택 사항) 데이터를 집계할 기준 시간입니다. 오프셋 시간은 00:00:00 에서 23:59:59 사이여야 합니다.
- 오프셋 시간대 - (선택 사항) 오프셋의 시간대입니다. 지정되지 않은 경우 기본 오프셋 시간대는 협정 세계시(UTC)입니다.

지원되는 시간대

- (UTC+00:00) 협정 세계시
- (UTC+01:00) 유럽 중부 표준시
- (UTC+02:00) 동유럽
- (UTC+03:00) 동부 아프리카 표준시
- (UTC+04:00) 근동 표준시
- (UTC+05:00) 파키스탄 라호르 표준시
- (UTC+05:30) 인도 표준시
- (UTC+06:00) 방글라데시 표준시
- (UTC+07:00) 베트남 표준시
- (UTC+08:00) 중국 대만 표준시

데이터(UTC+09:00) 일본 표준시

- (UTC+09:30) 오스트레일리아 중부 표준시
- (UTC+10:00) 오스트레일리아 동부 표준시
- (UTC+11:00) 솔로몬 표준시
- (UTC+12:00) 뉴질랜드 표준시
- (UTC-11:00) 미드웨이 제도 표준시
- (UTC-10:00) 하와이 표준시
- (UTC-09:00) 알래스카 표준시
- (UTC-08:00) 태평양 표준시
- (UTC-07:00) 피닉스 표준시
- (UTC-06:00) 미국 중부 표준시
- (UTC-05:00) 미국 동부 표준시
- (UTC-04:00) 푸에르토리코 및 미국령 버진아일랜드 표준시
- (UTC-03:00) 아르헨티나 표준시
- (UTC-02:00) 사우스조지아 표준시
- (UTC-01:00) 중앙아프리카 표준시

Example 오프셋 사용 사용자 지정 시간 간격(콘솔)

다음 예제는 2021년 2월 20일 오후 6시 30분(PST)에 오프셋을 사용하여 12시간 간격을 정의하는 방법을 보여줍니다.

오프셋을 사용하여 사용자 지정 구간을 정의하려면

1. 시간 간격의 경우 사용자 지정 간격을 선택합니다.
2. 시간 간격의 경우 다음 중 하나를 수행하세요.
 - **12**를 입력한 다음 시간을 선택합니다.
 - **720**을 입력한 다음 분을 선택합니다.
 - **43200**을 입력한 다음 초를 선택합니다.

Important

시간 간격은 단위에 관계없이 정수여야 합니다.

3. 오프셋 날짜로 2021/02/20을 선택합니다.
4. 오프셋 시간에는 **18:30:30**을 입력합니다.
5. 오프셋 시간대의 경우 (UTC-08:00) 태평양 표준시를 선택합니다.

2021년 7월 1일 오후 6시 30분 30분 (PST) 이전 또는 이전에 지표를 생성한 경우, 2021년 7월 1일 오후 6시 30분 (PST) 에 첫 번째 집계 결과를 얻을 수 있습니다. 두 번째 집계 결과는 2021년 7월 2일 오전 6시 30분 30분 (PST) 등입니다.

지표 정의 (AWS CLI)

AWS IoT SiteWise API를 사용하여 자산 모델의 지표를 정의할 때 다음 매개변수를 지정합니다.

- name - 속성의 이름입니다.
- dataType - 지표의 데이터 유형으로 DOUBLE 또는 STRING일 수 있습니다.
- externalId— (선택 사항) 사용자 정의 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
- expression - 지표 표현식입니다. 지표 표현식은 [집계 함수](#)를 사용하여 계층 구조의 모든 관련 자산에 대한 속성 데이터를 입력할 수 있습니다. 자세한 정보는 [공식 표현식 사용](#)을 참조하십시오.
- window - 이전 간격이 끝날 때 각 간격이 시작되는 지표의 연속 창에 대한 시간 간격 및 오프셋:
 - interval - 연속 창의 시간 간격입니다. 시간 간격은 1분과 1주 사이여야 합니다.
 - offsets - 연속 창의 오프셋입니다.

자세한 내용은 AWS IoT SiteWise API [TumblingWindow](#)참조를 참조하십시오.

Example 오프셋이 있는 사용자 지정 시간 간격(AWS CLI)

다음 예제는 2021년 2월 20일 오후 6시 30분(PST)에 오프셋을 사용하여 12시간 간격을 정의하는 방법을 보여줍니다.

```
{
  "window": {
    "tumbling": {
      "interval": "12h",
      "offset": " 2021-07-23T18:30:30-08"
    }
  }
}
```

2021년 7월 1일, 오후 6시 30분 30분 (PST) 이전 또는 이전에 지표를 생성하면 2021년 7월 1일 오후 6시 30분 (PST) 에 첫 번째 집계 결과를 얻을 수 있습니다. 두 번째 집계 결과는 2021년 7월 2일 오전 6시 30분 30분 (PST) 등입니다.

- **variables** – 표현식에 사용할 자산 또는 하위 자산의 기타 속성을 정의하는 변수 목록입니다. 각 변수 구조에는 표현식에 사용할 간단한 이름과 해당 변수에 연결할 속성을 지정하는 `value` 구조가 포함되어 있습니다. `value` 구조에는 다음 정보가 포함되어 있습니다.
- **propertyId** – 값을 가져올 속성의 ID입니다. 속성이 계층의 모델에 정의되어 있지 않고 현재 모델에 정의된 경우 ID 대신 속성 이름을 사용할 수 있습니다.
- **hierarchyId** – (선택 사항) 속성에 대한 하위 자산을 쿼리할 계층의 ID입니다. ID 대신 계층 정의의 이름을 사용할 수 있습니다. 이 값을 생략하면 현재 모델에서 속성을 AWS IoT SiteWise 찾습니다.

Important

지표는 정수, 실수, 부울 또는 문자열 유형의 속성만 입력할 수 있습니다. 부울은 0(거짓)과 1(참)로 변환됩니다.

지표의 표현식에서 지표 입력 변수를 정의하는 경우, 이러한 입력의 시간 간격은 출력 지표와 동일해야 합니다.

공식 표현식은 실수 또는 문자열 값만 출력할 수 있습니다. 중첩 표현식은 문자열과 같은 다른 데이터 유형을 출력할 수 있지만 공식 전체는 숫자나 문자열로 평가되어야 합니다. [jp 함수](#)를 사용하여 문자열을 숫자로 변환할 수 있습니다. 부울 값은 1(참) 또는 0(거짓)이어야 합니다. 자세한 정보는 [정의되지 않은 값, 무한 값 및 오버플로우 값](#)을 참조하세요.

- **unit** – (선택 사항) 속성에 대한 과학적 단위(예: mm 또는 섭씨)입니다.

Example 지표 정의 예

다음 예제에서는 자산의 온도 측정 데이터를 집계하여 시간당 최대 화씨 온도를 계산하는 지표 속성을 보여줍니다. 이 개체는 [메트릭이](#) 포함된 [AssetModel속성의](#) 예입니다. 이 객체를 [CreateAsset모델](#) 요청 페이로드의 일부로 지정하여 지표 속성을 생성할 수 있습니다. 자세한 정보는 [자산 모델 생성 \(AWS CLI\)](#)을 참조하세요.

```
{
  ...
  "assetModelProperties": [
    ...
  ]
}
```

```

{
  "name": "Max temperature",
  "dataType": "DOUBLE",
  "type": {
    "metric": {
      "expression": "max(temp_f)",
      "variables": [
        {
          "name": "temp_f",
          "value": {
            "propertyId": "Temperature F"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "1h"
        }
      }
    }
  },
  "unit": "Fahrenheit"
}
...
}

```

Example 관련 자산에서 데이터를 입력하는 지표 정의 예시

다음 예는 여러 풍력 터빈의 평균 전력 데이터를 집계하여 풍력 발전 단지의 총 평균 전력을 계산하는 지표 속성을 보여줍니다. 이 객체는 [메트릭이](#) 포함된 [AssetModel속성의](#) 예입니다. 이 객체를 [CreateAsset모델](#) 요청 페이로드의 일부로 지정하여 지표 속성을 생성할 수 있습니다.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Total Average Power",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "avg(power)",

```

```

    "variables": [
      {
        "name": "power",
        "value": {
          "propertyId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
          "hierarchyId": "Turbine Asset Model"
        }
      }
    ],
    "window": {
      "tumbling": {
        "interval": "5m"
      }
    }
  },
  "unit": "kWh"
}
...
}

```

공식 표현식 사용

수식 표현식을 사용하면 수학 함수를 정의하여 원시 산업 데이터를 변형하고 집계함으로써 작업에 대한 통찰력을 얻을 수 있습니다. 공식 표현식은 리터럴, 연산자, 함수 및 변수를 결합하여 데이터를 처리합니다. 공식 표현식을 사용하는 자산 속성을 정의하는 방법에 대한 자세한 내용은 [데이터 변환\(변환\)](#) 및 [속성과 기타 자산의 데이터 집계\(지표\)](#) 단원을 참조하세요. 변환과 지표는 공식 속성입니다.

주제

- [공식 표현식에서 변수 사용](#)
- [수식 표현식에 리터럴 사용](#)
- [공식 표현식에서 연산자 사용](#)
- [공식 표현식에 상수 사용](#)
- [공식 표현식에서 함수 사용](#)
- [공식 표현식 튜토리얼](#)

공식 표현식에서 변수 사용

변수는 수식 표현식의 AWS IoT SiteWise 자산 속성을 나타냅니다. 변수를 사용하여 표현식에 다른 자산 속성의 값을 입력하면 상수 속성([속성](#)), 원시 데이터 스트림([측정](#)) 및 기타 공식 속성의 데이터를 처리할 수 있습니다.

변수는 동일한 자산 모델 또는 관련 하위 자산 모델의 자산 속성을 나타낼 수 있습니다. 지표 공식만 하위 자산 모델의 변수를 입력할 수 있습니다.

콘솔과 API에서 변수를 서로 다른 이름으로 식별할 수 있습니다.

- AWS IoT SiteWise 콘솔 — 자산 속성 이름을 표현식의 변수로 사용합니다.
- AWS IoT SiteWise API (AWS CLI, AWS SDK) — 변수 이름과 자산 속성에 대한 참조가 필요한 [ExpressionVariable](#) 구조를 사용하여 변수를 정의합니다. 변수 이름에는 소문자, 숫자, 밑줄이 포함될 수 있습니다. 그런 다음 변수 이름을 사용하여 표현식의 자산 속성을 참조합니다.

변수 이름은 대소문자를 구분합니다.

자세한 내용은 [변환 정의](#) 및 [지표 정의](#)를 참조하세요.

변수를 사용하여 속성 참조

변수 값은 해당 변수가 참조하는 속성을 정의합니다. AWS IoT SiteWise 이를 위한 다양한 방법을 제공합니다.

- 속성 ID별: 속성의 고유 ID (UUID) 를 지정하여 식별할 수 있습니다.
- 이름별: 속성이 동일한 자산 모델에 있는 경우 속성 ID 필드에 속성 이름을 지정할 수 있습니다.
- 경로별: 변수 값은 경로를 통해 속성을 참조할 수 있습니다. 자세한 정보는 [경로를 사용하여 사용자 지정 복합 모델 속성 참조](#)을 참조하세요.

Note

AWS IoT SiteWise 콘솔에서는 변수를 지원하지 않습니다. AWS Command Line Interface (AWS CLI) 및 AWS SDK를 포함한 AWS IoT SiteWise API에서 사용됩니다.

응답으로 받은 변수에는 ID와 경로를 포함하여 값에 대한 전체 정보가 AWS IoT SiteWise 포함됩니다.

하지만 변수를 전달하는 경우 AWS IoT SiteWise (예: “create” 또는 “update” 호출), 다음 중 하나만 지정하면 됩니다. 예를 들어 경로를 지정하는 경우 ID를 제공할 필요가 없습니다.

수식 표현식에 리터럴 사용

공식 표현식에서 숫자 및 문자열 리터럴을 정의할 수 있습니다.

- 숫자

숫자와 과학 표기법을 사용하여 정수와 실수를 정의합니다. [E 표기법](#)을 사용하여 숫자를 과학적 표기법으로 표현할 수 있습니다.

예: 1, 2.0, .9, -23.1, 7.89e3, 3.4E-5

- 문자열

'(따옴표) 및 "(큰따옴표) 문자를 사용하여 문자열을 정의합니다. 시작과 끝의 따옴표 유형은 일치해야 합니다. 문자열을 선언할 때 사용하는 것과 일치하는 따옴표를 이스케이프하려면 해당 따옴표를 두 번 포함시키십시오. 이 문자는 문자열의 유일한 이스케이프 AWS IoT SiteWise 문자입니다.

예: 'active', "inactive", '{"temp": 52}', "{\"temp\": \"high\"}"

공식 표현식에서 연산자 사용

공식 표현식에는 다음과 같은 일반적인 수학 연산자를 사용할 수 있습니다.

연산자	설명
+	<p>두 피연산자가 모두 숫자인 경우 이 연산자는 왼쪽 및 오른쪽 피연산자를 더합니다.</p> <p>피연산자 중 하나가 문자열인 경우 이 연산자는 왼쪽 및 오른쪽 피연산자를 문자열로 연결합니다. 예를 들어, 표현식 <code>1 + 2 + " is three"</code>은 <code>"3 is three"</code>로 평가됩니다. 문자열은 최대 1,024자를 가질 수 있습니다. 문자열이 1,024자를 초과하는 경우 AWS IoT SiteWise에서는 해당 계산에 대한 데이터 포인트를 출력하지 않습니다.</p>

연산자	설명
-	<p>왼쪽 피연산자에서 오른쪽 피연산자를 뺍니다.</p> <p>이 연산자는 숫자 피연산자에만 사용할 수 있습니다.</p>
/	<p>왼쪽 피연산자를 오른쪽 피연산자로 나눕니다.</p> <p>이 연산자는 숫자 피연산자에만 사용할 수 있습니다.</p>
*	<p>왼쪽 피연산자와 오른쪽 피연산자를 곱합니다.</p> <p>이 연산자는 숫자 피연산자에만 사용할 수 있습니다.</p>
^	<p>왼쪽 피연산자를 오른쪽 피연산자의 거듭제곱으로 올립니다(지수).</p> <p>이 연산자는 숫자 피연산자에만 사용할 수 있습니다.</p>
%	<p>왼쪽 피연산자를 오른쪽 피연산자로 나눈 나머지를 반환합니다. 결과는 왼쪽 피연산자와 동일한 부호를 가집니다. 이 동작은 모듈로 작업과 다릅니다.</p> <p>이 연산자는 숫자 피연산자에만 사용할 수 있습니다.</p>
$x < y$	x가 y보다 작으면 1을 반환하고, 그렇지 않으면 0를 반환합니다.
$x > y$	x가 y보다 크면 1을 반환하고, 그렇지 않으면 0를 반환합니다.
$x <= y$	x가 y보다 작거나 같으면 1을 반환하고, 그렇지 않으면 0를 반환합니다.

연산자	설명
$x \geq y$	x가 y보다 크거나 같으면 1을 반환하고, 그렇지 않으면 0를 반환합니다.
$x == y$	x가 y와 같으면 1을 반환하고, 그렇지 않으면 0를 반환합니다.
$x != y$	x가 y와 같지 않으면 1을 반환하고, 그렇지 않으면 0를 반환합니다.
$!x$	<p>x가 0(거짓)으로 평가되면 1을 반환하고, 그렇지 않으면 0를 반환합니다.</p> <p>다음과 같은 경우 x는 거짓으로 평가됩니다.</p> <ul style="list-style-type: none"> x는 숫자형 피연산자이며, 0로 평가됩니다. x는 빈 문자열로 평가됩니다. x는 빈 배열로 평가됩니다. x는 None로 평가됩니다.
$x \text{ and } y$	<p>x가 0(거짓)로 평가되면 0을 반환합니다. 그렇지 않으면 y의 평가된 결과를 반환합니다.</p> <p>x 또는 y는 다음과 같은 경우 거짓으로 평가됩니다.</p> <ul style="list-style-type: none"> x 또는 y는 숫자형 피연산자이며, 0로 평가됩니다. x 또는 y는 빈 문자열로 평가됩니다. x 또는 y는 빈 배열로 평가됩니다. x 또는 y는 None으로 평가됩니다.

연산자	설명
<code>x or y</code>	<p><code>x</code>가 1(참)로 평가되면 1을 반환합니다. 그렇지 않으면 <code>y</code>의 평가된 결과를 반환합니다.</p> <p><code>x</code> 또는 <code>y</code>는 다음과 같은 경우 거짓으로 평가됩니다.</p> <ul style="list-style-type: none"> <code>x</code> 또는 <code>y</code>는 숫자형 피연산자이며, 0로 평가됩니다. <code>x</code> 또는 <code>y</code>는 빈 문자열로 평가됩니다. <code>x</code> 또는 <code>y</code>는 빈 배열로 평가됩니다. <code>x</code> 또는 <code>y</code>는 None으로 평가됩니다.
<code>not x</code>	<p><code>x</code>가 0(거짓)으로 평가되면 1을 반환하고, 그렇지 않으면 0를 반환합니다.</p> <p>다음과 같은 경우 <code>x</code>는 거짓으로 평가됩니다.</p> <ul style="list-style-type: none"> <code>x</code>는 숫자형 피연산자이며, 0로 평가됩니다. <code>x</code>는 빈 문자열로 평가됩니다. <code>x</code>는 빈 배열로 평가됩니다. <code>x</code>는 None로 평가됩니다.
<code>[]</code> <code>s[index]</code>	<p>문자열 <code>s</code>의 인덱스 <code>index</code>에 있는 문자를 반환합니다. 이 구문은 Python의 인덱스 구문과 동등합니다.</p> <p>Example 예제</p> <ul style="list-style-type: none"> "Hello!"[1] 에서 e를 반환합니다. "Hello!"[-2] 에서 o를 반환합니다.

연산자	설명
<p data-bbox="115 306 152 342">[]</p> <p data-bbox="115 388 435 424">s[start:end:step]</p>	<p data-bbox="829 226 1495 357">문자열 s의 한 조각을 반환합니다. 이 구문은 Python의 조각 구문과 동등합니다. 이 연산자에는 다음과 같은 인수가 사용됩니다.</p> <ul data-bbox="829 403 1495 840" style="list-style-type: none"> • start - (선택 사항) 조각의 포함 시작 인덱스입니다. 기본값은 0입니다. • end - (선택 사항) 조각의 배타적 끝 인덱스입니다. 기본값은 문자열 길이입니다. • step - (선택 사항) 조각의 각 단계마다 증가할 수 있습니다. 예를 들어, 다른 모든 문자가 포함된 조각을 반환하도록 2를 지정하거나 조각을 반대로 -1을 반환하도록 지정할 수 있습니다. 기본값은 1입니다. <p data-bbox="829 919 1495 1045">step 인수를 생략하여 기본값을 사용할 수 있습니다. 예를 들어, s[1:4:1]은 s[1:4]과 같습니다.</p> <p data-bbox="829 1092 1495 1218">인수는 정수이거나 없음 상수여야 합니다. 지정하는 경우 none, AWS IoT SiteWise 는 해당 인수의 디폴트 값을 사용합니다.</p> <p data-bbox="829 1264 1024 1299">Example 예제</p> <ul data-bbox="829 1346 1495 1812" style="list-style-type: none"> • "Hello!"[1:4] 에서 "ell"를 반환합니다. • "Hello!"[:2] 에서 "He"를 반환합니다. • "Hello!"[3:] 에서 "lo!"를 반환합니다. • "Hello!"[:-4] 에서 "He"를 반환합니다. • "Hello!"[::2] 에서 "Hlo"를 반환합니다. • "Hello!"[::-1] 에서 "!olleH"를 반환합니다.

공식 표현식에 상수 사용

표현식에는 다음과 같은 일반적인 수학 상수를 사용할 수 있습니다. 모든 상수는 대소문자를 구분하지 않습니다.

Note

상수와 이름이 같은 변수를 정의하면 변수가 상수보다 우선합니다.

상수	설명
pi	숫자 pi(π): 3.141592653589793
e	숫자 e: 2.718281828459045
true	숫자 1에 해당합니다. AWS IoT SiteWise에서는 불리언을 해당하는 숫자로 변환합니다.
false	숫자 0에 해당합니다. AWS IoT SiteWise에서는 불리언을 해당하는 숫자로 변환합니다.
none	값이 없는 것과 같습니다. 이 상수를 사용하여 조건부 표현식 의 결과로 아무것도 출력하지 않을 수 있습니다.

공식 표현식에서 함수 사용

다음 함수를 사용하여 공식 표현식의 데이터에 대해 작업할 수 있습니다.

변환과 지표는 서로 다른 함수를 지원합니다. 다음 표에는 각 유형의 공식 속성과 호환되는 함수 유형이 나와 있습니다.

Note

공식 표현식에는 최대 10개의 함수를 포함할 수 있습니다.

함수 유형	변형	지표
공식 표현식에 일반 함수 사용	 예	 네
공식 표현식에 비교 함수 사용	 네	 네
공식 표현식에 조건부 함수 사용	 네	 네
수식 표현식에 문자열 함수 사용	 네	 네
공식 표현식에서 집계 함수 사용	 아니요	 네
공식 표현식에 임시 함수 사용	 네	 네
공식 표현식에 날짜 및 시간 함수 사용	 네	 네

함수 구문

다음 구문을 사용하여 함수를 만들 수 있습니다.

일반 구문

일반 구문에서는 함수 이름 뒤에 0개 이상의 인수가 포함된 괄호가 옵니다.

`function_name(argument1, argument2, argument3, ...)`. 예를 들어, 일반 구문을 사용하는 함수는 `log(x)` 및 `contains(s, substring)`와 비슷할 수 있습니다.

UFCS(균일한 함수 호출 구문)

UFCS를 사용하면 객체 지향 프로그래밍의 메서드 호출 구문을 사용하여 함수를 호출할 수 있습니다. UFCS에서는 첫 번째 인수 다음에 점(`.`), 함수 이름, 나머지 인수(있는 경우)가 괄호 안에 옵니다.

`argument1.function_name(argument2, argument3, ...)`. 예를 들어, UFCS를 사용하는 함수는 `x.log()` 및 `s.contains(substring)`와 비슷할 수 있습니다.

UFCS를 사용하여 후속 함수를 연결할 수도 있습니다. AWS IoT SiteWise 현재 함수의 평가 결과를 다음 함수의 첫 번째 인수로 사용합니다.

예를 들어, `contains(lower(jp(message, '$.status')), 'fail')` 대신 `message.jp('$.status').lower().contains('fail')`를 사용할 수 있습니다.

자세한 내용은 [D 프로그래밍 언어](#) 웹 사이트를 방문하세요.

Note

모든 AWS IoT SiteWise 함수에 UFCS를 사용할 수 있습니다. AWS IoT SiteWise 함수는 대소문자를 구분하지 않습니다. 예를 들어, `lower(s)` 및 `Lower(s)`를 서로 바꿔서 사용할 수 있습니다.

공식 표현식에 일반 함수 사용

[변환](#)과 [지표](#)에서는 다음 함수를 사용하여 변환과 지표의 일반적인 수학 함수를 계산할 수 있습니다.

함수	설명
<code>abs(x)</code>	x의 절대값을 반환합니다.
<code>acos(x)</code>	x의 아크코사인을 반환합니다.

함수	설명
$\text{asin}(x)$	x 의 아크사인을 반환합니다.
$\text{atan}(x)$	x 의 아크탄젠트를 반환합니다.
$\text{cbrt}(x)$	x 의 세제곱근을 반환합니다.
$\text{ceil}(x)$	x 보다 큰 가장 가까운 정수를 반환합니다.
$\text{cos}(x)$	x 의 코사인을 반환합니다.
$\text{cosh}(x)$	x 의 쌍곡코사인을 반환합니다.
$\text{cot}(x)$	x 의 코탄젠트를 반환합니다.
$\text{exp}(x)$	e 를 x 의 거듭제곱으로 반환합니다.
$\text{expm1}(x)$	$\text{exp}(x) - 1$ 을 반환합니다. 이 함수를 사용하면 x 의 작은 값에 대해 $\text{exp}(x) - 1$ 를 더 정확하게 계산할 수 있습니다.
$\text{floor}(x)$	x 보다 작은 가장 가까운 정수를 반환합니다.
$\text{log}(x)$	x 의 \log_e (밑 e) 반환합니다.
$\text{log10}(x)$	x 의 \log_{10} (밑 10) 반환합니다.
$\text{log1p}(x)$	$\log(1 + x)$ 을 반환합니다. 이 함수를 사용하면 x 의 작은 값에 대해 $\log(1 + x)$ 를 더 정확하게 계산할 수 있습니다.
$\text{log2}(x)$	x 의 \log_2 (밑 2) 반환합니다.
$\text{pow}(x, y)$	x 를 y 의 거듭제곱으로 반환합니다. 이는 x^y 와 동일합니다.
$\text{signum}(x)$	x 의 부호를 반환합니다(음수를 입력하면 -1, 0을 입력하면 0, 양수를 입력하면 +1).
$\text{sin}(x)$	x 의 사인을 반환합니다.

함수	설명
$\sinh(x)$	x 의 쌍곡사인을 반환합니다.
\sqrt{x}	x 의 제곱근을 반환합니다.
$\tan(x)$	x 의 탄젠트를 반환합니다.
$\tanh(x)$	x 의 쌍곡탄젠트를 반환합니다.

공식 표현식에 비교 함수 사용

[변환과 지표에서는](#) 다음 비교 함수를 사용하여 두 값을 비교하고 출력값 1 (true) 또는 0 (false) 를 출력할 수 있습니다. AWS IoT SiteWise 문자열을 사전순으로 비교합니다.

함수	설명
$gt(x, y)$	<p>x가 y보다 크면 1을 반환하고, 그렇지 않으면 0을 반환합니다($x > y$).</p> <p>이 함수는 x 및 y가 숫자 및 문자열과 같이 호환되지 않는 유형인 경우 값을 반환하지 않습니다.</p>
$gte(x, y)$	<p>x가 y보다 크거나 같으면 1을 반환하고, 그렇지 않으면 0을 반환합니다($x \geq y$).</p> <p>AWS IoT SiteWise 의 상대 허용오차 범위 내에 있는 인수는 같다고 간주합니다. $1E-9$ 이것은 Python의 isclose 함수와 비슷하게 동작합니다.</p> <p>이 함수는 x 및 y가 숫자 및 문자열과 같이 호환되지 않는 유형인 경우 값을 반환하지 않습니다.</p>
$eq(x, y)$	<p>x가 y와 같으면 1을 반환하고, 그렇지 않으면 0을 반환합니다($x == y$).</p> <p>AWS IoT SiteWise 의 상대 허용오차 범위 내에 있는 인수는 같다고 간주합니다. $1E-9$ 이것은 Python의 isclose 함수와 비슷하게 동작합니다.</p>

함수	설명
	이 함수는 x 및 y 가 숫자 및 문자열과 같이 호환되지 않는 유형인 경우 값을 반환하지 않습니다.
<code>lt(x, y)</code>	<p>x가 y보다 작으면 1을 반환하고, 그렇지 않으면 0을 반환합니다($x < y$).</p> <p>이 함수는 x 및 y가 숫자 및 문자열과 같이 호환되지 않는 유형인 경우 값을 반환하지 않습니다.</p>
<code>lte(x, y)</code>	<p>x가 y보다 작거나 같으면 1을 반환하고, 그렇지 않으면 0을 반환합니다($x \leq y$).</p> <p>AWS IoT SiteWise의 상대 허용오차 범위 내에 있는 인수는 같다고 간주합니다. $1E-9$ 이것은 Python의 isclose 함수와 비슷하게 동작합니다.</p> <p>이 함수는 x 및 y가 숫자 및 문자열과 같이 호환되지 않는 유형인 경우 값을 반환하지 않습니다.</p>
<code>isnan(x)</code>	<p>x가 NaN와 같으면 1을 반환하고, 그렇지 않으면 0을 반환합니다.</p> <p>이 함수는 x가 문자열인 경우 값을 반환하지 않습니다.</p>

공식 표현식에 조건부 함수 사용

[변환 및 지표에서](#) 다음 함수를 사용하여 조건을 확인하고 조건이 true로 평가되는지 아니면 false로 평가되는지 여부에 관계없이 다양한 결과를 반환할 수 있습니다.

함수	설명
<code>if(condition, result_if_true, result_if_false)</code>	<code>condition</code> 을 평가하고 조건이 참으로 평가되면 <code>result_if_true</code> 를 반환하고 조건이 false로 평가되면 <code>result_if_false</code> 를 반환합니다.

함수	설명
	<p>condition 은 숫자여야 합니다. 이 함수는 0 및 빈 문자열을 false로 간주하고 그 밖의 모든 것(NaN 포함)을 true로 간주합니다. 부울은 0(거짓)과 1(참)로 변환됩니다.</p> <p>이 함수에서 none 상수를 반환하여 특정 조건에 대한 출력값을 무시할 수 있습니다. 즉, 조건을 충족하지 않는 데이터 포인트를 필터링할 수 있습니다. 자세한 정보는 데이터 포인트 필터링을 참조하세요.</p> <p>Example 예제</p> <ul style="list-style-type: none"> • if(0, x, y)에서는 변수 y를 반환합니다. • if(5, x, y)에서는 변수 x를 반환합니다. • if(gt(temp, 300), x, y)에서는 변수 temp가 300보다 크면 변수 x를 반환합니다. • if(gt(temp, 300), temp, none)에서는 300보다 크거나 같으면 변수 temp를 반환하고, temp가 300보다 작으면 none(값 없음)을 반환합니다. <p>하나 이상의 인수가 조건 함수인 중첩 조건 함수에는 UFCS를 사용하는 것이 좋습니다. if(condition, result_if_true)를 사용하여 조건을 평가하고 elif(condition, result_if_true, result_if_false)를 사용하여 추가 조건을 평가할 수 있습니다.</p> <p>예를 들어, if(condition1, result1_if_true, if(condition2, result2_if_true, result2_if_false)) 대신 if(condition1, result1_if_true).e</p>

함수	설명
	<p><code>lif(condition2, result2_if_true, result2_if_false)</code> 를 사용할 수 있습니다.</p> <p>중간 조건 함수를 추가로 연결할 수도 있습니다. 예를 들어, <code>if(condition1, result1_if_true).elif(condition2, result2_if_true).elif(condition3, result3_if_true, result3_if_false)</code> 와 같이 여러 <code>if</code> 문을 중첩하는 대신 <code>if(condition1, result1_if_true, if(condition2, result2_if_true, if(condition3, result3_if_true, result3_if_false)))</code> 를 사용할 수 있습니다.</p> <div data-bbox="829 898 1507 1163" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>⚠ Important</p> <p>UFCS와 함께 <code>elif(condition, result_if_true, result_if_false)</code> 를 사용해야 합니다.</p> </div>

수식 표현식에 문자열 함수 사용

[변환](#) 및 [지표](#)에서는 다음 함수를 사용하여 문자열에 대한 작업을 할 수 있습니다. 자세한 정보는 [공식에서 문자열 사용](#)을 참조하세요.

⚠ Important

공식 표현식은 실수 또는 문자열 값만 출력할 수 있습니다. 중첩 표현식은 문자열과 같은 다른 데이터 유형을 출력할 수 있지만 공식 전체는 숫자나 문자열로 평가되어야 합니다. [jp 함수](#)를 사용하여 문자열을 숫자로 변환할 수 있습니다. 부울 값은 1(참) 또는 0(거짓)이어야 합니다. 자세한 정보는 [정의되지 않은 값, 무한 값 및 오버플로우 값](#)을 참조하세요.

함수	설명
<code>len(s)</code>	문자열 <code>s</code> 의 길이를 반환합니다.
<code>find(s, substring)</code>	문자열 <code>s</code> 에 있는 문자열 <code>substring</code> 의 인덱스를 반환합니다.
<code>contains(s, substring)</code>	문자열 <code>s</code> 에 문자열 <code>substring</code> 가 포함되어 있으면 1을 반환하고, 그렇지 않으면 0을 반환합니다.
<code>upper(s)</code>	문자열 <code>s</code> 를 대문자 형태로 반환합니다.
<code>lower(s)</code>	문자열 <code>s</code> 를 소문자 형식으로 반환합니다.
<code>jp(s, json_path)</code>	<p>sJsonPath <code>json_path</code> 표현식으로 문자열을 평가하고 결과를 반환합니다.</p> <p>이 함수를 사용하여 다음을 수행합니다.</p> <ul style="list-style-type: none"> 직렬화된 JSON 구조에서 값, 배열 또는 객체를 추출합니다. 문자열을 숫자로 변환합니다. 예를 들어, 공식 <code>jp('111', '\$')</code>에서 숫자로 111을 반환합니다. <p>JSON 구조에서 문자열 값을 추출하여 숫자로 반환하려면 여러 중첩 <code>jp</code> 함수를 사용해야 합니다. 외부 <code>jp</code> 함수는 JSON 구조에서 문자열을 추출하고, 내부 <code>jp</code> 함수는 문자열을 숫자로 변환합니다.</p> <p>문자열 <code>json_path</code>는 문자열 리터럴을 포함해야 합니다. 즉, <code>json_path</code>는 문자열로 평가되는 표현식일 수 없습니다.</p>

함수	설명
	<p>Example 예제</p> <ul style="list-style-type: none"> • <code>jp({'status':"active","value":15}', '\$.value')</code> 에서 15를 반환합니다. • <code>jp({'measurement':{'reading':25,"confidence":0.95}}, '\$.measurement.reading')</code> 에서 25를 반환합니다. • <code>jp('[2,8,23]', '\$[2]')</code> 에서 23를 반환합니다. • <code>jp({'values':[3,6,7]}, '\$.values[1]')</code> 에서 6를 반환합니다. • <code>jp('111', '\$')</code>에서 111를 반환합니다. • <code>jp(jp({'measurement':{'reading':25,"confidence":"0.95"}}, '\$.measurement.confidence'), '\$')</code> 에서 0.95를 반환합니다.
<p><code>join(s0, s1, s2, s3, ...)</code></p>	<p>구분 기호와 함께 연결된 문자열을 반환합니다. 이 함수는 첫 번째 입력 문자열을 구분 기호로 사용하고 나머지 입력 문자열을 결합합니다. 이는 Java의 조인 (CharSequence 구분자, CharSequence... 요소) 함수와 비슷하게 동작합니다.</p> <p>Example 예제</p> <ul style="list-style-type: none"> • <code>join("-", "aa", "bb", "cc")</code>에서 <code>aa-bb-cc</code> 반환

함수	설명
<code>format(expression: "format")</code> 또는 <code>format("format", expression)</code>	<p>지정된 형식의 문자열을 반환합니다. 이 함수는 <code>expression</code> 값으로 평가한 다음 지정된 형식으로 값을 반환합니다. 이 함수는 Java의 형식 (문자열 형식, Object... args) 함수와 비슷하게 동작합니다. 지원되는 형식에 대한 자세한 내용은 Java 플랫폼, Standard Edition 7 API 사양의 클래스 포맷터에서 변환을 참조하세요.</p> <p>Example 예제</p> <ul style="list-style-type: none"> • <code>format(100+1: "d")</code> 에서 문자열 101을 반환합니다. • <code>format("The result is %d", 100+1)</code>에서 문자열 The result is 101을 반환합니다.

함수	설명
f'expression'	<p>연결된 문자열을 반환합니다. 이 형식이 지정된 함수를 사용하면 간단한 표현식을 사용하여 문자열을 연결하고 서식을 지정할 수 있습니다. 이러한 함수에는 중첩된 표현식이 포함될 수 있습니다. {}(중괄호)를 사용하여 표현식을 보간할 수 있습니다. 이것은 Python의 형식이 지정된 문자열 리터럴과 유사하게 동작합니다.</p> <p>Example 예제</p> <ul style="list-style-type: none"> f'abc{1+2: "f"}d' 에서 abc3.000000d 를 반환합니다. 이 예제 표현식을 평가하려면 다음을 수행하세요. <ol style="list-style-type: none"> format(1+2: "f") 에서 부동 소수점 숫자 3.000000를 반환합니다. join('', "abc", 1+2, 'd')에서 문자열 abc3.000000d 을 반환합니다. <pre>join('', "abc", format(1+2: "f"), 'd')</pre>와 같은 방법으로 표현식을 작성할 수도 있습니다.

공식 표현식에서 집계 함수 사용

[지표](#)에 한해, 다음 함수를 사용하여 각 시간 간격 동안 입력 값을 집계하고 단일 출력 값을 계산할 수 있습니다. 집계 함수는 연결된 자산의 데이터를 집계할 수 없습니다.

집계 함수 인수는 [변수](#), [숫자 리터럴](#), [시간 함수](#), 중첩 표현식 또는 집계 함수일 수 있습니다. 이 공식 `max(latest(x), latest(y), latest(z))`는 집계 함수를 인수로 사용하고 x, y, z 속성의 현재 최대 값을 반환합니다.

집계 함수에서 중첩 표현식을 사용할 수 있습니다. 중첩된 표현식을 사용할 때는 다음 규칙이 적용됩니다.

- 각 인수에는 변수가 하나만 있을 수 있습니다.

Example

예를 들어, $\text{avg}(x*(x-1))$ 및 $\text{sum}(x/2)/\text{avg}(y^2)$ 가 지원됩니다.

예를 들어, $\text{min}(x/y)$ 는 지원되지 않습니다.

- 각 인수에는 다단계 중첩 표현식이 있을 수 있습니다.

Example

예를 들어 $\text{sum}(\text{avg}(x^2)/2)$ 은 지원됩니다.

- 인수마다 변수가 다를 수 있습니다.

Example

예를 들어 $\text{sum}(x/2, y*2)$ 은 지원됩니다.

Note

- 표현식에 측정값이 포함된 경우 측정값의 현재 시간 간격 중 마지막 값을 AWS IoT SiteWise 사용하여 집계를 계산합니다.
- 표현식에 속성이 포함된 경우 속성의 최신 값을 AWS IoT SiteWise 사용하여 집계를 계산합니다.

함수	설명
$\text{avg}(x_0, \dots, x_n)$	현재 시간 간격 동안 해당 변수 값의 평균을 반환합니다. 이 함수는 지정된 변수에 현재 시간 간격 동안 하나 이상의 데이터 요소가 있는 경우에만 데이터 포인트를 출력합니다.
$\text{sum}(x_0, \dots, x_n)$	현재 시간 간격 동안 해당 변수 값의 합계를 반환합니다.

함수	설명
	이 함수는 지정된 변수에 현재 시간 간격 동안 하나 이상의 데이터 요소가 있는 경우에만 데이터 포인트를 출력합니다.
$\min(x_0, \dots, x_n)$	<p>현재 시간 간격 동안 해당 변수 값의 최소값을 반환합니다.</p> <p>이 함수는 지정된 변수에 현재 시간 간격 동안 하나 이상의 데이터 요소가 있는 경우에만 데이터 포인트를 출력합니다.</p>
$\max(x_0, \dots, x_n)$	<p>현재 시간 간격 동안 해당 변수 값의 최대값을 반환합니다.</p> <p>이 함수는 지정된 변수에 현재 시간 간격 동안 하나 이상의 데이터 요소가 있는 경우에만 데이터 포인트를 출력합니다.</p>
$\text{count}(x_0, \dots, x_n)$	<p>현재 시간 간격 동안 해당 변수에 대한 총 데이터 포인트 수를 반환합니다. 조건을 충족하는 데이터 포인트 수를 계산하는 방법에 대한 자세한 내용은 조건과 일치하는 데이터 포인트 계산 단원을 참조하세요.</p> <p>이 함수는 모든 시간 간격에 대한 데이터 포인트를 계산합니다.</p>
$\text{stdev}(x_0, \dots, x_n)$	<p>현재 시간 간격 동안 지정된 변수 값의 표준 편차를 반환합니다.</p> <p>이 함수는 지정된 변수에 현재 시간 간격 동안 하나 이상의 데이터 요소가 있는 경우에만 데이터 포인트를 출력합니다.</p>

공식 표현식에 임시 함수 사용

임시 함수를 사용하여 데이터 포인트의 타임스탬프를 기반으로 값을 반환합니다.

지표에서 시간 함수 사용

[지표](#)에서만 데이터 포인트의 타임스탬프를 기반으로 값을 반환하는 다음 함수를 사용할 수 있습니다.

시간 함수 인수는 로컬 자산 모델 또는 중첩 표현식의 속성이어야 합니다. 즉, 하위 자산 모델의 속성을 시간 함수에 사용할 수 없습니다.

시간 함수의 중첩 표현식을 사용할 수 있습니다. 중첩된 표현식을 사용할 때는 다음 규칙이 적용됩니다.

- 각 인수에는 변수가 하나만 있을 수 있습니다.

예를 들어 `latest(t*9/5 + 32)`은 지원됩니다.

- 인수는 집계 함수일 수 없습니다.

예를 들어 `first(sum(x))`은 지원되지 않습니다.

함수	설명
<code>first(x)</code>	현재 시간 간격 동안 해당 변수의 가장 오래된 타임스탬프가 있는 값을 반환합니다.
<code>last(x)</code>	현재 시간 간격 동안 해당 변수의 가장 최근 타임스탬프가 있는 값을 반환합니다.
<code>earliest(x)</code>	현재 시간 간격이 시작되기 전에 지정된 변수의 마지막 값을 반환합니다. 입력 속성의 기록에 데이터 포인트가 하나 이상 있는 경우 이 함수는 1시간 간격으로 데이터 포인트를 계산합니다. 세부 정보는 time-range-defintion 를 참조하세요.
<code>latest(x)</code>	주어진 변수의 마지막 값을 현재 시간 간격이 끝나기 전의 최신 타임스탬프와 함께 반환합니다. 입력 속성의 기록에 데이터 포인트가 하나 이상 있는 경우 이 함수는 1시간 간격으로 데이터

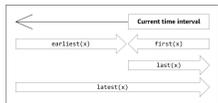
함수	설명
statetime(x)	<p>포인트를 계산합니다. 세부 정보는 time-range-defintion를 참조하세요.</p> <p>현재 시간 간격 동안 해당 변수가 양수인 시간(초)을 반환합니다. 비교 함수를 사용하면 statetime 함수가 사용할 변형 속성을 만들 수 있습니다.</p> <p>예를 들어 Idle 속성(0 또는 1)이 있는 경우, IdleTime = statetime(Idle) 표현식을 사용하여 시간 간격당 유휴 시간을 계산할 수 있습니다. 자세한 내용은 상태 시간 시나리오 예를 참조하세요.</p> <p>이 함수는 지표 속성을 입력 변수로 지원하지 않습니다.</p> <p>입력 속성의 기록에 데이터 포인트가 하나 이상 있는 경우 이 함수는 1시간 간격으로 데이터 포인트를 계산합니다.</p>

함수	설명
<p>TimeWeightedAvg(x, [interpolation])</p>	<p>포인트 간 시간 간격을 두고 가중치를 적용한 입력 데이터의 평균을 반환합니다. 계산 및 간격에 대한 자세한 내용은 시간 가중 함수 파라미터를 참조하세요.</p> <p>선택적 인수 <code>interpolation</code> 는 문자열 상수여야 합니다.</p> <ul style="list-style-type: none"> • <code>locf</code> - 이 값이 기본값입니다. 계산에서는 데이터 포인트 간 간격에 대해 마지막으로 관찰된 캐리 포워드 계산 알고리즘을 사용합니다. 이 접근 방식에서는 데이터 포인트가 다음 입력 데이터 포인트 타임스탬프까지 마지막으로 관측된 값으로 계산됩니다. <ul style="list-style-type: none"> 양호한 데이터 지점 이후의 값은 다음 데이터 포인트 타임스탬프까지의 값으로 추정됩니다. • <code>linear</code> - 계산에서는 데이터 포인트 간 간격에 대한 선형 보간 계산 알고리즘을 사용합니다. <ul style="list-style-type: none"> 양호한 두 데이터 포인트 사이의 값은 해당 데이터 포인트 값 간의 선형 보간으로 추정됩니다. 양호한 데이터 포인트와 불량 데이터 포인트 사이의 값 또는 마지막 양호한 데이터 포인트 이후의 값은 양호한 데이터 포인트로 추정됩니다.

함수	설명
<p><code>TimeWeightedStDev(x, [algo])</code></p>	<p>포인트 간 시간 간격으로 가중치를 적용한 입력 데이터의 표준 편차를 반환합니다.</p> <p>계산 및 간격에 대한 자세한 내용은 시간 가중 함수 파라미터를 참조하세요.</p> <p>계산에서는 데이터 포인트 간 간격에 대해 마지막으로 관찰된 캐리 포워드 계산 알고리즘을 사용합니다. 이 접근 방식에서는 데이터 포인트가 다음 입력 데이터 포인트 타임스탬프까지 마지막으로 관찰된 값으로 계산됩니다. 가중치는 데이터 포인트 또는 창 경계 사이의 시간 간격(초)으로 계산됩니다.</p> <p>선택적 인수 <code>algo</code>는 문자열 상수여야 합니다.</p> <ul style="list-style-type: none"> • <code>f</code> - 이 값이 기본값입니다. 주파수 가중치를 사용하여 편향 없는 가중치 표본 분산을 반환합니다. 여기서 값은 초 단위로 계산됩니다. <code>TimeWeight</code> 이 알고리즘은 일반적으로 표준 편차 이하로 가정되며 가중치 적용 표본에 대한 Bessel의 표준 편차 보정이라고 합니다. • <code>p</code> - 모집단 분산이라고도 하는 편향 가중 표본 분산을 반환합니다. <p>다음과 같은 경우 계산에 다음 공식이 사용됩니다.</p> <ul style="list-style-type: none"> • S_p = 모집단 표준 편차 • S_f = 빈도 표준 편차 • X_i = 수신 데이터 • ω_i = 시간 간격(초)과 동일한 가중치 • μ^* = 수신 데이터의 가중 평균

함수	설명
	<p>모집단 표준 편차 방정식:</p> $S_p^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i}$ <p>빈도 표준 편차 방정식:</p> $S_f^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i - 1}$

다음 다이어그램은 현재 시간 간격을 기준으로 시간 함수 first, latest, 및 earliest 를 AWS IoT SiteWise 계산하는 방법을 보여줍니다.



Note

- 의 시간 범위는 (현재 윈도우 시작 first(x), 현재 윈도우 종료] last(x) 입니다.
- 시간 범위는 (시작 시간, 현재 창 종료) 입니다. latest(x)
- 시간 범위는 (시작 시간, 이전 창 종료) 입니다. earliest(x)

시간 가중 함수 파라미터

집계 기간에 대해 계산된 시간 가중 함수는 다음 사항을 고려합니다.

- 기간 내부의 데이터 포인트
- 데이터 포인트 간 시간 간격
- 기간 전의 마지막 데이터 포인트
- 기간 뒤의 첫 번째 데이터 포인트(일부 알고리즘의 경우)

용어:

- 잘못된 데이터 포인트 - 품질이 좋지 않거나 숫자 값이 아닌 모든 데이터 포인트. 이는 기간 결과 계산에서 고려되지 않습니다.
- 잘못된 간격 - 잘못된 데이터 지점 이후의 간격입니다. 알려진 첫 번째 데이터 지점 이전의 간격도 잘못된 간격으로 간주됩니다.
- 양호한 데이터 포인트 - 품질이 우수하고 수치 값이 있는 모든 데이터 포인트.

Note

- AWS IoT SiteWise 변환과 GOOD 지표를 계산할 때만 품질 데이터를 사용합니다. UNCERTAIN 및 BAD 데이터 포인트를 무시합니다.
- 알려진 첫 번째 데이터 포인트 이전의 간격은 잘못된 간격으로 간주됩니다. 자세한 정보는 [the section called “공식 표현식 튜토리얼”](#)을 참조하세요.

마지막으로 알려진 데이터 포인트 이후의 간격은 무기한 계속되므로 이후의 모든 기간에 영향을 미칩니다. 새 데이터 포인트가 도착하면 함수는 간격을 다시 계산합니다.

위 규칙에 따라 기간 집계 결과가 계산되고 기간 경계로 제한됩니다. 기본적으로 함수는 전체 기간의 간격이 적절한 경우에만 기간 결과를 전송합니다.

기간의 적절한 간격이 기간 길이보다 작은 경우 함수는 기간을 전송하지 않습니다.

기간 결과에 영향을 미치는 데이터 포인트가 변경되면 함수는 데이터 포인트가 기간 외부에 있더라도 기간을 다시 계산합니다.

입력 속성의 기록에 데이터 요소가 하나 이상 있고 계산이 시작된 경우 함수는 시간 간격마다 시간 가중 집계 함수를 계산합니다.

Example 상태 시간 시나리오 예

다음 속성을 가진 자산이 있는 예를 고려하십시오.

- Idle - 0 또는 1인 측정입니다. 값이 1이면 머신이 유휴 상태입니다.
- Idle Time - 공식 `statetime(Idle)`를 사용하여 1분 간격으로 머신이 유휴 상태인 시간(초)을 계산하는 지표입니다.

Idle 속성은 다음과 같은 데이터 포인트를 갖습니다.

타임스탬프	2:00:00 PM	2:00:30 PM	2:01:15 PM	2:02:45 PM	2:04:00 PM
Idle	0	1	1	0	0

AWS IoT SiteWise 의 값에서 1분마다 Idle Time 속성을 계산합니다. Idle 이 계산이 완료되면 Idle Time 속성은 다음과 같은 데이터 포인트를 갖습니다.

타임스탬프	2:00:00 PM	2:01:00 PM	2:02:00 PM	2:03:00 PM	2:04:00 PM
Idle Time	N/A	30	60	45	0

AWS IoT SiteWise 1분이 Idle Time 끝날 때마다 다음과 같은 계산을 수행합니다.

- 2:00 PM(1:59 PM ~ 2:00 PM)
 - 2:00 PM 이전에는 Idle에 대한 데이터가 없으므로 데이터 포인트가 계산되지 않습니다.
- 2:01 PM(2:00 PM ~ 2:01 PM)
 - 2:00:00 PM에 머신이 활성 상태입니다(Idle이 0임).
 - 2:00:30 PM에 머신이 유휴 상태입니다(Idle가 1임).
 - Idle은 2:01:00 PM에 간격이 끝나기 전에 다시 변경되지 않으므로 Idle Time은 30초입니다.
- 2:02 PM(2:01 PM ~ 2:02 PM)
 - 2:01:00 PM에 머신이 유휴 상태입니다(2:00:30 PM의 마지막 데이터 포인트에 따라).
 - 2:01:15 PM에 머신이 유휴 상태입니다.
 - Idle은 2:02:00 PM에 간격이 끝나기 전에 다시 변경되지 않으므로 Idle Time은 60초입니다.
- 2:03 PM(2:02 PM ~ 2:03 PM)
 - 2:02:00 PM에 머신이 유휴 상태입니다(2:01:15 PM의 마지막 데이터 포인트에 따라).
 - 2:02:45 PM에 머신이 활성 상태입니다.
 - Idle은 2:03:00 PM에 간격이 끝나기 전에 다시 변경되지 않으므로 Idle Time은 45초입니다.
- 2:04 PM(2:03 PM ~ 2:04 PM)
 - 2:03:00 PM에 머신이 유휴 상태입니다(2:02:45 PM의 마지막 데이터 포인트에 따라).
 - Idle은 2:04:00 PM에 간격이 끝나기 전에 다시 변경되지 않으므로 Idle Time은 0초입니다.

Example 예제 TimeWeightedAvg 및 TimeWeightedStDev 시나리오

다음 표에는 이러한 1분 기간 지표 Avg(x), TimeWeightedAvg(x), TimeWeightedAvg(x, "linear"), stDev(x), timeWeightedStDev(x), timeWeightedStDev(x, 'p')에 대한 샘플 입력 및 출력이 나와 있습니다.

1분 집계 기간에 대한 샘플 입력:

Note

이러한 데이터 포인트는 모두 GOOD 품질이 우수합니다.

03:00:00	4.0
03:01:00	2.0
03:01:10	8.0
03:01:50	20.0
03:02:00	14.0
03:02:05	10.0
03:02:10	3.0
03:02:30	20.0
03:03:30	0.0

집계 결과 출력:

Note

없음 - 이 기간에 대한 결과가 생성되지 않았습니다.

Time	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:00:00	4	None	None	0	None	None
3:01:00	2	4	3	0	0	0
3:02:00	14	9	13	6	5.430610041581775	5.385164807134504
3:03:00	11	13	12.875	8.54400374531753	7.724054437220943	7.659416862050705
3:04:00	0	10	2.5	0	10.084389681792215	10
3:05:00	None	0	0	None	0	0

변환에서 시간 함수 사용

[변환](#)의 경우에만 `pretrigger()` 함수를 사용하여 현재 변환 계산을 시작한 속성 업데이트 이전에 변수에 대한 GOOD 품질 값을 검색할 수 있습니다.

제조업체가 기계 상태를 모니터링하는 데 사용하는 경우를 예로 들어 보겠습니다. AWS IoT SiteWise 제조업체는 다음 측정값과 변환을 사용하여 프로세스를 나타냅니다.

- 측정값 `current_state`는 0 또는 1이 될 수 있습니다.
 - 머신이 청소 상태인 경우, `current_state`는 1과 같습니다.
 - 머신이 제조 상태인 경우 `current_state`는 0과 같습니다.
- `if(pretrigger(current_state) == 1, timestamp(current_state) - timestamp(pretrigger(current_state)), none)`와 동일한 변환 `cleaning_state_duration`입니다. 이 변환은 머신이 청소 상태에 있었던 시간을 초 단위의 Unix epoch 형식으로 반환합니다. 자세한 내용은 [공식 표현식에 조건부 함수 사용](#) 및 [timestamp\(\)](#) 함수를 참조하세요.

머신이 예상보다 오래 청소 상태로 유지되는 경우 제조업체에서 머신을 조사할 수 있습니다.

또한 `pretrigger()` 함수를 다변량 변환에서 사용할 수 있습니다. 예를 들어, 이름이 x 및 y 인 측정값 두 개와 z 와 같은 변환 $x + y + \text{pretrigger}(y)$ 가 있다고 가정해 보겠습니다. 다음 표에는 오전 9시부터 오전 9시 15분까지의 x , y , z 에 대한 값이 나와 있습니다.

Note

- 이 예에서는 측정값이 시간순으로 도착한다고 가정합니다. 예를 들어 오전 9시의 x 값이 오전 9시 5분의 x 값보다 먼저 도착합니다.
- 오전 9시 5분의 데이터 포인트가 오전 9시의 데이터 포인트보다 먼저 도착하는 경우 z 는 오전 9시 5분에 계산되지 않습니다.
- 오전 9시 5분의 x 값이 오전 9시의 x 값보다 먼저 도달하고 y 값이 시간순으로 도달하면 오전 9시 5분에 z 는 $22 = 20 + 1 + 1$ 와 같습니다.

	오전 9시	오전 9시 5분	오전 9시 10분	오전 9시 15분
x	10	20		30
y	1	2	3	
$z = x + y + \text{pretrigger}(y)$	y 는 오전 9시 이전에는 데이터 포인트를 수신하지 않습니다. 따라서 오전 9시에 z 는 계산되지 않습니다.	$23 = 20 + 2 + 1$ $\text{pretrigger}(y)$ 는 1과 같습니다.	$25 = 20 + 3 + 2$ x 는 새 데이터 포인트를 받지 않습니다. $\text{pretrigger}(y)$ 는 2와 같습니다.	$36 = 30 + 3 + 3$ y 는 새 데이터 포인트를 받지 않습니다. 따라서 $\text{pretrigger}(y)$ 는 오전 9시 15분에는 3이 됩니다.

공식 표현식에 날짜 및 시간 함수 사용

[변환](#) 및 [지표](#)에서 날짜 및 시간 함수를 다음과 같은 방식으로 사용할 수 있습니다.

- 데이터 포인트의 현재 타임스탬프를 UTC 또는 현지 시간대로 검색합니다.

- year, month, day_of_month와 같은 인수를 사용하여 타임스탬프를 생성합니다.
- unix_time 인수를 사용하여 연 또는 월과 같은 시간 기간을 추출합니다.

함수	설명
now()	Unix epoch 형식으로 현재 날짜와 시간을 초 단위로 반환합니다.
timestamp()	<ul style="list-style-type: none"> • 변환에서 함수는 Unix epoch 형식으로 입력 메시지의 타임스탬프(초)를 반환합니다. <p>변환에서만 다음 중 하나를 수행할 수 있습니다.</p> <ul style="list-style-type: none"> • 변수를 함수에 인수로 제공합니다. 이 timestamp(<i>variable-name</i>) 함수는 Unix epoch 형식의 지정된 변수에 대한 최신 GOOD 품질 값의 타임스탬프(초)를 반환합니다. <p>예를 들어 각 온도 데이터 포인트를 섭씨에서 화씨로 변환하는 $9/5 * \text{Temperature_C}$ 공식을 사용하는 Temperature_F 라는 변환 속성이 자산에 있는 경우 timestamp(Temperature_F) 함수를 사용하여 Temperature_F 속성에 대한 최신 GOOD 품질 값의 타임스탬프를 가져올 수 있습니다.</p> <ul style="list-style-type: none"> • pretrigger() 함수를 함수의 인수로 사용합니다. timestamp(pretrigger(<i>variable-name</i>)) 함수는 Unix epoch 형식으로 현재 변환 계산을 시작한 속성 업데이트 이전의 지정된 변수에 대한 GOOD 품질 값의 타임스탬프(초)를 반환합니다. 자세한 정보는 변환에서 시간 함수 사용을 참조하세요.

함수	설명
	<ul style="list-style-type: none">지표에서 함수는 현재 기간 끝에서 검색된 타임스탬프(초)를 Unix epoch 형식으로 반환합니다.

함수	설명
<code>mktime(time_zone, year, month, day_of_month, hour, minute, second)</code>	<p>입력 시간을 Unix epoch 형식으로 초 단위로 반환합니다.</p> <p>이 함수 사용에 적용되는 요구 사항은 다음과 같습니다.</p> <ul style="list-style-type: none"> • 시간대 인수는 따옴표로 묶인 문자열('UTC') 이어야 합니다. 지정하지 않은 경우 기본 시간대는 UTC입니다. <p>시간대 인수는 첫 번째 또는 마지막 인수일 수 있습니다.</p> <ul style="list-style-type: none"> • 년, 월, 일, 시, 분, 두 번째 인수는 순서대로 입력해야 합니다. • 년, 월, 일 인수는 필수입니다. <p>이 함수 사용에는 다음 제한이 적용됩니다.</p> <ul style="list-style-type: none"> • year - 유효한 값은 1970~2250입니다. • month - 유효한 값은 1~12입니다. • day-of-month - 유효한 값은 1~31입니다. • hour - 유효한 값은 0~23입니다. • minute - 유효한 값은 0~59입니다. • second - 유효한 값은 0~60입니다. 부동 소수점 숫자일 수 있습니다. <p>예:</p> <ul style="list-style-type: none"> • <code>mktime(2020, 2, 29)</code> • <code>mktime('UTC+3', 2021, 12, 31, 22)</code> • <code>mktime(2022, 10, 13, 2, 55, 13.68, 'PST')</code>

함수	설명
<code>localtime(unix_time, time_zone)</code>	<p>Unix 시간을 기준으로 지정된 시간대의 연도, 월의 일, 요일, 연도의 일, 시간, 분 또는 초를 반환합니다.</p> <p>이 함수 사용에 적용되는 요구 사항은 다음과 같습니다.</p> <ul style="list-style-type: none"> 시간대 인수는 따옴표로 묶인 문자열('UTC')이어야 합니다. 지정하지 않은 경우 기본 시간대는 UTC입니다. Unix 시간 인수는 Unix epoch 형식의 시간(초)입니다. 유효한 범위는 1~31556889864403199입니다. 부동 소수점 숫자일 수 있습니다. <p>응답의 예: 2007-12-03T10:15:30+01:00[Europe/Paris]</p> <p><code>localtime(unix_time, time_zone)</code> 은 독립형 함수가 아닙니다. <code>year()</code>, <code>mon()</code>, <code>mday</code>, <code>wday()</code>, <code>yday()</code>, <code>hour()</code>, <code>minute()</code>, <code>sec()</code> 함수는 <code>localtime(unix_time, time_zone)</code> 를 인수로 사용합니다.</p> <p>예:</p> <ul style="list-style-type: none"> <code>year(localtime('GMT', 1605898608.8113723))</code> <code>now().localtime().year()</code> <code>timestamp().localtime('PST').year()</code> <code>localtime(1605289736, 'Europe/London').year()</code>
<code>year(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 연도를 반환합니다.

함수	설명
<code>mon(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 월을 반환합니다.
<code>mday(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 일을 반환합니다.
<code>wday(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 요일을 반환합니다.
<code>yday(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 연도의 일을 반환합니다.
<code>hour(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 시를 반환합니다.
<code>minute(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 분을 반환합니다.
<code>sec(localtime(unix_time, time_zone))</code>	<code>localtime(unix_time, time_zone)</code> 에서 초를 반환합니다.

지원되는 시간대 형식

다음과 같은 방법으로 시간대 인수를 지정할 수 있습니다.

- 시간대 오프셋 - UTC 또는 오프셋('+2' 또는 '-5')에 대해 'Z'를 지정합니다.
- 오프셋 ID - 시간대 약어와 오프셋을 결합합니다. 예: 'GMT+2' 및 'UTC-01:00'. 시간대 약어는 세 글자만 포함해야 합니다.
- 리전 기반 ID - 예: 'Etc/GMT+12' 및 'Pacific/Pago_Pago'.

지원되는 시간대 약어

날짜 및 시간 함수는 다음과 같은 세 글자로 된 시간대 약어를 지원합니다.

- EST - -05:00
- HST - -10:00

- MST - -07:00
- ACT - Australia/Darwin
- AET - Australia/Sydney
- AGT - America/Argentina/Buenos_Aires
- ART - Africa/Cairo
- AST - America/Anchorage
- BET - America/Sao_Paulo
- BST - Asia/Dhaka
- CAT - Africa/Harare
- CET - Europe/Paris
- CNT - America/St_Johns
- CST - America/Chicago
- CTT - Asia/Shanghai
- EAT - Africa/Addis_Ababa
- IET - America/Indiana/Indianapolis
- IST - Asia/Kolkata
- JST - Asia/Tokyo
- MIT - Pacific/Apia
- NET - Asia/Yerevan
- NST - Pacific/Auckland
- PLT - Asia/Karachi
- PRT - America/Puerto_Rico
- PST - America/Los_Angeles
- SST - Pacific/Guadalcanal
- VST - Asia/Ho_Chi_Minh

지원되는 지역 기반 ID

날짜 및 시간 함수는 UTC+ 00:00 과의 관계를 기준으로 구성된 다음과 같은 지역 기반 ID를 지원합니다.

- Etc/GMT+12(UTC-12:00)

- Pacific/Pago_Pago(UTC-11:00)
- Pacific/Samoa(UTC-11:00)
- Pacific/Niue(UTC-11:00)
- US/Samoa(UTC-11:00)
- Etc/GMT+11(UTC-11:00)
- Pacific/Midway(UTC-11:00)
- Pacific/Honolulu(UTC-10:00)
- Pacific/Rarotonga(UTC-10:00)
- Pacific/Tahiti(UTC-10:00)
- Pacific/Johnston(UTC-10:00)
- US/Hawaii(UTC-10:00)
- SystemV/HST10(UTC-10:00)
- Etc/GMT+10(UTC-10:00)
- Pacific/Marquesas(UTC-09:30)
- Etc/GMT+9(UTC-09:00)
- Pacific/Gambier(UTC-09:00)
- America/Atka(UTC-09:00)
- SystemV/YST9(UTC-09:00)
- America/Adak(UTC-09:00)
- US/Aleutian(UTC-09:00)
- Etc/GMT+8(UTC-08:00)
- US/Alaska(UTC-08:00)
- America/Juneau(UTC-08:00)
- America/Metlakatla(UTC-08:00)
- America/Yakutat(UTC-08:00)
- Pacific/Pitcairn(UTC-08:00)
- America/Sitka(UTC-08:00)
- America/Anchorage(UTC-08:00)
- SystemV/PST8(UTC-08:00)

- America/Nome(UTC-08:00)
- SystemV/YST9YDT(UTC-08:00)
- Canada/Yukon(UTC-07:00)
- US/Pacific-New(UTC-07:00)
- Etc/GMT+7(UTC-07:00)
- US/Arizona(UTC-07:00)
- America/Dawson_Creek(UTC-07:00)
- Canada/Pacific(UTC-07:00)
- PST8PDT(UTC-07:00)
- SystemV/MST7(UTC-07:00)
- America/Dawson(UTC-07:00)
- 멕시코/ (UTC-07:00) BajaNorte
- America/Tijuana(UTC-07:00)
- America/Creston(UTC-07:00)
- America/Hermosillo(UTC-07:00)
- America/Santa_Isabel(UTC-07:00)
- America/Vancouver(UTC-07:00)
- America/Ensenada(UTC-07:00)
- America/Phoenix(UTC-07:00)
- America/Whitehorse(UTC-07:00)
- America/Fort_Nelson(UTC-07:00)
- SystemV/PST8PDT(UTC-07:00)
- America/Los_Angele (UTC-07:00)
- US/Pacific(UTC-07:00)
- America/El_Salvador(UTC-06:00)
- America/Guatemala(UTC-06:00)
- America/Belize(UTC-06:00)
- America/Managua(UTC-06:00)
- America/Tegucigalpa(UTC-06:00)

- Etc/GMT+6(UTC-06:00)
- Pacific/Easter(UTC-06:00)
- 멕시코/ BajaSur (UTC-06:00)
- America/Regina(UTC-06:00)
- America/Denver(UTC-06:00)
- Pacific/Galapagos(UTC-06:00)
- America/Yellowknife(UTC-06:00)
- America/Swift_Current(UTC-06:00)
- America/Inuvik(UTC-06:00)
- America/Mazatlan(UTC-06:00)
- America/Boise(UTC-06:00)
- America/Costa_Rica(UTC-06:00)
- MST7MDT(UTC-06:00)
- SystemV/CST6(UTC-06:00)
- America/Chihuahua(UTC-06:00)
- America/Ojinaga(UTC-06:00)
- 칠레/ EasterIsland (UTC-06:00)
- US/Mountain(UTC-06:00)
- America/Edmonton(UTC-06:00)
- Canada/Mountain(UTC-06:00)
- America/Cambridge_Bay(UTC-06:00)
- Navajo(UTC-06:00)
- SystemV/MST7MDT(UTC-06:00)
- Canada/Saskatchewan(UTC-06:00)
- America/Shiprock(UTC-06:00)
- America/Panama(UTC-05:00)
- America/Chicago(UTC-05:00)
- America/Eirunepe(UTC-05:00)
- Etc/GMT+5(UTC-05:00)

- Mexico/General(UTC-05:00)
- America/Porto_Acre(UTC-05:00)
- America/Guayaquil(UTC-05:00)
- America/Rankin_Inlet(UTC-05:00)
- US/Central(UTC-05:00)
- America/Rainy_River(UTC-05:00)
- America/Indiana/Knox(UTC-05:00)
- America/North_Dakota/Beulah(UTC-05:00)
- America/Monterrey(UTC-05:00)
- America/Jamaica(UTC-05:00)
- America/Atikokan(UTC-05:00)
- America/Coral_Harbour(UTC-05:00)
- America/North_Dakota/Center(UTC-05:00)
- America/Cayman(UTC-05:00)
- America/Indiana/Tell_City(UTC-05:00)
- America/Mexico_City(UTC-05:00)
- America/Matamoros(UTC-05:00)
- CST6CDT(UTC-05:00)
- America/Knox_IN(UTC-05:00)
- America/Bogota(UTC-05:00)
- America/Menominee(UTC-05:00)
- America/Resolute(UTC-05:00)
- SystemV/EST5(UTC-05:00)
- Canada/Central(UTC-05:00)
- Brazil/Acre(UTC-05:00)
- America/Cancun(UTC-05:00)
- America/Lima(UTC-05:00)
- America/Bahia_Banderas(UTC-05:00)
- US/Indiana-Starke(UTC-05:00)

- America/Rio_Branco(UTC-05:00)
- SystemV/CST6CDT(UTC-05:00)
- Jamaica(UTC-05:00)
- America/Merida(UTC-05:00)
- America/North_Dakota/New_Salem(UTC-05:00)
- America/Winnipeg(UTC-05:00)
- America/Cuiaba(UTC-04:00)
- America/Marigot(UTC-04:00)
- America/Indiana/Petersburg(UTC-04:00)
- Chile/Continental(UTC-04:00)
- America/Grand_Turk(UTC-04:00)
- Cuba(UTC-04:00)
- Etc/GMT+4(UTC-04:00)
- America/Manaus(UTC-04:00)
- America/Fort_Wayne(UTC-04:00)
- America/St_Thomas(UTC-04:00)
- America/Anguilla(UTC-04:00)
- America/Havana(UTC-04:00)
- US/Michigan(UTC-04:00)
- America/Barbados(UTC-04:00)
- America/Louisville(UTC-04:00)
- America/Curacao(UTC-04:00)
- America/Guyana(UTC-04:00)
- America/Martinique(UTC-04:00)
- America/Puerto_Rico(UTC-04:00)
- America/Port_of_Spain(UTC-04:00)
- SystemV/AST4(UTC-04:00)
- America/Indiana/Vevay(UTC-04:00)
- America/Indiana/Vincennes(UTC-04:00)
- America/Kralendijk(UTC-04:00)

- America/Antigua(UTC-04:00)
- America/Indianapolis(UTC-04:00)
- America/Iqaluit(UTC-04:00)
- America/St_Vincent(UTC-04:00)
- America/Kentucky/Louisville(UTC-04:00)
- America/Dominica(UTC-04:00)
- America/Asuncion(UTC-04:00)
- EST5EDT(UTC-04:00)
- America/Nassa(UTC-04:00)
- America/Kentucky/Monticello(UTC-04:00)
- Brazil/West(UTC-04:00)
- America/Aruba(UTC-04:00)
- America/Indiana/Indianapolis(UTC-04:00)
- America/Santiago(UTC-04:00)
- America/La_Paz(UTC-04:00)
- America/Thunder_Bay(UTC-04:00)
- America/Indiana/Marengo(UTC-04:00)
- America/Blanc-Sablon(UTC-04:00)
- America/Santo_Domingo(UTC-04:00)
- US/Eastern(UTC-04:00)
- Canada/Eastern(UTC-04:00)
- America/Port-au-Prince(UTC-04:00)
- America/St_Barthelemy(UTC-04:00)
- America/Nipigon(UTC-04:00)
- US/East-Indiana(UTC-04:00)
- America/St_Lucia(UTC-04:00)
- America/Montserrat(UTC-04:00)
- America/Lower_Princes(UTC-04:00)
- America/Detroit(UTC-04:00)
- America/Tortola(UTC-04:00)

- America/Porto_Velho(UTC-04:00)
- America/Campo_Grande(UTC-04:00)
- America/Virgin(UTC-04:00)
- America/Pangnirtung(UTC-04:00)
- America/Montreal(UTC-04:00)
- America/Indiana/Winamac(UTC-04:00)
- America/Boa_Vista(UTC-04:00)
- America/Grenada(UTC-04:00)
- America/New_York(UTC-04:00)
- America/St_Kitts(UTC-04:00)
- America/Caracas(UTC-04:00)
- America/Guadeloupe(UTC-04:00)
- America/Toronto(UTC-04:00)
- SystemV/EST5EDT(UTC-04:00)
- America/Argentina/Catamarca(UTC-03:00)
- Canada/Atlantic(UTC-03:00)
- America/Argentina/Cordoba(UTC-03:00)
- America/Araguaina(UTC-03:00)
- America/Argentina/Salta(UTC-03:00)
- Etc/GMT+3(UTC-03:00)
- America/Montevideo(UTC-03:00)
- Brazil/Eas(UTC-03:00)
- America/Argentina/Mendoza(UTC-03:00)
- America/Argentina/Rio_Gallegos(UTC-03:00)
- America/Catamarca(UTC-03:00)
- America/Cordoba(UTC-03:00)
- America/Sao_Paulo(UTC-03:00)
- America/Argentina/Jujuy(UTC-03:00)
- America/Cayenne(UTC-03:00)
- America/Recife(UTC-03:00)

- America/Buenos_Aires(UTC-03:00)
- America/Paramaribo(UTC-03:00)
- America/Moncton(UTC-03:00)
- America/Mendoza(UTC-03:00)
- America/Santarem(UTC-03:00)
- Atlantic/Bermuda(UTC-03:00)
- America/Maceio(UTC-03:00)
- Atlantic/Stanley(UTC-03:00)
- America/Halifax(UTC-03:00)
- Antarctica/Rothera(UTC-03:00)
- America/Argentina/San_Luis(UTC-03:00)
- America/Argentina/Ushuaia(UTC-03:00)
- Antarctica/Palmer(UTC-03:00)
- America/Punta_Arenas(UTC-03:00)
- America/Glace_Bay(UTC-03:00)
- America/Fortaleza(UTC-03:00)
- America/Thule(UTC-03:00)
- America/Argentina/La_Rioja(UTC-03:00)
- America/Belem(UTC-03:00)
- America/Jujuy(UTC-03:00)
- America/Bahia(UTC-03:00)
- America/Goose_Bay(UTC-03:00)
- America/Argentina/San_Juan(UTC-03:00)
- 미국/아르헨티나/ ComodRivadavia (UTC-03:00)
- America/Argentina/Tucuman(UTC-03:00)
- America/Rosario(UTC-03:00)
- SystemV/AST4ADT(UTC-03:00)
- America/Argentina/Buenos_Aires(UTC-03:00)
- America/St_Johns(UTC-02:30)
- Canada/Newfoundland(UTC-02:30)

- America/Miquelon(UTC-02:00)
- Etc/GMT+2(UTC-02:00)
- America/Godthab(UTC-02:00)
- America/Noronha(UTC-02:00)
- 브라질/ DeNoronha (UTC-02:00)
- Atlantic/South_Georgia(UTC-02:00)
- Etc/GMT+1(UTC-01:00)
- Atlantic/Cape_Verde(UTC-01:00)
- Pacific/Kiritimati(UTC+14:00)
- Etc/GMT-14(UTC+14:00)
- Pacific/Fakaofu(UTC+13:00)
- Pacific/Enderbury(UTC+13:00)
- Pacific/Apia(UTC+13:00)
- Pacific/Tongatapu(UTC+13:00)
- Etc/GMT-13(UTC+13:00)
- NZ-CHAT(UTC+12:45)
- Pacific/Chatham(UTC+12:45)
- Pacific/Kwajalein(UTC+12:00)
- 남극/ McMurdo (UTC+ 12:00)
- Pacific/Wallis(UTC+12:00)
- Pacific/Fiji(UTC+12:00)
- Pacific/Funafuti(UTC+12:00)
- Pacific/Nauru(UTC+12:00)
- Kwajalein(UTC+12:00)
- NZ(UTC+12:00)
- Pacific/Wake(UTC+12:00)
- Antarctica/South_Pole(UTC+12:00)
- Pacific/Tarawa(UTC+12:00)
- Pacific/Auckland(UTC+12:00)
- Asia/Kamchatka(UTC+12:00)

- Etc/GMT-12(UTC+12:00)
- Asia/Anadyr(UTC+12:00)
- Pacific/Majuro(UTC+12:00)
- Pacific/Ponape(UTC+11:00)
- Pacific/Bougainville(UTC+11:00)
- Antarctica/Macquarie(UTC+11:00)
- Pacific/Pohnpei(UTC+11:00)
- Pacific/Efate(UTC+11:00)
- Pacific/Norfolk(UTC+11:00)
- Asia/Magadan(UTC+11:00)
- Pacific/Kosrae(UTC+11:00)
- Asia/Sakhalin(UTC+11:00)
- Pacific/Noumea(UTC+11:00)
- Etc/GMT-11(UTC+11:00)
- Asia/Srednekolymsk(UTC+11:00)
- Pacific/Guadalcanal(UTC+11:00)
- Australia/Lord_Howe(UTC+10:30)
- Australia/LHI(UTC+10:30)
- Australia/Hobart(UTC+10:00)
- Pacific/Yap(UTC+10:00)
- Australia/Tasmania(UTC+10:00)
- Pacific/Port_Moresby(UTC+10:00)
- Australia/ACT(UTC+10:00)
- Australia/Victoria(UTC+10:00)
- Pacific/Chuuk(UTC+10:00)
- Australia/Queensland(UTC+10:00)
- Australia/Canberra(UTC+10:00)
- Australia/Currie(UTC+10:00)
- Pacific/Guam(UTC+10:00)
- Pacific/Truk(UTC+10:00)

- Australia/NSW(UTC+10:00)
- Asia/Vladivostok(UTC+10:00)
- Pacific/Saipan(UTC+10:00)
- Antarctica/DumontDUrville(UTC+10:00)
- Australia/Sydney(UTC+10:00)
- Australia/Brisbane(UTC+10:00)
- Etc/GMT-10(UTC+10:00)
- Asia/Ust-Nera(UTC+10:00)
- Australia/Melbourne(UTC+10:00)
- Australia/Lindeman(UTC+10:00)
- Australia/North(UTC+09:30)
- Australia/Yancowinna(UTC+09:30)
- Australia/Adelaide(UTC+09:30)
- Australia/Broken_Hill(UTC+09:30)
- Australia/South(UTC+09:30)
- Australia/Darwin(UTC+09:30)
- Etc/GMT-9(UTC+09:00)
- Pacific/Palau(UTC+09:00)
- Asia/Chita(UTC+09:00)
- Asia/Dili(UTC+09:00)
- Asia/Jayapura(UTC+09:00)
- Asia/Yakutsk(UTC+09:00)
- Asia/Pyongyang(UTC+09:00)
- ROK(UTC+09:00)
- Asia/Seoul(UTC+09:00)
- Asia/Khandyga(UTC+09:00)
- Japan(UTC+09:00)
- Asia/Tokyo(UTC+09:00)
- Australia/Eucla(UTC+08:45)
- Asia/Kuching(UTC+08:00)

- Asia/Chungking(UTC+08:00)
- Etc/GMT-8(UTC+08:00)
- Australia/Perth(UTC+08:00)
- Asia/Macao(UTC+08:00)
- Asia/Maca(UTC+08:00)
- Asia/Choibalsan(UTC+08:00)
- Asia/Shanghai(UTC+08:00)
- Antarctica/Casey(UTC+08:00)
- Asia/Ulan_Bator(UTC+08:00)
- Asia/Chongqing(UTC+08:00)
- Asia/Ulaanbaatar(UTC+08:00)
- Asia/Taipei(UTC+08:00)
- Asia/Manila(UTC+08:00)
- PRC(UTC+08:00)
- Asia/Ujung_Pandang(UTC+08:00)
- Asia/Harbin(UTC+08:00)
- Singapore(UTC+08:00)
- Asia/Brunei(UTC+08:00)
- Australia/West(UTC+08:00)
- Asia/Hong_Kong(UTC+08:00)
- Asia/Makassar(UTC+08:00)
- Hongkong(UTC+08:00)
- Asia/Kuala_Lumpur(UTC+08:00)
- Asia/Irkutsk(UTC+08:00)
- Asia/Singapore(UTC+08:00)
- Asia/Pontianak(UTC+07:00)
- Etc/GMT-7(UTC+07:00)
- Asia/Phnom_Penh(UTC+07:00)
- Asia/Novosibirsk(UTC+07:00)
- Antarctica/Davis(UTC+07:00)

- Asia/Tomsk(UTC+07:00)
- Asia/Jakarta(UTC+07:00)
- Asia/Barnaul(UTC+07:00)
- Indian/Christmas(UTC+07:00)
- Asia/Ho_Chi_Minh(UTC+07:00)
- Asia/Hovd(UTC+07:00)
- Asia/Bangkok(UTC+07:00)
- Asia/Vientiane(UTC+07:00)
- Asia/Novokuznetsk(UTC+07:00)
- Asia/Krasnoyarsk(UTC+07:00)
- Asia/Saigon(UTC+07:00)
- Asia/Yangon(UTC+06:30)
- Asia/Rangoon(UTC+06:30)
- Indian/Cocos(UTC+06:30)
- Asia/Kashgar(UTC+06:00)
- Etc/GMT-6(UTC+06:00)
- Asia/Almaty(UTC+06:00)
- Asia/Dacca(UTC+06:00)
- Asia/Omsk(UTC+06:00)
- Asia/Dhaka(UTC+06:00)
- Indian/Chagos(UTC+06:00)
- Asia/Qyzylorda(UTC+06:00)
- Asia/Bishkek(UTC+06:00)
- Antarctica/Vostok(UTC+06:00)
- Asia/Urumqi(UTC+06:00)
- Asia/Thimbu(UTC+06:00)
- Asia/Thimphu(UTC+06:00)
- Asia/Kathmandu(UTC+05:45)
- Asia/Katmandu(UTC+05:45)
- Asia/Kolkata(UTC+05:30)

- Asia/Colombo(UTC+05:30)
- Asia/Calcutta(UTC+05:30)
- Asia/Aqtau(UTC+05:00)
- Etc/GMT-5(UTC+05:00)
- Asia/Samarkand(UTC+05:00)
- Asia/Karachi(UTC+05:00)
- Asia/Yekaterinburg(UTC+05:00)
- Asia/Dushanbe(UTC+05:00)
- Indian/Maldives(UTC+05:00)
- Asia/Oral(UTC+05:00)
- Asia/Tashkent(UTC+05:00)
- Antarctica/Mawson(UTC+05:00)
- Asia/Aqtobe(UTC+05:00)
- Asia/Ashkhabad(UTC+05:00)
- Asia/Ashgabat(UTC+05:00)
- Asia/Atyrau(UTC+05:00)
- Indian/Kerguelen(UTC+05:00)
- Iran(UTC+04:30)
- Asia/Tehran(UTC+04:30)
- Asia/Kabul(UTC+04:30)
- Asia/Yerevan(UTC+04:00)
- Etc/GMT-4(UTC+04:00)
- Etc/GMT-4(UTC+04:00)
- Asia/Dubai(UTC+04:00)
- Indian/Reunion(UTC+04:00)
- Europe/Saratov(UTC+04:00)
- Europe/Samara(UTC+04:00)
- Indian/Mahe(UTC+04:00)
- Asia/Baku(UTC+04:00)
- Asia/Muscat(UTC+04:00)

- Europe/Volgograd(UTC+04:00)
- Europe/Astrakhan(UTC+04:00)
- Asia/Tbilisi(UTC+04:00)
- Europe/Ulyanovsk(UTC+04:00)
- Asia/Aden(UTC+03:00)
- Africa/Nairobi(UTC+03:00)
- Europe/Istanbul(UTC+03:00)
- Etc/GMT-3(UTC+03:00)
- Europe/Zaporozhye(UTC+03:00)
- Israel(UTC+03:00)
- Indian/Comoro(UTC+03:00)
- Antarctica/Syowa(UTC+03:00)
- Africa/Mogadishu(UTC+03:00)
- Europe/Bucharest(UTC+03:00)
- Africa/Asmera(UTC+03:00)
- Europe/Mariehamn(UTC+03:00)
- Asia/Istanbul(UTC+03:00)
- Europe/Tiraspol(UTC+03:00)
- Europe/Moscow(UTC+03:00)
- Europe/Chisinau(UTC+03:00)
- Europe/Helsinki(UTC+03:00)
- Asia/Beirut(UTC+03:00)
- Asia/Tel_Aviv(UTC+03:00)
- Africa/Djibouti(UTC+03:00)
- Europe/Simferopol(UTC+03:00)
- Europe/Sofia(UTC+03:00)
- Asia/Gaza(UTC+03:00)
- Africa/Asmara(UTC+03:00)
- Europe/Riga(UTC+03:00)
- Asia/Baghdad(UTC+03:00)

- Asia/Damascus(UTC+03:00)
- Africa/Dar_es_Salaam(UTC+03:00)
- Africa/Addis_Ababa(UTC+03:00)
- Europe/Uzhgorod(UTC+03:00)
- Asia/Jerusalem(UTC+03:00)
- Asia/Riyadh(UTC+03:00)
- Asia/Kuwait(UTC+03:00)
- Europe/Kirov(UTC+03:00)
- Africa/Kampala(UTC+03:00)
- Europe/Minsk(UTC+03:00)
- Asia/Qatar UTC+03:00)
- Europe/Kiev(UTC+03:00)
- Asia/Bahrain(UTC+03:00)
- Europe/Vilnius(UTC+03:00)
- Indian/Antananarivo(UTC+03:00)
- Indian/Mayotte(UTC+03:00)
- Europe/Tallinn(UTC+03:00)
- Turkey(UTC+03:00)
- Africa/Juba(UTC+03:00)
- Asia/Nicosia(UTC+03:00)
- Asia/Famagusta(UTC+03:00)
- W-SU(UTC+03:00)
- EET(UTC+03:00)
- Asia/Hebron(UTC+03:00)
- Asia/Amman(UTC+03:00)
- Europe/Nicosia(UTC+03:00)
- Europe/Athens(UTC+03:00)
- Africa/Cairo(UTC+02:00)
- Africa/Mbabane(UTC+02:00)
- Europe/Brussels(UTC+02:00)

- Europe/Warsaw(UTC+02:00)
- CET(UTC+02:00)
- Europe/Luxembourg(UTC+02:00)
- Etc/GMT-2(UTC+02:00)
- Libya(UTC+02:00)
- Africa/Kigali(UTC+02:00)
- Africa/Tripoli(UTC+02:00)
- Europe/Kaliningrad(UTC+02:00)
- Africa/Windhoek(UTC+02:00)
- Europe/Malta(UTC+02:00)
- Europe/Busingen(UTC+02:00)
-
- Europe/Skopje(UTC+02:00)
- Europe/Sarajevo(UTC+02:00)
- Europe/Rome(UTC+02:00)
- Europe/Zurich(UTC+02:00)
- Europe/Gibraltar(UTC+02:00)
- Africa/Lubumbashi(UTC+02:00)
- Europe/Vaduz(UTC+02:00)
- Europe/Ljubljana(UTC+02:00)
- Europe/Berlin(UTC+02:00)
- Europe/Stockholm(UTC+02:00)
- Europe/Budapest(UTC+02:00)
- Europe/Zagreb(UTC+02:00)
- Europe/Paris(UTC+02:00)
- Africa/Ceuta(UTC+02:00)
- Europe/Prague(UTC+02:00)
- Antarctica/Troll(UTC+02:00)
- Africa/Gaborone(UTC+02:00)
- Europe/Copenhagen(UTC+02:00)

- Europe/Vienna(UTC+02:00)
- Europe/Tirane(UTC+02:00)
- MET(UTC+02:00)
- Europe/Amsterdam(UTC+02:00)
- Africa/Maputo(UTC+02:00)
- Europe/San_Marino(UTC+02:00)
- Poland(UTC+02:00)
- Europe/Andorra(UTC+02:00)
- Europe/Oslo(UTC+02:00)
- Europe/Podgorica(UTC+02:00)
- Africa/Bujumbura(UTC+02:00)
- Atlantic/Jan_Mayen(UTC+02:00)
- Africa/Maseru(UTC+02:00)
- Europe/Madrid(UTC+02:00)
- Africa/Blantyre(UTC+02:00)
- Africa/Lusaka(UTC+02:00)
- Africa/Harare(UTC+02:00)
- Africa/Khartoum(UTC+02:00)
- Africa/Johannesburg(UTC+02:00)
- Europe/Belgrade(UTC+02:00)
- Europe/Bratislava(UTC+02:00)
- Arctic/Longyearbyen(UTC+02:00)
- Egypt(UTC+02:00)
- Europe/Vatican(UTC+02:00)
- Europe/Monaco(UTC+02:00)
- Europe/London(UTC+01:00)
- Etc/GMT-1(UTC+01:00)
- Europe/Jersey(UTC+01:00)
- Europe/Guernsey(UTC+01:00)
- Europe/Isle_of_Man(UTC+01:00)

- Africa/Tunis(UTC+01:00)
- Africa/Malabo(UTC+01:00)
- GB-Eire(UTC+01:00)
- Africa/Lagos(UTC+01:00)
- Africa/Algiers(UTC+01:00)
- GB(UTC+01:00)
- Portugal(UTC+01:00)
- Africa/Sao_Tome(UTC+01:00)
- Africa/Ndjamena(UTC+01:00)
- Atlantic/Faeroe(UTC+01:00)
- Eire(UTC+01:00)
- Atlantic/Faroe(UTC+01:00)
- Europe/Dublin(UTC+01:00)
- Africa/Libreville(UTC+01:00)
- Africa/EI_Aaiun(UTC+01:00)
- Africa/EI_Aaiun(UTC+01:00)
- Africa/Douala(UTC+01:00)
- Africa/Brazzaville(UTC+01:00)
- Africa/Porto-Novo(UTC+01:00)
- Atlantic/Madeira(UTC+01:00)
- Europe/Lisbon(UTC+01:00)
- Atlantic/Canary(UTC+01:00)
- Africa/Casablanca(UTC+01:00)
- Europe/Belfast(UTC+01:00)
- Africa/Luanda(UTC+01:00)
- Africa/Kinshasa(UTC+01:00)
- Africa/Bangui(UTC+01:00)
- WET(UTC+01:00)
- Africa/Niamey(UTC+01:00)
- GMT(UTC+00:00)

- Etc/GMT-0(UTC+00:00)
- Atlantic/St_Helena(UTC+00:00)
- Etc/GMT+0(UTC+00:00)
- Africa/Banjul(UTC+00:00)
- Etc/GMT(UTC+00:00)
- Africa/Freetown(UTC+00:00)
- Africa/Bamako(UTC+00:00)
- Africa/Conakry(UTC+00:00)
- Universal(UTC+00:00)
- Africa/Nouakchott(UTC+00:00)
- UTC(UTC+00:00)
- Etc/Universal(UTC+00:00)
- Atlantic/Azores(UTC+00:00)
- Africa/Abidjan(UTC+00:00)
- Africa/Accra(UTC+00:00)
- Etc/UCT(UTC+00:00)
- GMT0(UTC+00:00)
- Zulu(UTC+00:00)Zulu (UTC+00:00)
- Africa/Ouagadougou(UTC+00:00)
- Atlantic/Reykjavik(UTC+00:00)
- Etc/Zulu(UTC+00:00)
- Iceland(UTC+00:00)
- Africa/Lome(UTC+00:00)
- Greenwich(UTC+00:00)
- Etc/GMT0(UTC+00:00)
- America/Danmarkshavn UTC+00:00)
- Africa/Dakar(UTC+00:00)
- Africa/Bissau(UTC+00:00)
- Etc/Greenwich(UTC+00:00)
- Africa/Timbuktu(UTC+00:00)

- UCT(UTC+00:00)
- Africa/Monrovia(UTC+00:00)
- Etc/UTC(UTC+00:00)

공식 표현식 튜토리얼

이 튜토리얼을 따라 AWS IoT SiteWise에서 공식 표현식을 사용할 수 있습니다.

주제

- [공식에서 문자열 사용](#)
- [데이터 포인트 필터링](#)
- [조건과 일치하는 데이터 포인트 계산](#)
- [공식의 지연 데이터](#)
- [공식의 데이터 품질](#)
- [정의되지 않은 값, 무한 값 및 오버플로우 값](#)

공식에서 문자열 사용

공식 표현식의 문자열에 대해 작업을 할 수 있습니다. 속성 및 측정 속성을 참조하는 변수에서 문자열을 입력할 수도 있습니다.

Important

공식 표현식은 실수 또는 문자열 값만 출력할 수 있습니다. 중첩 표현식은 문자열과 같은 다른 데이터 유형을 출력할 수 있지만 공식 전체는 숫자나 문자열로 평가되어야 합니다. [jp 함수](#)를 사용하여 문자열을 숫자로 변환할 수 있습니다. 부울 값은 1(참) 또는 0(거짓)이어야 합니다. 자세한 정보는 [정의되지 않은 값, 무한 값 및 오버플로우 값](#)을 참조하세요.

AWS IoT SiteWise 문자열 연산에 사용할 수 있는 다음과 같은 수식 표현식 기능을 제공합니다.

- [문자열 리터럴](#)
- [인덱스 연산자](#)(s[index])
- [조각 연산자](#)(s[start:end:step])
- [비교 함수](#)는 문자열을 [사전순으로](#) 비교하는 데 사용할 수 있습니다.

- 직렬화된 JSON 객체를 파싱하고 [문자열](#) 숫자로 변환할 수 있는 `jp` 함수를 포함하는 문자열 함수

데이터 포인트 필터링

[if 함수](#)를 사용하여 조건을 충족하지 않는 데이터 요소를 필터링할 수 있습니다. `if` 함수는 조건을 평가하여 `true` 및 `false` 결과에 대해 서로 다른 값을 반환합니다. [none 상수](#)를 `if` 함수의 한 케이스에 대한 출력값으로 사용하여 해당 케이스의 데이터 포인트를 버릴 수 있습니다.

조건과 일치하는 데이터 포인트를 필터링하려면

- `if` 함수를 사용하여 조건이 충족되는지 확인하는 조건을 정의하고 `result_if_true` 또는 `result_if_false` 값으로 `none`를 반환하는 변환을 만드세요.

Example 예: 물이 끓지 않는 데이터 포인트 필터링

머신에 있는 물의 온도(섭씨)를 제공하는 측정값(`temp_c`)이 있는 시나리오를 생각해 보세요. 다음 변환을 정의하여 물이 끓지 않는 데이터 포인트를 필터링할 수 있습니다.

- 변환: `boiling_temps = if(gte(temp_c, 100), temp_c, none)` - 섭씨 100도 이상이면 온도를 반환하고, 그렇지 않으면 데이터 포인트를 반환하지 않습니다.

조건과 일치하는 데이터 포인트 계산

[비교 함수](#)와 [sum\(\)](#)을 사용하여 조건이 참인 데이터 포인트 수를 계산할 수 있습니다.

조건과 일치하는 데이터 포인트를 계산하려면

1. 비교 함수를 사용하여 다른 속성에 대한 필터 조건을 정의하는 변환을 생성합니다.
2. 해당 조건이 충족되는 데이터 포인트를 합산하는 지표를 생성합니다.

Example 예: 물이 끓는 데이터 포인트 수 계산

머신에 있는 물의 온도(섭씨)를 제공하는 측정값(`temp_c`)이 있는 시나리오를 생각해 보세요. 다음과 같은 변환 및 지표 속성을 정의하여 물이 끓는 데이터 포인트 수를 계산할 수 있습니다.

- 변환: `is_boiling = gte(temp_c, 100)` - 온도가 섭씨 100도보다 높거나 같으면 1을 반환하고, 그렇지 않으면 0을 반환합니다.
- 지표: `boiling_count = sum(is_boiling)` - 물이 끓는 데이터 포인트 수를 계산합니다.

공식의 지연 데이터

AWS IoT SiteWise 최대 7일이 지난 데이터의 늦은 데이터 수집을 지원합니다. 지연 데이터를 AWS IoT SiteWise 수신하면 과거 기간의 지연 데이터를 입력하는 모든 지표에 대해 기존 값을 다시 계산합니다. 이러한 재계산으로 인해 데이터 처리 요금이 발생합니다.

Note

지연 데이터를 AWS IoT SiteWise 입력하는 속성을 계산할 때는 각 속성의 현재 수식 표현식을 사용합니다.

지표의 과거 창을 다시 AWS IoT SiteWise 계산한 후에는 해당 창의 이전 값을 대체합니다. 해당 지표에 대한 알림을 활성화한 경우 속성 값 AWS IoT SiteWise 알림도 발생합니다. 즉, 이전에 알림을 받은 동일한 속성 및 타임스탬프에 대해 새 속성 값 업데이트 알림을 받을 수 있습니다. 애플리케이션이나 데이터 레이크가 속성 값 알림을 사용하는 경우, 데이터가 정확하도록 이전 값을 새 값으로 업데이트해야 합니다.

공식의 데이터 품질

에서 AWS IoT SiteWise 각 데이터 포인트에는 품질 코드가 있으며, 품질 코드는 다음 중 하나일 수 있습니다.

- GOOD – 데이터가 어떤 문제의 영향도 받지 않습니다.
- BAD – 데이터가 센서 고장과 같은 문제의 영향을 받습니다.
- UNCERTAIN – 데이터가 센서 부정확과 같은 문제의 영향을 받습니다.

AWS IoT SiteWise 변환과 지표를 계산할 때는 GOOD 품질 데이터만 사용합니다. AWS IoT SiteWise 성공적인 계산을 위한 GOOD 품질 데이터만 출력합니다. 계산이 실패하면 해당 계산에 대한 데이터 포인트를 출력하지 AWS IoT SiteWise 않습니다. 이 문제는 계산의 결과로 정의되지 않은 값, 무한 값 또는 오버플로우 값이 나오는 경우에 발생할 수 있습니다.

데이터를 쿼리하고 데이터 품질별로 필터링하는 방법에 대한 자세한 내용은 [에서 데이터 쿼리 AWS IoT SiteWise](#) 단원을 참조하세요.

정의되지 않은 값, 무한 값 및 오버플로우 값

일부 공식 표현식 (예: $x / \sqrt{-1}$, 또는 $\log(0)$) 은 실수 체계에서 정의되지 않았거나 무한하거나 에서 지원하는 범위를 벗어난 값을 계산합니다. AWS IoT SiteWise 자산 속성의 표현식이 정의되

지 않은 값, 무한 값 또는 오버플로 값을 계산하는 경우 해당 계산에 대한 데이터 포인트가 AWS IoT SiteWise 출력되지 않습니다.

AWS IoT SiteWise 또한 수식 표현식의 결과로 숫자가 아닌 값을 계산하는 경우 데이터 포인트를 출력하지 않습니다. 즉, 문자열, 배열 또는 [none 상수](#)를 계산하는 공식을 정의하면 해당 계산에 대한 데이터 포인트가 AWS IoT SiteWise 에서 출력되지 않습니다.

Example 예제

다음 각 수식 표현식은 숫자로 표현할 AWS IoT SiteWise 수 없는 값을 생성합니다. AWS IoT SiteWise 이러한 수식 표현식을 계산할 때 데이터 포인트를 출력하지 않습니다.

- $x / 0$ 은 정의되지 않았습니다.
- $\log(0)$ 은 정의되지 않았습니다.
- $\text{sqrt}(-1)$ 는 실수 시스템에서는 정의되지 않습니다.
- "hello" + " world"는 문자열입니다.
- `jp({'values':[3,6,7]}, '$.values')`는 배열입니다.
- temp가 300보다 작을 때 `if(gte(temp, 300), temp, none)`는 none입니다.

사용자 지정 복합 모델 생성 (구성 요소)

사용자 지정 복합 모델 또는 구성 요소 (콘솔을 사용하는 경우)는 자산 모델 및 구성 요소 모델을 위한 또 다른 수준의 구성을 제공합니다. 속성을 그룹화하거나 다른 모델을 참조하여 모델을 구조화하는 데 사용할 수 있습니다. 사용자 지정 복합 모델 작업에 대한 자세한 내용은 [사용자 지정 복합 모델 \(구성 요소\)](#)를 참조하십시오.

기존 자산 모델 또는 구성 요소 모델 내에 사용자 지정 복합 모델을 생성합니다. 사용자 지정 복합 모델에는 두 가지 유형이 있습니다. 모델 내에서 관련 속성을 그룹화하려면 인라인 사용자 지정 복합 모델을 만들 수 있습니다. 자산 모델 또는 구성 요소 모델 내에서 구성 요소 모델을 참조하려면 구성 요소 모델 기반 사용자 지정 복합 모델을 만들 수 있습니다.

다음 섹션에서는 AWS IoT SiteWise API를 사용하여 사용자 지정 복합 모델을 만드는 방법을 설명합니다.

주제

- [인라인 컴포넌트 생성 \(콘솔\)](#)
- [인라인 사용자 지정 복합 모델 생성 \(AWS CLI\)](#)
- [component-model-based 구성 요소 생성 \(콘솔\)](#)

- [component-model-based 사용자 지정 복합 모델 생성 \(AWS CLI\)](#)

인라인 컴포넌트 생성 (콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자체 속성을 정의하는 인라인 구성 요소를 만들 수 있습니다.

Note

인라인 구성 요소이므로 이러한 속성은 현재 자산 모델에만 적용되며 다른 곳에서는 공유되지 않습니다.

재사용 가능한 모델을 만들어야 하는 경우 (예: 여러 자산 모델 간에 공유하거나 한 자산 모델 내에 여러 인스턴스를 포함하려는 경우) 대신 구성 요소 모델을 기반으로 구성 요소를 만들어야 합니다. 자세한 내용은 다음 섹션을 참조하십시오.

구성 요소를 만들려면 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 구성 요소를 추가할 에셋 모델을 선택합니다.
4. 속성 탭에서 구성 요소를 선택합니다.
5. 구성 요소 생성을 선택합니다.
6. 구성 요소 생성 페이지에서 다음을 수행합니다.
 - a. 구성 요소 이름 (예: **ServoMotor** 또는) 을 입력합니다 **ServoMotor Model**. 이 이름은 이 지역 내 계정의 모든 구성 요소에서 고유해야 합니다.
 - b. (선택 사항) 모델의 속성 정의를 추가합니다. 속성은 거의 변경되지 않는 정보를 나타냅니다. 자세한 정보는 [정적 데이터\(속성\) 정의](#)을 참조하세요.
 - c. (선택 사항) 모델의 측정 정의를 추가합니다. 측정값은 장비의 데이터 스트림을 나타냅니다. 자세한 정보는 [장비의 데이터 스트림 정의\(측정값\)](#)을 참조하세요.
 - d. (선택 사항) 모델의 변환 정의를 추가합니다. 변환은 데이터를 한 양식에서 다른 양식으로 매핑하는 공식입니다. 자세한 정보는 [데이터 변환\(변환\)](#)을 참조하세요.
 - e. (선택 사항) 모델의 지표 정의를 추가합니다. 지표는 일정 간격으로 데이터를 집계하는 공식입니다. 지표는 연결된 자산에서 데이터를 입력할 수 있으므로 작업 또는 작업의 하위 집합을 나타내는 값을 계산할 수 있습니다. 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)을 참조하세요.

- f. 구성 요소 생성을 선택합니다.

인라인 사용자 지정 복합 모델 생성 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자체 속성을 정의하는 인라인 사용자 지정 복합 모델을 만들 수 있습니다.

[CreateAssetModelComposite](#) 모델을 사용하여 속성이 있는 인라인 모델을 만들 수 있습니다. 이 작업에는 다음과 같은 구조의 페이로드가 필요합니다.

Note

이 모델은 인라인 복합 모델이므로 이러한 속성은 현재 자산 모델에만 적용되며 다른 곳에서는 공유되지 않습니다. 필드를 “인라인”으로 만드는 이유는 필드에 값을 제공하지 않기 때문입니다. `composedAssetModelId` 재사용 가능한 모델을 생성해야 하는 경우 (예: 여러 자산 모델 간에 공유하거나 하나의 자산 모델 내에 여러 인스턴스를 포함하려는 경우) 구성 요소 모델 기반 복합 모델을 대신 만들어야 합니다. 자세한 내용은 다음 섹션을 참조하십시오.

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "assetModelCompositeModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

```
    ]
  }
```

component-model-based 구성 요소 생성 (콘솔)

AWS IoT SiteWise 콘솔을 사용하여 구성 요소 모델을 기반으로 구성 요소를 만들 수 있습니다.

component-model-based 구성 요소를 만들려면 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 구성 요소를 추가할 에셋 모델을 선택합니다.
4. 속성 탭에서 구성 요소를 선택합니다.
5. 구성 요소 생성을 선택합니다.
6. 구성 요소 생성 페이지에서 다음을 수행합니다.
 - a. 구성 요소의 기반으로 사용할 구성 요소 모델을 선택합니다.
 - b. 구성 요소 이름 (예: **ServoMotor** 또는) 을 입력합니다 **ServoMotor Model**. 이 이름은 이 지역 내 계정의 모든 구성 요소에서 고유해야 합니다.
 - c. 구성 요소 생성을 선택합니다.

component-model-based 사용자 지정 복합 모델 생성 (AWS CLI)

를 사용하여 자산 모델 내에 component-model-based 사용자 지정 복합 모델을 만들 수 있습니다. AWS CLI component-model-based 사용자 지정 복합 모델은 다른 곳에서 이미 정의한 구성 요소 모델에 대한 참조입니다.

[CreateAssetModelComposite](#) **모델** 작업을 사용하여 component-model-based 사용자 지정 복합 모델을 만들 수 있습니다. 이 작업에는 다음과 같은 구조의 페이로드가 필요합니다.

Note

이 예제에서 의 `composedAssetModelId` 값은 기존 구성 요소 모델의 자산 모델 ID 또는 외부 ID입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오. 구성 요소 모델을 만드는 방법에 대한 예는 [을 참조하십시오](#) [구성 요소 모델 생성 \(AWS CLI\)](#).

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "composedAssetModelId": component model ID
}
```

component-model-based 사용자 지정 복합 모델은 참조일 뿐이므로 이름 외에는 고유한 속성이 없습니다.

동일한 컴포넌트의 여러 인스턴스를 자산 모델에 추가하려는 경우 (예: 서보 모터가 여러 개 있는 CNC 기계), 각각 고유한 이름을 갖지만 모두 동일한 참조를 갖는 component-model-based 사용자 정의 복합 모델을 여러 개 추가할 수 composedAssetModelId 있습니다.

부품을 다른 부품 안에 중첩할 수 있습니다. 이렇게 하려면 이 예제와 같이 component-model-based 복합 모델을 구성 요소 모델 중 하나에 추가할 수 있습니다.

자산 만들기

자산 모델에서 자산을 생성할 수 있습니다. 자산을 생성하려면 먼저 자산 모델이 있어야 합니다. 자산 모델을 생성하지 않은 경우 [자산 모델 생성](#) 단원을 참조하세요.

Note

ACTIVE 모델에서만 자산을 만들 수 있습니다. 모델의 상태가 ACTIVE가 아닌 경우 해당 모델에서 자산을 만들기 전에 최대 몇 분 정도 기다려야 할 수 있습니다. 자세한 정보는 [자산 및 모델 상태](#)를 참조하세요.

주제

- [자산 생성\(콘솔\)](#)
- [자산 만들기 \(\)AWS CLI](#)
- [새 자산 구성](#)

자산 생성(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산을 생성할 수 있습니다.

자산을 생성하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 자산 생성을 선택합니다.
4. 자산 생성 페이지에서 다음을 수행합니다.
 - a. 모델에서 자산을 생성할 자산 모델을 선택합니다.

Note

모델이 ACTIVE(활성) 상태가 아닌 경우 활성 상태가 될 때까지 기다리거나, 모델이 FAILED(실패) 상태인 경우 문제를 해결해야 합니다.

- b. 자산의 이름을 입력합니다.
- c. (선택 사항) 자산에 대한 태그를 추가합니다. 자세한 정보는 [리소스에 태그 지정하기 AWS IoT SiteWise](#)을 참조하세요.
- d. 자산 생성을 선택합니다.

에셋을 만들면 AWS IoT SiteWise 콘솔이 새 에셋 페이지로 이동합니다. 이 페이지에서 자산의 상태를 볼 수 있습니다. 상태는 처음에 CREATING(생성 중)입니다. 이 페이지는 자동으로 업데이트되므로 자산 상태가 업데이트될 때까지 기다릴 수 있습니다.

Note

자산 생성 프로세스에는 최대 1분이 소요될 수 있습니다. 상태가 활성이 된 후에는 자산에 대한 업데이트 작업을 수행할 수 있습니다. 자세한 정보는 [자산 및 모델 상태](#)을 참조하세요.

자산을 생성한 후 [새 자산 구성](#) 단원을 참조하세요.

자산 만들기 ()AWS CLI

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 모델에서 자산을 생성할 수 있습니다.

자산을 만들려면 `assetModelId`가 있어야 합니다. 자산 모델을 만들었지만 모르는 경우에는 `assetModelId` [ListAssetModels](#) API를 사용하여 모든 자산 모델을 확인하십시오.

자산 모델에서 자산을 생성하려면 다음 매개변수와 함께 [CreateAssetAPI](#)를 사용하십시오.

- `assetName` - 새 자산의 이름입니다. 쉽게 식별할 수 있도록 자산 이름을 지정하십시오.
- `assetModelId` - 자산의 ID입니다. UUID 형식의 실제 ID 또는 UUID 형식이 있는 `externalId:myExternalId` 경우 해당 ID입니다. 자세한 내용은 [AWS IoT SiteWise 사용 설명서](#)의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

자산을 만들려면 ()AWS CLI

- 다음 명령을 실행하여 자산을 생성합니다. `## ###` 에셋 이름으로 바꾸고 에셋 `## ID# ## ### ID` 또는 외부 ID로 바꾸십시오.

```
aws iotsitewise create-asset \
  --asset-name asset-name \
  --asset-model-id asset-model-id
```

이 작업은 다음과 같은 형식으로 새 자산의 세부 정보와 상태가 포함된 응답을 반환합니다.

```
{
  "assetId": "String",
  "assetArn": "String",
  "assetStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

자산이 생성될 때까지 자산의 state는 CREATING입니다.

Note

자산 생성 프로세스에는 최대 1분이 소요될 수 있습니다. 에셋 상태를 확인하려면 에셋 ID를 파라미터로 사용하는 [DescribeAsset](#) 작업을 사용하십시오. `assetId` 자산의 state가 ACTIVE이면 자산에 대한 업데이트 작업을 수행할 수 있습니다. 자세한 정보는 [자산 및 모델 상태](#)를 참조하십시오.

자산을 생성한 후 [새 자산 구성](#) 단원을 참조하세요.

새 자산 구성

다음과 같은 선택적 작업을 사용하여 자산 구성을 마칩니다.

- [산업 데이터 스트림을 자산 속성에 매핑](#) 자산에 측정 속성이 있는 경우.
- [속성 값 업데이트](#) 자산에 고유한 속성 값이 있는 경우.
- [자산 연결 및 연결 해제](#) 자산이 상위 자산인 경우.

자산 검색

AWS IoT SiteWise Console 검색 기능을 사용하여 메타데이터 및 실시간 속성 값 필터를 기반으로 자산을 찾을 수 있습니다.

필수 조건

AWS IoT SiteWise 산업 데이터를 더 잘 구성하고 모델링하려면 AWS IoT TwinMaker 통합할 수 있는 권한이 필요합니다. 권한을 AWS IoT SiteWise 부여한 경우 [ExecuteQuery](#) API를 사용하세요. 권한을 부여하지 않았는데 시작하는 데 AWS IoT SiteWise 도움이 필요한 경우 [참조하십시오 AWS IoT SiteWise 및 AWS IoT TwinMaker 통합](#).

고급 검색: AWS IoT SiteWise Console

메타데이터 검색

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창의 자산 아래에서 고급 검색을 선택합니다.
3. 고급 검색에서 메타데이터 검색 옵션을 선택합니다.
4. 매개변수를 입력합니다. 효율적인 검색을 위해 가능한 한 많은 필드를 채우십시오.
 - a. 자산 이름 — 전체 자산 이름을 입력하거나 광범위한 검색을 위해 일부 이름을 입력합니다.
 - b. 부동산 이름 — 전체 부동산 이름을 입력하거나 광범위한 검색을 위해 일부 이름을 입력합니다.
 - c. 연산자 — 다음 중에서 연산자를 선택합니다.

• =

- <
 - >
 - <=
 - >=
- d. 속성 값 — 이 값을 속성의 최신 값과 비교합니다.
- e. 속성 값 유형 — 속성의 데이터 유형입니다. 다음 중에서 선택합니다.
- 더블
 - 정수
 - 문자열
 - 부울
5. 검색을 선택합니다.
6. 검색 결과 테이블의 이름 열에서 자산을 선택합니다. 그러면 해당 자산의 세부 자산 페이지로 이동합니다.

The screenshot shows the AWS IoT SiteWise console interface for searching assets. The search criteria are as follows:

Asset name	Property name	Operator	Property value	Property value type
Level-2	power_max	>	20	Double

The search results are displayed in a table with 2 results:

Name	Asset id	Description
Level-2-asset-1	d0e9019b-9c38-4316-b574-38317aa38143	
Level-2-asset-2	b9c0d2fc-1527-42ce-8ba2-d1a4e8ff43de	Example description

부분 검색

자산 검색에는 모든 매개변수를 제공할 필요가 없습니다. 다음은 메타데이터 검색 옵션을 사용한 부분 검색의 몇 가지 예입니다.

- 이름을 기준으로 에셋 찾기:
 - 에셋 이름 필드에만 값을 입력합니다.
 - 속성 이름 및 속성 값 필드는 비어 있습니다.
- 특정 이름을 가진 속성을 포함하는 자산 찾기:
 - 속성 이름 필드에만 값을 입력합니다.
 - 에셋 이름 및 프로퍼티 값 필드는 비어 있습니다.
- 해당 속성의 최신 값을 기준으로 자산을 찾으십시오.
 - 속성 이름 및 속성 값 필드에 값을 입력합니다.
 - 연산자 및 속성 값 유형을 선택합니다.

쿼리 빌더 검색

1. AWS IoT SiteWise Console로 이동합니다.
2. 탐색 창의 자산 아래에서 고급 검색을 선택합니다.
3. 고급 검색에서 쿼리 작성기 옵션을 선택합니다.
4. 쿼리 작성기 창에서 `asset_name`, `asset_id` 및 를 검색할 SQL 쿼리를 작성합니다 `asset_description`.
5. 검색을 선택합니다.
6. 검색 결과 테이블의 이름 열에서 자산을 선택합니다. 그러면 해당 자산의 세부 자산 페이지로 이동합니다.

The screenshot shows the AWS IoT SiteWise 'Assets' page. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a region dropdown set to 'N. Virginia'. Below the navigation, the page title is 'Assets' with a 'Create asset' button. A brief description states: 'Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.'

The 'Advanced search' section is active, showing a 'Query builder' tab. The query entered is:


```
SELECT a.asset_id, a.asset_name, a.asset_description
FROM asset a, asset_property p, latest_value_time_series ts
WHERE a.asset_name LIKE '%asset-2%' AND a.property_name = 'temperature_f' AND ts.double_value > 50.0
```

 Below the query builder, there are 'Clear' and 'Search' buttons. The search results section shows 2 results in a table:

Name	Asset id	Description
Level-2a-asset-2	4fed596d-e903-4338-86db-34ca9301233a	Generator #3
Level-2b-asset-2	b4ac2b24-4fce-4a72-9fea-ef6d0f741e8d	Generator #2

Note

- SQL 쿼리의 SELECT 절에는 검색 결과 테이블의 자산이 유효한지 확인하기 위해 `asset_name` 및 `asset_id` 필드가 포함되어야 합니다.
- 쿼리 빌더는 결과 테이블에 이름, 자산 ID 및 설명만 표시합니다. SELECT 조항에 필드를 더 추가해도 결과 테이블에 열이 더 추가되지 않습니다.

산업 데이터 스트림을 자산 속성에 매핑

자산 자산에 속성 별칭을 정의할 수 있습니다. 이렇게 하면 자산 데이터를 수집하거나 검색할 때 자산 자산을 식별하는 데 도움이 됩니다. 자산에 측정 속성이 있는 경우 속성 별칭을 정의하여 데이터 스트림을 해당 측정 속성에 매핑할 수 있습니다.

이 프로세스를 수행하려면 속성 별칭을 알고 있어야 합니다.

- [SiteWise 에지 게이트웨이의 OPC-UA 데이터 소스를 사용하여 OPC-UA 서버에서 데이터를 수집하는 경우 속성 별칭은 다음으로 시작하는 Objects 노드 아래의 변수 경로입니다.](#) /

Example

변수 경로가 인 경우 속성 별칭은 `company/windfarm/3/turbine/7/temperature.` /
`company/windfarm/3/turbine/7/temperature`

OPC-UA 정보 아키텍처에 대한 자세한 내용은 OPC UA 온라인 참조의 [정보 모델 및 주소 간격 매핑](#)을 참조하십시오.

참고

- OPC-UA 소스의 데이터 스트림 접두사를 구성하는 경우 해당 소스의 모든 데이터 스트림에 대한 속성 별칭에 해당 접두사를 포함시켜야 합니다.

Example

가 접두사인 경우 이전 `/RentonWA` 별칭은 `입니다.` `/RentonWA/company/windfarm/3/turbine/7/temperature`

- 속성 별칭은 최대 1,000바이트까지 포함할 수 있습니다. OPC-UA 변수 경로에는 최대 4,096바이트가 포함될 수 있습니다. 현재는 경로가 긴 OPC-UA 변수의 데이터 수집을 AWS IoT SiteWise 지원하지 않습니다.

- [엣지 게이트웨이의 Modbus TCP 데이터 소스를 사용하여 Modbus 서버에서 데이터를 수집하는 경우 속성 별칭은 다음과 같습니다.](#) SiteWise

모드버스 `register set tag name`

이 값을 사용하여 이 레지스터 세트의 데이터를 자산 속성으로 보낼 수 있습니다.

- [AWS IoT 규칙이나 API를](#) 사용하는 등 다른 소스에서 데이터를 수집하는 경우 속성 별칭을 정의해야 합니다. 디바이스 구성에 적용 가능한 속성 별칭 이름 지정 시스템을 정의할 수 있습니다. 예를 들어 AWS IoT 사물의 데이터를 수집하는 경우 속성 별칭에 사물 이름을 포함시켜 데이터 스트림을 고유하게 식별할 수 있습니다. 이 예제에 대한 자세한 내용은 [사물에서 AWS IoT 데이터 수집 튜토리얼](#)을 참조하십시오.

속성 별칭은 지역 및 AWS 계정 내에서 고유해야 합니다. AWS IoT SiteWise 속성 별칭을 다른 자산 속성에 이미 있는 별칭으로 설정하면 오류가 반환됩니다.

데이터 스트림 경로가 동일한 OPC-UA 소스가 여러 개 있는 경우 각 소스의 경로에 접두사를 추가하여 고유한 별칭을 형성하십시오. 자세한 정보는 [데이터 소스 구성](#)을 참조하십시오.

Note

이 단원에서는 측정 속성의 속성 별칭을 설정하는 방법을 설명합니다. 외부 경보 상태 속성의 속성 별칭을 설정하는 방법에 대한 자세한 내용은 [외부 경보 상태 스트림 매핑](#) 단원을 참조하십시오.

주제

- [속성 별칭 설정\(콘솔\)](#)
- [속성 별칭 설정 \(\)AWS CLI](#)

속성 별칭 설정(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산 자산의 별칭을 설정할 수 있습니다.

속성 별칭을 설정하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 속성 별칭을 설정할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.
5. 별칭을 설정할 속성을 찾은 다음, 속성 별칭을 입력합니다.

The screenshot shows a configuration form for a property named "Wind Speed". The "Name" field contains the path "/company/windfarm/3/turbine/7/speed" and is circled in red. Below the field, it says "Must be less than 2048 characters." To the right, the "Notification status" is set to "DISABLED". Below that, there is a notification topic: "Notification will be published to topic \$aws/sitesite/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE".

6. 저장을 선택합니다.

속성 별칭 설정 ()AWS CLI

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 자산의 별칭을 설정합니다.

이 절차를 완료하려면 자산의 `assetId` 및 속성의 `propertyId`를 알고 있어야 합니다. 외부 ID를 사용할 수도 있습니다. 자산을 생성했는데 모르는 경우 [ListAssets](#) API를 사용하여 특정 모델의 모든 자산을 나열하십시오. `assetId` [DescribeAsset](#) 작업을 사용하여 속성 ID를 포함한 자산의 속성을 볼 수 있습니다.

[UpdateAsset속성](#) 작업을 사용하여 데이터 스트림을 자산 속성에 매핑할 수 있습니다. 다음 파라미터를 지정합니다.

- `assetId`— 자산의 ID 또는 외부 ID. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
- `propertyId`— 자산 자산의 ID 또는 외부 ID.
- `propertyAlias` - 속성 별칭에 대한 데이터 스트림의 경로입니다.
- `propertyNotificationState` - 속성 값 알림 상태는 ENABLED 또는 DISABLED입니다. 속성의 별칭을 업데이트할 때 속성의 기존 알림 상태를 지정합니다. [DescribeAsset속성](#) 작업을 통해 기존 알림 상태를 검색할 수 있습니다.

이 파라미터를 생략하면 새 알림 상태는 DISABLED가 됩니다. 속성 알림에 대한 자세한 내용은 [다른 AWS 서비스와의 상호작용](#) 단원을 참조하세요.

속성 별칭을 설정하려면 ()AWS CLI

1. 다음 명령을 실행하여 속성의 현재 알림 상태를 검색합니다. `asset-id`와 `property-id`를 자산 속성의 ID로 바꿉니다.

```
aws iotsitewise describe-asset-property \
  --asset-id asset-id \
  --property-id property-id
```

이 작업은 다음과 같은 형식으로 자산 속성의 세부 정보가 포함된 응답을 반환합니다. 속성 알림 상태는 JSON 객체에서 `assetProperty.notification.state`입니다.

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetName": "Wind Turbine 7",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
```

```

"assetProperty": {
  "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "name": "Wind Speed",
  "notification": {
    "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE",
    "state": "ENABLED"
  },
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
}
}
}

```

2. 다음 명령을 실행하여 자산 속성의 별칭을 설정합니다. *property-alias*를 속성 별칭으로 바꾸고 *notification-state*를 알림 상태로 바꾸거나 `--property-notification-state`를 생략하여 알림을 비활성화합니다. 선택적으로 자산의 단위를 새 `##` 및 `--property-unit`로 업데이트할 수 있습니다.

```

aws iotsitewise update-asset-property \
  --asset-id asset-id \
  --property-id property-id \
  --property-alias property-alias \
  --property-notification-state notification-state \
  --property-unit unit

```

3. 별칭이 설정되었는지 확인하려면 다음 명령을 실행하여 속성의 세부 정보를 검색하세요. *asset-id*와 *property-id*를 자산 속성의 ID로 바꿉니다.

```

aws iotsitewise describe-asset-property \
  --asset-id asset-id \
  --property-id property-id

```

이 작업은 다음과 같은 형식으로 자산 속성의 세부 정보가 포함된 응답을 반환합니다. 속성 별칭은 JSON 객체의 `assetProperty.alias`에 있으며 이 예제에서는 `myAlias`로 설정되어 있습니다.

```

{
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetName": "Wind Turbine 7",

```

```

"assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
"assetProperty": {
  "alias": "myAlias",
  "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "name": "Wind Speed",
  "notification": {
    "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE",
    "state": "ENABLED"
  },
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
}
}

```

속성 값 업데이트

자산은 속성의 기본 값을 포함하여 자산 모델의 속성을 상속합니다. 일부 경우에는 자산 제조업체 속성과 같이 자산 모델의 기본 속성을 유지해야 할 수 있습니다. 다른 일부 경우에는 자산의 위도 및 경도와 같이 상속된 속성을 업데이트해야 할 수 있습니다.

Updating an attribute value (console)

AWS IoT SiteWise 콘솔을 사용하여 속성 자산 속성의 값을 업데이트할 수 있습니다.

속성의 값을 업데이트하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 속성을 업데이트할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.

- 업데이트할 속성을 찾은 다음, 새 값을 입력합니다.

- 저장을 선택합니다.

Updating an attribute value (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 속성 값을 업데이트할 수 있습니다.

이 절차를 완료하려면 자산의 `assetId` 및 속성의 `propertyId`를 알고 있어야 합니다. 외부 ID를 사용할 수도 있습니다. 자산을 생성했는데 모르는 경우 [ListAssets](#) API를 사용하여 특정 모델의 모든 자산을 나열하십시오. `assetId` [DescribeAsset](#) 작업을 사용하여 속성 ID를 포함한 자산의 속성을 볼 수 있습니다.

[BatchPutAssetProperty](#) 작업을 사용하여 자산에 속성 값을 할당할 수 있습니다. 이 작업을 사용하여 한 번에 여러 속성을 설정할 수 있습니다. 이 작업의 페이로드에는 각각 자산 ID, 속성 ID, 속성 값을 포함하는 항목 목록이 포함되어 있습니다.

속성 값을 업데이트하려면 (AWS CLI)

- `batch-put-payload.json`이라는 파일을 만들고 다음 JSON 개체를 파일에 복사합니다. 이 페이로드 예제에서는 풍력 터빈의 위도와 경도를 설정하는 방법을 보여줍니다. ID, 값 및 타임스탬프를 업데이트하여 사용 사례에 대한 페이로드를 수정합니다.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          }
        }
      ]
    }
  ]
}
```

```

        "timestamp": {
            "timeInSeconds": 1575691200
        }
    ],
    },
    {
        "entryId": "windfarm3-turbine7-longitude",
        "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
        "propertyValues": [
            {
                "value": {
                    "doubleValue": 122.3491
                },
                "timestamp": {
                    "timeInSeconds": 1575691200
                }
            }
        ]
    }
]
}

```

- 페이로드의 각 항목은 고유한 문자열로 정의할 수 있는 entryId를 포함합니다. 요청 항목이 실패하면 각 오류에 해당 요청의 entryId가 포함되므로 다시 시도할 요청을 알 수 있습니다.
- 속성 값을 설정하려면 각 속성 속성 목록에 하나의 timestamp-quality-value (TQV) 구조를 포함할 수 있습니다. propertyValues 이 구조에는 새 value와 현재 timestamp가 포함되어야 합니다.
 - value -설정되는 속성 유형에 따라 다음 필드 중 하나를 포함하는 구조입니다.
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue
 - timestamp— 현재 Unix 에포크 시간을 초 단위로 포함하는 구조입니다. timeInSeconds AWS IoT SiteWise 과거에 7일 이상 존재했거나 이후에 5분 이상 존재한 타임스탬프가 있는 데이터 포인트를 거부합니다.

[Value용 페이로드를 준비하는 방법에 대한 자세한 내용은 을 참조하십시오. BatchPut AssetProperty API를 사용한 데이터 수집 AWS IoT SiteWise](#)

2. 다음 명령을 실행하여 속성 값을 다음으로 보내십시오. AWS IoT SiteWise

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

자산 연결 및 연결 해제

자산의 모델이 하위 자산 모델 계층 구조를 정의하는 경우, 하위 자산을 자산에 연결할 수 있습니다. 상위 자산은 연결된 자산에서 데이터에 액세스하고 집계할 수 있습니다. 계층적 자산 모델에 대한 자세한 내용은 [자산 모델 계층 구조 정의](#) 단원을 참조하세요.

주제

- [자산 연결 및 연결 해제\(콘솔\)](#)
- [자산 연결 및 연결 해제 \(\)AWS CLI](#)

자산 연결 및 연결 해제(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산을 연결하고 분리할 수 있습니다.

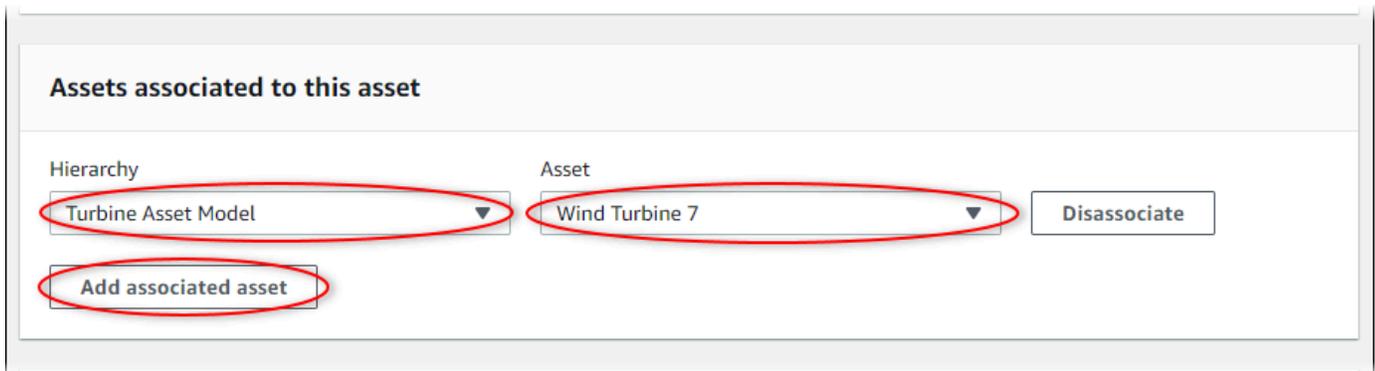
자산을 연결하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 하위 자산을 연결할 상위 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.
5. 이 자산에 연결된 자산에서 연결된 자산 추가를 선택합니다.



6. 계층 구조에서 상위 자산과 하위 자산 간의 관계를 정의하는 계층 구조를 선택합니다.
7. 자산에서 연결할 하위 자산을 선택합니다.
8. 저장을 선택합니다.

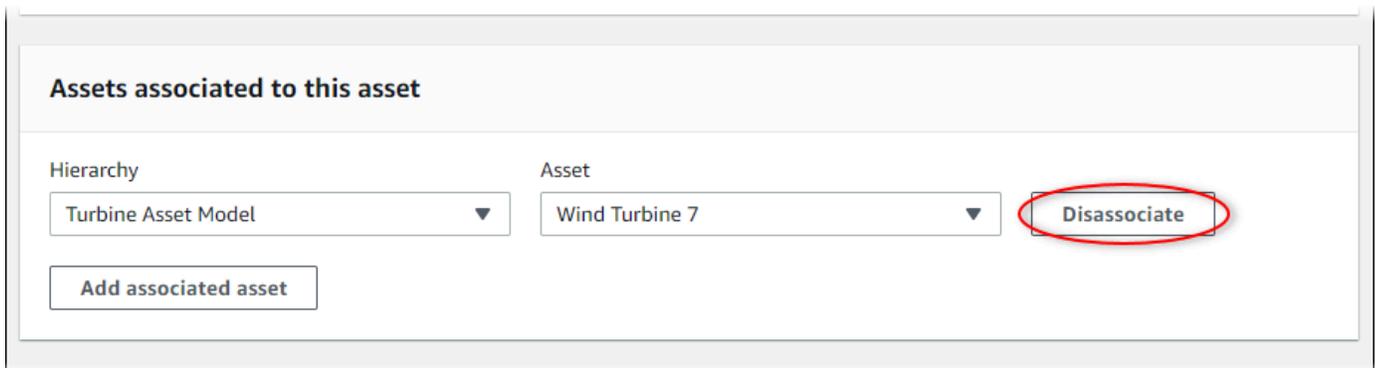
자산의 연결을 해제하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 하위 자산의 연결을 해제할 상위 자산을 선택합니다.

i Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.
5. 이 자산에 연결된 자산에서 자산에 대한 연결 해제를 선택합니다.



6. 저장을 선택합니다.

자산 연결 및 연결 해제 ()AWS CLI

AWS Command Line Interface (AWS CLI) 를 사용하여 자산을 연결하거나 분리할 수 있습니다.

이 절차를 수행하려면 상위 자산 모델에서 하위 자산 모델에 대한 관계를 정의하는 계층 구조 (hierarchyId)의 ID를 알고 있어야 합니다. [DescribeAsset](#)작업을 사용하여 응답에서 계층 ID를 찾을 수 있습니다.

계층 구조 ID를 찾으려면

- 다음 명령을 실행하여 상위 자산을 설명합니다. `## ## ID# ## ### ID ## ## ID#` 바꾸십시오.

```
aws iotsitewise describe-asset --asset-id parent-asset-id
```

이 작업은 자산의 세부 정보가 포함된 응답을 반환합니다. 응답에는 다음과 같은 구조의 `assetHierarchies` 목록이 포함됩니다.

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

계층 구조 ID는 자산 계층 구조 목록에 있는 계층 구조의 `id` 값입니다.

계층 구조 ID를 확인한 후에는 자산을 해당 계층 구조와 연결하거나 연결 해제할 수 있습니다.

하위 자산을 상위 자산에 연결하려면 [AssociateAssets](#)작업을 사용하십시오. 상위 자산에서 하위 자산을 분리하려면 [DisassociateAssets](#)작업을 사용하십시오. 다음 파라미터를 지정합니다. 이 파라미터는 두 작업에 모두 동일합니다.

- `assetId`— 상위 자산의 ID 또는 외부 ID.
- `hierarchyId`— 상위 자산의 계층 ID 또는 외부 ID.

- `childAssetId`— 하위 자산의 ID 또는 외부 ID.

자산을 연결하려면 (AWS CLI)

- 다음 명령을 실행하여 하위 자산을 상위 자산에 연결합니다. `## ## ID, ## ID ### ## ID# # ID# #####`.

```
aws iotsitewise associate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

에셋 연결을 끊으려면 AWS CLI()

- 다음 명령을 실행하여 상위 자산에서 하위 자산의 연결을 해제합니다. `## ## ID, ## ID # ## ## ID# # ID# #####`.

```
aws iotsitewise disassociate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

자산 및 모델 업데이트

에서 자산, 자산 모델 및 구성 요소 모델을 AWS IoT SiteWise 업데이트하여 이름과 정의를 수정할 수 있습니다. 이러한 업데이트 작업은 비동기적으로 진행되며 적용되기까지 시간이 걸립니다. AWS IoT SiteWise 추가로 변경하기 전에 자산 또는 모델의 상태를 확인하십시오. 업데이트된 자산이나 모델을 계속 사용하려면 변경 사항이 전파될 때까지 기다려야 합니다.

주제

- [자산 업데이트](#)
- [자산 모델 및 구성 요소 모델 업데이트](#)
- [커스텀 복합 모델 \(컴포넌트\) 업데이트](#)

자산 업데이트

AWS IoT SiteWise 콘솔 또는 API를 사용하여 자산 이름을 업데이트할 수 있습니다.

자산을 업데이트할 때 자산의 상태는 변경 사항이 전파될 때까지 UPDATING입니다. 자세한 정보는 [자산 및 모델 상태](#)를 참조하세요.

주제

- [자산 업데이트\(콘솔\)](#)
- [자산 업데이트 \(AWS CLI\)](#)

자산 업데이트(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산 세부 정보를 업데이트할 수 있습니다.

자산을 업데이트하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 업데이트할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.
5. 자산의 이름을 업데이트합니다.
6. (선택 사항) 이 페이지에서 자산에 대한 기타 정보를 업데이트합니다. 자세한 내용은 다음 자료를 참조하세요.
 - [산업 데이터 스트림을 자산 속성에 매핑](#)
 - [속성 값 업데이트](#)
 - [다른 AWS 서비스와의 상호작용](#)
7. 저장을 선택합니다.

자산 업데이트 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 이름을 업데이트할 수 있습니다.

[UpdateAsset](#) 작업을 사용하여 자산을 업데이트합니다. 다음 파라미터를 지정합니다.

- `assetId` - 자산의 ID입니다. UUID 형식의 실제 ID (있는 `externalId:myExternalId` 경우) 입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.
- `assetName` - 자산의 새 이름입니다.

에셋 이름을 업데이트하려면 ()AWS CLI

- 다음 명령을 실행하여 자산의 이름을 업데이트합니다. `asset-id# ###` ID 또는 외부 ID로 바꾸십시오. `## ### ## # #####` 업데이트하십시오.

```
aws iotsitewise update-asset \
  --asset-id asset-id \
  --asset-name asset-name
```

자산 모델 및 구성 요소 모델 업데이트

AWS IoT SiteWise 콘솔 또는 API를 사용하여 자산 모델 또는 구성 요소 모델을 업데이트할 수 있습니다.

기존 속성의 유형이나 데이터 유형 또는 기존 지표의 창은 변경할 수 없습니다. 또한 모델 유형을 자산 모델에서 구성 요소 모델로 또는 그 반대로 변경할 수 없습니다.

Important

- 자산 모델이나 구성 요소 모델에서 속성을 제거하면 해당 속성에 대한 이전 데이터가 모두 AWS IoT SiteWise 삭제됩니다. 구성 요소 모델의 경우 이는 해당 구성 요소 모델을 사용하는 모든 자산 모델에 영향을 미치므로 변경 사항이 얼마나 광범위하게 적용될 수 있는지 이해하는 데 특히 주의해야 합니다.
- 자산 모델에서 계층 정의를 제거하면 해당 계층 구조의 모든 AWS IoT SiteWise 자산이 분리됩니다.

자산 모델을 업데이트하면 해당 모델을 기반으로 하는 모든 자산에 기본 모델의 변경 사항이 반영됩니다. 변경 사항이 전파될 때까지 각 자산에는 UPDATING 상태가 있습니다. 이러한 자산과 상호 작용하기 전에 자산의 상태가 ACTIVE 상태로 돌아올 때까지 기다려야 합니다. 이 기간 동안 업데이트된 자산 모델의 상태는 PROPAGATING가 됩니다.

구성 요소 모델을 업데이트하면 해당 구성 요소 모델을 통합하는 모든 자산 모델에 변경 내용이 반영됩니다. 구성 요소 모델 변경 내용이 적용되기 전까지는 이전 단락에 설명된 대로 영향을 받는 각 자산 모델의 UPDATING 상태가 유지되고 관련 자산이 업데이트되는 시기가 뒤따릅니다. PROPAGATING 해당 자산 모델과 상호 작용하려면 해당 자산 모델이 ACTIVE 상태로 돌아올 때까지 기다려야 합니다. 이 기간 동안 업데이트된 구성 요소 모델의 상태는 다음과 같습니다 PROPAGATING.

자세한 정보는 [자산 및 모델 상태](#)를 참조하세요.

주제

- [자산 또는 구성 요소 모델 업데이트 \(콘솔\)](#)
- [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#)

자산 또는 구성 요소 모델 업데이트 (콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산 모델 또는 구성 요소 모델을 업데이트할 수 있습니다.

에셋 모델 또는 구성 요소 모델을 업데이트하려면 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 업데이트할 자산 모델 또는 구성 요소 모델을 선택합니다.
4. 편집을 선택합니다.
5. 모델 편집 페이지에서 다음 중 하나를 수행합니다.
 - 모델 세부 정보에서 모델의 이름을 변경합니다.
 - 속성 정의를 변경합니다. 기존 속성의 데이터 형식은 변경할 수 없습니다. 자세한 정보는 [정적 데이터\(속성\) 정의](#)를 참조하세요.
 - 측정 정의를 변경합니다. 기존 측정의 데이터 형식은 변경할 수 없습니다. 자세한 정보는 [장비의 데이터 스트림 정의\(측정값\)](#)를 참조하세요.
 - 변환 정의를 변경합니다. 자세한 정보는 [데이터 변환\(변환\)](#)을 참조하세요.
 - 지표 정의를 변경합니다. 기존 지표의 시간 간격은 변경할 수 없습니다. 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)를 참조하세요.

- (에셋 모델만 해당) 계층 정의를 변경하십시오. 기존 계층 구조의 계층 구조 모델은 변경할 수 없습니다. 자세한 정보는 [자산 모델 계층 구조 정의](#)를 참조하세요.

6. 저장(Save)을 선택합니다.

자산 또는 구성 요소 모델 업데이트 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 모델 또는 구성 요소 모델을 업데이트할 수 있습니다.

[UpdateAsset모델](#) API를 사용하여 자산 모델 또는 구성 요소 모델의 이름, 설명 및 속성을 업데이트하십시오. 자산 모델의 경우에만 계층 구조를 업데이트할 수 있습니다. 다음 파라미터를 지정합니다.

- `assetModelId` – 자산의 ID입니다. UUID 형식의 실제 ID (있는 `externalId:myExternalId` 경우) 입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

페이로드에 업데이트된 모델을 지정합니다. 자산 모델 또는 구성 요소 모델의 예상 형식에 대해 알아보려면 [참조하십시오 자산 모델 생성](#).

Warning

[UpdateAsset모델](#) API는 기존 모델을 페이로드에 제공한 모델로 덮어씁니다. 모델의 속성이나 계층이 삭제되지 않도록 하려면 업데이트된 모델 페이로드에 해당 ID와 정의를 포함해야 합니다. [모델의 기존 구조를 쿼리하는 방법을 알아보려면 모델 작업을 참조하십시오. DescribeAsset](#)

Note

다음 프로시저는 유형의 복합 모델만 업데이트할 수 AWS/ALARM 있습니다. CUSTOM복합 모델을 업데이트하려면 [UpdateAssetModelComposite모델](#)을 대신 사용하십시오. 자세한 정보는 [커스텀 복합 모델 \(컴포넌트\) 업데이트](#)을 참조하세요.

자산 모델 또는 구성 요소 모델을 업데이트하려면 (AWS CLI)

1. 다음 명령을 실행하여 기존 모델 정의를 검색합니다. `asset-model-id#` 업데이트할 자산 모델 또는 구성 요소 모델의 ID 또는 외부 ID로 바꾸십시오.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

이 작업은 모델의 세부 정보가 포함된 응답을 반환합니다. 응답의 구조는 다음과 같습니다.

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel,
  "assetModelCompositeModelSummaries": Array of AssetModelCompositeModelSummary,
  "assetModelCreationDate": "String",
  "assetModelLastUpdateDate": "String",
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  },
  "assetModelType": "String"
}
```

자세한 내용은 [DescribeAsset모델](#) 작업을 참조하십시오.

2. update-asset-model.json이라는 파일을 만들고 이전 명령의 응답을 파일에 복사합니다.
3. update-asset-model.json의 JSON 객체에서 다음 키-값 페어를 제거합니다.
 - assetModelId
 - assetModelArn
 - assetModelCompositeModelSummaries
 - assetModelCreationDate
 - assetModelLastUpdateDate
 - assetModelStatus
 - assetModelType

[UpdateAsset모델](#) 작업에는 다음과 같은 구조의 페이로드가 필요합니다.

```
{
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel
}
```

4. update-asset-model.json에서 다음을 수행합니다.

- 자산 모델의 이름(assetModelName)을 변경합니다.
- 자산 모델의 설명(assetModelDescription)을 변경, 추가 또는 제거합니다.
- 자산 모델의 속성(assetModelProperties)을 변경, 추가 또는 제거합니다. 기본 속성의 dataType 또는 기존 지표의 window는 변경할 수 없습니다. 자세한 정보는 [데이터 속성 정의](#)를 참조하세요.
- 자산 모델의 계층 구조(assetModelHierarchies)를 변경, 추가 또는 제거합니다. 기존 계층 구조의 childAssetModelId는 변경할 수 없습니다. 자세한 정보는 [자산 모델 계층 구조 정의](#)를 참조하세요.
- 자산 모델의 복합 모델 유형 AWS/ALARM (assetModelCompositeModels) 을 변경, 추가 또는 제거합니다. 경보는 다른 속성을 모니터링하므로 장비나 프로세스에 주의가 필요한 시점을 식별할 수 있습니다. 각 경보 정의는 경보가 사용하는 속성 집합을 표준화하는 복합 모델입니다. 자세한 내용은 [경보로 데이터 모니터링](#) 및 [자산 모델에서 경보 정의](#) 섹션을 참조하세요.

5. 다음 명령을 실행하여 update-asset-model.json에 저장된 정의로 자산 모델을 업데이트합니다. `asset-model-id# ## ### ID#` 바꾸십시오.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://model-payload.json
```

커스텀 복합 모델 (컴포넌트) 업데이트

AWS IoT SiteWise API를 사용하여 사용자 지정 복합 모델을 업데이트하거나 AWS IoT SiteWise 콘솔을 사용하여 구성 요소를 업데이트할 수 있습니다.

주제

- [구성 요소 업데이트 \(콘솔\)](#)
- [사용자 지정 복합 모델 업데이트 \(AWS CLI\)](#)

구성 요소 업데이트 (콘솔)

AWS IoT SiteWise 콘솔을 사용하여 구성 요소를 업데이트할 수 있습니다.

구성 요소를 업데이트하려면 (콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 구성 요소가 있는 에셋 모델을 선택합니다.
4. 속성 탭에서 구성 요소를 선택합니다.
5. 업데이트하려는 구성 요소를 선택합니다.
6. 편집을 선택합니다.
7. 구성 요소 편집 페이지에서 다음 중 하나를 수행하십시오.
 - 모델 세부 정보에서 모델의 이름을 변경합니다.
 - 속성 정의를 변경합니다. 기존 속성의 데이터 형식은 변경할 수 없습니다. 자세한 정보는 [정적 데이터\(속성\) 정의](#)를 참조하세요.
 - 측정 정의를 변경합니다. 기존 측정의 데이터 형식은 변경할 수 없습니다. 자세한 정보는 [장비의 데이터 스트림 정의\(측정값\)](#)를 참조하세요.
 - 변환 정의를 변경합니다. 자세한 정보는 [데이터 변환\(변환\)](#)을 참조하세요.
 - 지표 정의를 변경합니다. 기존 지표의 시간 간격은 변경할 수 없습니다. 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)를 참조하세요.
8. 저장(Save)을 선택합니다.

사용자 지정 복합 모델 업데이트 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 사용자 지정 복합 모델을 업데이트할 수 있습니다.

이름 또는 설명을 업데이트하려면 [UpdateAssetModelComposite모델](#) 작업을 사용합니다. 인라인 사용자 지정 복합 모델의 경우에만 속성을 업데이트할 수도 있습니다. 사용자 지정 복합 모델은 참조된 구

성 요소 모델이 관련 속성을 제공하므로 component-model-based 사용자 지정 복합 모델의 속성을 업데이트할 수 없습니다.

⚠ Important

사용자 지정 복합 모델에서 속성을 제거하면 해당 속성에 대한 이전 데이터가 모두 AWS IoT SiteWise 삭제됩니다. 기존 속성의 유형이나 데이터 유형은 변경할 수 없습니다.

기존 복합 모델 속성을 동일한 name 새 복합 모델 속성으로 바꾸려면 다음과 같이 하십시오.

1. 기존 속성 전체를 제거한 상태로 UpdateAssetModelCompositeModel 요청을 제출하십시오.
2. 새 속성이 포함된 두 번째 UpdateAssetModelCompositeModel 요청을 제출하세요. 새 자산 속성은 이전 자산과 name 동일하며 AWS IoT SiteWise 새 고유 자산을 생성합니다. id.

사용자 지정 복합 모델 업데이트하기 (AWS CLI)

1. 기존 복합 모델 정의를 검색하려면 다음 명령을 실행합니다. `composite-model-id#` 업데이트할 사용자 지정 복합 모델의 ID 또는 외부 ID로 바꾸고, `asset-model-id#` 사용자 지정 복합 모델과 관련된 자산 모델로 바꾸십시오. 자세한 내용은 사용 설명서를 참조하십시오. AWS IoT SiteWise

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-composite-model-id composite-model-id \
  --asset-model-id asset-model-id
```

자세한 내용은 [DescribeAssetModelComposite모델](#) 작업을 참조하십시오.

2. 라는 `update-custom-composite-model.json` 파일을 만든 다음 이전 명령의 응답을 파일에 복사합니다.
3. 다음 필드를 `update-custom-composite-model.json` 제외한 모든 키-값 쌍을 JSON 객체에서 제거합니다.

- `assetModelCompositeModelName`
- `assetModelCompositeModelDescription`(있는 경우)
- `assetModelCompositeModelProperties`(있는 경우)

4. `update-custom-composite-model.json`에서 다음을 수행합니다.

- 의 값을 변경합니다 `assetModelCompositeModelName`.
- 값을 추가 또는 `assetModelCompositeModelDescription` 제거하거나 변경합니다.
- 인라인 사용자 지정 복합 모델만 해당: 에서 `assetModelCompositeModelProperties` 자산 모델의 속성을 변경, 추가 또는 제거합니다.

이 파일에 필요한 형식에 대한 자세한 내용은 [UpdateAssetModelCompositeModel의](#) 요청 구문을 참조하십시오.

5. 다음 명령을 실행하여 사용자 지정 복합 모델을 저장된 정의로 `update-custom-composite-model.json` 업데이트합니다. `##-##-id# ## ### ID#` 바꾸고, `asset-model-id# ## ## ### ID#` 대체합니다.

```
aws iotsitewise update-asset-model-composite-model \
--asset-model-composite-model-id composite-model-id \
--asset-model-id asset-model-id \
--cli-input-json file://update-custom-composite-model.json
```

자산 및 모델 삭제

자산과 모델을 모두 사용한 AWS IoT SiteWise 다음에는 삭제할 수 있습니다. 삭제 작업은 비동기적으로 진행되며 완료하는 데 시간이 걸립니다. AWS IoT SiteWise

주제

- [자산 삭제](#)
- [자산 모델 삭제](#)

자산 삭제

AWS IoT SiteWise 콘솔 또는 API를 사용하여 자산을 삭제할 수 있습니다.

자산을 삭제하려면 먼저 하위 자산의 연결을 해제하고 상위 자산에서 해당 자산의 연결을 해제해야 합니다. 자세한 정보는 [자산 연결 및 연결 해제](#)를 참조하세요. AWS Command Line Interface (AWS CLI)를 사용하는 경우 [ListAssociatedAssets](#) 작업을 사용하여 자산의 하위 항목을 나열할 수 있습니다.

자산을 삭제할 때 자산의 상태는 변경 사항이 전파될 때까지 DELETING입니다. 자세한 정보는 [자산 및 모델 상태](#)를 참조하세요. 자산을 삭제한 후에는 해당 자산을 쿼리할 수 없습니다. 이 경우 API는 HTTP 404 응답을 반환합니다.

Important

AWS IoT SiteWise 삭제된 자산의 모든 속성 데이터를 삭제합니다.

주제

- [자산 삭제\(콘솔\)](#)
- [에셋 삭제 \(AWS CLI\)](#)

자산 삭제(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산을 삭제할 수 있습니다.

자산을 삭제하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 삭제할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 자산에 연결된 자산이 있는 경우 각 자산을 삭제합니다. 자산의 이름을 선택하여 자산 페이지로 이동한 후, 이 페이지에서 자산을 삭제할 수 있습니다.
5. 자산 페이지에서 삭제를 선택합니다.
6. 에셋 삭제 대화 상자에서 다음을 수행합니다.
 - a. **Delete**를 입력하여 삭제를 확인합니다.
 - b. 삭제를 선택합니다.

에셋 삭제 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산을 삭제할 수 있습니다.

[DeleteAsset](#)작업을 사용하여 자산을 삭제합니다. 다음 파라미터를 지정하십시오.

- `assetId` – 자산의 ID입니다. UUID 형식의 실제 ID (있는 `externalId:myExternalId` 경우) 입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

자산 삭제하기 ()AWS CLI

1. 다음 명령을 실행하여 자산의 계층 구조를 나열합니다. `asset-id#` 자산의 ID 또는 외부 ID로 바꾸십시오.

```
aws iotsitewise describe-asset --asset-id asset-id
```

이 작업은 자산의 세부 정보가 포함된 응답을 반환합니다. 응답에는 다음과 같은 구조의 `assetHierarchies` 목록이 포함됩니다.

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

자세한 내용은 [DescribeAsset](#)작업을 참조하십시오.

2. 각 계층 구조에 대해 다음 명령을 실행하여 해당 계층 구조와 연결된 자산의 하위 항목을 나열합니다. `asset-id#` 자산의 ID 또는 외부 ID로 바꾸고, 계층 ID를 `## ### ID ## ## ID#` 바꾸십시오.

```
aws iotsitewise list-associated-assets \
  --asset-id asset-id \
  --hierarchy-id hierarchy-id
```

[자세한 내용은 에셋 작업을 참조하십시오. ListAssociated](#)

- 다음 명령을 실행하여 연결된 각 자산을 삭제한 다음 자산을 삭제합니다. `asset-id# ###` ID 또는 외부 ID로 바꾸십시오.

```
aws iotsitewise delete-asset --asset-id asset-id
```

자산 모델 삭제

AWS IoT SiteWise 콘솔 또는 API를 사용하여 자산 모델을 삭제할 수 있습니다.

자산 모델을 삭제하려면 먼저 자산 모델에서 생성된 모든 자산을 삭제해야 합니다.

자산 모델을 삭제할 때 자산 모델의 상태는 변경 사항이 전파될 때까지 DELETING입니다. 자세한 정보는 [자산 및 모델 상태](#)를 참조하세요. 자산 모델을 삭제한 후에는 해당 자산 모델을 쿼리할 수 없습니다. 이 경우 API는 HTTP 404 응답을 반환합니다.

주제

- [자산 모델 삭제\(콘솔\)](#)
- [자산 모델 삭제 \(AWS CLI\)](#)

자산 모델 삭제(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산 모델을 삭제할 수 있습니다.

자산 모델을 삭제하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 삭제할 자산 모델을 선택합니다.
4. 모델에 자산이 있는 경우 각 자산을 삭제합니다. 자산의 이름을 선택하여 자산 페이지로 이동한 후, 이 페이지에서 자산을 삭제할 수 있습니다. 자세한 정보는 [자산 삭제\(콘솔\)](#)를 참조하세요.
5. 모델 페이지에서 삭제를 선택합니다.
6. 모델 삭제 대화 상자에서 다음을 수행합니다.
 - a. **Delete**를 입력하여 삭제를 확인합니다.
 - b. 삭제를 선택합니다.

자산 모델 삭제 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 모델을 삭제할 수 있습니다.

[DeleteAsset모델](#) 작업을 사용하여 자산 모델을 삭제합니다. 다음 파라미터를 지정하십시오.

- `assetModelId` – 자산의 ID입니다. UUID 형식의 실제 ID 또는 UUID 형식이 있는 `externalId:myExternalId` 경우 UUID 형식입니다. 자세한 내용은 AWS IoT SiteWise 사용 설명서의 [외부 ID를 사용하여 개체 참조](#)를 참조하십시오.

자산 모델을 삭제하려면 ()AWS CLI

1. 다음 명령을 실행하여 모델에서 생성된 모든 자산을 나열합니다. `asset-model-id#` 에셋 모델의 ID 또는 외부 ID로 대체합니다.

```
aws iotsitewise list-assets --asset-model-id asset-model-id
```

자세한 내용은 작업을 참조하십시오. [ListAssets](#)

2. 이전의 명령이 모델에서 자산을 반환하는 경우 각 자산을 삭제합니다. 자세한 정보는 [에셋 삭제 \(AWS CLI\)](#)을 참조하세요.
3. 다음 명령을 실행하여 자산 모델을 삭제합니다. `asset-model-id# ## ### ID` 또는 외부 ID로 바꾸십시오.

```
aws iotsitewise delete-asset-model --asset-model-id asset-model-id
```

자산 및 모델을 사용한 대량 작업

많은 수의 자산 또는 자산 모델로 작업하려면 대량 작업을 사용하여 리소스를 대량으로 가져오고 다른 위치로 내보내십시오. 예를 들어 Amazon S3 버킷에 자산 또는 자산 모델을 정의하는 데이터 파일을 생성하고 대량 가져오기를 사용하여 Amazon S3 버킷에서 이를 생성하거나 업데이트할 수 AWS IoT SiteWise 있습니다. 또는 많은 수의 자산 또는 자산 모델이 있는 경우 Amazon S3로 내보낼 수 있습니다. AWS IoT SiteWise

Note

AWS IoT TwinMaker API에서 작업을 AWS IoT SiteWise 호출하여 대량 작업을 수행합니다. AWS IoT TwinMaker 작업 영역을 AWS IoT TwinMaker 설정하거나 만들지 않고도 이 작업을

수행할 수 있습니다. AWS IoT SiteWise 콘텐츠를 배치할 수 있는 Amazon S3 버킷만 있으면 됩니다.

주제

- [주요 개념 및 용어](#)
- [지원되는 기능](#)
- [대량 작업 사전 요구 사항](#)
- [대량 가져오기 작업 실행](#)
- [대량 내보내기 작업 실행](#)
- [작업 진행 상황 추적 및 오류 처리](#)
- [메타데이터 가져오기 예제](#)
- [메타데이터 내보내기 예제](#)
- [AWS IoT SiteWise 메타데이터 전송 작업 스키마](#)

주요 개념 및 용어

AWS IoT SiteWise 대량 가져오기 및 내보내기 기능은 다음 개념과 용어를 기반으로 합니다.

- **가져오기:** Amazon S3 버킷의 파일에서 자산 또는 자산 모델을 다음으로 이동하는 AWS IoT SiteWise 작업입니다.
- **내보내기:** Amazon S3 버킷에서 AWS IoT SiteWise 자산 또는 자산 모델을 이동하는 작업입니다.
- **소스:** 콘텐츠를 이동하려는 위치의 시작 위치입니다.

예를 들어 Amazon S3 버킷은 가져오기 AWS IoT SiteWise 소스이고 내보내기 소스입니다.

- **대상:** 콘텐츠를 이동하려는 원하는 위치입니다.

예를 들어 Amazon S3 버킷은 내보내기 AWS IoT SiteWise 대상이고 가져오기 목적지입니다.

- **AWS IoT SiteWise 스키마:** 이 스키마는 메타데이터를 가져오고 내보내는 데 사용됩니다 AWS IoT SiteWise.
- **최상위 리소스:** 자산 또는 자산 모델과 같이 개별적으로 만들거나 업데이트할 수 있는 AWS IoT SiteWise 리소스입니다.
- **하위 리소스:** 최상위 AWS IoT SiteWise 리소스 내에 중첩된 리소스입니다. 예로는 속성, 계층 구조, 복합 모델 등이 있습니다.

- 메타데이터: 리소스를 성공적으로 가져오거나 내보내는 데 필요한 주요 정보입니다. 메타데이터의 예로는 자산 정의 및 자산 모델 등이 있습니다.
- 메타데이터 TransferJob: 실행 시 생성되는 CreateMetadataTransferJob 객체입니다.

지원되는 기능

이 항목에서는 대량 작업을 실행할 때 수행할 수 있는 작업에 대해 설명합니다. 대량 작업은 다음 기능을 지원합니다.

- 최상위 리소스 생성: ID를 정의하지 않거나 ID가 기존 ID와 일치하지 않는 자산 또는 자산 모델을 가져오면 새 리소스로 생성됩니다.
- 최상위 리소스 교체: ID가 이미 존재하는 것과 일치하는 자산 또는 자산 모델을 가져오면 기존 리소스가 대체됩니다.
- 하위 리소스 생성, 교체 또는 삭제: 가져오기로 자산 또는 자산 모델과 같은 최상위 리소스가 대체되면 속성, 계층 구조 또는 복합 모델 같은 모든 하위 리소스가 새 정의로 대체됩니다.

예를 들어 대량 가져오기 중에 자산 모델을 업데이트하고 업데이트된 버전에서 원본에 없었던 속성을 정의하면 새 속성이 생성됩니다. 이미 존재하는 속성을 정의하는 경우 기존 속성이 업데이트됩니다. 업데이트된 자산 모델에서 원본에 있던 속성이 누락되면 해당 속성이 삭제됩니다.

- 최상위 리소스 삭제 없음: 대량 작업으로는 자산 또는 자산 모델이 삭제되지 않습니다. 대량 작업에서는 생성 또는 업데이트만 가능합니다.

대량 작업 사전 요구 사항

이 섹션에서는 로컬 시스템 간에 리소스를 교환하기 위한 AWS Identity and Access Management (IAM) 권한을 비롯한 대량 작업 사전 요구 사항을 설명합니다. AWS 서비스 대량 작업을 시작하기 전에 다음 사전 요구 사항을 완료하십시오.

- Amazon S3 버킷을 생성하여 리소스를 저장합니다. Amazon S3 사용에 대한 자세한 내용은 [Amazon S3란 무엇입니까?](#) 를 참조하십시오.

IAM 권한

대량 작업을 수행하려면 Amazon S3와 로컬 시스템 간에 AWS 리소스를 교환할 수 있는 권한을 포함하는 AWS Identity and Access Management (IAM) 정책을 생성해야 합니다. AWS IoT SiteWise IAM 정책을 만드는 방법에 대한 자세한 내용은 [IAM 정책 만들기](#) 섹션을 참조하세요.

대량 작업을 수행하려면 다음 정책이 필요합니다.

AWS IoT SiteWise 정책

이 정책은 대량 작업에 필요한 AWS IoT SiteWise API 작업에 대한 액세스를 허용합니다.

```
{
  "Sid": "SiteWiseApiAccess",
  "Effect": "Allow",
  "Action": [
    "iotsitewise:CreateAsset",
    "iotsitewise:CreateAssetModel",
    "iotsitewise:UpdateAsset",
    "iotsitewise:UpdateAssetModel",
    "iotsitewise:UpdateAssetProperty",
    "iotsitewise:ListAssets",
    "iotsitewise:ListAssetModels",
    "iotsitewise:ListAssetProperties",
    "iotsitewise:ListAssetModelProperties",
    "iotsitewise:ListAssociatedAssets",
    "iotsitewise:DescribeAsset",
    "iotsitewise:DescribeAssetModel",
    "iotsitewise:DescribeAssetProperty",
    "iotsitewise:AssociateAssets",
    "iotsitewise:DisassociateAssets",
    "iotsitewise:AssociateTimeSeriesToAssetProperty",
    "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
    "iotsitewise:BatchPutAssetPropertyValue",
    "iotsitewise:BatchGetAssetPropertyValue",
    "iotsitewise:TagResource",
    "iotsitewise:UntagResource",
    "iotsitewise:ListTagsForResource",
    "iotsitewise:CreateAssetModelCompositeModel",
    "iotsitewise:UpdateAssetModelCompositeModel",
    "iotsitewise:DescribeAssetModelCompositeModel",
    "iotsitewise>DeleteAssetModelCompositeModel",
    "iotsitewise:ListAssetModelCompositeModels",
    "iotsitewise:ListCompositionRelationships",
    "iotsitewise:DescribeAssetCompositeModel"
  ],
  "Resource": "*"
}
```

AWS IoT TwinMaker 정책

이 정책은 대량 작업에 사용하는 AWS IoT TwinMaker API 작업에 대한 액세스를 허용합니다.

```
{
  "Sid": "MetadataTransferJobApiAccess",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:CreateMetadataTransferJob",
    "iottwinmaker:CancelMetadataTransferJob",
    "iottwinmaker:GetMetadataTransferJob",
    "iottwinmaker:ListMetadataTransferJobs"
  ],
  "Resource": "*"
}
```

아마존 S3 정책

이 정책은 대량 작업을 위한 메타데이터 전송을 위해 Amazon S3 버킷에 대한 액세스를 제공합니다.

For a specific Amazon S3 bucket

대량 작업 메타데이터 작업에 특정 버킷 하나를 사용하는 경우 이 정책은 해당 버킷에 대한 액세스를 제공합니다.

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

To allow any Amazon S3 bucket

대량 작업 메타데이터 작업에 다양한 버킷을 사용하려는 경우 이 정책은 모든 버킷에 대한 액세스를 제공합니다.

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}
```

가져오기 및 내보내기 작업의 문제 해결에 대한 자세한 내용은 [을 참조하십시오](#) [대량 가져오기 및 내보내기 문제 해결](#).

대량 가져오기 작업 실행

대량 가져오기는 메타데이터를 작업 공간으로 이동하는 AWS IoT SiteWise 작업입니다. 예를 들어 대량 가져오기는 로컬 파일 또는 Amazon S3 버킷의 파일에서 AWS IoT SiteWise 작업 공간으로 메타데이터를 이동할 수 있습니다.

1단계: 가져올 파일을 준비합니다.

AWS IoT SiteWise 네이티브 포맷 파일을 다운로드하여 에셋과 에셋 모델을 импорт하십시오. 자세한 내용은 [AWS IoT SiteWise 메타데이터 전송 작업 스키마](#) 섹션을 참조하세요.

2단계: 준비된 파일을 Amazon S3에 업로드

Amazon S3에 파일을 업로드합니다. 자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 [Amazon S3에 파일 업로드](#)를 참조하십시오.

메타데이터 가져오기 (콘솔)

를 사용하여 메타데이터를 대량으로 가져올 수 있습니다. AWS IoT SiteWise Console [1단계: 가져올 파일을 준비합니다](#). 이어서 [2단계: 준비된 파일을 Amazon S3에 업로드](#) 가져올 준비가 된 파일을 준비하세요.

Amazon S3에서 다음으로 데이터 가져오기 AWS IoT SiteWise Console

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 대량 작업 새로 만들기를 선택합니다.
3. 새 가져오기를 선택하여 가져오기 프로세스를 시작합니다.
4. 메타데이터 가져오기 페이지에서:
 - Amazon S3 버킷과 파일을 보려면 [Amazon S3 찾아보기] 를 선택합니다.
 - 준비된 가져오기 파일이 들어 있는 Amazon S3 버킷으로 이동합니다.
 - 가져올 파일을 선택합니다.
 - 선택한 파일을 검토하고 [가져오기] 를 선택합니다.
5. 의 SiteWise 메타데이터 대량 작업 페이지에는 새로 생성된 가져오기 작업이 작업 진행 테이블에 AWS IoT SiteWise Console 표시됩니다.

메타데이터 가져오기 (AWS CLI)

가져오기 작업을 수행하려면 다음 절차를 사용하십시오.

Amazon S3에서 다음으로 데이터 가져오기 AWS CLI

1. 에 따라 가져오려는 리소스를 지정하는 메타데이터 파일을 생성합니다 [AWS IoT SiteWise 메타데이터 전송 작업 스키마](#). Amazon S3 버킷에 이 파일을 저장합니다.

가져올 메타데이터 파일의 예는 을 참조하십시오 [메타데이터 가져오기 예제](#).

2. 이제 요청 본문을 사용하여 JSON 파일을 생성합니다. 요청 본문은 전송 작업의 소스와 대상을 지정합니다. 이 파일은 이전 단계의 파일과는 별개입니다. Amazon S3 버킷을 원본 및 `iotsitewise` 대상으로 지정해야 합니다.

다음 예제는 요청 본문을 보여줍니다.

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
```

```

    "sources": [{
      "type": "s3",
      "s3Configuration": {
        "location": "arn:aws:s3:::your-S3-bucket-name/
your_import_metadata.json"
      }
    }],
    "destination": {
      "type": "iotsitewise"
    }
  }

```

3. 다음 AWS CLI 명령을 CreateMetadataTransferJob 실행하여 를 호출합니다. 이 예시에서는 이전 단계의 요청 본문 파일 이름이 createMetadataTransferJobExport.json 지정됩니다.

```

aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
--cli-input-json file://createMetadataTransferJobImport.json

```

그러면 메타데이터 전송 작업이 생성되고 선택한 리소스를 전송하는 프로세스가 시작됩니다.

대량 내보내기 작업 실행

대량 내보내는 작업 AWS IoT SiteWise 공간에서 Amazon S3 버킷으로 메타데이터를 이동하는 작업입니다.

AWS IoT SiteWise 콘텐츠를 Amazon S3로 대량 내보내는 경우 필터를 지정하여 내보내려는 특정 자산 모델 및 자산을 제한할 수 있습니다.

필터는 JSON 요청의 소스 iotSiteWiseConfiguration 섹션 내 섹션에서 지정해야 합니다.

Note

요청에 여러 필터를 포함할 수 있습니다. 일괄 작업에서는 모든 필터와 일치하는 자산 모델 및 자산을 내보냅니다.

필터를 제공하지 않으면 대량 작업에서 모든 자산 모델 및 자산을 내보냅니다.

Example 필터가 포함된 요청 본문

```

{

```

```

"metadataTransferJobId": "your-transfer-job-id",
"sources": [
  {
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID"
          }
        },
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID",
            "includeAssets": true
          }
        },
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID",
            "includeOffspring": true
          }
        }
      ]
    }
  }
],
"destination": {
  "type": "s3",
  "s3Configuration": {
    "location": "arn:aws:s3:::your-S3-bucket-location"
  }
}
}

```

메타데이터 내보내기 (콘솔)

다음 절차는 콘솔 내보내기 작업을 설명합니다.

에서 내보내기 작업을 생성합니다. AWS IoT SiteWise Console

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 대량 작업 새로 만들기를 선택합니다.

3. 새 내보내기를 선택하여 내보내기 프로세스를 시작합니다.
4. 메타데이터 내보내기 페이지에서:
 - 내보내기 작업의 이름을 입력합니다. Amazon S3 버킷에서 내보낸 파일에 사용되는 이름입니다.
 - 내보낼 리소스를 선택하면 작업에 대한 필터가 설정됩니다.
 - 모든 자산 및 자산 모델을 내보냅니다. 예셋 및 예셋 모델에 필터를 사용합니다.
 - 자산 내보내기. 자산을 기준으로 필터링합니다.
 - 내보내기 필터에 사용할 자산을 선택합니다.
 - (선택 사항) 자손 또는 관련 자산 모델을 추가합니다.
 - 자산 모델을 내보냅니다. 자산 모델을 기준으로 필터링합니다.
 - 내보내기 필터에 사용할 자산 모델을 선택합니다.
 - (선택 사항) 하위 자산이나 관련 자산 또는 둘 다를 추가합니다.
 - 다음을 선택합니다.
 - Amazon S3 버킷으로 이동합니다.
 - Amazon S3 버킷과 파일을 보려면 [Amazon S3 찾아보기] 를 선택합니다.
 - 파일을 배치해야 하는 Amazon S3 버킷으로 이동합니다.
 - 다음을 선택합니다.
 - 내보내기 작업을 검토하고 내보내기를 선택합니다.
5. 의 SiteWise 메타데이터 대량 작업 페이지에는 새로 생성된 가져오기 작업이 작업 진행 테이블에 AWS IoT SiteWise Console 표시됩니다.

메타데이터를 내보낼 때 필터를 사용하는 다양한 방법에 대한 내용은 을 참조하십시오 [메타데이터 내보내기 예제](#).

메타데이터 내보내기 (AWS CLI)

다음 절차는 AWS CLI 내보내기 작업을 설명합니다.

Amazon AWS IoT SiteWise S3로 데이터 내보내기

1. 요청 본문을 사용하여 JSON 파일을 생성합니다. 요청 본문은 전송 작업의 소스와 대상을 지정합니다. 다음 예시는 요청 본문의 예를 보여줍니다.

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iotsitewise"
  }],
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3::your-S3-bucket-location"
    }
  }
}
```

Amazon S3 버킷을 메타데이터 전송 작업의 대상으로 지정해야 합니다.

Note

이 예제는 모든 자산 모델 및 자산을 내보냅니다. 내보내기를 특정 자산 모델 또는 자산으로 제한하려면 요청 본문에 필터를 포함시킬 수 있습니다. 내보내기 필터 적용에 대한 자세한 내용은 [참조하십시오 메타데이터 내보내기 예제](#).

2. 다음 단계에서 사용할 수 있도록 요청 본문 파일을 저장합니다. 이 예시에서 파일의 이름은 `createMetadataTransferJobExport.json`으로 지정됩니다.
3. 다음 AWS CLI 명령을 `CreateMetadataTransferJob` 실행하여 를 호출합니다.

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
  --cli-input-json file://createMetadataTransferJobExport.json
```

입력 JSON 파일을 자체 전송 파일 `createMetadataTransferJobExport.json` 이름으로 바꿉니다.

작업 진행 상황 추적 및 오류 처리

대량 프로세스 작업은 처리하는 데 시간이 걸립니다. 각 작업은 요청을 AWS IoT SiteWise 받은 순서대로 처리됩니다. 각 one-at-a-time 계정별로 처리됩니다. 작업이 완료되면 대기열에 있는 다음 작업자가 자동으로 처리를 시작합니다. AWS IoT SiteWise 작업을 비동기적으로 해결하고 작업이 진행됨에 따라 각 작업의 상태를 업데이트합니다. 각 작업에는 리소스 상태가 포함된 상태 필드와 오류 메시지 (해당하는 경우)가 있습니다.

상태는 다음 값 중 하나일 수 있습니다.

- VALIDATING— 제출된 파일 형식 및 내용을 포함하여 작업을 검증합니다.
- PENDING— 작업이 대기열에 있습니다. AWS IoT SiteWise 콘솔에서 이 상태의 작업을 취소할 수 있지만 다른 모든 상태는 끝까지 계속됩니다.
- RUNNING— 작업 처리 중. 가져오기 파일에 정의된 대로 리소스를 생성 및 업데이트하거나 선택한 내보내기 작업 필터를 기반으로 리소스를 내보냅니다. 취소한 경우 이 작업에서 가져온 리소스는 삭제되지 않습니다. 자세한 정보는 [작업 진행 상황 및 세부 정보 검토 \(콘솔\)](#)을 참조하세요.
- CANCELLING— 작업이 현재 취소되고 있습니다.
- ERROR— 하나 이상의 리소스를 처리하지 못했습니다. 자세한 내용은 세부 작업 보고서를 확인하세요. 자세한 정보는 [오류 세부 정보 검사 \(콘솔\)](#)을 참조하세요.
- COMPLETED— 작업이 오류 없이 완료되었습니다.
- CANCELLED— 작업이 취소되고 대기열에 추가되지 않습니다. RUNNING작업을 취소한 경우 취소 당시 이 작업에서 이미 가져온 리소스는 삭제되지 않습니다. AWS IoT SiteWise

주제

- [작업 진행 상황 추적](#)
- [오류 검사](#)

작업 진행 상황 추적

작업 진행 상황 및 세부 정보 검토 (콘솔)

대량 작업을 [메타데이터 내보내기 \(콘솔\)](#) 시작하려면 [메타데이터 가져오기 \(콘솔\)](#) 또는 을 참조하십시오.

AWS IoT SiteWise 콘솔의 Job 진행 상황 개요:

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 대량 작업 새로 만들기를 선택합니다.
3. AWS IoT SiteWise 콘솔의 작업 진행 테이블에는 대량 작업 목록이 표시됩니다.
4. Job type 열은 익스포트 작업인지 임포트 작업인지를 설명합니다. 가져온 날짜 열에는 작업이 시작된 날짜가 표시됩니다.
5. 상태 열에는 작업 상태가 표시됩니다. 작업을 선택하여 작업에 대한 세부 정보를 볼 수 있습니다.

6. 선택한 작업에 성공 시 성공이 표시되고, 작업이 실패하면 실패 목록이 표시됩니다. 각 리소스 유형과 함께 오류 설명도 표시됩니다.

AWS IoT SiteWise 콘솔의 Job 세부 정보 개요:

AWS IoT SiteWise 콘솔의 작업 진행 테이블에는 대량 작업 목록이 표시됩니다.

1. 작업을 선택하면 자세한 내용을 볼 수 있습니다.
2. 가져오기 작업의 경우는 가져오기 파일의 Amazon S3 위치를 Data source ARN 나타냅니다.
3. 내보내기 작업의 경우는 내보내기 후 파일의 Amazon S3 위치를 Data destination ARN 나타냅니다.
4. Status 및 Status reason에는 현재 작업에 대한 추가 세부 정보가 제공됩니다. 자세한 내용은 [작업 진행 상황 추적 및 오류 처리](#) 섹션을 참조하세요.
5. 는 프로세스 대기열에서의 작업 위치를 Queued position 나타냅니다. 작업은 한 번에 하나씩 처리됩니다. 대기열 위치가 1이면 작업이 다음에 처리된다는 의미입니다.
6. 작업 세부 정보 페이지에는 작업 진행 횟수도 표시됩니다.
 - 작업 진행 횟수 유형은 다음과 같습니다.
 - i. Total resources— 이전 프로세스에 있는 총 자산 수를 나타냅니다.
 - ii. Succeeded— 프로세스 중에 성공적으로 이전된 자산의 수를 나타냅니다.
 - iii. Failed— 프로세스 중에 실패한 자산 수를 나타냅니다.
 - iv. Skipped— 프로세스 중에 건너뛰었던 자산의 수를 나타냅니다.
7. 작업 상태가 PENDING 또는 VALIDATING 이면 모든 작업 진행 상태가 로 표시됩니다. - 이는 작업 진행 횟수가 평가 중임을 나타냅니다.
8. 작업 상태에는 처리를 위해 제출된 작업 Total resources 수가 RUNNING 표시됩니다. 세부 개수 (Succeeded, Failed, 및 Skipped) 는 처리된 리소스에 적용됩니다. 작업 상태가 또는 일 때까지는 세부 카운트의 Total resources 합계가 수보다 작습니다. COMPLETED ERROR
9. 작업 상태가 OR인 경우 Total resources 개수는 세부 개수 (ERRORSucceeded,, 및) 의 합계와 같습니다. COMPLETED Failed Skipped
10. 작업 상태가 ERROR 인 경우 Job failure 테이블에서 특정 오류 및 실패에 대한 세부 정보를 확인하십시오. 자세한 내용은 [오류 세부 정보 검사 \(콘솔\)](#) 섹션을 참조하세요.

작업 진행 상황 및 세부 정보 검토 (AWS CLI)

일괄 작업을 시작한 후 다음 API 작업을 사용하여 작업 상태를 확인하거나 업데이트할 수 있습니다.

- 특정 작업에 대한 정보를 검색하려면 [GetMetadataTransferJob](#) API 작업을 사용하십시오.

GetMetadataTransferJob API를 사용하여 정보 검색:

- 전송 작업을 생성하고 실행합니다. `GetMetadataTransferJob` API를 호출하세요.

Example AWS CLI 명령:

```
aws iottwinmaker get-metadata-transfer-job \
  --metadata-transfer-job-id your_metadata_transfer_job_id \
  --region your_region
```

- `GetMetadataTransferJob` API는 다음 파라미터가 포함된 `MetadataTransferJobProgress` 객체를 반환합니다.

- `SuccesedCount` — 프로세스에서 성공적으로 전송된 자산의 수를 나타냅니다.
- `FailedCount` — 프로세스 중에 실패한 자산의 수를 나타냅니다.
- `SkippedCount` — 프로세스 중에 건너뛰었던 자산의 수를 나타냅니다.
- 총 개수 — 이전 프로세스에 포함된 총 자산 수를 나타냅니다.

이 매개변수는 작업 진행 상태를 나타냅니다. 상태가 `RUNNING` 인 경우 아직 처리해야 할 리소스 수를 추적하는 데 도움이 됩니다.

스키마 유효성 검사 오류가 발생하거나 `FailedCount`가 1보다 크거나 같으면 작업 진행 상태로 바뀝니다. `ERROR` 작업에 대한 전체 오류 보고서는 Amazon S3 버킷에 저장됩니다. 자세한 내용은 [오류 검사](#) 섹션을 참조하세요.

- 현재 작업을 나열하려면 [ListMetadataTransferJobs](#) API 작업을 사용하십시오.

JSON 파일을 사용하여 반환된 작업을 현재 상태를 기준으로 필터링합니다. 다음 절차를 참조하십시오.

- 사용할 필터를 지정하려면 사용하려는 AWS CLI 입력 JSON 파일을 생성하십시오.

```
{
  "sourceType": "s3",
  "destinationType": "iottwinmaker",
```

```
"filters": [{
  "state": "COMPLETED"
}]
}
```

유효한 state 값 목록은 AWS IoT TwinMaker API 참조 안내서의 [ListMetadataTransferJobs 필터](#)를 참조하십시오.

- 다음 AWS CLI 예제 명령에서 JSON 파일을 인수로 사용합니다.

```
aws iottwinmaker list-metadata-transfer-job --region your_region \
  --cli-input-json file://ListMetadataTransferJobsExample.json
```

- 작업을 취소하려면 [CancelMetadataTransferJob](#) API 작업을 사용합니다. 이 API는 이미 내보내거나 가져온 리소스에 영향을 주지 않고 특정 메타데이터 전송 작업을 취소합니다.

```
aws iottwinmaker cancel-metadata-transfer-job \
  --region your_region \
  --metadata-transfer-job-id job-to-cancel-id
```

오류 검사

오류 세부 정보 검사 (콘솔)

AWS IoT SiteWise 콘솔의 오류 세부 정보:

- [AWS IoT SiteWise 콘솔](#)로 이동합니다.
- 대량 작업 목록은 의 작업 진행률 표를 참조하십시오. AWS IoT SiteWise Console
- 작업을 선택하면 작업 세부 정보를 볼 수 있습니다.
- 작업 상태가 OR인 경우 Total resources 개수는 세부 개수 (ERROR, SucceededFailed, 및 Skipped) 의 합계와 같습니다. COMPLETED
- 작업 상태가 ERROR 인 경우 Job failure 테이블에서 특정 오류 및 실패에 대한 세부 정보를 확인하십시오.
- Job failure 테이블에는 작업 보고서의 내용이 표시됩니다. 이 Resource type 필드는 다음과 같은 오류 또는 실패의 위치를 나타냅니다.

- 예를 들어 Resource type 필드에 유효성 검사 오류가 있으면 가져오기 템플릿과 메타데이터 스키마 파일 형식이 일치하지 않는 것입니다. Bulk operations template 자세한 정보는 [AWS IoT SiteWise 메타데이터 전송 작업 스키마](#)을 참조하세요.
- AssetResource type 필드에 오류가 발생하면 다른 자산과의 충돌로 인해 해당 자산이 생성되지 않았음을 나타냅니다. AWS IoT SiteWise 리소스 오류 및 충돌에 대한 자세한 내용은 [일반적인 오류](#)를 참조하십시오.

오류 세부 정보 검사 (AWS CLI)

전송 작업 중에 발생한 오류를 처리하고 진단하려면 GetMetadataTransferJob API 작업 사용에 대한 다음 절차를 참조하십시오.

1. 전송 작업을 생성하고 실행한 후 [GetMetadataTransferJob](#) 다음을 호출합니다.

```
aws iottwinmaker get-metadata-transfer-job \
  --metadata-transfer-job-id your_metadata_transfer_job_id \
  --region us-east-1
```

2. 작업 상태가 켜지면 작업 결과 확인을 시작할 수 있습니다. COMPLETED
3. GetMetadataTransferJob 호출하면 라는 [MetadataTransferJobProgress](#) 객체가 반환됩니다.

MetadataTransferJobProgress 객체에는 다음과 같은 매개변수가 포함됩니다.

- FailedCount: 전송 프로세스 중에 실패한 자산의 수를 나타냅니다.
 - SkippedCount: 이전 프로세스 중에 건너뛰었던 자산의 수를 나타냅니다.
 - 성공 개수: 이전 프로세스 중에 성공한 자산 수를 나타냅니다.
 - 총계: 이전 프로세스에 관련된 자산의 총 수를 나타냅니다.
4. 또한 API 호출은 미리 서명된 URL이 포함된 요소를 reportUrl 반환합니다. 전송 작업에 추가 조사가 필요한 문제가 있는 경우 이 URL을 방문하세요.

메타데이터 가져오기 예제

이 섹션에서는 한 번의 대량 가져오기 작업으로 자산 모델 및 자산을 가져오기 위한 메타데이터 파일을 생성하는 방법을 보여 줍니다.

대량 가져오기의 예

한 번의 대량 가져오기 작업으로 여러 자산 모델 및 자산을 가져올 수 있습니다. 다음 예제는 이를 위한 메타데이터 파일을 생성하는 방법을 보여줍니다.

이 예제 시나리오에서는 작업실에 산업용 로봇이 포함된 다양한 작업장이 있습니다.

이 예제에서는 두 가지 자산 모델을 정의합니다.

- RobotModel1: 이 자산 모델은 작업 현장에 있는 특정 유형의 로봇을 나타냅니다. 로봇에는 측정 속성이 있습니다 Temperature.
- WorkCell1: 이 자산 모델은 작업 현장 중 한 곳에 있는 로봇 컬렉션을 나타냅니다. 자산 모델은 계층을 정의하여 작업셀에 로봇이 포함된 관계를 나타냅니다. robotHierarchyOEM1

이 예제에서는 일부 자산도 정의합니다.

- WorkCell11: 보스턴 사이트 내 작업실
- RobotArm123456: 작업실 안에 있는 로봇
- RobotArm987654: 작업실 안에 있는 또 다른 로봇

다음 JSON 메타데이터 파일은 이러한 자산 모델 및 자산을 정의합니다. 이 메타데이터를 사용하여 대량 가져오기를 실행하면 계층 관계를 AWS IoT SiteWise 포함하여 자산 모델 및 자산이 생성됩니다.

가져오기용 메타데이터 파일

```
{
  "assetModels": [
    {
      "assetModelExternalId": "Robot.OEM1.3536",
      "assetModelName": "RobotModel1",
      "assetModelProperties": [
        {
          "dataType": "DOUBLE",
          "externalId": "Temperature",
          "name": "Temperature",
          "type": {
            "measurement": {
              "processingConfig": {
                "forwardingConfig": {
                  "state": "ENABLED"
                }
              }
            }
          }
        }
      ]
    }
  ]
}
```



```

    }
  ]
}
}
}

```

모델 및 자산의 초기 온보딩 예제

이 예제 시나리오에서는 회사에 산업용 로봇이 있는 다양한 작업장이 있습니다.

이 예제에서는 여러 자산 모델을 정의합니다.

- **Sample_Enterprise**— 이 자산 모델은 사이트가 속한 회사를 나타냅니다. 자산 모델은 사이트와 기업 간의 관계를 나타내는 계층 구조를 정의합니다. **Enterprise to Site**
- **Sample_Site**— 이 자산 모델은 회사 내 제조 현장을 나타냅니다. 자산 모델은 라인과 사이트 간의 관계를 나타내는 계층 구조를 정의합니다. **Site to Line**
- **Sample_Welding Line**— 이 자산 모델은 작업장 내의 조립 라인을 나타냅니다. 자산 모델은 로봇과 라인 간의 관계를 나타내는 계층 구조를 정의합니다. **Line to Robot**
- **Sample_Welding Robot**— 이 자산 모델은 작업 현장의 특정 유형의 로봇을 나타냅니다.

또한 이 예제에서는 자산 모델을 기반으로 자산을 정의합니다.

- **Sample_AnyCompany Motor**— 이 자산은 **Sample_Enterprise** 자산 모델에서 생성됩니다.
- **Sample_Chicago**— 이 자산은 **Sample_Site** 자산 모델에서 생성됩니다.
- **Sample_Welding Line 1**— 이 자산은 **Sample_Welding Line** 자산 모델에서 생성됩니다.
- **Sample_Welding Robot 1**— 이 자산은 **Sample_Welding Robot** 자산 모델에서 생성됩니다.
- **Sample_Welding Robot 2**— 이 자산은 **Sample_Welding Robot** 자산 모델에서 생성됩니다.

다음 JSON 메타데이터 파일은 이러한 자산 모델 및 자산을 정의합니다. 이 메타데이터를 사용하여 대량 가져오기를 실행하면 계층 관계를 AWS IoT SiteWise 포함하여 자산 모델 및 자산이 생성됩니다.

가져올 에셋 및 모델을 온보딩하기 위한 JSON 파일로

```

{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",

```

```

"assetModelName": "Sample_Welding Robot",
"assetModelProperties": [
  {
    "dataType": "STRING",
    "externalId": "External_Id_Welding_Robot_Serial_Number",
    "name": "Serial Number",
    "type": {
      "attribute": {
        "defaultValue": "-"
      }
    },
    "unit": "-"
  },
  {
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Cycle_Count",
    "name": "CycleCount",
    "type": {
      "measurement": {}
    },
    "unit": "EA"
  },
  {
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Joint_1_Current",
    "name": "Joint 1 Current",
    "type": {
      "measurement": {}
    },
    "unit": "Amps"
  },
  {
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
    "name": "Max Joint 1 Current",
    "type": {
      "metric": {
        "expression": "max(joint1current)",
        "variables": [
          {
            "name": "joint1current",
            "value": {
              "propertyExternalId":
                "External_Id_Welding_Robot_Joint_1_Current"
            }
          }
        ]
      }
    }
  }
]

```

```

        }
    },
    ],
    "window": {
        "tumbling": {
            "interval": "5m"
        }
    }
},
"unit": "Amps"
}
],
{
"assetModelExternalId": "External_Id_Welding_Line",
"assetModelName": "Sample_Welding Line",
"assetModelProperties": [
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Line_Availability",
        "name": "Availability",
        "type": {
            "measurement": {}
        },
        "unit": "%"
    }
],
"assetModelHierarchies": [
    {
        "externalId": "External_Id_Welding_Line_TO_Robot",
        "name": "Line to Robot",
        "childAssetModelExternalId": "External_Id_Welding_Robot"
    }
]
},
{
"assetModelExternalId": "External_Id_Site",
"assetModelName": "Sample_Site",
"assetModelProperties": [
    {
        "dataType": "STRING",
        "externalId": "External_Id_Site_Street_Address",
        "name": "Street Address",

```

```

        "type": {
            "attribute": {
                "defaultValue": "-"
            }
        },
        "unit": "-"
    }
],
"assetModelHierarchies": [
    {
        "externalId": "External_Id_Site_T0_Line",
        "name": "Site to Line",
        "childAssetModelExternalId": "External_Id_Welding_Line"
    }
]
},
{
    "assetModelExternalId": "External_Id_Enterprise",
    "assetModelName": "Sample_Enterprise",
    "assetModelProperties": [
        {
            "dataType": "STRING",
            "name": "Company Name",
            "externalId": "External_Id_Enterprise_Company_Name",
            "type": {
                "attribute": {
                    "defaultValue": "-"
                }
            },
            "unit": "-"
        }
    ],
    "assetModelHierarchies": [
        {
            "externalId": "External_Id_Enterprise_T0_Site",
            "name": "Enterprise to Site",
            "childAssetModelExternalId": "External_Id_Site"
        }
    ]
}
],
"assets": [
    {
        "assetExternalId": "External_Id_Welding_Robot_1",

```

```

    "assetName": "Sample_Welding Robot 1",
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [
      {
        "externalId": "External_Id_Welding_Robot_Serial_Number",
        "attributeValue": "S1000"
      },
      {
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "alias": "AnyCompany/Chicago/Welding Line/S1000/Count"
      },
      {
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "alias": "AnyCompany/Chicago/Welding Line/S1000/1/Current"
      }
    ]
  },
  {
    "assetExternalId": "External_Id_Welding_Robot_2",
    "assetName": "Sample_Welding Robot 2",
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [
      {
        "externalId": "External_Id_Welding_Robot_Serial_Number",
        "attributeValue": "S2000"
      },
      {
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "alias": "AnyCompany/Chicago/Welding Line/S2000/Count"
      },
      {
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "alias": "AnyCompany/Chicago/Welding Line/S2000/1/Current"
      }
    ]
  },
  {
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetProperties": [
      {
        "externalId": "External_Id_Welding_Line_Availability",
        "alias": "AnyCompany/Chicago/Welding Line/Availability"
      }
    ]
  }
}

```

```

    }
  ],
  "assetHierarchies": [
    {
      "externalId": "External_Id_Welding_Line_T0_Robot",
      "childAssetExternalId": "External_Id_Welding_Robot_1"
    },
    {
      "externalId": "External_Id_Welding_Line_T0_Robot",
      "childAssetExternalId": "External_Id_Welding_Robot_2"
    }
  ]
},
{
  "assetExternalId": "External_Id_Site_Chicago",
  "assetName": "Sample_Chicago",
  "assetModelExternalId": "External_Id_Site",
  "assetHierarchies": [
    {
      "externalId": "External_Id_Site_T0_Line",
      "childAssetExternalId": "External_Id_Welding_Line_1"
    }
  ]
},
{
  "assetExternalId": "External_Id_Enterprise_AnyCompany",
  "assetName": "Sample_AnyEnterprise Motor",
  "assetModelExternalId": "External_Id_Enterprise",
  "assetHierarchies": [
    {
      "externalId": "External_Id_Enterprise_T0_Site",
      "childAssetExternalId": "External_Id_Site_Chicago"
    }
  ]
}
]
}

```

다음 스크린샷은 이전 코드 예제를 AWS IoT SiteWise Console 실행한 후에 표시되는 모델의 스크린샷입니다.

Models (4)

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Filter instances

Name	Status	Model type	Date created	Date modified
Sample_Enterprise	ACTIVE	Asset model	November 10, 2023 at 11:22:13 (UT...)	November 10, 202...
Sample_Site	ACTIVE	Asset model	November 10, 2023 at 11:21:57 (UT...)	November 10, 202...
Sample_Welding Line	ACTIVE	Asset model	November 10, 2023 at 11:21:40 (UT...)	November 10, 202...
Sample_Welding Robot	ACTIVE	Asset model	November 10, 2023 at 11:21:24 (UT...)	November 10, 202...

다음 스크린샷은 이전 코드 예제를 실행한 AWS IoT SiteWise Console 후에 표시되는 모델, 자산 및 계층 구조입니다.

Assets (1)

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Filter top level assets

Name	Description	Status	Date created	Date modified
Sample_AnyEnterprise Motor		ACTIVE	November 10, 2023 at 11:23:06 (UTC-5:00)	November 10, 2023 at 11:23:06 (UTC-...
Sample_Chicago		ACTIVE	November 10, 2023 at 11:22:57 (UTC-5:00)	November 10, 2023 at 11:22:57 (UTC-...
Sample_Welding Line 1		ACTIVE	November 10, 2023 at 11:22:48 (UTC-5:00)	November 10, 2023 at 11:22:48 (UTC-...
Sample_Welding Robot 1		ACTIVE	November 10, 2023 at 11:22:39 (UTC-5:00)	November 10, 2023 at 11:22:39 (UTC-...
Sample_Welding Robot 2		ACTIVE	November 10, 2023 at 11:22:30 (UTC-5:00)	November 10, 2023 at 11:22:30 (UTC-...

추가 자산 온보딩 예제

이 예시에서는 계정의 기존 자산 모델로 가져올 추가 자산을 정의합니다.

- Sample_Welding Line 2— 이 자산은 Sample_Welding Line 자산 모델에서 생성됩니다.
- Sample_Welding Robot 3— 이 자산은 Sample_Welding Robot 자산 모델에서 생성됩니다.
- Sample_Welding Robot 4— 이 자산은 Sample_Welding Robot 자산 모델에서 생성됩니다.

이 예제의 초기 자산을 만들려면 [참조하십시오 모델 및 자산의 초기 온보딩 예제.](#)

다음 JSON 메타데이터 파일은 이러한 자산 모델 및 자산을 정의합니다. 이 메타데이터를 사용하여 대량 가져오기를 실행하면 계층 관계를 AWS IoT SiteWise 포함하여 자산 모델 및 자산이 생성됩니다.

추가 자산을 온보딩하기 위한 JSON 파일

```
{
  "assets": [
    {
      "assetExternalId": "External_Id_Welding_Robot_3",
      "assetName": "Sample_Welding Robot 3",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
        {
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "attributeValue": "S3000"
        },
        {
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/Count"
        },
        {
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/1/Current"
        }
      ]
    },
    {
      "assetExternalId": "External_Id_Welding_Robot_4",
      "assetName": "Sample_Welding Robot 4",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
        {
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "attributeValue": "S4000"
        },
        {
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "alias": "AnyCompany/Chicago/Welding Line/S4000/Count"
        },
        {
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "alias": "AnyCompany/Chicago/Welding Line/S4000/1/Current"
        }
      ]
    },
    {
```

```

    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
      {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "childAssetExternalId": "External_Id_Welding_Robot_1"
      },
      {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "childAssetExternalId": "External_Id_Welding_Robot_2"
      },
      {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "childAssetExternalId": "External_Id_Welding_Robot_3"
      }
    ]
  },
  {
    "assetExternalId": "External_Id_Welding_Line_2",
    "assetName": "Sample_Welding Line 2",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
      {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "childAssetExternalId": "External_Id_Welding_Robot_4"
      }
    ]
  },
  {
    "assetExternalId": "External_Id_Site_Chicago",
    "assetName": "Sample_Chicago",
    "assetModelExternalId": "External_Id_Site",
    "assetHierarchies": [
      {
        "externalId": "External_Id_Site_T0_Line",
        "childAssetExternalId": "External_Id_Welding_Line_1"
      },
      {
        "externalId": "External_Id_Site_T0_Line",
        "childAssetExternalId": "External_Id_Welding_Line_2"
      }
    ]
  }
}

```

```
]
}
```

다음 스크린샷은 이전 코드 예제를 실행한 AWS IoT SiteWise Console 후 에 표시되는 모델, 자산 및 계층 구조입니다.

The screenshot shows the AWS IoT SiteWise Assets console. At the top, there's a breadcrumb 'IoT SiteWise > Assets' and a 'Create asset' button. Below is a search bar 'Filter top level assets' and a table of assets. The table has columns for Name, Description, Status, Date created, and Date modified. The assets are listed in a tree structure:

- Sample_AnyCompany Motor (ACTIVE, Nov 09, 2023 at 19:18:05)
 - Sample_Chicago (ACTIVE, Nov 09, 2023 at 19:17:56)
 - Sample_Welding Line 1 (ACTIVE, Nov 09, 2023 at 19:17:48)
 - Sample_Welding Robot 2 (ACTIVE, Nov 09, 2023 at 19:17:39)
 - Sample_Welding Robot 3 (ACTIVE, Nov 09, 2023 at 20:40:02)
 - Sample_Welding Robot 1 (ACTIVE, Nov 09, 2023 at 19:17:30)
 - Sample_Welding Line 2 (ACTIVE, Nov 09, 2023 at 20:40:20)
 - Sample_Welding Robot 4 (ACTIVE, Nov 09, 2023 at 20:40:11)

새 속성 온보딩 예제

이 예제는 기존 자산 모델의 새 속성을 정의합니다. 추가 자산 및 모델을 [추가 자산 온보딩 예제](#) 온보딩하려면 을 참조하십시오.

- **Joint 1 Temperature**— 이 속성이 Sample_Welding Robot 자산 모델에 추가됩니다. 이 새 속성은 자산 모델에서 생성된 각 자산에도 전파됩니다 Sample_Welding Robot.

기존 자산 모델에 새 속성을 추가하려면 다음 JSON 메타데이터 파일 예제를 참조하십시오. JSON에서 볼 수 있듯이 기존 Sample_Welding Robot 자산 모델 정의 전체가 새 속성과 함께 제공되어야 합니다. 기존 정의의 전체 속성 목록이 제공되지 않은 경우 생략된 속성을 AWS IoT SiteWise 삭제합니다.

새 속성을 온보딩하기 위한 JSON 파일

이 예제는 자산 모델에 새 속성을 Joint 1 Temperature 추가합니다.

```
{
  "assetModels": [
    {
```

```

"assetModelExternalId": "External_Id_Welding_Robot",
"assetModelName": "Sample_Welding Robot",
"assetModelProperties": [
  {
    "dataType": "STRING",
    "externalId": "External_Id_Welding_Robot_Serial_Number",
    "name": "Serial Number",
    "type": {
      "attribute": {
        "defaultValue": "-"
      }
    },
    "unit": "-"
  },
  {
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Cycle_Count",
    "name": "CycleCount",
    "type": {
      "measurement": {}
    },
    "unit": "EA"
  },
  {
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Joint_1_Current",
    "name": "Joint 1 Current",
    "type": {
      "measurement": {}
    },
    "unit": "Amps"
  },
  {
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
    "name": "Max Joint 1 Current",
    "type": {
      "metric": {
        "expression": "max(joint1current)",
        "variables": [
          {
            "name": "joint1current",
            "value": {

```

```

        "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
    }
    },
    "window": {
        "tumbling": {
            "interval": "5m"
        }
    }
},
"unit": "Amps"
},
{
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Joint_1_Temperature",
    "name": "Joint 1 Temperature",
    "type": {
        "measurement": {}
    },
    "unit": "degC"
}
]
}
]
}

```

메타데이터 내보내기 예제

AWS IoT SiteWise 콘텐츠를 Amazon S3로 대량 내보내는 경우 필터를 지정하여 내보내려는 특정 자산 모델 및 자산을 제한할 수 있습니다.

요청 본문의 `iotSiteWiseConfiguration` 섹션 내 `sources` 섹션에서 필터를 지정합니다.

Note

필터를 여러 개 포함할 수 있습니다. 일괄 작업에서는 필터 중 하나와 일치하는 모든 자산 모델 또는 자산을 내보냅니다.

필터를 제공하지 않으면 작업에서 모든 자산 모델 및 자산을 내보냅니다.

```
{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [{
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [{
        "list of filters"
      }]
    }
  ]},
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::your-S3-bucket-location"
    }
  }
}
```

자산 모델별 필터링

특정 자산 모델을 필터링할 수 있습니다. 해당 모델을 사용하는 모든 자산을 포함하거나 계층 구조 내의 모든 자산 모델을 포함할 수도 있습니다. 자산과 계층 구조를 모두 포함할 수는 없습니다.

계층 구조에 대한 자세한 정보는 [자산 모델 계층 구조 정의](#) 단원을 참조하십시오.

Asset model

이 필터에는 지정된 자산 모델이 포함됩니다.

```
"filterByAssetModel": {
  "assetModelId": "asset model ID"
}
```

Asset model and its assets

이 필터에는 지정된 자산 모델과 해당 자산 모델을 사용하는 모든 자산이 포함됩니다.

```
"filterByAssetModel": {
  "assetModelId": "asset model ID",
  "includeAssets": true
}
```

Asset model and its hierarchy

이 필터에는 지정된 자산 모델과 해당 계층 구조의 모든 관련 자산 모델이 포함됩니다.

```
"filterByAssetModel": {
  "assetModelId": "asset model ID",
  "includeOffspring": true
}
```

자산별 필터링

특정 자산을 필터링할 수 있습니다. 자산 모델 또는 계층 구조 내의 모든 관련 자산을 포함할 수도 있습니다. 자산 모델과 계층 구조를 모두 포함할 수는 없습니다.

계층 구조에 대한 자세한 정보는 [자산 모델 계층 구조 정의](#) 단원을 참조하십시오.

Asset

이 필터에는 지정된 자산이 포함됩니다.

```
"filterByAsset": {
  "assetId": "asset ID"
}
```

Asset and its asset model

이 필터에는 지정된 자산과 해당 자산이 사용하는 자산 모델이 포함됩니다.

```
"filterByAsset": {
  "assetId": "asset ID",
  "includeAssetModel": true
}
```

Asset and its hierarchy

이 필터에는 지정된 자산과 해당 계층 구조의 모든 관련 자산이 포함됩니다.

```
"filterByAsset": {
  "assetId": "asset ID",
  "includeOffspring": true
}
```

}

AWS IoT SiteWise 메타데이터 전송 작업 스키마

자체 대량 가져오기 및 내보내기 작업을 수행할 때 AWS IoT SiteWise 메타데이터 전송 작업 스키마를 참조용으로 사용하세요.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTSiteWise",
  "description": "Metadata transfer job resource schema for IoTSiteWise",
  "definitions": {
    "Name": {
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "Description": {
      "type": "string",
      "minLength": 1,
      "maxLength": 2048,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "ID": {
      "type": "string",
      "minLength": 36,
      "maxLength": 36,
      "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
    },
    "ExternalId": {
      "type": "string",
      "minLength": 2,
      "maxLength": 128,
      "pattern": "[a-zA-Z0-9_][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9_]+"
    },
    "AttributeValue": {
      "description": "The value of the property attribute.",
      "type": "string",
      "minLength": 1,
      "maxLength": 1024,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    }
  }
}
```

```

    },
    "PropertyUnit": {
      "description": "The unit of measure (such as Newtons or RPM) of the asset
property.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "PropertyAlias": {
      "description": "The property alias that identifies the property.",
      "type": "string",
      "minLength": 1,
      "maxLength": 1000,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "AssetProperty": {
      "description": "The asset property's definition, alias, unit, and notification
state.",
      "type": "object",
      "additionalProperties": false,
      "anyOf": [
        {
          "required": [
            "id"
          ]
        },
        {
          "required": [
            "externalId"
          ]
        }
      ],
      "properties": {
        "id": {
          "description": "The ID of the asset property.",
          "$ref": "#/definitions/ID"
        },
        "externalId": {
          "description": "The ExternalID of the asset property.",
          "$ref": "#/definitions/ExternalId"
        },
        "alias": {
          "$ref": "#/definitions/PropertyAlias"
        }
      }
    }
  }
}

```

```

    },
    "unit": {
      "$ref": "#/definitions/PropertyUnit"
    },
    },
    "attributeValue": {
      "$ref": "#/definitions/AttributeValue"
    },
    },
    "retainDataOnAliasChange": {
      "type": "string",
      "default": "TRUE",
      "enum": [
        "TRUE",
        "FALSE"
      ]
    },
    },
    "propertyNotificationState": {
      "description": "The MQTT notification state (ENABLED or DISABLED) for this
asset property.",
      "type": "string",
      "enum": [
        "ENABLED",
        "DISABLED"
      ]
    }
  }
},
"AssetHierarchy": {
  "description": "A hierarchy specifies allowed parent/child asset relationships.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id",
        "childAssetId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetId"
      ]
    }
  ],
},
{

```

```
    "required": [
      "id",
      "childAssetExternalId"
    ],
  },
  {
    "required": [
      "externalId",
      "childAssetExternalId"
    ]
  }
],
"properties": {
  "id": {
    "description": "The ID of a hierarchy in the parent asset's model.",
    "$ref": "#/definitions/ID"
  },
  "externalId": {
    "description": "The ExternalID of a hierarchy in the parent asset's model.",
    "$ref": "#/definitions/ExternalId"
  },
  "childAssetId": {
    "description": "The ID of the child asset to be associated.",
    "$ref": "#/definitions/ID"
  },
  "childAssetExternalId": {
    "description": "The ExternalID of the child asset to be associated.",
    "$ref": "#/definitions/ExternalId"
  }
}
},
"Tag": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "key",
    "value"
  ],
  "properties": {
    "key": {
      "type": "string"
    },
    "value": {
      "type": "string"
    }
  }
}
```

```

    }
  }
},
"AssetModelType": {
  "type": "string",
  "default": null,
  "enum": [
    "ASSET_MODEL",
    "COMPONENT_MODEL"
  ]
},
"AssetModelCompositeModel": {
  "description": "Contains a composite model definition in an asset model. This composite model definition is applied to all assets created from the asset model.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id"
      ]
    },
    {
      "required": [
        "externalId"
      ]
    }
  ],
  "required": [
    "name",
    "type"
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset model composite model.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model composite model.",
      "$ref": "#/definitions/ExternalId"
    },
    "parentId": {
      "description": "The ID of the parent asset model composite model.",
      "$ref": "#/definitions/ID"
    }
  }
}

```

```

    },
    "parentExternalId": {
      "description": "The ExternalID of the parent asset model composite model.",
      "$ref": "#/definitions/ExternalId"
    },
    },
    "composedAssetModelId": {
      "description": "The ID of the composed asset model.",
      "$ref": "#/definitions/ID"
    },
    },
    "composedAssetModelExternalId": {
      "description": "The ExternalID of the composed asset model.",
      "$ref": "#/definitions/ExternalId"
    },
    },
    "description": {
      "description": "A description for the asset composite model.",
      "$ref": "#/definitions/Description"
    },
    },
    "name": {
      "description": "A unique, friendly name for the asset composite model.",
      "$ref": "#/definitions/Name"
    },
    },
    "type": {
      "description": "The type of the composite model. For alarm composite models,
this type is AWS/ALARM.",
      "$ref": "#/definitions/Name"
    },
    },
    "properties": {
      "description": "The property definitions of the asset model.",
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetModelProperty"
      }
    }
  }
},
"AssetModelProperty": {
  "description": "Contains information about an asset model property.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id"
      ]
    }
  ]
}

```

```

    },
    {
      "required": [
        "externalId"
      ]
    }
  ],
  "required": [
    "name",
    "dataType",
    "type"
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset model property.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model property.",
      "$ref": "#/definitions/ExternalId"
    },
    "name": {
      "description": "The name of the asset model property.",
      "$ref": "#/definitions/Name"
    },
    "dataType": {
      "description": "The data type of the asset model property.",
      "$ref": "#/definitions/DataType"
    },
    "dataTypeSpec": {
      "description": "The data type of the structure for this property.",
      "$ref": "#/definitions/Name"
    },
    "unit": {
      "description": "The unit of the asset model property, such as Newtons or
RPM.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "type": {
      "description": "The property type",
      "$ref": "#/definitions/PropertyType"
    }
  }
}

```

```

    }
  }
},
"DataType": {
  "type": "string",
  "enum": [
    "STRING",
    "INTEGER",
    "DOUBLE",
    "BOOLEAN",
    "STRUCT"
  ]
},
"PropertyType": {
  "description": "Contains a property type, which can be one of attribute,
measurement, metric, or transform.",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "attribute": {
      "$ref": "#/definitions/Attribute"
    },
    "transform": {
      "$ref": "#/definitions/Transform"
    },
    "metric": {
      "$ref": "#/definitions/Metric"
    },
    "measurement": {
      "$ref": "#/definitions/Measurement"
    }
  }
},
"Attribute": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "defaultValue": {
      "type": "string",
      "minLength": 1,
      "maxLength": 1024,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    }
  }
}

```

```
  },
  "Transform": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "expression",
      "variables"
    ],
    "properties": {
      "expression": {
        "description": "The mathematical expression that defines the transformation function.",
        "type": "string",
        "minLength": 1,
        "maxLength": 1024
      },
      "variables": {
        "description": "The list of variables used in the expression.",
        "type": "array",
        "items": {
          "$ref": "#/definitions/ExpressionVariable"
        }
      },
      "processingConfig": {
        "$ref": "#/definitions/TransformProcessingConfig"
      }
    }
  },
  "TransformProcessingConfig": {
    "description": "The processing configuration for the given transform property.",
    "type": "object",
    "additionalProperties": false,
    "required": [
      "computeLocation"
    ],
    "properties": {
      "computeLocation": {
        "description": "The compute location for the given transform property.",
        "$ref": "#/definitions/ComputeLocation"
      },
      "forwardingConfig": {
        "description": "The forwarding configuration for a given property.",
        "$ref": "#/definitions/ForwardingConfig"
      }
    }
  }
}
```

```

    }
  },
  "Metric": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "expression",
      "variables",
      "window"
    ],
    "properties": {
      "expression": {
        "description": "The mathematical expression that defines the metric aggregation function.",
        "type": "string",
        "minLength": 1,
        "maxLength": 1024
      },
      "variables": {
        "description": "The list of variables used in the expression.",
        "type": "array",
        "items": {
          "$ref": "#/definitions/ExpressionVariable"
        }
      },
      "window": {
        "description": "The window (time interval) over which AWS IoT SiteWise computes the metric's aggregation expression",
        "$ref": "#/definitions/MetricWindow"
      },
      "processingConfig": {
        "$ref": "#/definitions/MetricProcessingConfig"
      }
    }
  },
  "MetricProcessingConfig": {
    "description": "The processing configuration for the metric.",
    "type": "object",
    "additionalProperties": false,
    "required": [
      "computeLocation"
    ],
    "properties": {
      "computeLocation": {

```

```
        "description": "The compute location for the given metric property.",
        "$ref": "#/definitions/ComputeLocation"
    }
}
},
"ComputeLocation": {
    "type": "string",
    "enum": [
        "EDGE",
        "CLOUD"
    ]
},
"ForwardingConfig": {
    "type": "object",
    "additionalProperties": false,
    "required": [
        "state"
    ],
    "properties": {
        "state": {
            "type": "string",
            "enum": [
                "ENABLED",
                "DISABLED"
            ]
        }
    }
},
"MetricWindow": {
    "description": "Contains a time interval window used for data aggregate
computations (for example, average, sum, count, and so on).",
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "tumbling": {
            "description": "The tumbling time interval window.",
            "type": "object",
            "additionalProperties": false,
            "required": [
                "interval"
            ],
            "properties": {
                "interval": {
                    "description": "The time interval for the tumbling window.",
```



```

    ]
  },
  {
    "required": [
      "propertyExternalId"
    ]
  }
],
"properties": {
  "propertyId": {
    "$ref": "#/definitions/ID"
  },
  "propertyExternalId": {
    "$ref": "#/definitions/ExternalId"
  },
  "hierarchyId": {
    "$ref": "#/definitions/ID"
  },
  "hierarchyExternalId": {
    "$ref": "#/definitions/ExternalId"
  }
}
},
"Measurement": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "processingConfig": {
      "$ref": "#/definitions/MeasurementProcessingConfig"
    }
  }
}
},
"MeasurementProcessingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "forwardingConfig"
  ],
  "properties": {
    "forwardingConfig": {
      "description": "The forwarding configuration for the given measurement property.",
      "$ref": "#/definitions/ForwardingConfig"
    }
  }
}

```

```

    }
  },
  "AssetModelHierarchy": {
    "description": "Contains information about an asset model hierarchy.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "id",
          "childAssetModelId"
        ]
      },
      {
        "required": [
          "id",
          "childAssetModelExternalId"
        ]
      },
      {
        "required": [
          "externalId",
          "childAssetModelId"
        ]
      },
      {
        "required": [
          "externalId",
          "childAssetModelExternalId"
        ]
      }
    ],
    "required": [
      "name"
    ],
    "properties": {
      "id": {
        "description": "The ID of the asset model hierarchy.",
        "$ref": "#/definitions/ID"
      },
      "externalId": {
        "description": "The ExternalID of the asset model hierarchy.",
        "$ref": "#/definitions/ExternalId"
      }
    }
  },

```

```
    "name": {
      "description": "The name of the asset model hierarchy.",
      "$ref": "#/definitions/Name"
    },
    "childAssetModelId": {
      "description": "The ID of the asset model. All assets in this hierarchy must
be instances of the child AssetModelId asset model.",
      "$ref": "#/definitions/ID"
    },
    "childAssetModelExternalId": {
      "description": "The ExternalID of the asset model. All assets in this
hierarchy must be instances of the child AssetModelId asset model.",
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"AssetModel": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "assetModelId"
      ]
    },
    {
      "required": [
        "assetModelExternalId"
      ]
    }
  ],
  "required": [
    "assetModelName"
  ],
  "properties": {
    "assetModelId": {
      "description": "The ID of the asset model.",
      "$ref": "#/definitions/ID"
    },
    "assetModelExternalId": {
      "description": "The ID of the asset model.",
      "$ref": "#/definitions/ExternalId"
    },
    "assetModelName": {
```

```

    "description": "A unique, friendly name for the asset model.",
    "$ref": "#/definitions/Name"
  },
  "assetModelDescription": {
    "description": "A description for the asset model.",
    "$ref": "#/definitions/Description"
  },
  "assetModelType": {
    "description": "The type of the asset model.",
    "$ref": "#/definitions/AssetModelType"
  },
  "assetModelProperties": {
    "description": "The property definitions of the asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelProperty"
    }
  },
  "assetModelCompositeModels": {
    "description": "The composite asset models that are part of this asset model.
Composite asset models are asset models that contain specific properties.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelCompositeModel"
    }
  },
  "assetModelHierarchies": {
    "description": "The hierarchy definitions of the asset model. Each hierarchy
specifies an asset model whose assets can be children of any other assets created from
this asset model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetModelHierarchy"
    }
  },
  "tags": {
    "description": "A list of key-value pairs that contain metadata for the asset
model.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/Tag"
    }
  }
}

```

```
},
"Asset": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "assetId",
        "assetModelId"
      ]
    },
    {
      "required": [
        "assetExternalId",
        "assetModelId"
      ]
    },
    {
      "required": [
        "assetId",
        "assetModelExternalId"
      ]
    },
    {
      "required": [
        "assetExternalId",
        "assetModelExternalId"
      ]
    }
  ],
  "required": [
    "assetName"
  ],
  "properties": {
    "assetId": {
      "description": "The ID of the asset",
      "$ref": "#/definitions/ID"
    },
    "assetExternalId": {
      "description": "The external ID of the asset",
      "$ref": "#/definitions/ExternalId"
    },
    "assetModelId": {
      "description": "The ID of the asset model from which to create the asset.",
```

```
    "$ref": "#/definitions/ID"
  },
  "assetModelExternalId": {
    "description": "The ExternalID of the asset model from which to create the
asset.",
    "$ref": "#/definitions/ExternalId"
  },
  "assetName": {
    "description": "A unique, friendly name for the asset.",
    "$ref": "#/definitions/Name"
  },
  "assetDescription": {
    "description": "A description for the asset",
    "$ref": "#/definitions/Description"
  },
  "assetProperties": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetProperty"
    }
  },
  "assetHierarchies": {
    "type": "array",
    "items": {
      "$ref": "#/definitions/AssetHierarchy"
    }
  },
  "tags": {
    "description": "A list of key-value pairs that contain metadata for the
asset.",
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/Tag"
    }
  }
}
},
"additionalProperties": false,
"properties": {
  "assetModels": {
    "type": "array",
    "uniqueItems": false,
```

```
    "items": {
      "$ref": "#/definitions/AssetModel"
    }
  },
  "assets": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/Asset"
    }
  }
}
```

경보로 데이터 모니터링

장비 또는 프로세스가 최적화되지 않았을 때 \팀원들에게 경고 알림을 보내도록 데이터에 대한 경보를 구성할 수 있습니다. 특정 메트릭 값이 상한 범위와 하한 범위 내에 있어야 컴퓨터 또는 프로세스 성능이 최적화됩니다. 메트릭이 작동 범위를 벗어나면 장비 운영자가 문제를 해결할 수 있도록 알림을 보내야 합니다. 경보를 사용하여 문제를 신속하게 식별하고 운영자에게 알려 장비 및 프로세스 성능을 극대화할 수 있습니다.

주제

- [경보 유형](#)
- [경보 상태](#)
- [경보 상태 속성](#)
- [자산 모델에서 경보 정의](#)
- [자산에 대한 경보 구성](#)
- [경보에 대응](#)
- [외부 경보 상태 수집](#)

경보 유형

AWS 클라우드에서 탐지하는 경보와 외부 프로세스로 탐지하는 경보를 정의할 수 있습니다. AWS IoT SiteWise 다음과 같은 유형의 경보를 지원합니다.

- AWS IoT Events 알람

AWS IoT Events 알람은 수신을 감지하는 알람입니다. AWS IoT Events AWS IoT SiteWise 의 알람 모델에 자산 속성 값을 전송합니다. AWS IoT Events 그런 다음 경보 상태를 로 AWS IoT Events 보냅니다 AWS IoT SiteWise. 경보 감지 시기, 경보 상태 변경 시 알림 대상 등의 옵션을 구성할 수 있습니다. 경보 상태가 변경될 때 발생하는 [AWS IoT Events 작업](#)을 정의할 수도 있습니다.

의 AWS IoT Events 알람은 알람 모델의 인스턴스입니다. 경보 모델은 경보의 임계값 및 심각도, 경보 상태가 변경될 때 수행할 조치 등을 지정합니다. 경보 모델의 각 특성을 구성할 때는 경보가 모니터링하는 자산 모델의 특성(attribute) 속성을 지정합니다. 자산 모델을 기반으로 하는 모든 자산은 경보의 해당 특성을 AWS IoT Events 평가할 때 속성 값을 사용합니다. 자세한 정보는 AWS IoT Events 개발자 안내서의 [경보 사용](#)을 참조하세요.

상태가 변경되면 AWS IoT Events 경보에 응답할 수 있습니다. 예를 들어 경보가 활성화되면 경보를 확인하거나 일시 중지할 수 있습니다. 또한 경보를 활성화, 비활성화, 재설정할 수 있습니다.

SiteWise 모니터 사용자는 모니터 포털에서 AWS IoT Events SiteWise 경보를 시각화하고 구성하고 이에 응답할 수 있습니다. 자세한 내용을 알아보려면 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [경보를 통한 모니터링](#)을 참조하세요.

Note

AWS IoT Events 이러한 경보를 평가하고 및 간에 데이터를 전송하는 경우 요금이 부과됩니다. AWS IoT SiteWise AWS IoT Events 자세한 내용은 [AWS IoT Events 요금](#)을 참조하십시오.

• 외부 경보

외부 경보는 외부에서 평가하는 경보입니다. AWS IoT SiteWise 경보 상태를 보고하는 데이터 소스가 있는 경우 외부 경보를 사용합니다. 외부 경보에는 경보 상태 데이터를 수집하는 데 사용하는 측정값 속성이 포함되어 있습니다.

상태가 바뀌면 외부 경보를 확인하거나 일시 중지할 수 없습니다.

SiteWise 모니터 사용자는 SiteWise 모니터 포털에서 외부 경보의 상태를 볼 수 있지만 이러한 경보를 구성하거나 이에 응답할 수는 없습니다.

AWS IoT SiteWise 외부 경보의 상태를 평가하지 않습니다.

경보 상태

산업용 경보에는 모니터링하는 장비 또는 프로세스의 상태에 대한 정보와 경보 상태에 대한 작업자의 대응에 대한 정보(선택 사항)가 포함됩니다.

AWS IoT Events 경보를 정의할 때 승인 흐름을 활성화할지 여부를 지정합니다. 확인 흐름은 기본적으로 활성화되어 있습니다. 이 옵션을 활성화하면 작업자는 경보를 확인하고 경보에 대한 세부 정보 또는 경보를 해결하기 위해 취한 조치를 메모로 남길 수 있습니다. 작업자가 활성 경보가 비활성 상태가 되기 전에 확인하지 않으면 경보가 래치 상태가 됩니다. 래치 상태는 경보가 활성화되었지만 인식되지 않았음을 나타내므로 작업자는 장비 또는 프로세스를 점검하고 래치 경보를 확인해야 합니다.

경보의 상태는 다음과 같습니다:

- 정상(Normal) - 경보가 활성화되었지만 비활성 상태입니다. 산업 프로세스 또는 장비가 예상대로 작동합니다.
- 활성(Active) - 경보가 활성 상태입니다. 산업 공정 또는 장비가 작동 범위를 벗어났으므로 주의가 필요합니다.
- 확인됨(Acknowledged) - 작업자가 경보 상태를 확인했습니다.

이 상태는 확인 흐름을 활성화한 경보에만 적용됩니다.

- 래치 상태(Latched) - 경보가 정상으로 돌아왔지만 활성 상태였으며 작업자가 이를 확인하지 않았습니다. 산업 공정 또는 장비에서 경보를 정상으로 재설정하려면 작업자의 주의가 필요합니다.

이 상태는 확인 흐름을 활성화한 경보에만 적용됩니다.

- 일시 중지됨(SnoozeDisabled) - 작업자가 경보를 일시 중지했기 때문에 경보가 비활성화되었습니다. 운영자는 경보가 일시 정지되는 기간을 정의합니다. 이 기간이 지나면 경보는 정상 상태로 돌아옵니다.
- 비활성화됨(Disabled) - 경보가 비활성화되어 감지되지 않습니다.

경보 상태 속성

AWS IoT SiteWise 경보 상태 데이터를 문자열에 직렬화된 JSON 객체로 저장합니다. 이 객체에는 작업자 대응 조치, 경보가 평가하는 규칙 등 경보에 대한 상태 및 추가 정보가 포함됩니다.

이름 및 구조 유형 AWS/ALARM_STATE로 경보 상태 속성을 식별할 수 있습니다. 자세한 정보는 [자산 모델에서 경보 정의](#)를 참조하세요.

경보 상태 데이터 객체에는 다음과 같은 정보가 포함되어 있습니다.

`stateName`

경보 상태 자세한 정보는 [경보 상태](#)를 참조하세요.

데이터 유형: STRING

`customerAction`

(선택 사항) 경보에 대한 작업자의 대응 정보가 들어 있는 객체입니다. 작업자는 경보를 활성화, 비활성화, 확인, 일시 중지할 수 있습니다. 이러한 작업을 하면 경보 상태 데이터에 응답과 응답 시 남길 수 있는 메모가 포함됩니다. 이 객체에는 다음 정보가 포함되어 있어야 합니다.

actionName

작업자가 경보에 대응하기 위해 취하는 조치의 이름입니다. 다음 스트링 중 하나를 포함합니다.

- ENABLE
- DISABLE
- SNOOZE
- ACKNOWLEDGE
- RESET

데이터 유형: STRING

enable

(선택 사항) 작업자가 경보를 활성화할 때 `customerAction`에 표시되는 객체입니다. 작업자가 경보를 활성화하면 경보 상태가 `Normal`로 바뀝니다. 이 객체에는 다음 정보가 포함되어 있어야 합니다.

note

(선택 사항) 고객이 경보를 활성화할 때 남기는 메모입니다.

데이터 유형: STRING

최대 길이: 128자

disable

(선택 사항) 작업자가 경보를 비활성화할 때 `customerAction`에 표시되는 객체입니다. 작업자가 경보를 비활성화하면 경보 상태가 `Disabled`로 바뀝니다. 이 객체에는 다음 정보가 포함되어 있어야 합니다.

note

(선택 사항) 고객이 경보를 비활성화할 때 남기는 메모입니다.

데이터 유형: STRING

최대 길이: 128자

acknowledge

(선택 사항) 작업자가 경보를 확인할 때 `customerAction`에 표시되는 객체입니다. 작업자가 경보를 활성화하면 경보 상태가 `Acknowledged`로 바뀝니다. 이 객체에는 다음 정보가 포함되어 있어야 합니다.

note

(선택 사항) 고객이 경보를 확인할 때 남기는 메모입니다.

데이터 유형: STRING

최대 길이: 128자

snooze

(선택 사항) 작업자가 경보를 일시 중지할 때 `customerAction`에 표시되는 객체입니다. 작업자가 경보를 활성화하면 경보 상태가 `SnoozeDisabled`로 바뀝니다. 이 객체에는 다음 정보가 포함되어 있어야 합니다.

snoozeDuration

작업자가 경보를 일시 중지하는 시간(초)입니다. 이 시간이 지나면 경보가 `Normal` 상태로 바뀝니다.

데이터 유형: INTEGER

note

(선택 사항) 고객이 경보를 일시 중지할 때 남기는 메모입니다.

데이터 유형: STRING

최대 길이: 128자

ruleEvaluation

(선택 사항) 경보를 평가하는 규칙에 대한 정보가 포함된 객체입니다. 이 객체에는 다음 정보가 포함되어 있어야 합니다.

simpleRule

비교 연산자를 사용하여 속성 값을 임계값과 비교하는 간단한 규칙에 대한 정보가 들어 있는 객체입니다. 이 객체에는 다음 정보가 포함되어 있어야 합니다.

inputProperty

경보에서 평가하는 속성의 값입니다.

데이터 유형: DOUBLE

operator

이 경보가 속성을 임계값과 비교하는 데 사용하는 비교 연산자입니다. 다음 스트링 중 하나를 포함합니다.

- < - 미만
- <= - 이하
- == - 같음
- != - 같지 않음
- >= - 이상
- > - 초과

데이터 유형: STRING

threshold

이 경보가 속성값과 비교하는 임계값입니다.

데이터 유형: DOUBLE

자산 모델에서 경보 정의

자산 모델은 산업 데이터 및 경보의 표준화를 촉진합니다. 자산 모델에 경보 정의를 정의하여 자산 모델을 기반으로 모든 자산에 대한 경보를 표준화할 수 있습니다.

복합 자산 모델을 사용하여 이 자산 모델에 대한 경보를 정의할 수 있습니다. 복합 자산 모델은 다른 자산 모델의 특정 속성 집합을 표준화하는 자산 모델입니다. 복합 자산 모델은 자산 모델에 특정 속성이 존재하도록 합니다. 경보에는 유형, 상태, 소스 속성(선택 사항)이 있으므로, 경보 복합 모델은 이러한 속성이 존재하도록 강제합니다.

각 복합 자산 모델에는 복합 모델이 지원하는 속성을 정의하는 유형이 있습니다. 경보 복합 모델은 경보 유형, 경보 상태, 경보 소스(선택 사항)의 속성을 정의합니다. 복합 모델이 포함된 자산 모델에서 자산을 생성하는 경우, 자산에는 자산 모델에서 지정한 속성과 함께 복합 모델의 속성이 포함됩니다.

복합 모델의 각 속성에는 해당 복합 모델 유형을 식별하는 이름이 있어야 합니다. 복합 모델 속성은 복잡한 데이터 유형의 속성을 지원합니다. 이러한 속성에는 속성의 복합 데이터 유형을 지정하는 STRUCT 데이터 유형과 dataTypeSpec 특성이 있습니다. 복합 데이터 유형 속성에는 문자열로 직렬화된 JSON 데이터가 포함됩니다.

경보 복합 모델에는 다음과 같은 속성이 있습니다. 각 속성에는 이 유형의 복합 모델에 대해 이를 식별하는 이름이 있어야 합니다.

경보 유형

경보 유형. 다음 중 하나를 지정하세요.

- IOT_EVENTS— AWS IoT Events 알람. AWS IoT SiteWise 이 경보의 상태를 AWS IoT Events 평가하기 위해 데이터를 전송합니다. 이 경보 정의에 대한 경보 모델을 정의하려면 AWS IoT Events 경보 소스 속성을 지정해야 합니다.
- EXTERNAL - 외부 경보입니다. 경보 상태를 측정값으로 수집합니다.

속성 이름: AWS/ALARM_TYPE

속성 유형: [attribute](#)

데이터 유형: STRING

경보 상태

경보 상태에 대한 시계열 데이터입니다. 경보에 대한 상태 및 기타 정보를 포함하는 문자열로 직렬화된 객체입니다. 자세한 정보는 [경보 상태 속성](#)을 참조하세요.

속성 이름: AWS/ALARM_STATE

속성 유형: [measurement](#)

데이터 유형: STRUCT

데이터 구조 유형: AWS/ALARM_STATE

경보 소스

(선택 사항) 경보 상태를 평가하는 리소스의 Amazon 리소스 이름(ARN)입니다. 경보의 경우 이는 AWS IoT Events 경보 모델의 ARN입니다.

속성 이름: AWS/ALARM_SOURCE

속성 유형: [attribute](#)

데이터 유형: STRING

Example 경보 복합 모델 예시

다음 자산 모델은 온도를 모니터링하는 경보가 있는 보일러를 나타냅니다. AWS IoT SiteWise 온도 데이터를 AWS IoT Events 전송하여 경보를 감지합니다.

```
{
  "assetModelName": "Boiler",
  "assetModelDescription": "A boiler that alarms when its temperature exceeds its
limit.",
  "assetModelProperties": [
    {
      "name": "Temperature",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "High Temperature",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    }
  ],
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "IOT_EVENTS"
            }
          }
        }
      ],
    }
  ]
}
```

```

    "name": "AWS/ALARM_STATE",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/ALARM_STATE",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "AWS/ALARM_SOURCE",
    "dataType": "STRING",
    "type": {
      "attribute": {}
    }
  }
]
}

```

주제

- [AWS IoT Events 알람 정의](#)
- [외부 경보 정의](#)

AWS IoT Events 알람 정의

AWS IoT Events 경보를 만들면 자산 속성 값을 로 AWS IoT SiteWise AWS IoT Events 전송하여 경보 상태를 평가합니다. AWS IoT Events 알람 정의는 에서 정의한 알람 모델에 따라 달라집니다 AWS IoT Events. 자산 모델에서 AWS IoT Events 경보를 정의하려면 경보 모델을 경보 소스 속성으로 지정하는 AWS IoT Events 경보 복합 모델을 정의합니다.

AWS IoT Events 경보는 경보 임계값 및 경보 알림 설정과 같은 입력에 따라 달라집니다. 이러한 입력을 자산 모델의 속성(attribute)으로 정의합니다. 그런 다음 모델에 따라 각 자산에서 이러한 입력을 사용자 지정할 수 있습니다. AWS IoT SiteWise 콘솔에서 이러한 속성을 자동으로 생성할 수 있습니다. AWS CLI 또는 API로 경보를 정의하는 경우 자산 모델에서 이러한 속성을 수동으로 정의해야 합니다.

또한 사용자 지정 경보 알림 동작과 같이 경보가 감지될 때 발생하는 다른 작업을 정의할 수 있습니다. 예를 들어, Amazon SNS 주제에 푸시 알림을 보내는 작업을 구성할 수 있습니다. 정의할 수 있는 작업에 대한 자세한 내용은 AWS IoT Events 개발자 안내서의 [다른 AWS 서비스](#) 사용을 참조하십시오.

자산 모델을 업데이트하거나 삭제할 때의 AWS IoT Events 경보 모델이 이 자산 모델과 관련된 자산 속성을 모니터링하고 있는지 확인할 AWS IoT SiteWise 수 있습니다. 이렇게 하면 AWS IoT Events 경보에서 현재 사용 중인 자산 속성을 삭제할 수 없습니다. 에서 AWS IoT SiteWise이 기능을 활성화하려면 `iotevents:ListInputRoutings` 권한이 있어야 합니다. 이 AWS IoT SiteWise 권한을 사용하면 에서 지원하는 [ListInput AWS IoT Events라우팅](#) API 작업을 호출할 수 있습니다. 자세한 정보는 [\(선택 사항\) ListInputRoutings 권한](#)을 참조하세요.

Note

경보 알림 기능은 중국(베이징) 리전에서 사용할 수 없습니다.

주제

- [경보 알림 요구 사항](#)
- [AWS IoT Events 경보 정의 \(콘솔\)AWS IoT SiteWise](#)
- [AWS IoT Events 알람 정의 \(AWS IoT Events 콘솔\)](#)
- [AWS IoT Events 알람 정의 \(AWS CLI\)](#)

경보 알림 요구 사항

AWS IoT Events AWS 계정의 AWS Lambda 함수를 사용하여 알람 알림을 보냅니다. 경보 알림을 활성화하려면 경보와 AWS 동일한 지역에 이 Lambda 함수를 생성해야 합니다. 이 Lambda 함수는 [Amazon Simple Notification Service\(SNS\)](#)를 사용하여 문자 알림을 보내고, [Amazon Simple Email Service\(Amazon SES\)](#)를 사용하여 이메일 알림을 보냅니다. 경보를 생성할 때 AWS IoT Events 경보가 알림을 보내는 데 사용하는 프로토콜과 설정을 구성합니다.

AWS IoT Events 계정에서 이 Lambda 함수를 생성하는 데 사용할 수 있는 AWS CloudFormation 스택 템플릿을 제공합니다. 자세한 정보는 AWS IoT Events 개발자 안내서의 [Lambda 함수 오류](#)를 참조하세요.

AWS IoT Events 경보 정의 (콘솔)AWS IoT SiteWise

AWS IoT SiteWise 콘솔을 사용하여 기존 자산 모델에 대한 AWS IoT Events 경보를 정의할 수 있습니다. 새 자산 모델에 AWS IoT Events 경보를 정의하려면 자산 모델을 만든 다음 다음 단계를 완료하십시오. 자세한 정보는 [자산 모델 생성](#)을 참조하세요.

⚠ Important

각 경보에는 경보와 비교할 임계값을 지정하는 속성(attribute)이 필요합니다. 경보를 정의하려면 먼저 자산 모델에서 임계값 속성(attribute)을 정의해야 합니다.

풍력 터빈이 최대 풍속 등급인 50mph를 초과할 때 이를 감지하는 경보를 정의하려는 경우를 예로 들어 보겠습니다. 경보를 정의하기 전에 기본값 50으로 속성(attribute)(최대 풍속)을 정의해야 합니다.

자산 모델에 대한 AWS IoT Events 경보를 정의하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 경보를 정의할 자산 모델을 선택합니다.
4. 경보 탭을 선택합니다.
5. 경보 추가를 선택합니다.
6. 경보 유형 옵션 섹션에서 AWS IoT Events 경보를 선택합니다.
7. 경보 세부 정보 섹션에서 다음을 수행합니다.
 - a. 경보의 이름을 입력합니다.
 - b. (선택 사항) 경보에 대한 설명을 입력합니다.
8. 임계값 정의 섹션에서는 경보가 탐지되는 시기와 경보의 심각도를 정의합니다. 다음을 따릅니다.
 - a. 경보가 탐지되는 속성을 선택합니다. 이 속성이 새 값을 수신할 때마다 값에 AWS IoT SiteWise AWS IoT Events 전송하여 경보 상태를 평가합니다.
 - b. 속성을 임계값과 비교하는 데 사용할 연산자를 선택합니다. 다음 옵션 중 하나를 선택합니다.
 - < 미만
 - <= 이하
 - == 같음
 - != 같지 않음
 - >= 이상
 - > 초과
 - c. Value에서 임계값으로 사용할 속성 속성을 선택합니다. AWS IoT Events 속성 값을 이 속성의 값과 비교합니다.

- d. 경보의 심각도를 입력합니다. 팀에서 이해할 수 있는 숫자를 사용하여 이 경보의 심각도를 반영합니다.
9. (선택 사항) 알림 설정 - 선택 사항 섹션에서 다음을 수행합니다.

- a. 활성화를 선택합니다.

 Note

비활성을 선택하면 사용자와 팀이 경보 알림을 받지 않게 됩니다.

- b. 수신자에서 수신자를 선택합니다.

 Important

AWS IAM Identity Center 사용자에게 알림 알림을 보낼 수 있습니다. 이 기능을 사용하려면 IAM Identity Center를 활성화해야 합니다. IAM Identity Center는 한 번에 한 AWS 지역에서만 활성화할 수 있습니다. 즉, IAM Identity Center를 활성화한 리전에 서만 경보 알림을 정의할 수 있습니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 [시작하기](#)를 참조하세요.

- c. 프로토콜에서 다음 옵션 중에 선택합니다.
- 이메일 및 문자 메시지 - IAM Identity Center 사용자에게 SMS 메시지와 이메일 메시지를 통해 알림을 전달합니다.
 - 이메일 - 경보가 IAM Identity Center 사용자에게 이메일 메시지를 통해 알림을 전달합니다.
 - 텍스트 - 경보가 IAM Identity Center 사용자에게 SMS 메시지를 통해 알림을 전달합니다.
- d. 발신자에서 발신자를 선택합니다.

 Important

Amazon Simple Email Service(Amazon SES)에서 발신자 이메일 주소를 확인해야 합니다. 자세한 내용은 Amazon Simple Email Service 개발자 가이드에서 [Amazon SES에서 이메일 주소 확인](#)을 참조하세요.

10. 기본 자산 상태 섹션에서 이 자산 모델에서 생성된 경보의 기본 상태를 설정할 수 있습니다.

Note

이후 단계에서 이 자산 모델에서 생성한 자산에 대해 이 경보를 활성화하거나 비활성화합니다.

11. 고급 설정 섹션에서 권한, 추가 알림 설정, 경고 상태 조치, SiteWise 모니터의 경고 모델 및 승인 흐름을 구성할 수 있습니다.

Note

AWS IoT Events 알람에는 다음과 같은 서비스 역할이 필요합니다.

- 경고 상태 값을 AWS IoT Events 전송하는 역할을 위임합니다. AWS IoT SiteWise
- Lambda로 데이터를 전송하는 것으로 AWS IoT Events 가정하는 역할입니다. 경보가 알림을 보내는 경우에만 이 역할이 필요합니다.

권한 섹션에서 다음을 수행합니다.

- a. AWS IoT Events 역할의 경우 기존 역할을 사용하거나 필요한 권한이 있는 역할을 생성합니다. 이 역할에는 `iotevents.amazonaws.com`가 역할을 맡을 수 있도록 허용하는 `iotsitewise:BatchPutAssetPropertyValue` 권한과 신뢰 관계가 필요합니다.
- b. AWS IoT Events Lambda 역할의 경우 기존 역할을 사용하거나 필요한 권한이 있는 역할을 생성합니다. 이 역할에는 `iotevents.amazonaws.com`가 역할을 맡을 수 있도록 허용하는 `lambda:InvokeFunction` 및 `sso-directory:DescribeUser` 권한과 신뢰 관계가 필요합니다.

12. (선택 사항) 추가 알림 설정 섹션에서 다음을 수행합니다.

- a. 수신자 속성(attribute)에서 알림 수신자를 지정하는 값을 가진 속성(attribute)을 정의합니다. IAM Identity Center 사용자를 수신자로 선택할 수 있습니다.

속성(attribute)을 생성하거나 자산 모델에서 기존 속성(attribute)을 사용할 수 있습니다.

- 새 수신자 속성(attribute) 생성을 선택하는 경우 속성(attribute)에 대해 수신자 속성(attribute) 이름과 수신자 기본값 - 선택 사항을 지정합니다.
- 기존 수신자 속성(attribute) 사용을 선택한 경우 수신자 속성(attribute) 이름에서 속성(attribute)을 선택합니다. 경보는 사용자가 선택한 속성(attribute) 기본값을 사용합니다.

이 자산 모델에서 만든 각 자산의 기본값을 재정의할 수 있습니다.

- b. 사용자 지정 메시지 속성(attribute)의 경우 기본 상태 변경 메시지 외에 전송할 사용자 지정 메시지를 지정하는 값을 가진 속성(attribute)을 정의합니다. 예를 들어 팀이 이 경보를 해결하는 방법을 이해하는 데 도움이 되는 메시지를 지정할 수 있습니다.

속성(attribute)을 생성하거나 자산 모델에서 기존 속성(attribute)을 사용하도록 선택할 수 있습니다.

- 새 사용자 지정 메시지 속성(attribute) 생성을 선택한 경우 속성(attribute)에 대해 사용자 지정 메시지 속성(attribute) 이름과 사용자 지정 메시지 기본값 - 선택 사항을 지정합니다.
- 기존 사용자 지정 메시지 속성(attribute) 사용을 선택한 경우 사용자 지정 메시지 속성(attribute) 이름에서 속성(attribute)을 선택합니다. 경보는 사용자가 선택한 속성(attribute) 기본값을 사용합니다.

이 자산 모델에서 만든 각 자산의 기본값을 재정의할 수 있습니다.

- c. Lambda 함수 관리에서 다음 중 하나를 수행합니다.

- 새 Lambda 함수를 AWS IoT SiteWise 생성하려면 AWS 관리형 템플릿에서 새 Lambda 생성을 선택하십시오.
- 기존 Lambda 함수를 사용하려면 기존 Lambda 사용을 선택하고 함수 이름을 선택합니다.

자세한 정보는 AWS IoT Events 개발자 안내서의 [경보 알림 관리](#)를 참조하세요.

13. (선택 사항) 상태 작업 설정 섹션에서 다음을 수행합니다.

- a. 작업 편집을 선택합니다.
- b. 경보 상태 작업 추가에서 작업을 추가한 다음 저장을 선택합니다.

최대 10개의 작업을 추가할 수 있습니다.

AWS IoT Events 경보가 활성 상태일 때 작업을 수행할 수 있습니다. 내장 액션을 정의하여 타이머를 사용하거나 변수를 설정하거나 데이터를 다른 AWS 리소스로 보낼 수 있습니다. 자세한 정보는 AWS IoT Events 개발자 안내서의 [지원되는 작업](#)을 참조하세요.

14. (선택 사항) SiteWise 모니터의 알람 모델 관리 - 선택 사항에서 활성 또는 비활성을 선택합니다.

이 옵션을 사용하면 SiteWise Monitor에서 알람 모델을 업데이트할 수 있습니다. 이 옵션은 기본적으로 활성화되어 있습니다.

15. 확인 흐름에서 활성 또는 비활성을 선택합니다. 확인 흐름에 대한 자세한 내용은 [경보 상태](#) 섹션을 참조합니다.
16. 경보 추가를 선택합니다.

Note

AWS IoT SiteWise 콘솔은 애셋 모델에 경보를 추가하기 위해 여러 API 요청을 보냅니다. 경보 추가를 선택하면 콘솔에서 이러한 API 요청의 진행 상황을 보여주는 대화 상자가 열립니다. 각 API 요청이 성공하거나 API 요청이 실패할 때까지 이 페이지에 머무르세요. 요청이 실패하면 대화 상자를 닫고 문제를 해결한 다음 경보 추가를 선택하여 다시 시도합니다.

AWS IoT Events 알람 정의 (AWS IoT Events 콘솔)

AWS IoT Events 콘솔을 사용하여 기존 자산 모델에 대한 AWS IoT Events 경보를 정의할 수 있습니다. 새 자산 모델에 AWS IoT Events 경보를 정의하려면 자산 모델을 만든 다음 다음 단계를 완료하십시오. 자세한 정보는 [자산 모델 생성](#)을 참조하세요.

Important

각 경보에는 경보와 비교할 임계값을 지정하는 속성(attribute)이 필요합니다. 경보를 정의하려면 먼저 자산 모델에서 임계값 속성(attribute)을 정의해야 합니다. 풍력 터빈이 최대 풍속 등급인 50mph를 초과할 때 이를 감지하는 경보를 정의하려는 경우를 예로 들어 보겠습니다. 경보를 정의하기 전에 기본값 50으로 속성(attribute)(최대 풍속)을 정의해야 합니다.

자산 모델에 대한 AWS IoT Events 경보를 정의하려면

1. [AWS IoT Events 콘솔](#)로 이동합니다.
2. 탐색 창에서 경보 모델을 선택합니다.
3. 경보 모델 생성을 선택하세요.
4. 경보의 이름을 입력합니다.

5. (선택 사항) 경보에 대한 설명을 입력합니다.
6. 경보 대상 섹션에서 다음을 수행합니다.
 - a. 대상 옵션에서 AWS IoT SiteWise 자산 속성을 선택합니다.
 - b. 경보를 추가하고자 하는 자산 모델을 선택합니다.
7. 임계값 정의 섹션에서는 경보가 탐지되는 시기와 경보의 심각도를 정의합니다. 다음을 따릅니다.
 - a. 경보가 탐지되는 속성을 선택합니다. 이 속성이 새 값을 수신할 때마다 값에 AWS IoT SiteWise AWS IoT Events 전송하여 경보 상태를 평가합니다.
 - b. 속성을 임계값과 비교하는 데 사용할 연산자를 선택합니다. 다음 옵션 중 하나를 선택합니다.
 - < 미만
 - <= 이하
 - == 같음
 - != 같지 않음
 - >= 이상
 - > 초과
 - c. Value에서 임계값으로 사용할 속성 속성을 선택합니다. AWS IoT Events 속성 값을 이 속성의 값과 비교합니다.
 - d. 경보의 심각도를 입력합니다. 팀에서 이해할 수 있는 숫자를 사용하여 이 경보의 심각도를 반영합니다.
8. (선택 사항) 알림 설정 - 선택 사항 섹션에서 다음을 수행합니다.
 - a. 프로토콜에서 다음 옵션 중에 선택합니다.
 - 이메일 및 문자 메시지 - IAM Identity Center 사용자에게 SMS 메시지와 이메일 메시지를 통해 알림을 전달합니다.
 - 이메일 - 경보가 IAM Identity Center 사용자에게 이메일 메시지를 통해 알림을 전달합니다.
 - 텍스트 - 경보가 IAM Identity Center 사용자에게 SMS 메시지를 통해 알림을 전달합니다.
 - b. 발신자에서 발신자를 선택합니다.

⚠ Important

Amazon Simple Email Service(Amazon SES)에서 발신자 이메일 주소를 확인해야 합니다. 자세한 내용은 Amazon Simple Email Service 개발자 가이드에서 [Amazon SES에서 이메일 주소 확인](#)을 참조하세요.

- c. 수신자 속성(attribute) - 선택 사항에서 속성(attribute)을 선택합니다. 경보는 사용자가 선택한 속성(attribute) 기본값을 사용합니다.
 - d. 사용자 지정 메시지 속성(attribute) - 선택 사항에서 속성(attribute)을 선택합니다. 경보는 사용자가 선택한 속성(attribute) 기본값을 사용합니다.
9. 인스턴스 섹션에서 이 경보의 기본 상태를 지정합니다. 이후 단계에서 이 자산 모델에서 생성한 모든 자산에 대해 이 경보를 활성화하거나 비활성화할 수 있습니다.
 10. 고급 설정 설정에서 권한, 추가 알림 설정, 경보 상태 동작, SiteWise 모니터의 경보 모델 및 승인 흐름을 구성할 수 있습니다.

i Note

AWS IoT Events 알람에는 다음과 같은 서비스 역할이 필요합니다.

- 경보 상태 값을 AWS IoT Events 전송하는 역할을 위임합니다. AWS IoT SiteWise
- Lambda로 데이터를 전송하는 것으로 AWS IoT Events 가정하는 역할입니다. 경보가 알림을 보내는 경우에만 이 역할이 필요합니다.

- a. 확인 흐름 섹션에서 활성화됨 또는 비활성화됨을 선택합니다. 확인 흐름에 대한 자세한 내용은 [경보 상태](#) 섹션을 참조합니다.
- b. 권한 섹션에서 다음을 수행합니다.
 - i. AWS IoT Events 역할의 경우 기존 역할을 사용하거나 필요한 권한이 있는 역할을 생성합니다. 이 역할에는 `iotevents.amazonaws.com`가 역할을 맡을 수 있도록 허용하는 `iotsitewise:BatchPutAssetPropertyValue` 권한과 신뢰 관계가 필요합니다.
 - ii. Lambda 역할의 경우 기존 역할을 사용하거나 필요한 권한이 있는 역할을 생성합니다. 이 역할에는 `iotevents.amazonaws.com`가 역할을 맡을 수 있도록 허용하는 `lambda:InvokeFunction` 및 `sso-directory:DescribeUser` 권한과 신뢰 관계가 필요합니다.
- c. (선택 사항) 추가 알림 설정 창에서 다음을 수행합니다.

- (Lambda 함수 관리에 대해 다음 중 하나를 수행합니다.
 - 새 Lambda 함수를 AWS IoT Events 생성하려면 새 Lambda 함수 생성을 선택하십시오.
 - 기존 Lambda 함수를 사용하려면 기존 Lambda 함수 사용을 선택하고 함수 이름을 선택합니다.

자세한 정보는 AWS IoT Events 개발자 안내서의 [경보 알림 관리](#)를 참조하세요.

d. (선택 사항) 상태 작업 설정 - 선택 사항 섹션에서 다음을 수행합니다.

- 경보 상태 작업업에서 작업을 추가하고 저장을 선택합니다.

최대 10개의 작업을 추가할 수 있습니다.

AWS IoT Events 경보가 활성화 상태일 때 작업을 수행할 수 있습니다. 내장 액션을 정의하여 타 이머를 사용하거나 변수를 설정하거나 데이터를 다른 AWS 리소스로 보낼 수 있습니다. 자세한 정보는 AWS IoT Events 개발자 안내서의 [지원되는 작업](#)을 참조하세요.

11. 생성을 선택합니다.

Note

AWS IoT Events 콘솔은 애셋 모델에 경보를 추가하기 위해 여러 API 요청을 보냅니다. 경 보 추가를 선택하면 콘솔에서 이러한 API 요청의 진행 상황을 보여주는 대화 상자가 열립 니다. 각 API 요청이 성공하거나 API 요청이 실패할 때까지 이 페이지에 머무르세요. 요청 이 실패하면 대화 상자를 닫고 문제를 해결한 다음 경보 추가를 선택하여 다시 시도합니 다.

AWS IoT Events 알람 정의 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 속성을 모니터링하는 AWS IoT Events 경 보를 정의할 수 있습니다. 새 자산 모델이나 기존 자산 모델에서 경보를 정의할 수 있습니다. 자산 모델 에 경보를 정의한 후에는 경보를 생성하고 자산 모델에 연결합니다. AWS IoT Events 이 프로세스에서 는 다음을 수행합니다.

단계

- [1단계: 자산 모델에서 경보 정의](#)
- [2단계: AWS IoT Events 알람 모델 정의](#)
- [3단계: ~ 사이의 AWS IoT SiteWise 데이터 흐름 활성화 AWS IoT Events](#)

1단계: 자산 모델에서 경보 정의

새 자산 모델 또는 기존 자산 모델에 경보 정의 및 연결된 속성을 추가합니다.

자산 모델(CLI)에서 경보를 정의하려면

1. `asset-model-payload.json`이라는 파일을 생성합니다. 다른 섹션의 단계에 따라 자산 모델의 세부 정보를 파일에 추가하되, 자산 모델을 만들거나 업데이트하라는 요청을 제출하지 마세요. 이 섹션에서는 `asset-model-payload.json` 파일의 자산 모델 세부 정보에 경보 정의를 추가합니다.
 - 자산 모델을 만드는 방법에 대한 자세한 내용은 [자산 모델 생성 \(AWS CLI\)](#)을 참조하세요.
 - 기존 자산 모델을 업데이트하는 방법에 대한 자세한 내용은 [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#) 섹션을 참조합니다.

Note

자산 모델은 경보로 모니터링할 자산 속성을 포함하여 하나 이상의 자산 속성을 정의해야 합니다.

2. 경보 복합 모델(`assetModelCompositeModels`)을 자산 모델에 추가합니다. AWS IoT Events 알람 복합 모델은 `IOT_EVENTS` 유형을 지정하고 알람 소스 속성을 지정합니다. 에서 알람 모델을 생성한 후 알람 소스 속성을 추가합니다 AWS IoT Events.

Important

알람 복합 모델은 나중에 생성하는 AWS IoT Events 알람 모델과 이름이 같아야 합니다. 경보 모델 이름에는 영숫자만 사용할 수 있습니다. 경보 모델에 동일한 이름을 사용할 수 있도록 고유한 영숫자 이름을 지정합니다.

```
{
  ...
  "assetModelCompositeModels": [
```

```

{
  "name": "BoilerTemperatureHighAlarm",
  "type": "AWS/ALARM",
  "properties": [
    {
      "name": "AWS/ALARM_TYPE",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "IOT_EVENTS"
        }
      }
    },
    {
      "name": "AWS/ALARM_STATE",
      "dataType": "STRUCT",
      "dataTypeSpec": "AWS/ALARM_STATE",
      "type": {
        "measurement": {}
      }
    }
  ]
}

```

3. 자산 모델에 경보 임계값 속성(attribute)을 추가합니다. 이 임계값에 사용할 기본값을 지정합니다. 이 모델을 기반으로 각 자산에서 이 기본값을 재정의할 수 있습니다.

Note

경보 임계값 속성(attribute)은 INTEGER 또는 DOUBLE여야 합니다.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Temperature Max Threshold",
      "dataType": "DOUBLE",
      "type": {

```

```

    "attribute": {
      "defaultValue": "105.0"
    }
  }
}
]
}

```

4. (선택 사항) 자산 모델에 경보 알림 속성(attribute)을 추가합니다. 이러한 속성은 경보 상태가 변경될 때 알림을 보내는 데 AWS IoT Events 사용하는 IAM Identity Center 수신자 및 기타 입력을 지정합니다. 이 모델을 기반으로 각 자산에서 이러한 기본값을 재정의할 수 있습니다.

Important

AWS IAM Identity Center 사용자에게 경보 알림을 보낼 수 있습니다. 이 기능을 사용하려면 IAM Identity Center를 활성화해야 합니다. IAM Identity Center는 한 번에 한 AWS 지역에서만 활성화할 수 있습니다. 즉, IAM Identity Center를 활성화한 리전에서만 경보 알림을 정의할 수 있습니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 [시작하기](#)를 참조하세요.

다음을 따릅니다.

- a. IAM Identity Center 아이덴티티 스토어의 ID를 지정하는 속성(attribute)을 추가합니다. IAM ID 센터 [ListInstances](#) API 작업을 사용하여 ID 저장소를 나열할 수 있습니다. 이 작업은 IAM Identity Center를 활성화한 리전에서만 작동합니다.

```
aws sso-admin list-instances
```

그런 다음 아이덴티티 스토어 ID(예:d-123EXAMPLE)를 속성(attribute)의 기본값으로 지정합니다.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {

```

```

    "attribute": {
      "defaultValue": "d-123EXAMPLE"
    }
  }
}
]
}

```

- b. 알림을 받는 IAM Identity Center 사용자의 ID를 지정하는 속성(attribute)을 추가합니다. 기본 알림 수신자를 정의하려면 IAM Identity Center 사용자 ID를 기본값으로 추가합니다. IAM Identity Center 사용자 ID를 가져오려면 다음 중 하나를 수행합니다.
 - i. IAM ID 센터 [ListUsers](#) API를 사용하여 사용자 이름을 알고 있는 사용자의 ID를 가져올 수 있습니다. *d-123EXAMPLE*을 아이덴티티 스토어의 ID로 바꾸고, *Name*을 사용자의 사용자 이름으로 바꿉니다.

```

aws identitystore list-users \
  --identity-store-id d-123EXAMPLE \
  --filters AttributePath=UserName,AttributeValue=Name

```

- ii. [IAM Identity Center 콘솔](#)을 사용하여 사용자를 검색하고 사용자 ID를 찾습니다.

그런 다음 사용자 ID(예: 123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE)를 속성(attribute)의 기본값으로 지정하거나 기본값 없이 속성(attribute)을 정의합니다.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "userId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
        }
      }
    }
  ]
}

```

- c. (선택 사항) SMS(텍스트) 메시지 알림의 기본 발신자 ID를 지정하는 속성(attribute)을 추가합니다. 발신자 ID는 Amazon Simple Notification Service(SNS)가 보내는 메시지에 메시지 발신자로 표시됩니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서의 [Amazon SNS로 SMS 메시징을 위한 발신자 ID 요청](#)을 참조하세요.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    }
  ]
}
```

- d. (선택 사항) 이메일 알림에서 발신 주소로 사용할 기본 이메일 주소를 지정하는 속성(attribute)을 추가합니다.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "fromAddress",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "my.factory@example.com"
        }
      }
    }
  ]
}
```

- e. (선택 사항) 이메일 알림에 사용할 기본 제목을 지정하는 속성(attribute)을 추가합니다.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "emailSubject",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "[ALERT] High boiler temperature"
        }
      }
    }
  ]
}
```

- f. (선택 사항) 알림에 포함할 추가 메시지를 지정하는 속성(attribute)을 추가합니다. 기본적으로 알림 메시지에는 경보에 대한 정보가 포함됩니다. 사용자에게 자세한 정보를 제공하는 추가 메시지를 포함할 수도 있습니다.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "additionalMessage",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Turn off the power before you check the alarm."
        }
      }
    }
  ]
}
```

5. 자산 모델을 만들거나 기존 자산 모델을 업데이트합니다. 다음 중 하나를 수행하십시오.
- 다음 명령을 실행하여 자산 모델을 생성합니다.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-
payload.json
```

- 다음 명령을 실행하여 기존 자산 모델을 업데이트합니다. *asset-model-id*를 자산 모델의 ID로 바꿉니다.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://asset-model-payload.json
```

명령을 실행한 후 응답에서 `assetModelId`를 기록합니다.

예: 보일러 자산 모델

다음 자산 모델은 온도 데이터를 보고하는 보일러를 나타냅니다. 이 자산 모델은 보일러 과열을 감지하는 경보를 정의합니다.

```
{  
  "assetModelName": "Boiler Model",  
  "assetModelDescription": "Represents a boiler.",  
  "assetModelProperties": [  
    {  
      "name": "Temperature",  
      "dataType": "DOUBLE",  
      "unit": "C",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature Max Threshold",  
      "dataType": "DOUBLE",  
      "type": {  
        "attribute": {  
          "defaultValue": "105.0"  
        }  
      }  
    },  
    {  
      "name": "identityStoreId",  
      "dataType": "STRING",  
      "type": {  
        "attribute": {  
          "defaultValue": "d-123EXAMPLE"  
        }  
      }  
    }  
  ]  
}
```

```
    }
  },
  {
    "name": "userId",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
      }
    }
  },
  {
    "name": "senderId",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "MyFactory"
      }
    }
  },
  {
    "name": "fromAddress",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "my.factory@example.com"
      }
    }
  },
  {
    "name": "emailSubject",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "[ALERT] High boiler temperature"
      }
    }
  },
  {
    "name": "additionalMessage",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Turn off the power before you check the alarm."
      }
    }
  }
}
```

```

    }
  }
}
],
"assetModelHierarchies": [

],
"assetModelCompositeModels": [
  {
    "name": "BoilerTemperatureHighAlarm",
    "type": "AWS/ALARM",
    "properties": [
      {
        "name": "AWS/ALARM_TYPE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "IOT_EVENTS"
          }
        }
      },
      {
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
          "measurement": {}
        }
      }
    ]
  }
]
}
]
}
}

```

2단계: AWS IoT Events 알람 모델 정의

에서 알람 모델을 생성합니다 AWS IoT Events. AWS IoT Events에서는 표현식을 사용하여 알람 모델의 값을 지정합니다. 표현식을 사용하여 평가할 값을 지정하고 AWS IoT SiteWise 경보에 대한 입력으로 사용할 수 있습니다. 자산 속성 값을 경보 모델로 AWS IoT SiteWise 보낼 때 표현식을 AWS IoT Events 평가하여 속성 값 또는 자산 ID를 가져옵니다. 경보 모델에서는 다음 표현식을 사용할 수 있습니다.

- 자산 속성 값

자산 속성의 값을 가져오려면 다음 표현식을 사용합니다. **### ##** 모델의 ModelId ID로 바꾸고 **PropertyID# ###** ID로 바꾸십시오.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.propertyValue.value
```

• 자산 ID

자산의 ID를 가져오려면 다음 표현식을 사용합니다. **### ##** 모델의 ModelId ID로 바꾸고 **PropertyID# ###** ID로 바꾸십시오.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.assetId
```

Note

알람 모델을 만들 때 값으로 평가되는 표현식 대신 리터럴을 정의할 수 있습니다. AWS IoT SiteWise 이렇게 하면 자산 모델에 정의하는 속성(attribute)의 수를 줄일 수 있습니다. 하지만 값을 리터럴로 정의하면 자산 모델을 기반으로 자산의 해당 값을 사용자 지정할 수 없습니다. 또한 AWS IoT SiteWise Monitor 사용자는 자산에 대해서만 경보 설정을 구성할 수 있으므로 경보를 사용자 지정할 수 없습니다.

AWS IoT Events 알람 모델 (CLI) 을 만들려면

1. 에서 AWS IoT Events 알람 모델을 생성할 때는 다음을 포함하여 경보가 사용하는 각 속성의 ID를 지정해야 합니다.
 - 복합 자산 모델의 경보 상태 속성
 - 경보가 모니터링하는 속성
 - 임계값 속성(attribute)
 - (선택 사항) IAM Identity Center 아이덴티티 스토어 ID 속성(attribute)
 - (선택 사항) IAM Identity Center 사용자 ID 속성(attribute)
 - (선택 사항) SMS 발신자 ID 속성(attribute)
 - (선택 사항) 이메일 발신 주소 속성(attribute)
 - (선택 사항) 이메일 제목 속성(attribute)
 - (선택 사항) 추가 메시지 속성(attribute)

다음 명령을 실행하여 자산 모델에서 이러한 속성의 ID를 검색합니다. *asset-model-id*를 이전 단계의 자산 모델의 ID로 바꿉니다.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

이 작업은 자산 모델의 세부 사항이 포함된 응답을 반환합니다. 경보가 사용하는 각 속성의 ID를 기록해 둡니다. 다음 단계에서 AWS IoT Events 경보 모델을 생성할 때 이러한 ID를 사용합니다.

2. 에서 알람 모델을 AWS IoT Events 생성하십시오. 다음을 따릅니다.

- a. `alarm-model-payload.json`이라는 파일을 생성합니다.
- b. 다음 JSON 객체를 파일에 복사합니다.
- c. 경보의 이름(`alarmModelName`), 설명(`alarmModelDescription`), 그리고 심각도 (`severity`)를 입력합니다. 심각도에서는 회사의 심각도 수준을 반영하는 정수를 지정합니다.

Important

경보 모델은 이전에 자산 모델에서 정의한 경보 복합 모델과 이름이 같아야 합니다. 경보 모델 이름에는 영숫자만 사용할 수 있습니다.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3
}
```

d. 비교 규칙(`alarmRule`)을 경보에 추가합니다. 이 규칙은 모니터링할 속성(`inputProperty`), 비교할 임계값(`threshold`), 사용할 비교 연산자 (`comparisonOperator`)를 정의합니다.

- *### ##* 모델의 ModelId ID로 대체합니다.
- *PropertyId###* 알람이 모니터링하는 속성의 ID로 바꾸십시오.
- *#### ###* 속성 속성의 AttributeId ID로 대체합니다.
- *GREATER*를 속성 값을 임계값과 비교하는 데 사용할 연산자로 바꿉니다. 다음 옵션 중 하나를 선택합니다.

- LESS
- LESS_OR_EQUAL
- EQUAL
- NOT_EQUAL
- GREATER_OR_EQUAL
- GREATER

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  }
}
```

- e. 경보 상태가 변경될 때 경보 상태를 AWS IoT SiteWise 로 보내는 작업 (alarmEventActions)을 추가합니다.

Note

고급 구성에서 경보 상태가 변경될 때 수행할 추가 작업을 정의할 수 있습니다. 예를 들어 AWS Lambda 함수를 호출하거나 MQTT 주제에 게시할 수 있습니다. 자세한 내용은 AWS IoT Events 개발자 안내서의 [다른 AWS 서비스](#) 사용을 참조하십시오.

- **### ##** 모델의 ModelId ID로 바꾸십시오.
- **PropertyId###** 알람이 모니터링하는 속성의 ID로 바꾸십시오.
- **## StateProperty** 복합 모델의 알람 Id를 알람 상태 속성의 ID로 대체합니다.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "`alarmStatePropertyId`"
        }
      }
    ]
  }
}
```

- f. (선택 사항) 경보 알림 설정을 구성합니다. 경보 알림 작업은 계정의 Lambda 함수를 사용하여 경보 알림을 전송합니다. 자세한 정보는 [경보 알림 요구 사항](#)을 참조하세요. 경보 알림 설정에서 IAM Identity Center 사용자에게 전송하도록 SMS 및 이메일 알림을 구성할 수 있습니다. 다음을 따릅니다.
- i. alarm-model-payload.json의 페이로드에 경보 알림 구성(alarmNotification)을 추가합니다.
- **## NotificationFunction Arn#** 경보 알림을 처리하는 Lambda 함수의 ARN으로 대체하십시오.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
```

```

"severity": 3,
"alarmRule": {
  "simpleRule": {
    "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
    "comparisonOperator": "GREATER",
    "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
  }
},
"alarmEventActions": {
  "alarmActions": [
    {
      "iotSiteWise": {
        "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
        "propertyId": "'alarmStatePropertyId'"
      }
    }
  ]
},
"alarmNotification": {
  "notificationActions": [
    {
      "action": {
        "lambdaAction": {
          "functionArn": "alarmNotificationFunctionArn"
        }
      }
    }
  ]
}
}

```

ii. (선택 사항) 경보 상태가 변경될 때 IAM Identity Center 사용자에게 전송하도록 SMS 알림(smsConfigurations)을 구성합니다.

- *ID# IAM ID ## ID ##### ID# ### ## ID# #####. StoreId AttributeId*
- *### IdAttribute ID#* IAM ID 센터 사용자의 ID가 포함된 속성의 ID로 교체합니다.
- *### IdAttribute ID#* Amazon SNS 발신자 ID가 포함된 속성의 ID로 바꾸거나 senderId 페이로드에서 제거합니다.

- **## MessageAttribute Id#** 추가 메시지가 포함된 속성의 ID로 바꾸거나 additionalMessage 페이로드에서 제거하십시오.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        }
      },
      {
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.va"
                }
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

        "userId":
        "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
    }
    },
    "senderId":
    "$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
    "additionalMessage":
    "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.
    }
    ]
    }
    ]
    }
}

```

iii. (선택 사항) 경보 상태가 변경될 때 IAM Identity Center 사용자에게 전송하도록 이메일 알림(emailConfigurations)을 구성합니다.

- ID를 IAM *Identity* Center ID 스토어 ID 속성 속성의 StoreId AttributeId ID로 대체하십시오.
- *### IdAttribute ID#* IAM ID 센터 사용자 ID 속성 속성의 ID로 교체합니다.
- *From AddressAttribute Id#* "from" 주소 속성 속성의 ID로 바꾸거나 from 페이로드에서 제거하십시오.
- *### SubjectAttribute ID# ###* 제목 속성 속성의 ID로 바꾸거나 subject 페이로드에서 제거합니다.
- *## MessageAttribute ID# ##* 메시지 속성 속성의 ID로 바꾸거나 additionalMessage 페이로드에서 제거합니다.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
      "$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
      "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  }
}

```

```

    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        },
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                  "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                }
              }
            ],
            "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
            "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
          }
        ],
        "emailConfigurations": [
          {
            "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value"
            "recipients": {

```

```

        "to": [
          {
            "ssoIdentity": {
              "identityStoreId":
                "$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
              "userId":
                "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
            }
          }
        ],
        "content": {
          "subject":
            "$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
          "additionalMessage":
            "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
        }
      }
    ]
  }
}

```

- g. (선택 사항) 경보 기능(AlarmCapabilities)을 alarm-model-payload.json의 페이로드에 추가합니다. 이 객체에서 자산 모델을 기반으로 확인 흐름의 활성화 여부와 자산의 기본 활성화 상태를 지정할 수 있습니다. 확인 흐름에 대한 자세한 내용은 [경보 상태](#) 섹션을 참조합니다.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
        "$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
        "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {

```

```

    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ],
    "alarmNotification": {
      "notificationActions": [
        {
          "action": {
            "lambdaAction": {
              "functionArn": "alarmNotificationFunctionArn"
            }
          },
          "smsConfigurations": [
            {
              "recipients": [
                {
                  "ssoIdentity": {
                    "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                    "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                  }
                }
              ],
              "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
              "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
            }
          ],
          "emailConfigurations": [
            {
              "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
              "recipients": {
                "to": [
                  {
                    "ssoIdentity": {

```

```

        "identityStoreId":
        "$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
        "userId":
        "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
    }
    }
    ],
    },
    "content": {
        "subject":
        "$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
        "additionalMessage":
        "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
    }
    }
    ]
    }
    ],
    },
    "alarmCapabilities": {
        "initializationConfiguration": {
            "disabledOnInitialization": false
        },
        "acknowledgeFlow": {
            "enabled": true
        }
    }
}
}

```

- h. 데이터 전송을 AWS IoT Events 위임할 수 있는 IAM 서비스 역할 (roleArn) 을 추가합니다. AWS IoT SiteWise이 역할에는 iotevents.amazonaws.com가 역할을 맡을 수 있도록 허용하는 iotsitewise:BatchPutAssetPropertyValue 권한과 신뢰 관계가 필요합니다. 알림을 보내려면 이 역할에도 lambda:InvokeFunction 및 sso-directory:DescribeUser 권한이 필요합니다. 자세한 정보는 AWS IoT Events 개발자 안내서에서 [경보 서비스 역할](#)을 참조하세요.

- 를 roleArn 이러한 작업을 수행할 AWS IoT Events 수 있는 역할의 ARN으로 대체하십시오.

```

{
    "alarmModelName": "BoilerTemperatureHighAlarm",

```

```

"alarmModelDescription": "Detects when the boiler temperature is high.",
"severity": 3,
"alarmRule": {
  "simpleRule": {
    "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
    "comparisonOperator": "GREATER",
    "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
  }
},
"alarmEventActions": {
  "alarmActions": [
    {
      "iotSiteWise": {
        "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
        "propertyId": "'alarmStatePropertyId'"
      }
    }
  ]
},
"alarmNotification": {
  "notificationActions": [
    {
      "action": {
        "lambdaAction": {
          "functionArn": "alarmNotificationFunctionArn"
        }
      },
      "smsConfigurations": [
        {
          "recipients": [
            {
              "ssoIdentity": {
                "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
              }
            }
          ],
          "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",

```

```

        "additionalMessage":
        "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
    }
  ],
  "emailConfigurations": [
    {
      "from":
        "$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
      "recipients": {
        "to": [
          {
            "ssoIdentity": {
              "identityStoreId":
                "$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
              "userId":
                "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
            }
          }
        ]
      },
      "content": {
        "subject":
          "$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
        "additionalMessage":
          "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
      }
    }
  ]
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": false
  }
},
"roleArn": "arn:aws:iam::123456789012:role/MyIoTEventsAlarmRole"
}

```

- i. 다음 명령을 실행하여 페이로드로부터 AWS IoT Events 경보 모델을 생성합니다. alarm-model-payload.json

```
aws iotevents create-alarm-model --cli-input-json file://alarm-model-payload.json
```

- j. 이 작업은 경보 모델(alarmModelArn)의 ARN이 포함된 응답을 반환합니다. 이 ARN을 복사하여 다음 단계에서 자산 모델의 경보 정의에 설정합니다.

3단계: ~ 사이의 AWS IoT SiteWise 데이터 흐름 활성화 AWS IoT Events

AWS IoT SiteWise 및 AWS IoT Events에서 필요한 리소스를 생성한 후 리소스 간 데이터 흐름을 활성화하여 경보를 활성화할 수 있습니다. 이 섹션에서는 이전 단계에서 만든 경보 모델을 사용하도록 자산 모델의 경보 정의를 업데이트합니다.

AWS IoT SiteWise 와 AWS IoT Events (CLI) 사이의 데이터 흐름을 활성화하려면

- 경보 모델을 자산 모델에서 경보 소스로 설정합니다. 다음을 따릅니다.
 - a. 다음 명령을 실행하여 기존 자산 모델 정의를 검색합니다. *asset-model-id*를 자산 모델의 ID로 바꿉니다.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

이 작업은 자산 모델의 세부 사항이 포함된 응답을 반환합니다.

- b. `update-asset-model-payload.json`이라는 파일을 만들고 이전 명령의 응답을 파일에 복사합니다.
- c. `update-asset-model-payload.json` 파일에서 다음 키-값 페어를 제거합니다.
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
- d. 이전에 정의한 경보 복합 모델에 경보 소스 속성(AWS/ALARM_SOURCE)을 추가합니다. *ModelArn***alarm# ##** 소스 속성의 값을 설정하는 알람 모델의 ARN으로 대체합니다.

```
{
  ...
  "assetModelCompositeModels": [
```

```

...
{
  "name": "BoilerTemperatureHighAlarm",
  "type": "AWS/ALARM",
  "properties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "name": "AWS/ALARM_TYPE",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "IOT_EVENTS"
        }
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "name": "AWS/ALARM_STATE",
      "dataType": "STRUCT",
      "dataTypeSpec": "AWS/ALARM_STATE",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "AWS/ALARM_SOURCE",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "alarmModelArn"
        }
      }
    }
  ]
}
]
}

```

- e. 다음 명령을 실행하여 `update-asset-model-payload.json` 파일에 저장된 정의로 자산 모델을 업데이트합니다. *asset-model-id*를 자산 모델의 ID로 바꿉니다.

```

aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \

```

```
--cli-input-json file://update-asset-model-payload.json
```

이제 자산 모델이 AWS IoT Events에서 이를 감지하는 경보를 정의합니다. 경보는 이 자산 모델을 기반으로 모든 자산의 대상 자산을 모니터링합니다. 각 자산에 대한 경보를 구성하여 각 자산의 임계값 또는 IAM Identity Center 수신자와 같은 속성을 사용자 지정할 수 있습니다. 자세한 정보는 [자산에 대한 경보 구성](#)을 참조하세요.

외부 경보 정의

외부 경보에는 AWS IoT SiteWise외부에서 감지한 경보 상태가 포함됩니다.

외부 경보 정의(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 기존 자산 모델에 외부 경보를 정의할 수 있습니다. 새 자산 모델에 외부 경보를 정의하려면 자산 모델을 만든 다음 다음 단계를 완료합니다. 자세한 정보는 [자산 모델 생성](#)을 참조하세요.

자산 모델에 경보를 정의하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택합니다.
3. 경보를 정의할 자산 모델을 선택합니다.
4. 경보 정의 탭을 선택합니다.
5. 경보 추가를 선택합니다.
6. 경보 유형 옵션에서 외부 경보를 선택합니다.
7. 경보의 이름을 입력합니다.
8. (선택 사항) 경보에 대한 설명을 입력합니다.
9. 경보 추가를 선택합니다.

외부 경보 정의(CLI)

를 사용하여 새 자산 모델 또는 기존 자산 모델에 외부 경보를 정의할 수 있습니다. AWS CLI

자산 모델에 외부 경보를 추가하려면 자산 모델에 경보 복합 모델을 추가합니다. 외부 경보 복합 모델은 EXTERNAL 유형을 지정하며 경보 소스 속성을 지정하지 않습니다. 다음 예제 복합 경보는 외부 온도 경보를 정의합니다.

```

{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "EXTERNAL"
            }
          }
        },
        {
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",
          "type": {
            "measurement": {}
          }
        }
      ]
    }
  ]
}

```

신규 또는 기존 자산 모델에 복합 모델을 추가하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [자산 모델 생성 \(AWS CLI\)](#)
- [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#)

외부 경보를 정의한 후에는 자산 모델을 기반으로 자산에 경보 상태를 수집할 수 있습니다. 자세한 정보는 [외부 경보 상태 수집](#)을 참조하세요.

자산에 대한 경보 구성

자산 모델에 AWS IoT Events 경보를 정의한 후 자산 모델을 기반으로 각 자산에 경보를 구성할 수 있습니다. 임계값과 경보의 알림 설정을 편집할 수 있습니다. 각 값은 자산의 속성(attribute)이므로 속성(attribute)의 기본값을 업데이트하여 이러한 값을 구성할 수 있습니다.

Note

경보에 대해서는 이러한 값을 구성할 수 있지만 외부 AWS IoT Events 경보에는 구성할 수 없습니다.

주제

- [임계값 구성\(콘솔\)](#)
- [임계값 구성 \(AWS CLI\)](#)
- [알림 설정 구성\(콘솔\)](#)
- [알림 설정 구성\(CLI\)](#)

임계값 구성(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 경보의 임계값을 지정하는 속성 값을 업데이트할 수 있습니다.

경보의 임계값 업데이트(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 경보 임계값을 업데이트할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.
5. 경보가 임계값으로 사용하는 속성(attribute)을 찾은 다음 새 값을 입력합니다.
6. 저장을 선택합니다.

임계값 구성 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 경보의 임계값을 지정하는 속성 값을 업데이트 할 수 있습니다.

이 절차를 완료하려면 자산의 `assetId` 및 속성의 `propertyId`를 알고 있어야 합니다. 외부 ID를 사용할 수도 있습니다. 자산을 생성했는데 모르는 경우 [ListAssets](#) API를 사용하여 특정 모델의 모든 자산을 나열하십시오. `assetId` [DescribeAsset](#) 작업을 사용하여 속성 ID를 포함한 자산의 속성을 볼 수 있습니다.

[BatchPutAssetProperty](#) 값을 사용하여 자산에 속성 값을 할당할 수 있습니다. 이 작업을 사용하여 한 번에 여러 속성을 설정할 수 있습니다. 이 작업의 페이로드에는 각각 자산 ID, 속성 ID, 속성 값을 포함하는 항목 목록이 포함되어 있습니다.

속성 값을 업데이트하려면 (AWS CLI)

1. `batch-put-payload.json`이라는 파일을 만들고 다음 JSON 개체를 파일에 복사합니다. 이 페이로드 예제에서는 풍력 터빈의 위도와 경도를 설정하는 방법을 보여줍니다. ID, 값 및 타임스탬프를 업데이트하여 사용 사례에 대한 페이로드를 수정합니다.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
      "propertyValues": [
```

```

    {
      "value": {
        "doubleValue": 122.3491
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      }
    }
  ]
}
]
}

```

- 페이로드의 각 항목은 고유한 문자열로 정의할 수 있는 entryId를 포함합니다. 요청 항목이 실패하면 각 오류에 해당 요청의 entryId가 포함되므로 다시 시도할 요청을 알 수 있습니다.
- 속성 값을 설정하려면 각 속성 속성 목록에 하나의 timestamp-quality-value (TQV) 구조를 포함할 수 있습니다. propertyValues 이 구조에는 새 value와 현재 timestamp가 포함되어야 합니다.
 - value -설정되는 속성 유형에 따라 다음 필드 중 하나를 포함하는 구조입니다.
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue
 - timestamp— 현재 Unix 에포크 시간을 초 단위로 포함하는 구조입니다. timeInSeconds AWS IoT SiteWise 과거에 7일 이상 존재했거나 이후에 5분 이상 존재한 타임스탬프가 있는 데이터 포인트를 거부합니다.

[Value용 페이로드를 준비하는 방법에 대한 자세한 내용은 을 참조하십시오. BatchPut AssetProperty API를 사용한 데이터 수집 AWS IoT SiteWise](#)

2. 다음 명령을 실행하여 속성 값을 다음으로 보내십시오. AWS IoT SiteWise

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

알림 설정 구성(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 경보에 대한 알림 설정을 지정하는 속성 값을 업데이트할 수 있습니다.

경보의 알림 설정을 업데이트하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 경보 설정을 업데이트할 자산을 선택합니다.
4. 편집을 선택합니다.
5. 변경하려는 알림 설정에 경보가 사용하는 속성(attribute)을 찾은 다음 새 값을 입력합니다.
6. 저장을 선택합니다.

알림 설정 구성(CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 경보의 알림 설정을 지정하는 속성 값을 업데이트할 수 있습니다.

이 절차를 완료하려면 자산의 `assetId` 및 속성의 `propertyId`를 알고 있어야 합니다. 외부 ID를 사용할 수도 있습니다. 자산을 생성했는데 모르는 경우 [ListAssets](#) API를 사용하여 특정 모델의 모든 자산을 나열하십시오. `assetId` [DescribeAsset](#) 작업을 사용하여 속성 ID를 포함한 자산의 속성을 볼 수 있습니다.

[BatchPutAssetProperty](#) 값을 사용하여 자산에 속성 값을 할당할 수 있습니다. 이 작업을 사용하여 한 번에 여러 속성을 설정할 수 있습니다. 이 작업의 페이로드에는 각각 자산 ID, 속성 ID, 속성 값을 포함하는 항목 목록이 포함되어 있습니다.

속성 값을 업데이트하려면 (AWS CLI)

1. `batch-put-payload.json`이라는 파일을 만들고 다음 JSON 개체를 파일에 복사합니다. 이 페이로드 예제에서는 풍력 터빈의 위도와 경도를 설정하는 방법을 보여줍니다. ID, 값 및 타임스탬프를 업데이트하여 사용 사례에 대한 페이로드를 수정합니다.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
```

```

"assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
"propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
"propertyValues": [
  {
    "value": {
      "doubleValue": 47.6204
    },
    "timestamp": {
      "timeInSeconds": 1575691200
    }
  }
],
{
  "entryId": "windfarm3-turbine7-longitude",
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
  "propertyValues": [
    {
      "value": {
        "doubleValue": 122.3491
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      }
    }
  ]
}
]
}

```

- 페이로드의 각 항목은 고유한 문자열로 정의할 수 있는 entryId를 포함합니다. 요청 항목이 실패하면 각 오류에 해당 요청의 entryId가 포함되므로 다시 시도할 요청을 알 수 있습니다.
- 속성 값을 설정하려면 각 속성 속성 목록에 하나의 timestamp-quality-value (TQV) 구조를 포함할 수 있습니다. propertyValues 이 구조에는 새 value와 현재 timestamp가 포함되어야 합니다.
 - value -설정되는 속성 유형에 따라 다음 필드 중 하나를 포함하는 구조입니다.
 - booleanValue
 - doubleValue
 - integerValue

- stringValue
- timestamp— 현재 Unix 에포크 시간을 초 단위로 포함하는 구조입니다. timeInSeconds AWS IoT SiteWise 과거에 7일 이상 존재했거나 이후에 5분 이상 존재한 타임스탬프가 있는 데이터 포인트를 거부합니다.

[Value용 페이로드를 준비하는 방법에 대한 자세한 내용은 을 참조하십시오. BatchPut AssetProperty API를 사용한 데이터 수집 AWS IoT SiteWise](#)

2. 다음 명령을 실행하여 속성 값을 다음으로 보내십시오. AWS IoT SiteWise

```
aws iotsitewise batch-put-asset-property-value -\cli-input-json file://batch-put-payload.json
```

경보에 대응

AWS IoT Events 알람 상태가 변경되면 다음과 같이 알람에 응답할 수 있습니다.

- 문제를 처리하고 있음을 나타내는 경보를 확인합니다.
- 경보를 일시 정지하여 일시적으로 비활성화합니다.
- 경보를 비활성화하면 다시 활성화할 때까지 영구적으로 비활성화됩니다.
- 경보 상태를 감지하려면 비활성화된 경보를 활성화합니다.
- 경보를 재설정하 상태 및 최신 값을 지울 수 있습니다.

AWS IoT SiteWise 콘솔이나 AWS IoT Events API를 사용하여 경보에 응답할 수 있습니다.

Note

경보에는 응답할 수 있지만 외부 AWS IoT Events 경보에는 응답할 수 없습니다.

주제

- [경보에 대응\(콘솔\)](#)
- [경보에 응답\(API\)](#)

경보에 대응(콘솔)

AWS IoT SiteWise 콘솔을 사용하여 알람을 확인, 일시 중지, 비활성화 또는 활성화할 수 있습니다.

주제

- [경보 확인\(콘솔\)](#)
- [경보 일시 중지\(콘솔\)](#)
- [경보 비활성화\(콘솔\)](#)
- [경보 활성화\(콘솔\)](#)
- [경보 재설정\(콘솔\)](#)

경보 확인(콘솔)

문제를 처리하고 있음을 나타내는 경보를 확인할 수 있습니다.

Note

경보를 확인하려면 경보에서 확인 흐름을 활성화해야 합니다. AWS IoT SiteWise 콘솔에서 경보를 정의하는 경우 이 옵션이 기본적으로 활성화됩니다.

경보를 확인하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 경보를 확인할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 경보 탭을 선택합니다.
5. 확인할 경보를 선택한 다음 작업을 선택하여 응답 조치 메뉴를 엽니다.
6. 확인을 선택합니다. 경보 상태가 확인됨으로 변경됩니다.

경보 일시 중지(콘솔)

경보를 일시 중지하여 일시적으로 비활성화할 수 있습니다. 경보를 일시 중지할 기간을 지정합니다.

경보를 일시 중지하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 경보를 일시 중지할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 경보 탭을 선택합니다.
5. 일시 중지할 경보를 선택한 다음 작업을 선택하여 응답 조치 메뉴를 엽니다.
6. 일시 정지를 선택합니다. 일시 중지 기간을 지정하는 모델이 열립니다.
7. 일시 중지 길이를 선택하거나 사용자 지정 일시 중지 길이를 입력합니다.
8. 저장을 선택합니다. 경보 상태가 일시 정지 상태로 바뀝니다.

경보 비활성화(콘솔)

경보가 더 이상 감지되지 않도록 비활성화할 수 있습니다. 경보를 비활성화한 후 경보가 감지되도록 하려면 다시 활성화해야 합니다.

경보를 비활성화하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 경보를 사용하지 않을 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 경보 탭을 선택합니다.
5. 비활성화할 경보를 선택한 다음 작업을 선택하여 응답 조치 메뉴를 엽니다.

6. 비활성화를 선택합니다. 경보 상태가 비활성화됨으로 변경됩니다.

경보 활성화(콘솔)

경보를 비활성화하거나 일시 중지한 후에 다시 감지하도록 활성화할 수 있습니다.

경보를 사용하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 경보를 사용하고자 하는 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 경보 탭을 선택합니다.
5. 활성화할 경보를 선택한 다음 경보를 선택하여 응답 조치 메뉴를 엽니다.
6. 활성화를 선택합니다. 경보 상태가 정상으로 변경됩니다.

경보 재설정(콘솔)

경보를 재설정하여 상태 및 최신 값을 지울 수 있습니다.

경보를 재설정하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 경보를 재설정하고자 하는 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 경보 탭을 선택합니다.
5. 활성화할 경보를 선택한 다음 경보를 선택하여 응답 조치 메뉴를 엽니다.
6. 재설정을 선택합니다. 경보 상태가 정상으로 변경됩니다.

경보에 응답(API)

AWS IoT Events API를 사용하여 알람을 확인, 일시 중지, 비활성화, 활성화 또는 재설정할 수 있습니다. 자세한 내용은 AWS IoT Events API 참조에 있는 다음 운영을 참조하세요.

- [BatchAcknowledge알람](#)
- [BatchSnooze알람](#)
- [BatchDisable알람](#)
- [BatchEnable알람](#)
- [BatchReset알람](#)

자세한 정보는 AWS IoT Events 개발자 안내서에서 [경보에 응답하기](#)를 참조하세요.

외부 경보 상태 수집

외부 경보는 외부에서 평가하는 경보입니다. AWS IoT SiteWise에 수집하려는 경보 상태를 보고하는 데이터 소스가 있는 경우 외부 경보를 사용할 수 있습니다.

경보 상태 속성에는 경보 상태 데이터 값에 대한 특정 형식이 필요합니다. 각 데이터 값은 문자열로 직렬화된 JSON 객체여야 합니다. 그런 다음 직렬화된 문자열을 문자열 값으로 수집합니다. 자세한 정보는 [경보 상태 속성](#)을 참조하세요.

Example 경보 상태 데이터 값 예시(직렬화되지 않음)

```
{
  "stateName": "Active"
}
```

Example 경보 상태 데이터 값 예시(직렬화)

```
{\"stateName\": \"Active\"}
```

Note

데이터 소스에서 이 형식의 데이터를 보고할 수 없거나 데이터를 수집하기 전에 이 형식으로 변환할 수 없는 경우, 경보 속성을 사용하지 않도록 선택할 수 있습니다. 대신, 문자열 데이터

유형을 사용하여 측정 속성으로 데이터를 수집 수 있습니다. 자세한 내용은 [장비의 데이터 스트림 정의\(측정값\)](#) 및 [에 데이터 수집 AWS IoT SiteWise](#) 섹션을 참조하세요.

외부 경보 상태 스트림 매핑

속성 별칭을 정의하여 데이터 스트림을 경보 상태 속성에 매핑할 수 있습니다. 이를 통해 데이터를 수집하거나 검색할 때 경보 상태 속성을 쉽게 식별할 수 있습니다. 속성 별칭에 대한 자세한 내용은 [산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.

주제

- [외부 경보 상태 스트림 매핑\(콘솔\)](#)
- [외부 경보 상태 스트림 매핑 \(AWS CLI\)](#)

외부 경보 상태 스트림 매핑(콘솔)

속성 별칭을 정의하여 데이터 스트림을 경보 상태 속성에 매핑할 수 있습니다. 이를 통해 데이터를 수집하거나 검색할 때 경보 상태 속성을 쉽게 식별할 수 있습니다. 속성 별칭에 대한 자세한 내용은 [산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.

AWS IoT SiteWise 콘솔을 사용하여 알람 상태 속성의 별칭을 설정할 수 있습니다.

경보 상태 속성에 속성 별칭을 설정하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 속성 별칭을 설정할 자산을 선택합니다.

Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.
5. 알람으로 스크롤하여 섹션을 확장합니다.
6. 외부 경보에서 속성 별칭에 별칭을 입력합니다 (선택 사항).
7. 저장을 선택합니다.

외부 경보 상태 스트림 매핑 ()AWS CLI

속성 별칭을 정의하여 데이터 스트림을 경보 상태 속성에 매핑할 수 있습니다. 이를 통해 데이터를 수집하거나 검색할 때 경보 상태 속성을 쉽게 식별할 수 있습니다. 속성 별칭에 대한 자세한 내용은 [산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.

AWS Command Line Interface (AWS CLI) 를 사용하여 알람 상태 속성의 별칭을 설정할 수 있습니다.

이 절차를 완료하려면 자산의 `assetId` 및 속성의 `propertyId`를 알고 있어야 합니다. 외부 ID를 사용할 수도 있습니다. 자산을 생성했는데 모르는 경우 [ListAssets](#)API를 사용하여 특정 모델의 모든 자산을 나열하십시오. `assetId` [DescribeAsset](#)작업을 사용하여 속성 ID를 포함한 자산의 속성을 볼 수 있습니다.

Note

[DescribeAsset](#)응답에는 자산의 복합 자산 모델 목록이 포함됩니다. 각 경보는 복합 모델입니다. `propertyId`를 찾으려면 경보의 복합 모델을 찾은 다음 해당 복합 모델에서 `AWS/ALARM_STATE` 속성을 찾습니다.

속성 별칭을 설정하여 그렇게 하는 방법에 대한 자세한 내용은 [속성 별칭 설정 \(\)AWS CLI](#)을 참조하세요.

경보 상태 데이터 수집

경보 상태 속성은 경보 상태를 직렬화된 JSON 문자열로 예상합니다. 에서 AWS IoT SiteWise외부 경보에 경보 상태를 인제스트하려면 이 직렬화된 문자열을 타임스탬프가 있는 문자열 값으로 인제스트합니다. 다음 예제는 활성 경보의 상태 데이터 값을 보여줍니다.

```
{\"stateName\": \"Active\"}
```

경보 자산 속성을 식별하려면 다음 중 하나를 지정합니다.

- 데이터를 보낼 경보 자산 속성의 `assetId` 및 `propertyId`.
- 데이터 스트림 별칭(예: `/company/windfarm/3/turbine/7/temperature/high`)인 `propertyAlias`. 이 옵션을 사용하려면 먼저 경보 속성의 별칭을 설정해야 합니다. 경보 상태 속성의 속성 별칭을 설정하는 방법을 알아보려면 [외부 경보 상태 스트림 매핑](#) 섹션을 참조합니다.

다음 예제 [BatchPutAssetPropertyValue](#) API 페이로드는 외부 경보의 상태를 포맷하는 방법을 보여줍니다. 이 외부 경보는 풍력 터빈의 분당 회전수(RPM) 측정값이 너무 높을 때 보고합니다.

Example 알람 상태 BatchPutAssetPropertyValue 데이터의 페이로드 예시

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature/high",
      "propertyValues": [
        {
          "value": {
            "stringValue": "{\"stateName\":\"Active\"}"
          },
          "timestamp": {
            "timeInSeconds": 1607550262
          }
        }
      ]
    }
  ]
}
```

BatchPutAssetPropertyValue API를 사용하여 데이터를 수집하는 방법에 대한 자세한 내용은 [API를 사용한 데이터 수집 AWS IoT SiteWise](#) 섹션을 참조하세요.

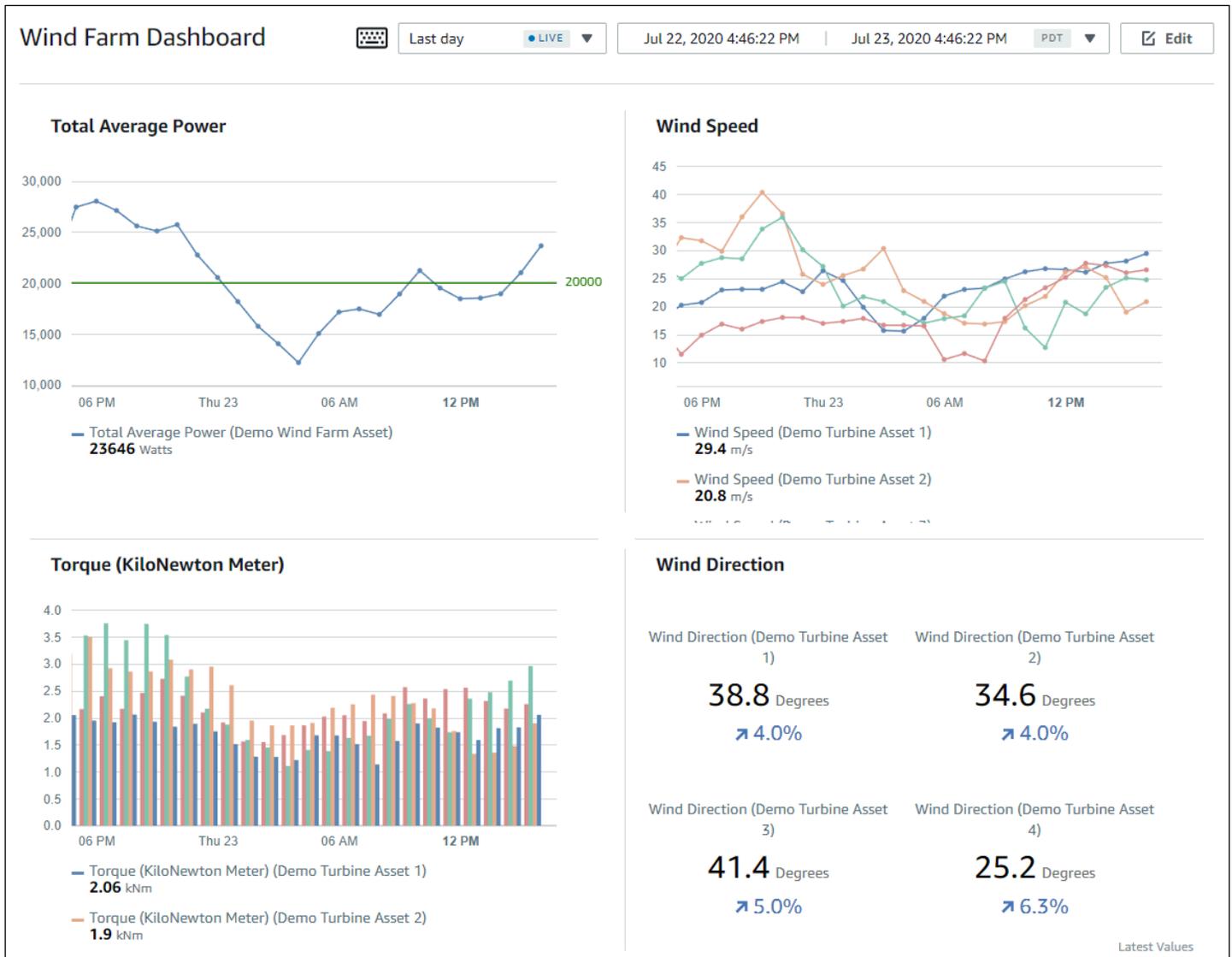
데이터를 수집하는 다른 방법에 대한 자세한 내용은 [에 데이터 수집 AWS IoT SiteWise](#) 섹션을 참조하세요.

를 사용하여 데이터 모니터링 AWS IoT SiteWise Monitor

Monitor 웹 포털을 만들어 프로세스, 장치 및 장비의 데이터를 SiteWise 모니터링하는 AWS IoT SiteWise 데 사용할 수 있습니다. SiteWise 모니터는 관리형 웹 애플리케이션의 AWS IoT SiteWise 형태로 포털을 생성하는 데 사용할 수 있는 기능입니다. 이러한 포털을 사용하여 운영 데이터를 보고 공유할 수 있습니다. 대시보드로 프로젝트를 만들어, AWS IoT에 연결된 프로세스, 디바이스 및 장비의 데이터를 시각화할 수 있습니다.

프로세스 엔지니어 같은 도메인 전문가는 이러한 포털을 사용해 운영 데이터에 대한 통찰을 빠르게 확보하여 디바이스와 장비의 동작을 파악할 수 있습니다.

다음은 풍력발전단지에 대한 데이터를 표시하는 대시보드의 예입니다.



시간 경과에 따른 데이터를 AWS IoT SiteWise 캡처하므로 SiteWise Monitor를 사용하여 시간 경과에 따른 운영 데이터 또는 특정 시점에서 마지막으로 보고된 값을 볼 수 있습니다. 이렇게 하면 다른 방법으로는 얻기 힘든 통찰을 얻을 수 있습니다.

SiteWise 모니터 역할

네 가지 역할이 SiteWise 모니터와 상호 작용합니다.

AWS 관리자

AWS 관리자는 AWS IoT SiteWise 콘솔을 사용하여 포털을 생성합니다. AWS 관리자는 포털 관리자를 할당하고 포털 사용자를 추가할 수도 있습니다. 포털 관리자는 나중에 포털 사용자를 프로젝트에 소유자 또는 뷰어로 할당합니다. AWS 관리자는 AWS 콘솔에서만 작업합니다.

포털 관리자

각 SiteWise 모니터 포털에는 한 명 이상의 포털 관리자가 있습니다. 포털 관리자는 포털을 사용하여 자산 및 대시보드 컬렉션이 포함된 프로젝트를 생성합니다. 그런 다음 포털 관리자가 각 프로젝트에 자산과 소유자를 할당합니다. 프로젝트에 대한 액세스를 제어하면 포털 관리자가 프로젝트 소유자와 뷰어가 볼 수 있는 자산을 제한할 수 있습니다.

프로젝트 소유자

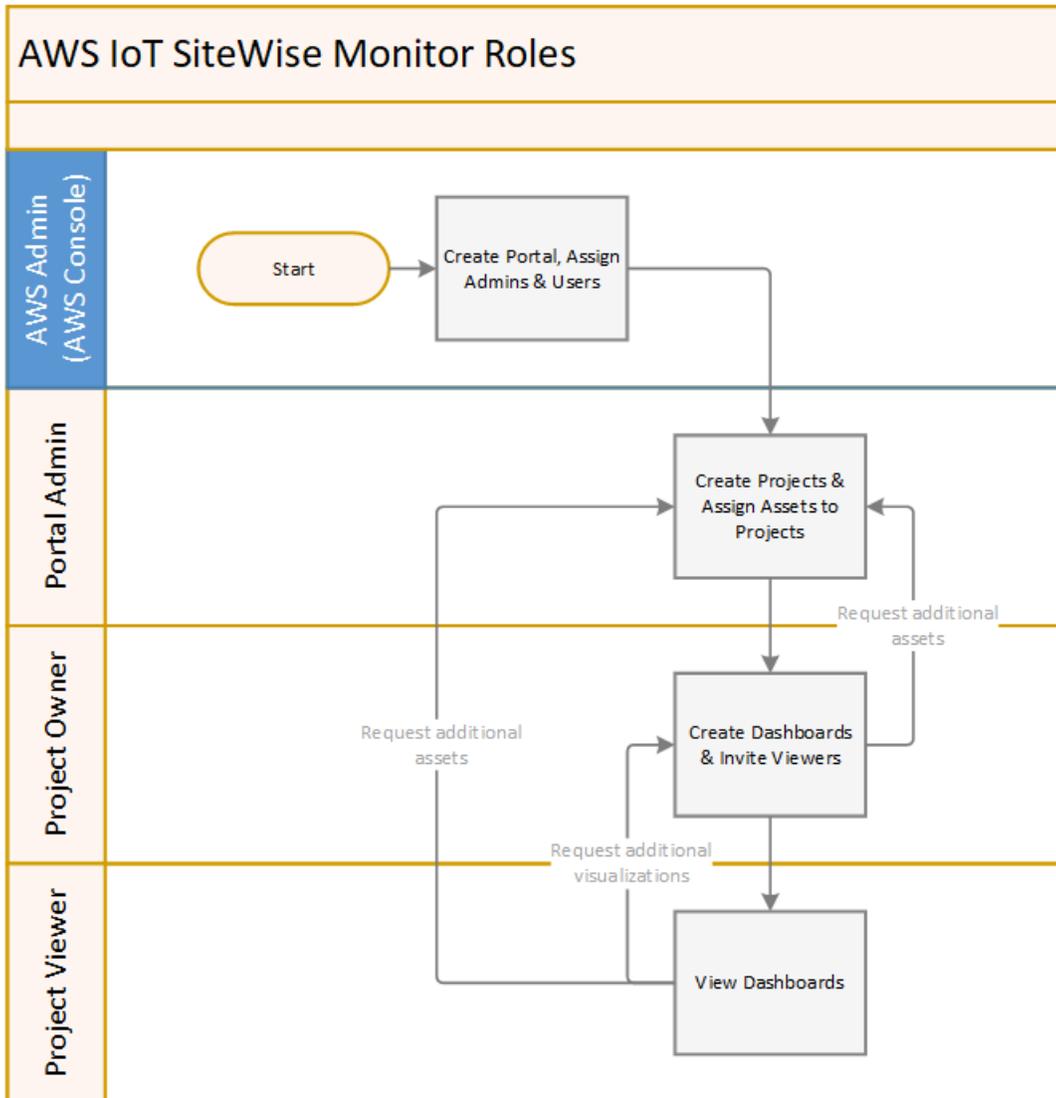
각 SiteWise 모니터 프로젝트에는 소유자가 있습니다. 프로젝트 소유자는 운영 데이터를 일관된 방식으로 나타내기 위해 대시보드 형태로 시각화를 생성합니다. 대시보드를 공유할 준비가 되면 프로젝트 소유자가 프로젝트에 뷰어를 초대할 수 있습니다. 프로젝트 소유자는 프로젝트에 다른 소유자를 할당할 수도 있습니다. 프로젝트 소유자는 경보에 대한 임계값 및 알림 설정을 구성할 수 있습니다.

프로젝트 뷰어

각 SiteWise 모니터 프로젝트에는 뷰어가 있습니다. 프로젝트 뷰어는 포털에 연결하여 프로젝트 소유자가 생성한 대시보드를 볼 수 있습니다. 각 대시보드에서 프로젝트 뷰어는 시간 범위를 조정하여 운영 데이터를 더 잘 이해할 수 있습니다. 프로젝트 뷰어는 액세스 권한이 있는 프로젝트의 대시보드만 볼 수 있습니다. 프로젝트 뷰어는 경보를 확인하고 일시 중지할 수 있습니다.

조직에 따라 같은 사람이 여러 역할을 수행할 수 있습니다.

다음 이미지는 SiteWise 모니터 포털에서 이러한 네 가지 역할이 상호 작용하는 방식을 보여줍니다.



AWS IAM Identity Center 또는 IAM을 사용하여 데이터에 액세스할 수 있는 사용자를 관리할 수 있습니다. 데이터 사용자는 IAM Identity Center 또는 IAM 자격 증명을 사용하여 데스크톱 또는 모바일 브라우저에서 SiteWise Monitor에 로그인할 수 있습니다.

SAML 페더레이션

IAM Identity Center 및 IAM은 [SAML\(Security Assertion Markup Language\) 2.0](#)을 사용한 ID 페더레이션을 지원합니다. SAML 2.0은 많은 외부 ID 제공업체 (IdPs) 가 사용자를 인증하고 ID 및 보안 정보를 서비스 제공업체 (SP) 에 전달하는 데 사용하는 개방형 표준입니다. SP는 일반적으로 애플리케이션 또는 서비스입니다. SAML 페더레이션을 사용하면 SiteWise Monitor 포털 관리자와 사용자가 회사 사용자 이름 및 암호와 같은 외부 자격 증명으로 할당된 포털에 로그인할 수 있습니다.

모니터 포털에 액세스하기 위해 SAML 기반 페더레이션을 사용하도록 IAM ID 센터 및 IAM을 구성할 수 있습니다. SiteWise

IAM Identity Center

포털 관리자와 사용자는 회사 사용자 이름과 비밀번호를 사용하여 AWS 액세스 포털에 로그인할 수 있습니다. 그러면 할당된 SiteWise 모니터 포털로 이동할 수 있습니다. IAM Identity Center는 인증서를 사용하여 ID 공급자와 간에 SAML 신뢰 관계를 설정합니다. AWS자세한 내용은 AWS IAM Identity Center 사용 설명서의 [SCIM 프로파일 및 SAML 2.0 구현](#)을 참조하세요.

IAM

포털 관리자와 사용자는 할당된 SiteWise 모니터 포털에 접근하기 위해 임시 보안 자격 증명을 요청할 수 있습니다. IAM에서 SAML ID 공급자 ID를 생성하여 ID 공급자와 사이에 신뢰 관계를 설정합니다. AWS자세한 내용은 IAM [사용 설명서의 API 액세스를 AWS위한 SAML 기반 페더레이션 사용을 참조하십시오](#).

포털 관리자와 사용자는 회사 포털에 로그인하고 관리 콘솔로 이동하는 옵션을 선택할 수 있습니다. AWS 그러면 할당된 SiteWise 모니터 포털로 이동할 수 있습니다. 회사 포털은 ID 공급자와 AWS ID 공급자 간의 신뢰 교환을 처리합니다. 자세한 내용은 IAM 사용 [설명서의 SAML 2.0 연동 사용자의 AWS 관리 콘솔 액세스 활성화](#)를 참조하십시오.

Note

포털에 사용자나 관리자를 추가할 때 제한된 IP와 같이 사용자 권한을 제한하는 IAM 정책을 생성하지 마십시오. 권한이 제한된 연결된 정책은 포털에 연결할 수 없습니다. AWS IoT SiteWise

SiteWise 모니터 개념

SiteWise Monitor를 사용하려면 다음 개념을 잘 알고 있어야 합니다.

Portal

AWS IoT SiteWise Monitor 포털은 AWS IoT SiteWise 데이터를 시각화하고 공유하는 데 사용할 수 있는 웹 애플리케이션입니다. 포털에는 관리자가 한 명 이상 있으며 프로젝트가 0개 이상 포함되어 있습니다.

프로젝트

각 SiteWise 모니터 포털에는 프로젝트 세트가 포함되어 있습니다. 각 프로젝트에는 연결된 AWS IoT SiteWise 자산의 하위 집합이 있습니다. 프로젝트 소유자는 하나 이상의 대시보드를 생성하여 해당 자산과 연결된 데이터를 일관된 방식으로 볼 수 있습니다. 프로젝트 소유자는 프로젝트에 뷰어를 초대하여 프로젝트의 자산 및 대시보드를 보도록 허용할 수 있습니다. 프로젝트는 SiteWise Monitor 내 공유의 기본 단위입니다. 프로젝트 소유자는 AWS 관리자로부터 포털 접근 권한을 부여 받은 사용자를 초대할 수 있습니다. 해당 포털에 있는 프로젝트를 해당 사용자와 공유하려면 먼저 사용자에게 포털에 대한 액세스 권한이 있어야 합니다.

자산

산업 AWS IoT SiteWise 장비에서 데이터를 수집하면 장치, 장비 및 프로세스가 각각 자산으로 표시됩니다. 각 자산에는 관련 속성 및 경보가 있습니다. 포털 관리자는 각 프로젝트에 자산 집합을 할당합니다.

속성

속성은 자산과 관련된 시계열 데이터입니다. 예를 들어 장비에는 일련 번호, 위치, 제조업체 및 모델, 설치 날짜가 있을 수 있습니다. 또한 가용성, 성능, 품질, 온도, 압력 등에 대한 시계열 값도 있을 수 있습니다.

경보

경보는 속성을 모니터링하여 장비가 작동 범위를 벗어나는 시기를 식별합니다. 각 경보는 모니터링할 임계값과 속성을 정의합니다. 속성이 임계값을 초과하면 경보가 활성화되어 사용자 또는 팀원이 문제를 해결해야 함을 알립니다. 프로젝트 소유자는 경보의 임계값 및 알림 설정을 사용자 지정할 수 있습니다. 프로젝트 뷰어는 경보를 확인하고 일시 중지할 수 있으며 경보에 대한 세부 정보 또는 경보를 해결하기 위해 취한 조치를 메시지로 남길 수 있습니다.

대시보드

각 프로젝트에는 대시보드 집합이 포함되어 있습니다. 대시보드는 자산 집합의 값에 대한 시각화 집합을 제공합니다. 프로젝트 소유자는 대시보드와 대시보드에 포함된 시각화를 생성합니다. 프로젝트 소유자가 대시보드 집합을 공유할 준비가 되면 소유자는 프로젝트에 뷰어를 초대하여 프로젝트의 모든 대시보드에 대한 액세스 권한을 부여할 수 있습니다. 다른 대시보드에 대해 다른 뷰어 집합을 사용하려면 프로젝트 간에 대시보드를 나누어야 합니다. 뷰어는 대시보드를 볼 때 특정 데이터를 볼 수 있도록 시간 범위를 사용자 지정할 수 있습니다.

시각화

각 대시보드에서 프로젝트 소유자는 프로젝트와 연결된 속성과 경보를 표시하는 방법을 결정합니다. 가용성은 꺾은선형 차트로 가장 잘 표현되고 다른 값은 막대 차트나 핵심 성능 지표(KPI)로 표시

될 수 있습니다. 경보는 상태 그리드 및 상태 타임라인으로 표시하는 것이 가장 좋습니다. 프로젝트 소유자는 각 시각화를 사용자 지정하여 해당 자산의 데이터를 가장 잘 이해할 수 있도록 합니다.

시작하기 AWS IoT SiteWise Monitor

기관 AWS 관리자인 경우 AWS IoT SiteWise 콘솔에서 포털을 생성합니다. 기관 구성원이 AWS IoT SiteWise 데이터를 볼 수 있도록 포털을 생성하려면 다음 단계를 완료하세요.

1. 포털 구성 및 생성
2. 포털 관리자 추가 및 초대 이메일 보내기
3. 포털 사용자 추가

포털을 생성한 후 포털 관리자는 AWS IoT SiteWise 자산을 보고 포털의 프로젝트에 할당할 수 있습니다. 그러면 프로젝트 소유자가 대시보드를 만들어 자산 속성을 시각화할 수 있는데, 이는 프로젝트 뷰어가 디바이스, 프로세스 및 장비의 작동 방식을 이해하는 데 도움이 됩니다.

Note

포털에 사용자나 관리자를 추가할 때 제한된 IP와 같이 사용자 권한을 제한하는 AWS Identity and Access Management (IAM) 정책을 생성하지 마세요. 권한이 제한된 연결된 정책은 AWS IoT SiteWise 포털에 연결할 수 없습니다.

풍력 발전소 데이터를 사용하여 특정 시나리오의 프로젝트, 대시보드 및 여러 사용자가 포함된 포털을 설정하는 데 필요한 단계를 안내하는 자습서를 따를 수 있습니다. 자세한 정보는 [모니터에서 SiteWise 풍력 발전 단지 데이터 시각화 및 공유](#)를 참조하세요.

주제

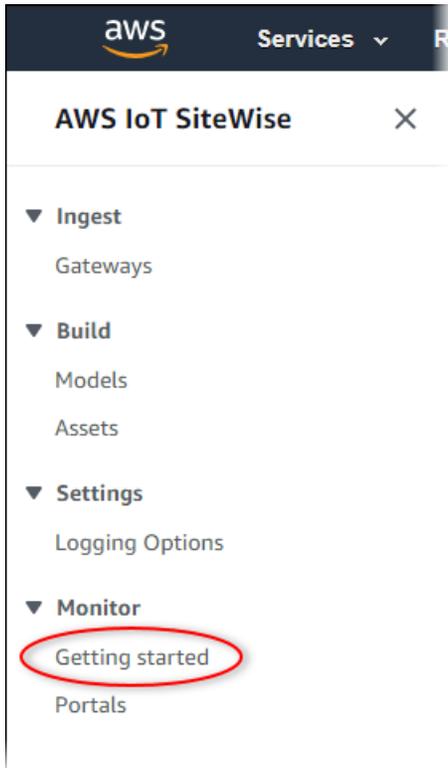
- [포털 생성](#)
- [포털 구성](#)
- [관리자 초대](#)
- [포털 사용자 추가](#)

포털 생성

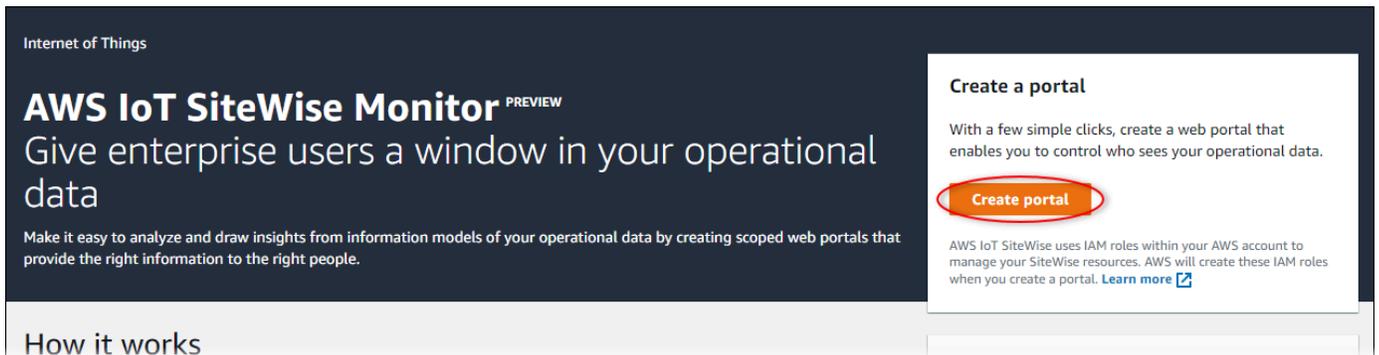
AWS IoT SiteWise 콘솔에서 SiteWise 모니터 포털을 생성합니다.

포털 생성

1. [AWS IoT SiteWise 콘솔](#)에 로그인합니다.
2. 탐색 창에서 모니터, 시작하기를 선택합니다.



3. Create Portal(포털 생성)을 선택합니다.



그런 다음 포털을 구성하기 위한 몇 가지 기본 정보를 입력해야 합니다.

포털 구성

사용자는 포털을 사용하여 데이터를 봅니다. 포털의 이름, 설명, 브랜딩, 사용자 인증, 지원 연락처 이메일 및 권한을 사용자 지정할 수 있습니다.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configurationStep 2 - optional
Additional featuresStep 3
Invite administratorsStep 4
Assign users

Portal configuration

Each web portal provides enterprise users with access to your IoT SiteWise assets. [Learn more](#)

Portal details

Portal name

Choose a portal name to identify the web portal to your users. Company name is recommended.

example-factory-1

Name should be 1-128 characters and only contain A-Z a-z 0-9 _ and -.

Description - optional

Create a description of your portal

Example Corp Factory #1 in Renton, WA

Description should contain a maximum of 2048 characters.

Portal branding

You can provide your logo image to display your brand in this web portal.

Logo image

Upload a square, high-resolution .png file. The image is displayed on a dark background.

Choose file

The file size must be less than 1 MB.

User authentication

Your users can sign in to this portal with their AWS Single Sign-On (AWS SSO) or AWS Identity and Access Management (IAM) credentials. If you choose AWS SSO, you must enable the service for your AWS account.

 You haven't enabled AWS SSO in your account yet. When you create your first portal user, this automatically enables AWS SSO in your AWS account.

Create user

AWS SSO

Your users can sign in to the portal with their corporate usernames and passwords.

IAM

Your users can sign in to the portal with their IAM credentials.

Support contact email

You can provide an email address for cases where there's a problem or issue with this portal and your users need to contact support to resolve.

Email

support@example.com

Tags

This resource doesn't have any tags.

Add tag

You can add up to 50 more tags.

Permissions

SiteWise Monitor assumes this role to give permissions to your federated users to access AWS IoT SiteWise resources. [Learn](#)

포털을 구성하려면

1. 프로젝트의 이름을 입력합니다.
2. (선택 사항) 포털에 대한 설명을 입력합니다. 포털이 여러 개인 경우 의미 있는 설명을 사용하여 각 포털에 포함된 내용을 추적할 수 있습니다.
3. (선택 사항) 이미지를 업로드하여 포털에 브랜드를 표시합니다. 정사각형의 PNG 이미지를 선택합니다. 정사각형이 아닌 이미지를 업로드하는 경우 포털에서 이미지의 크기를 정사각형으로 축소합니다.
4. 다음 옵션 중 하나를 선택합니다:
 - 포털 사용자가 회사 사용자 이름과 암호를 사용하여 이 포털에 로그인하는 경우 IAM Identity Center를 선택합니다.

계정에서 IAM Identity Center를 활성화하지 않은 경우, 다음을 수행하세요.

- a. 사용자 생성을 선택합니다.
- b. 사용자 생성 페이지에서 첫 번째 포털을 생성하려면 사용자의 이메일 주소, 이름, 성을 입력한 다음 사용자 생성을 선택합니다.

The screenshot shows a 'Create user' dialog box with the following fields and values:

- Email address:** janedoe@example.com
- First name:** Jane
- Last name:** Doe

At the bottom right, there are two buttons: 'Cancel' and 'Create user'. The 'Create user' button is highlighted with a red circle.

Note

- AWS 첫 번째 포털 사용자를 생성하면 계정에서 IAM ID 센터가 자동으로 활성화됩니다.
- 한 번에 한 리전에서 IAM Identity Center를 구성할 수 있습니다. SiteWise 모니터는 IAM ID 센터용으로 구성된 지역에 연결됩니다. 즉, IAM Identity Center

액세스에는 하나의 리전을 사용하지만 어느 리전에서든 포털을 생성할 수 있습니다.

- 포털 사용자가 IAM 자격 증명으로 이 포털에 로그인하는 경우 IAM을 선택하세요.

⚠ Important

사용자 또는 역할에는 포털에 로그인할 수 있는 `iotsitewise:DescribePortal` 권한이 있어야 합니다.

5. 포털에 문제가 있고 이를 해결하는 데 도움이 필요한 경우 포털 사용자가 문의할 수 있는 이메일 주소를 입력합니다.
6. (선택 사항) 포털에 대한 태그를 추가합니다. 자세한 정보는 [리소스에 태그 지정하기 AWS IoT SiteWise](#)을 참조하세요.
7. 다음 옵션 중 하나를 선택합니다:
 - 새 서비스 역할 생성 및 사용을 선택합니다. 기본적으로 SiteWise Monitor는 각 포털에 대한 서비스 역할을 자동으로 생성합니다. 이 역할을 통해 포털 사용자는 AWS IoT SiteWise 리소스에 접근할 수 있습니다. 자세한 정보는 [서비스 역할 사용 대상 AWS IoT SiteWise Monitor](#)을 참조하세요.
 - 기존 서비스 역할 사용을 선택하고 대상 역할을 선택합니다.
8. 다음을 선택합니다.
9. (선택 사항) 포털의 경보를 활성화합니다. 자세한 정보는 [포털의 경보 활성화](#)을 참조하세요.
10. 생성을 선택합니다. AWS IoT SiteWise 포털을 생성합니다.

i Note

콘솔을 닫으면 관리자와 사용자를 추가하여 설치 프로세스를 완료할 수 있습니다. 자세한 정보는 [포털 관리자 추가 또는 제거](#)을 참조하세요. 이 포털을 유지용하지 않으려면 리소스를 사용하지 않도록 삭제하세요. 자세한 정보는 [포털 삭제](#)을 참조하세요.

상태 열은 다음 값 중 하나일 수 있습니다.

- CREATING - AWS IoT SiteWise 는 포털 생성 요청을 처리하는 중입니다. 이 프로세스는 완료하는데 몇 분 정도 걸립니다.

- 업데이트 - AWS IoT SiteWise 는 포털 업데이트 요청을 처리하는 것입니다. 이 프로세스는 완료하는 데 몇 분 정도 걸립니다.
- PENDING - AWS IoT SiteWise 는 DNS 레코드 전파가 완료되기를 기다리는 중입니다. 이 프로세스는 완료하는 데 몇 분 정도 걸립니다. 대기 중 상태일 때 포털을 삭제할 수 있습니다.
- 삭제 - AWS IoT SiteWise 는 포털 삭제 요청을 처리하는 것입니다. 이 프로세스는 완료하는 데 몇 분 걸릴 수 있습니다.
- 활성화 - 포털이 활성화되면 포털 사용자가 포털에 액세스할 수 있습니다.
- 실패 AWS IoT SiteWise - 포털 생성, 업데이트 또는 삭제 요청을 처리할 수 없습니다. Amazon CloudWatch Logs로 로그를 AWS IoT SiteWise 전송하도록 설정한 경우 이러한 로그를 사용하여 문제를 해결할 수 있습니다. 자세한 내용은 [CloudWatch 로그를 AWS IoT SiteWise 사용한 모니터링을 참조하십시오](#).

포털이 생성되면 메시지가 나타납니다.

Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-11111EXAMPLE.app.iotsitewise.aws

다음으로, 한 명 이상의 포털 관리자를 포털에 초대해야 합니다. 지금까지 포털을 만들었지만 아무도 액세스할 수 없습니다.

관리자 초대

새 포털에서 시작하려면 포털 관리자를 할당해야 합니다. 포털 관리자는 프로젝트를 만들고, 프로젝트 소유자를 선택하고, 프로젝트에 자산을 할당합니다. 포털 관리자는 모든 AWS IoT SiteWise 자산을 볼 수 있습니다.

사용자 인증 서비스에 따라 다음 옵션 중 하나를 선택합니다.

IAM Identity Center

SiteWise Monitor를 처음 사용하는 경우 이전에 생성한 사용자를 포털 관리자로 선택할 수 있습니다. 다른 사용자를 포털 관리자로 추가하려는 경우 이 페이지에서 IAM Identity Center 사용자를 생성할 수 있습니다. 또는 외부 자격 증명 공급자를 IAM Identity Center에 연결할 수 있습니다. 자세한 내용은 [AWS IAM Identity Center 사용 설명서](#)를 참조하십시오.

관리자를 초대하려면

1. 포털 관리자로 초대할 사용자의 확인란을 선택합니다. 그러면 포털 관리자 목록에 사용자가 추가됩니다.

Note

IAM Identity Center를 자격 증명 스토어로 사용하고 AWS Organizations 관리 계정에 로그인한 경우 사용자 생성을 선택하여 IAM Identity Center 사용자를 생성할 수 있습니다. IAM Identity Center는 새 사용자에게 비밀번호를 설정하라는 이메일을 보냅니다. 그런 다음 사용자를 포털에 관리자로 할당할 수 있습니다. 자세한 내용은 [IAM Identity Center에서 자격 증명 관리](#) 섹션을 참조하세요.

2. (선택 사항) 선택한 사용자에게 초대 보내기를 선택합니다. 이메일 클라이언트가 열리고 메시지 본문에 초대장이 채워집니다.

이메일을 포털 관리자에게 보내기 전에 사용자 지정할 수 있습니다. 나중에 포털 관리자에게 이메일을 보낼 수도 있습니다. 처음으로 SiteWise Monitor를 사용해 보고 새 IAM Identity Center 또는 IAM 사용자 또는 역할을 포털 관리자로 추가하는 경우에는 이메일을 보내지 않아도 됩니다.

3. 관리자로 초대하지 않을 사용자를 추가하는 경우 해당 사용자의 확인란 선택을 취소합니다.
4. 포털 관리자 초대를 마쳤으면 다음을 선택합니다.

IAM

포털 관리자가 될 사용자 또는 역할을 선택할 수 있습니다. 다른 사용자나 역할을 포털 관리자로 추가하려는 경우 IAM 콘솔에서 사용자 또는 역할을 생성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 계정에서 IAM 사용자 생성](#) 및 [IAM 역할 생성](#)을 참조하세요.

관리자를 초대하려면

1. 다음을 따릅니다.
 - IAM 사용자를 선택하여 IAM 사용자를 포털 관리자로 추가합니다.
 - IAM 역할을 선택하여 IAM 역할을 포털 관리자로 추가합니다.
2. 포털 관리자로 지정하려는 사용자 또는 확인란을 선택합니다. 그러면 포털 관리자 목록에 사용자 또는 역할이 추가됩니다.
3. 관리자로 지정하지 않으려는 사용자 또는 역할을 추가하는 경우 해당 사용자 또는 역할의 확인란 선택을 취소합니다.
4. 포털 관리자 초대를 마쳤으면 다음을 선택합니다.

⚠ Important

사용자 또는 역할에는 포털에 로그인할 수 있는 `iotsitewise:DescribePortal` 권한이 있어야 합니다.

ℹ Note

IAM Identity Center를 자격 증명 스토어로 사용하고 AWS Organizations 관리 계정에 로그인한 경우 사용자 생성을 선택하여 IAM Identity Center 사용자를 생성할 수 있습니다. IAM Identity Center는 새 사용자에게 비밀번호를 설정하라는 이메일을 보냅니다. 그런 다음 사용자를 포털에 관리자로 할당할 수 있습니다. 자세한 내용은 [IAM Identity Center에서 자격 증명 관리](#) 섹션을 참조하세요.

나중에 포털 관리자 목록을 변경할 수 있습니다. 자세한 정보는 [포털 관리자 추가 또는 제거](#)를 참조하세요.

ℹ Note

포털 관리자만 프로젝트를 생성하고 프로젝트에 자산을 할당할 수 있으므로 포털 관리자를 한 명 이상 지정해야 합니다.

마지막 단계로, 새 포털에 액세스할 수 있는 사용자를 추가합니다.

포털 사용자 추가

포털에 액세스할 수 있는 사용자를 통제합니다. 각 포털에서 포털 관리자는 하나 이상의 프로젝트를 생성하고 포털 사용자를 각 프로젝트의 소유자 또는 뷰어로 할당합니다. 각 프로젝트 소유자는 프로젝트를 소유하거나 볼 수 있는 추가 포털 사용자를 초대할 수 있습니다.

사용자 인증 서비스에 따라 다음 옵션 중 하나를 선택합니다.

IAM Identity Center

사용자 목록에 사용자를 추가하려면 다음 단계를 완료하세요.

포털 사용자를 추가하려면

1. 사용자 목록에서 포털에 추가할 사용자를 선택합니다. 그러면 포털 사용자 목록에 사용자가 추가됩니다. SiteWise Monitor를 처음 사용하는 경우 포털 관리자를 포털 사용자로 추가할 필요가 없습니다.

Note

IAM Identity Center를 자격 증명 스토어로 사용하고 AWS Organizations 관리 계정에 로그인한 경우 사용자 생성을 선택하여 IAM Identity Center 사용자를 생성할 수 있습니다. IAM Identity Center는 새 사용자에게 비밀번호를 설정하라는 이메일을 보냅니다. 그런 다음 사용자를 포털에 사용자로 할당할 수 있습니다. 자세한 내용은 [IAM Identity Center에서 자격 증명 관리](#) 섹션을 참조하세요.

2. 포털에 액세스하지 못하는 사용자를 추가하는 경우 해당 사용자의 확인란 선택을 취소합니다.
3. 사용자 선택을 마치면 사용자 할당을 선택합니다.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
[Invite administrators](#)

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users (2) Create user

Find resources

<input type="checkbox"/>	Display name	Email
<input type="checkbox"/>	Jane Doe	janedoe@example.com
<input checked="" type="checkbox"/>	John Doe	johndoe@example.com

Selected users (1)

Cancel Previous **Assign users**

IAM

IAM 사용자 또는 IAM 역할 목록에 추가하려는 사용자 또는 역할이 있으면 다음 단계를 완료하세요.

포털 사용자를 추가하려면

1. 다음 옵션이 있습니다.

- IAM 사용자를 선택하여 IAM 사용자를 포털 사용자로 추가합니다.
- IAM 역할을 선택하여 IAM 역할을 포털 사용자로 추가합니다.

SiteWise 모니터를 처음 사용하는 경우 포털 관리자를 포털 사용자로 추가할 필요가 없습니다.

2. 포털 사용자로 지정하려는 사용자 또는 역할의 확인란을 선택합니다. 그러면 포털 사용자 목록에 사용자 또는 역할이 추가됩니다.
3. 포털에 액세스하지 못하는 사용자를 추가하는 경우 해당 사용자의 확인란 선택을 취소합니다.
4. 사용자 선택을 마치면 사용자 할당을 선택합니다.

⚠ Important

사용자 또는 역할에는 포털에 로그인할 수 있는 `iotsitewise:DescribePortal` 권한이 있어야 합니다.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
[Invite administrators](#)

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	raspberrypi-testing	11-08-2019

Portal users (1) [Remove](#)

Cancel Previous **Assign users**

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
Invite administrators

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users **Roles**

IAM roles (66) [Manage roles in IAM console](#)

Find role name

<input type="checkbox"/>	Name	Date created
<input type="checkbox"/>	[REDACTED]	
<input checked="" type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_EcKt-2Oar	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_GTnd004Wr	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	

► Portal users (2) Remove

Cancel Previous **Assign users**

축하합니다! 포털을 성공적으로 생성하고 포털 관리자를 할당했으며, 해당 포털을 사용할 수 있는 사용자를 초대하여 할당했습니다. 이제 포털 관리자가 프로젝트를 생성하고 해당 프로젝트에 자산을 할당할 수 있습니다. 그런 다음 프로젝트 소유자는 대시보드를 생성하여 각 프로젝트 자산에 대한 데이터를 시각화할 수 있습니다.

나중에 포털 사용자 목록을 변경할 수 있습니다. 자세한 정보는 [포털 사용자 추가 또는 제거](#)를 참조하세요.

포털을 변경해야 하는 경우 [SiteWise 모니터 포털 관리](#) 단원을 참조하십시오.

포털에서 시작하려면 SiteWise Monitor 애플리케이션 가이드에서 [시작하기](#)를 참고하세요.

대시보드 생성(AWS Command Line Interface)

AWS CLI를 사용하여 대시보드에서 시각화(또는 위젯)를 정의하는 경우, `dashboardDefinition` JSON 문서에 다음 정보를 지정해야 합니다. 이 정의는 [CreateDashboard](#) 및 [UpdateDashboard](#) 연산의 매개변수입니다.

widgets

각각 다음 정보를 포함하는 위젯 정의 구조의 목록입니다.

type

위젯의 유형입니다. AWS IoT SiteWise는 다음 유형의 위젯을 제공합니다.

- `sc-line-chart` - 선형 차트입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [선형 차트](#)를 참조하세요.
- `sc-scatter-chart` - 산점도 차트입니다. 자세한 정보는 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [산점도 차트](#)를 참조하세요.
- `sc-bar-chart` - 막대 차트입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [막대 차트](#)를 참조하세요.
- `sc-status-grid` - 자산 속성의 최신 값을 그리드로 표시하는 상태 위젯입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [상태 위젯](#)을 참조하세요.
- `sc-status-timeline` - 자산 속성의 과거 값을 타임라인으로 표시하는 상태 위젯입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [상태 위젯](#)을 참조하세요.
- `sc-kpi` - 핵심 성능 지표(KPI) 시각화입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [KPI 위젯](#)을 참조하세요.
- `sc-table` - 테이블 위젯입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [테이블 위젯](#)을 참조하세요.

title

위젯의 제목입니다.

x

그리드의 왼쪽을 기준으로 한 위젯의 수평 위치입니다. 이 값은 대시보드 그리드에서 위젯의 위치를 나타냅니다.

y

그리드의 위쪽을 기준으로 한 위젯의 수직 위치입니다. 이 값은 대시보드 그리드에서 위젯의 위치를 나타냅니다.

width

대시보드 그리드의 공백 수로 표시되는 위젯의 너비입니다.

height

대시보드 그리드의 공백 수로 표시되는 위젯의 높이입니다.

metrics

이 위젯에 대한 데이터 스트림을 각각 정의하는 지표 구조의 목록입니다. 각 목록 구조에는 다음 정보가 포함되어야 합니다.

label

이 지표에 대해 표시할 레이블입니다.

type

이 지표의 데이터 소스 유형입니다. AWS IoT SiteWise는 다음과 같은 지표 유형을 제공합니다.

- `iotsitewise` - 대시보드는 AWS IoT SiteWise의 자산 속성에 대한 데이터를 가져옵니다. 이 옵션을 선택하는 경우 이 지표에 대한 `assetId` 및 `propertyId`를 정의해야 합니다.

assetId

(선택 사항) AWS IoT SiteWise의 자산 ID입니다.

이 지표에서 `type`으로 `iotsitewise`를 선택한 경우 이 필드가 필요합니다.

propertyId

(선택 사항) AWS IoT SiteWise의 자산 속성 ID입니다.

이 지표에서 `type`으로 `iotsitewise`를 선택한 경우 이 필드가 필요합니다.

analysis

(선택 사항) 위젯에 표시할 분석(예: 추세선)을 정의하는 구조입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [추세선 구성](#)을 참조하세요. 위젯의 속성별로 각 유형의 추세선을 하나씩 추가할 수 있습니다. 분석 구조에는 다음 정보가 포함되어 있습니다.

trends

(선택 사항) 이 위젯의 추세 분석을 각각 정의하는 추세 구조 목록입니다. 목록의 각 구조에는 다음 정보가 포함되어 있습니다.

type

추세선 유형. 다음 옵션을 선택합니다.

- `linear-regression`— 선형 회귀선을 표시합니다. SiteWise Monitor는 [최소 제곱법](#)을 사용하여 선형 회귀를 계산합니다.

annotations

(선택 사항) 위젯의 임계값을 정의하는 주석 구조입니다. 자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 [임계값 구성](#)을 참조하세요. 위젯당 최대 6개의 주석을 추가할 수 있습니다. 주석 구조에는 다음 정보가 포함되어 있습니다.

y

(선택 사항) 이 위젯의 수평 임계값을 각각 정의하는 주석 구조 목록입니다. 목록의 각 구조에는 다음 정보가 포함되어 있습니다.

comparisonOperator

임계값에 대한 비교 연산자입니다. 다음 중 하나를 선택합니다.

- `LT` - value보다 작은 데이터 포인트가 하나 이상 있는 속성을 강조 표시합니다.
- `GT` - value보다 큰 데이터 포인트가 하나 이상 있는 속성을 강조 표시합니다.
- `LTE` - value보다 작거나 같은 데이터 포인트가 하나 이상 있는 속성을 강조 표시합니다.
- `GTE` - value보다 크거나 같은 데이터 포인트가 하나 이상 있는 속성을 강조 표시합니다.
- `EQ` - value와 같은 데이터 포인트가 하나 이상 있는 속성을 강조 표시합니다.

value

데이터 포인트를 `comparisonOperator`와 비교하기 위한 임계값입니다.

color

(선택 사항) 임계값 색상의 6자리 16진수 코드입니다. 시각화에서는 임계값 규칙을 충족하는 데이터 포인트가 하나 이상 있는 속성에 대한 속성 범례를 이 색상으로 표시합니다. 기본값은 검정색(`#000000`)입니다.

showValue

(선택 사항) 위젯 여백에 임계값을 표시할지 여부입니다. 기본값은 true입니다.

properties

(선택 사항) 위젯 속성의 기본 사전입니다. 이 구조의 멤버는 컨텍스트에 따라 다릅니다. AWS IoT SiteWise에서는 `properties`를 사용하는 다음 위젯을 제공합니다.

- [선형 차트](#), [산점도 차트](#) 및 [막대 차트](#)에는 다음과 같은 속성이 있습니다.

colorDataAcrossThresholds

(선택 사항) 이 위젯의 임계값을 초과하는 데이터의 색상을 변경할지 여부입니다. 이 옵션을 활성화하면 임계값을 초과하는 데이터가 선택한 색상으로 표시됩니다. 기본값은 true입니다.

- [상태 그리드](#)에는 다음과 같은 속성이 있습니다.

labels

(선택 사항) 상태 그리드에 표시할 레이블을 정의하는 구조입니다. 레이블 구조에는 다음 정보가 포함되어 있습니다.

showValue

(선택 사항) 이 위젯의 각 자산 속성에 대한 단위 및 값을 표시할지 여부입니다. 기본값은 true입니다.

Example 대시보드 정의 예

다음 예제에서는 JSON 파일에 저장된 페이로드로 대시보드를 정의합니다.

```
aws iotsitewise create-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \
  --dashboard-name "Wind Farm Dashboard" \
  --dashboard-definition file:///dashboard-definition.json
```

`dashboard-definition.json`에 대한 다음 JSON 예제는 다음과 같은 시각화 위젯으로 대시보드를 정의합니다.

- 대시보드의 왼쪽 상단에 풍력 발전 단지의 총 전력을 시각화하는 꺾은선형 차트입니다. 이 선형 차트에는 풍력 발전 단지가 최소 예상 출력보다 적은 전력을 출력하는 경우를 나타내는 임계값이 포함되어 있습니다. 이 선형 차트에는 선형 회귀 추세선도 포함되어 있습니다.

- 대시보드의 오른쪽 상단에 네 개의 터빈 풍속을 시각화하는 막대 차트입니다.

Note

이 예제에서는 대시보드의 선형 차트와 막대 차트 시각화를 나타냅니다. 이 대시보드는 [풍력 발전 단지 대시보드의 예](#)와 유사합니다.

```
{
  "widgets": [
    {
      "type": "sc-line-chart",
      "title": "Total Average Power",
      "x": 0,
      "y": 0,
      "height": 3,
      "width": 3,
      "metrics": [
        {
          "label": "Power",
          "type": "iotsitewise",
          "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
          "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
          "analysis": {
            "trends": [
              {
                "type": "linear-regression"
              }
            ]
          }
        }
      ],
      "annotations": {
        "y": [
          {
            "comparisonOperator": "LT",
            "value": 20000,
            "color": "#D13212",
            "showValue": true
          }
        ]
      }
    }
  ]
}
```

```

    }
  },
  {
    "type": "sc-bar-chart",
    "title": "Wind Speed",
    "x": 3,
    "y": 3,
    "height": 3,
    "width": 3,
    "metrics": [
      {
        "label": "Turbine 1",
        "type": "iotsitewise",
        "assetId": "a1b2c3d4-5678-90ab-cdef-2a2a2EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
      },
      {
        "label": "Turbine 2",
        "type": "iotsitewise",
        "assetId": "a1b2c3d4-5678-90ab-cdef-2b2b2EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
      },
      {
        "label": "Turbine 3",
        "type": "iotsitewise",
        "assetId": "a1b2c3d4-5678-90ab-cdef-2c2c2EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
      },
      {
        "label": "Turbine 4",
        "type": "iotsitewise",
        "assetId": "a1b2c3d4-5678-90ab-cdef-2d2d2EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
      }
    ]
  }
]
}

```

포털의 경보 활성화

포털에서 지원하는 알람 기능을 활성화하여 AWS IoT Events 포털 관리자가 모니터 포털에서 AWS IoT Events 알람 모델을 생성, 편집, 삭제할 수 있습니다 SiteWise . 프로젝트 소유자는 경보를 구성할

수 있습니다. 프로젝트 뷰어는 경고 세부 정보를 볼 수 있습니다. 이 섹션에서는 AWS IoT SiteWise 콘솔을 사용하여 포털의 알람 기능을 활성화하는 방법을 설명합니다.

Important

- 포털에서는 외부 경보를 생성할 수 없습니다.
- 경고 알림을 보내려면 사용자 인증 서비스로 IAM Identity Center를 선택해야 합니다.
- 중국 (베이징) 에서는 알람 알림 기능을 사용할 수 없습니다. AWS 리전

포털을 구성하고 생성할 때 2단계 추가 기능에서 경고 및 경고 알림을 활성화할 수 있습니다. 사용자 인증 서비스에 따라 다음 옵션 중 하나를 선택합니다.

IAM Identity Center

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2- optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - *optional*

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

Enable alarms
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template

Use an existing role

Enable alarm notifications
If enabled, alarms can send email or SMS notifications.

Sender
Specify the email address that sends alarm notifications. To edit or add a sender, go to the [Amazon SES console](#).

AWS Lambda role
Choose an IAM role that allows AWS Lambda to send data to Amazon SES and Amazon SNS. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template

Use an existing role

AWS Lambda function
Choose an AWS Lambda function to manage alarm notifications. To edit the function, go to the [AWS Lambda console](#).

Create a lambda from an AWS managed template

Use an existing lambda

[Previous](#) [Create](#)

포털의 경보를 활성화하려면

1. (선택 사항) 경보 활성화를 선택합니다.
 - AWS IoT SiteWise 액세스 역할의 경우 기존 역할을 사용하거나 필요한 권한이 있는 역할을 생성합니다. 이 역할에는 `iot.amazonaws.com` 및 `iotevents.amazonaws.com`가 역할을 맡을 수 있도록 허용하는 `iotevents:BatchPutMessage` 권한과 신뢰 관계가 필요합니다.
2. (선택 사항) 경보 알림 활성화를 선택합니다.
 - a. 발신자에서 발신자를 선택합니다.

⚠ Important

Amazon SES에서 발신자 이메일 주소를 확인해야 합니다. 자세한 내용은 Amazon Simple Email Service 개발자 가이드에서 [Amazon SES에서 이메일 주소 확인](#)을 참조하십시오.

- b. AWS Lambda 역할의 경우 기존 역할을 사용하거나 필요한 권한이 있는 역할을 생성합니다. 이 역할에는 `iotevents.amazonaws.com` 및 `lambda.amazonaws.com`가 역할을 맡을 수 있도록 허용하는 `lambda:InvokeFunction` 및 `sso-directory:DescribeUser` 권한과 신뢰 관계가 필요합니다.
- c. AWS Lambda 함수의 경우 기존 Lambda 함수를 선택하거나 경보 알림을 관리하는 함수를 생성합니다. 자세한 정보는 AWS IoT Events 개발자 안내서의 [경보 알림 관리](#)를 참조하십시오.

IAM

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2- optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - optional

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

Enable alarms
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

Create a role from an AWS managed template

Use an existing role

ⓘ Alarms created in the portal can't send notifications. If you want to send alarm notifications, choose **Previous**. Then, on the **Portal configuration** page, choose **AWS SSO for User authentication**.

Previous **Create**

포털의 경보를 활성화하려면

- (선택 사항) 경보 활성화를 선택합니다.

- AWS IoT SiteWise 액세스 역할의 경우 기존 역할을 사용하거나 필요한 권한이 있는 역할을 생성합니다. 이 역할에는 `iot.amazonaws.com` 및 `iotevents.amazonaws.com`가 역할을 맡을 수 있도록 허용하는 `iotevents:BatchPutMessage` 권한과 신뢰 관계가 필요합니다.

SiteWise Monitor의 경보에 대한 자세한 내용은 AWS IoT SiteWise 애플리케이션 가이드의 [경보를 사용한 모니터링](#)을 참조하십시오.

엣지에서 포털 활성화

엣지에서 포털을 활성화하면 계정에 데이터 처리 팩이 활성화되어 있는 모든 SiteWise 엣지 게이트웨이에서 이 포털을 사용할 수 있습니다.

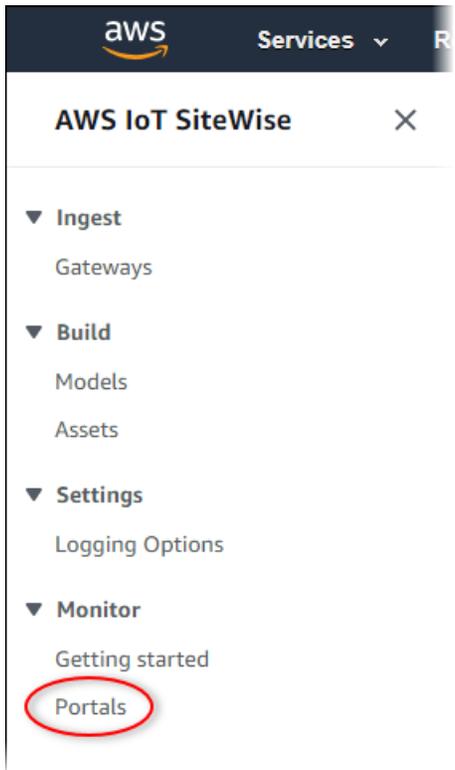
엣지에서 포털을 활성화하려면

1. 엣지 구성 섹션에서 엣지에서 이 포털 활성화를 켜세요.
2. 생성을 선택하세요.

SiteWise 모니터 포털 관리

포털 세부 정보를 업데이트하거나, 관리자를 변경하거나, 포털에 사용자를 추가해야 할 수 있습니다. 이 섹션에서는 SiteWise Monitor 포털에 대한 이러한 기본 관리 작업을 완료하는 방법을 설명합니다.

1. [AWS IoT SiteWise 콘솔](#)에 로그인합니다.
2. 탐색 창에서 모니터, 포털을 선택합니다.



3. 포털을 선택한 다음 세부 정보 보기(또는 포털 이름)를 선택합니다.
4. 다음 관리 작업을 수행할 수 있습니다.
 - [포털의 이름, 설명, 브랜딩, 지원 이메일 및 권한 변경](#)
 - [포털 관리자 추가 또는 제거](#)
 - [포털 관리자에게 이메일 초대 보내기](#)
 - [포털 사용자 추가 또는 제거](#)
 - [포털 삭제](#)

포털 생성 방법에 대한 자세한 내용은 [시작하기 AWS IoT SiteWise Monitor](#) 단원을 참조하십시오.

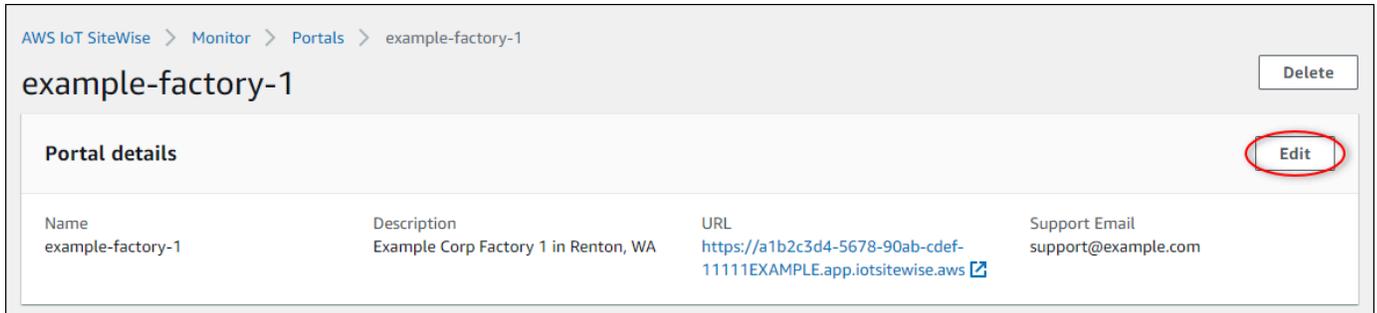
주제

- [포털의 이름, 설명, 브랜딩, 지원 이메일 및 권한 변경](#)
- [포털 관리자 추가 또는 제거](#)
- [포털 관리자에게 이메일 초대 보내기](#)
- [포털 사용자 추가 또는 제거](#)
- [포털 삭제](#)

포털의 이름, 설명, 브랜딩, 지원 이메일 및 권한 변경

포털의 이름, 설명, 브랜딩, 지원 이메일 및 권한을 변경할 수 있습니다.

1. 포털 세부 정보 페이지의 포털 세부 정보 섹션에서 편집을 선택합니다.

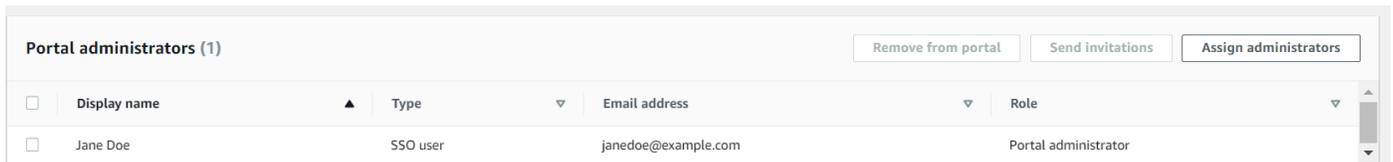


2. 이름, 설명, 포털 브랜딩, 지원 연락처 이메일 또는 권한을 업데이트합니다.
3. 작업을 마쳤으면 저장을 선택합니다.

포털 관리자 추가 또는 제거

몇 단계를 거쳐 포털의 관리자 역할을 할 사용자를 추가하거나 제거할 수 있습니다. 사용자 인증 서비스에 따라 다음 옵션 중 하나를 선택합니다.

IAM Identity Center



포털 관리자를 추가하려면

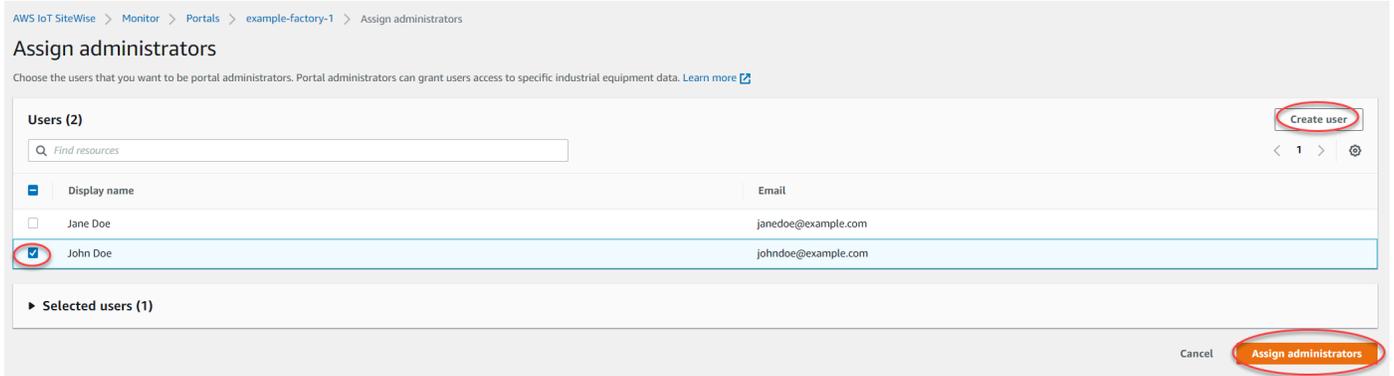
1. 포털 세부 정보 페이지의 포털 관리자 섹션에서 관리자 할당을 선택합니다.
2. 관리자 할당 페이지에서 포털에 관리자로 추가할 사용자의 확인란을 선택합니다.

Note

IAM Identity Center를 자격 증명 스토어로 사용하고 AWS Organizations 관리 계정에 로그인한 경우 사용자 생성을 선택하여 IAM Identity Center 사용자를 생성할 수 있습니다. IAM Identity Center는 새 사용자에게 비밀번호를 설정하라는 이메일을 보냅니다.

니다. 그런 다음 사용자를 포털에 관리자로 할당할 수 있습니다. 자세한 내용은 [IAM Identity Center에서 자격 증명 관리](#) 섹션을 참조하세요.

3. 관리자 할당을 선택합니다.



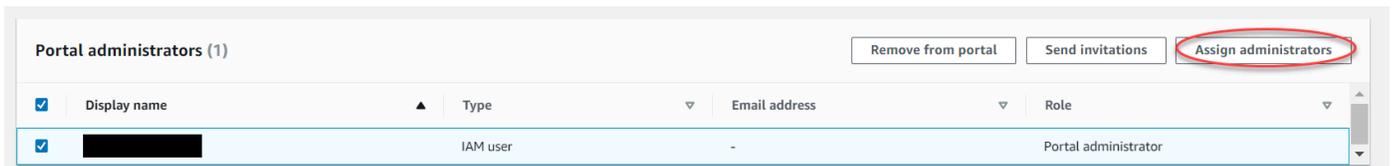
포털 관리자를 제거하려면

- 포털 세부 정보 페이지의 포털 관리자 섹션에서 제거할 각 사용자의 확인란을 선택한 다음 포털에서 제거를 선택합니다.

Note

포털 관리자를 한 명 이상 선택하는 것이 좋습니다.

IAM



포털 관리자를 추가하려면

1. 포털 세부 정보 페이지의 포털 관리자 섹션에서 관리자 할당을 선택합니다.
2. 관리자 할당 페이지에서 다음을 수행하세요.
 - IAM 사용자를 포털 관리자로 추가하려면 IAM 사용자를 선택합니다.
 - IAM 역할을 포털 관리자로 추가하려면 IAM 역할을 선택합니다.

3. 포털 관리자로 초대할 사용자 또는 역할의 확인란을 선택합니다. 그러면 포털 관리자 목록에 사용자 또는 역할이 추가됩니다.
4. 관리자 할당을 선택합니다.

Important

사용자 또는 역할에는 포털에 로그인할 수 있는 `iotsitewise:DescribePortal` 권한이 있어야 합니다.

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

 IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

Users Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	raspberrypi-testing	11-08-2019

► Portal administrators (1) [Remove](#)

Cancel [Assign administrators](#)

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

 IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

Users **Roles**

IAM roles (66) [Manage roles in IAM console](#)

Find role name

<input type="checkbox"/>	Name	Date created
<input type="checkbox"/>	[REDACTED]	
<input checked="" type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_EcKT-2Oar	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_GTnd004Wr	03-16-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_rHINLNCs-	03-11-2021
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	
<input type="checkbox"/>	[REDACTED]	

► Portal administrators (2) [Remove](#)

Cancel [Assign administrators](#)

포털 관리자를 제거하려면

- 포털 세부 정보 페이지의 포털 관리자 섹션에서 제거할 각 사용자의 확인란을 선택한 다음 포털에서 제거를 선택합니다.

Note

포털 관리자 없이 포털을 종료하는 것은 권장하지 않습니다.

포털 관리자에게 이메일 초대 보내기

포털 관리자에게 이메일 초대를 보낼 수 있습니다.

- 포털 세부 정보 페이지의 포털 관리자 섹션에서 포털 관리자의 확인란을 선택합니다.

Portal administrators (1)			Remove from portal	Send invitations	Assign users
<input checked="" type="checkbox"/>	Display name	Email address	Role		
<input checked="" type="checkbox"/>	John Doe	john.doe@example.com	Portal administrator		

- 초대 보내기를 선택합니다. 이메일 클라이언트가 열리고 메시지 본문에 초대장이 채워집니다.

이메일을 포털 관리자에게 보내기 전에 사용자 지정할 수 있습니다.

포털 사용자 추가 또는 제거

포털에 액세스할 수 있는 사용자를 통제합니다. 포털 사용자는 SiteWise 모니터 포털 내 사용자 목록에 표시됩니다. 이 목록에서 포털 관리자는 프로젝트 소유자를 추가할 수 있으며 프로젝트 소유자는 프로젝트 뷰어를 추가할 수 있습니다.

Note

포털 관리자와 포털 사용자는 사용자를 추가하거나 제거해야 하는 경우 포털의 지원 이메일을 통해 사용자에게 연락할 수 있습니다.

사용자 인증 서비스에 따라 다음 옵션 중 하나를 선택합니다.

IAM Identity Center

Portal users (1)					Remove from portal	Assign users
<input type="checkbox"/>	Display name	Type	Email address	Role		
<input type="checkbox"/>	John Doe	SSO user	johndoe@example.com	Portal viewer		

포털 사용자를 추가하려면

1. 포털 세부 정보 페이지의 포털 사용자 섹션에서 사용자 할당을 선택합니다.
2. 사용자 할당 페이지에서 포털에 추가할 사용자의 확인란을 선택합니다.

 Note

IAM Identity Center를 자격 증명 스토어로 사용하고 AWS Organizations 관리 계정에 로그인한 경우 사용자 생성을 선택하여 IAM Identity Center 사용자를 생성할 수 있습니다. IAM Identity Center는 새 사용자에게 비밀번호를 설정하라는 이메일을 보냅니다. 그런 다음 사용자를 포털에 사용자로 할당할 수 있습니다. 자세한 내용은 [IAM Identity Center에서 자격 증명 관리](#) 섹션을 참조하세요.

3. 사용자 할당을 선택합니다.

AWS IoT SiteWise > Monitor > Portals > example-factory-1 > Assign users

Assign users

Users (2) Create user

Find resources

<input type="checkbox"/>	Display name	Email
<input checked="" type="checkbox"/>	John Doe	johndoe@example.com
<input type="checkbox"/>	Jane Doe	janedoe@example.com

Selected users (1)

Cancel Assign users

포털 사용자를 제거하려면

- 포털 세부 정보 페이지의 포털 사용자 섹션에서 포털에서 제거할 사용자의 확인란을 선택한 다음 포털에서 제거를 선택합니다.

IAM

Portal users (1)					Remove from portal	Assign users
<input type="checkbox"/>	Display name	Type	Email address	Role		
<input type="checkbox"/>	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	IAM role	-	Portal viewer		

포털 사용자를 추가하려면

1. 포털 세부 정보 페이지의 포털 사용자 섹션에서 사용자 할당을 선택합니다.
2. 사용자 할당 페이지에서 다음을 수행합니다.
 - IAM 사용자를 선택하여 IAM 사용자를 포털 사용자로 추가합니다.
 - IAM 역할을 선택하여 IAM 역할을 포털 사용자로 추가합니다.
3. 포털 사용자로 추가하려는 사용자 또는 역할의 확인란을 선택합니다. 그러면 포털 사용자 목록에 사용자 또는 역할이 추가됩니다.
4. 사용자 할당을 선택합니다.

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users Roles

IAM users (1) [Manage users in IAM console](#)

<input checked="" type="checkbox"/>	Name	Date created
<input checked="" type="checkbox"/>	[REDACTED]	11-08-2019

▶ Portal users (1)

Cancel

The screenshot shows the 'Assign users' page in the AWS IoT SiteWise console. The breadcrumb trail is 'AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users'. The 'Roles' tab is active. Under 'IAM roles (66)', a search bar is present. A table lists roles with columns for 'Name' and 'Date created'. The role 'AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1' is selected. Below the table, there is a section for 'Portal users (2)' with a 'Remove' button. At the bottom right, there are 'Cancel' and 'Assign users' buttons.

포털 사용자를 제거하려면

- 포털 세부 정보 페이지의 포털 사용자 섹션에서 포털에서 제거할 사용자의 확인란을 선택한 다음 포털에서 제거를 선택합니다.

⚠ Important

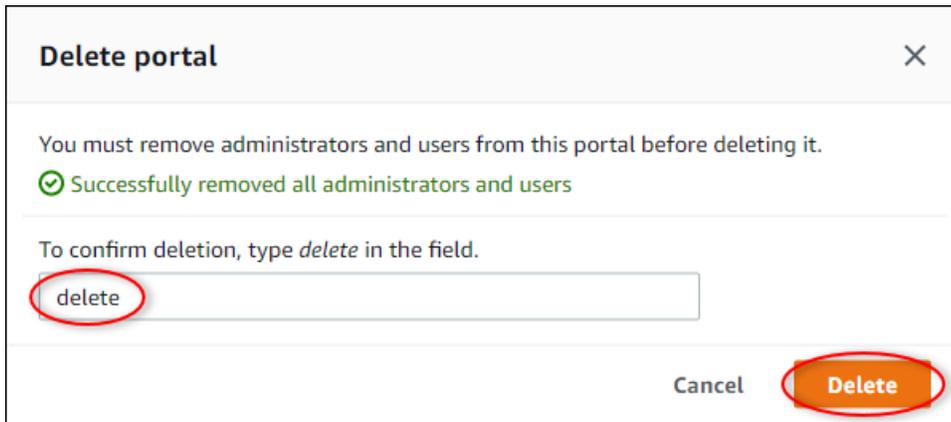
사용자 또는 역할에는 포털에 로그인할 수 있는 `iotsitewise:DescribePortal` 권한이 있어야 합니다.

포털 삭제

테스트 목적으로 포털을 생성했거나 이미 존재하는 포털의 복제본을 생성한 경우 포털을 삭제할 수 있습니다.

ℹ Note

포털을 삭제하려면 먼저 포털의 모든 대시보드와 프로젝트를 수동으로 삭제해야 합니다. 자세한 내용은 SiteWise Monitor 애플리케이션 가이드의 [프로젝트 삭제 및 대시보드 삭제](#)를 참조하십시오.



4. 삭제를 선택합니다.

IoT 대시보드 애플리케이션을 통한 데이터 모니터링

IoT 대시보드 애플리케이션은 운영 데이터를 시각화하고 상호 작용할 수 있는 오픈 소스 대시보드 애플리케이션입니다. 를 활용하여 IoT 대시보드 애플리케이션을 AWS Cloud Development Kit (AWS CDK) 배포할 수 있습니다.

다음은 IoT 대시보드 애플리케이션의 사용자 지정 가능한 데이터 시각화 기능의 예입니다.

- 단일 라인 차트에서 여러 속성을 지원합니다.
- 자산 및 속성에 대한 향상된 검색.

제조, 물류, 에너지 및 기타 산업의 고객은 IoT 대시보드 애플리케이션을 사용하여 장비 성능 추적, 운영 효율성 최적화, 데이터 기반 의사 결정과 같은 특정 문제를 해결할 수 있습니다. 자세한 내용은 [IoT 대시보드 애플리케이션용 GitHub 리포지토리를](#) 참조하십시오.

에서 데이터 쿼리 AWS IoT SiteWise

AWS IoT SiteWise API 작업을 사용하여 특정 기간 동안 자산 속성의 현재 값, 과거 값 및 집계를 쿼리할 수 있습니다.

이러한 기능을 사용하여 데이터에 대한 통찰력을 얻으십시오. 예를 들어, 특정 자산 가치를 지닌 모든 자산을 검색하거나 데이터를 맞춤 표현해 보세요. 또한 API 작업을 사용하여 AWS IoT SiteWise 자산에 저장된 산업 데이터와 통합되는 소프트웨어 솔루션을 개발할 수 있습니다. AWS IoT SiteWise Monitor에서 라이브 자산 데이터를 탐색할 수도 있습니다. SiteWise Monitor를 구성하는 방법을 알아보려면 [을 참조하십시오](#) [을 사용하여 데이터 모니터링 AWS IoT SiteWise Monitor](#).

이 섹션에서 설명하는 작업은 타임스탬프, 품질, 값 (TQV) 구조를 포함하는 속성값 객체를 반환합니다.

- timestamp는 현재 Unix epoch 시간(초)과 오프셋(나노초)을 포함합니다.
- quality는 데이터 요소의 품질을 나타내는 다음 문자열 중 하나를 포함합니다.
 - GOOD – 데이터가 어떤 문제의 영향도 받지 않습니다.
 - BAD – 데이터가 센서 고장과 같은 문제의 영향을 받습니다.
 - UNCERTAIN – 데이터가 센서 부정확과 같은 문제의 영향을 받습니다.
- value는 속성 유형에 따라 다음 필드 중 하나를 포함합니다.
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue

주제

- [현재 자산의 자산 가치 조회](#)
- [과거 자산 속성 값 쿼리](#)
- [자산 속성 집계 쿼리](#)
- [AWS IoT SiteWise 쿼리 언어](#)

현재 자산의 자산 가치 조회

이 자습서에서는 자산 속성의 현재 값을 가져오는 두 가지 방법을 보여줍니다. AWS IoT SiteWise 콘솔을 사용하거나 AWS Command Line Interface (AWS CLI) 에서 API를 사용할 수 있습니다.

주제

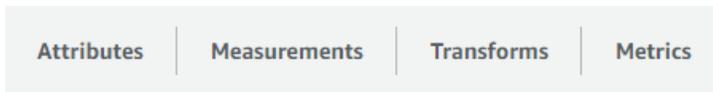
- [자산 속성의 현재 값 쿼리 \(콘솔\)](#)
- [자산 속성의 현재 값 쿼리 \(AWS CLI\)](#)

자산 속성의 현재 값 쿼리 (콘솔)

AWS IoT SiteWise 콘솔을 사용하여 자산 속성의 현재 가치를 볼 수 있습니다.

자산 속성의 현재 값을 가져오려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택합니다.
3. 쿼리할 속성이 있는 자산을 선택합니다.
4. 화살표 아이콘을 선택하여 자산 계층을 확장하여 자산을 찾을 수 있습니다.
5. 속성 유형에 대한 탭을 선택합니다. 예를 들어 측정 속성의 현재 값을 보려면 측정을 선택합니다.



6. 보려는 속성을 찾습니다. 현재 값이 최신 값 옆에 나타납니다.

자산 속성의 현재 값 쿼리 (AWS CLI)

AWS Command Line Interface (AWS CLI) 를 사용하여 자산 속성의 현재 값을 쿼리할 수 있습니다.

[GetAssetPropertyValue](#)작업을 사용하면 자산 속성의 현재 값을 쿼리할 수 있습니다.

자산 속성을 식별하려면 다음 중 하나를 지정하십시오.

- 데이터가 전송되는 자산 `propertyId` 자산의 소유자. `assetId`
- 데이터 스트림 별칭(예: `/company/windfarm/3/turbine/7/temperature`)인 `propertyAlias`. 이 옵션을 사용하려면 먼저 자산 속성의 별칭을 설정해야 합니다. 속성 별칭을 설정하려면 을 참조하십시오 [산업 데이터 스트림을 자산 속성에 매핑](#).

자산 속성의 현재 값을 가져오려면 ()AWS CLI

- 다음 명령을 실행하여 자산 속성의 현재 값을 가져옵니다. `asset-id`를 자산의 ID로 바꾸고 `property-id`를 속성의 ID로 바꿉니다.

```
aws iotsitewise get-asset-property-value \
  --asset-id asset-id \
  --property-id property-id
```

이 작업은 다음 형식으로 속성의 현재 TQV를 포함하는 응답을 반환합니다.

```
{
  "propertyValue": {
    "value": {
      "booleanValue": Boolean,
      "doubleValue": Number,
      "integerValue": Number,
      "stringValue": "String"
    },
    "timestamp": {
      "timeInSeconds": Number,
      "offsetInNanos": Number
    },
    "quality": "String"
  }
}
```

과거 자산 속성 값 쿼리

AWS IoT SiteWise API [GetAssetPropertyValueHistory](#) 작업을 사용하여 자산 속성의 과거 값을 쿼리할 수 있습니다.

자산 자산을 식별하려면 다음 중 하나를 지정하십시오.

- 데이터가 전송되는 자산 `propertyId` 자산의 소유자. `assetId`
- 데이터 스트림 별칭(예: `/company/windfarm/3/turbine/7/temperature`)인 `propertyAlias`. 이 옵션을 사용하려면 먼저 자산 속성의 별칭을 설정해야 합니다. 속성 별칭을 설정하려면 [참조하십시오 산업 데이터 스트림을 자산 속성에 매핑](#).

다음 파라미터를 전달하여 결과를 구체화하세요.

- `startDate` – Unix Epoch 시간(초)으로 표시되며 과거 데이터를 쿼리하기 시작할 범위의 시작(제외)입니다.

- `endDate` – Unix Epoch 시간(초)으로 표시되며 과거 데이터를 쿼리하기 시작할 범위의 끝(포함)입니다.
- `maxResults` – 하나의 요청에서 반환할 최대 결과 수입니다. 기본값은 20개의 결과입니다.
- `nextToken` – 이 작업의 이전 호출에서 반환된 페이지 매김 토큰입니다.
- `timeOrdering` – 반환된 값에 적용할 정렬은 ASCENDING 또는 DESCENDING 입니다.
- `qualities` – 결과 필터링 기준이 되는 품질은 GOOD, BAD, 또는 UNCERTAIN입니다.

주제

- [자산 자산의 가치 기록 쿼리 \(\)AWS CLI](#)

자산 자산의 가치 기록 쿼리 ()AWS CLI

자산 속성의 가치 기록을 쿼리하려면 (AWS CLI)

1. 다음 명령을 실행하여 자산 속성에 대한 값 기록을 가져옵니다. 이 명령은 특정한 10분 간격 동안 속성의 기록을 쿼리합니다. `asset-id`를 자산의 ID로 바꾸고 `property-id`를 속성의 ID로 바꿉니다. 날짜 매개 변수를 쿼리할 간격으로 바꿉니다.

```
aws iotsitewise get-asset-property-value-history \
  --asset-id asset-id \
  --property-id property-id \
  --start-date 1575216000 \
  --end-date 1575216600
```

이 작업은 속성의 과거 TQV가 다음 형식으로 포함된 응답을 반환합니다.

```
{
  "assetPropertyValueHistory": [
    {
      "value": {
        "booleanValue": Boolean,
        "doubleValue": Number,
        "integerValue": Number,
        "stringValue": "String"
      },
      "timestamp": {
        "timeInSeconds": Number,
        "offsetInNanos": Number
      }
    }
  ]
}
```

```

    },
    "quality": "String"
  }
],
"nextToken": "String"
}

```

- 값 항목이 더 있는 경우 nextToken 필드의 페이지 매김 토큰을 후속 작업 호출에 전달할 수 있습니다. [GetAssetPropertyValueHistory](#)

자산 속성 집계 쿼리

AWS IoT SiteWise 여러 시간 간격으로 계산된 기본 지표 집합인 집계된 자산 속성 값을 자동으로 계산합니다. AWS IoT SiteWise 자산 속성에 대해 분, 시간, 일마다 다음과 같은 집계를 계산합니다.

- average(평균) – 시간 간격 동안 속성 값의 평균입니다.
- count(개수) – 일정 기간에 걸친 속성의 데이터 포인트 수입니다.
- maximum(최대) – 시간 간격 동안 속성 값의 최대값입니다.
- minimum(최소) – 시간 간격 동안 속성 값의 최소값입니다.
- standard deviation(표준 편차) – 시간 간격 동안 속성 값의 표준 편차입니다.
- sum(합계) – 시간 간격 동안 속성 값의 합계입니다.

문자열이나 불리언과 같은 숫자가 아닌 속성의 경우 집계된 개수만 계산합니다. AWS IoT SiteWise

자산 데이터에 대한 사용자 지정 지표를 계산할 수도 있습니다. 지표 속성을 사용하면 작업별 집계를 정의할 수 있습니다. 지표 속성은 API에 대해 미리 계산되지 않은 추가 집계 함수와 시간 간격을 제공합니다. AWS IoT SiteWise 자세한 정보는 [속성과 기타 자산의 데이터 집계\(지표\)](#)을 참조하세요.

주제

- [자산 속성 \(API\) 에 대한 집계](#)
- [자산 속성에 대한 집계 \(\)AWS CLI](#)

자산 속성 (API) 에 대한 집계

AWS IoT SiteWise API를 사용하여 자산 속성에 대한 집계를 가져올 수 있습니다.

[GetAssetPropertyAggregates](#) 작업을 사용하여 자산 속성의 집계를 쿼리할 수 있습니다.

자산 속성을 식별하려면 다음 중 하나를 지정하십시오.

- 데이터가 전송되는 자산 `propertyId` 자산의 소유자. `assetId`
- 데이터 스트림 별칭(예: `/company/windfarm/3/turbine/7/temperature`)인 `propertyAlias`. 이 옵션을 사용하려면 먼저 자산 속성의 별칭을 설정해야 합니다. 속성 별칭을 설정하려면 [참조하십시오 산업 데이터 스트림을 자산 속성에 매핑](#).

또한 다음과 같은 필수 파라미터를 전달해야 합니다.

- `aggregateTypes` – 검색할 집계 목록입니다. AVERAGE, COUNT, MAXIMUM, MINIMUM, STANDARD_DEVIATION, SUM 중 하나를 지정할 수 있습니다.
- `resolution` – 1m (1분), 1h (1시간) 또는 1d (1일) 등 지표를 검색할 시간 간격입니다.
- `startDate` – Unix Epoch 시간(초)으로 표시되며 과거 데이터를 쿼리하기 시작할 범위의 시작(제외)입니다.
- `endDate` – Unix Epoch 시간(초)으로 표시되며 과거 데이터를 쿼리하기 시작할 범위의 끝(포함)입니다.

다음 파라미터 중 하나를 전달하여 결과를 구체화할 수도 있습니다.

- `maxResults` – 하나의 요청에서 반환할 최대 결과 수입니다. 기본값은 20개의 결과입니다.
- `nextToken` – 이 작업의 이전 호출에서 반환된 페이지 매김 토큰입니다.
- `timeOrdering` – 반환된 값에 적용할 정렬은 ASCENDING 또는 DESCENDING 입니다.
- `qualities` – 결과 필터링 기준이 되는 품질은 GOOD, BAD, 또는 UNCERTAIN입니다.

Note

이 [GetAssetPropertyAggregates](#) 작업은 이 섹션에 설명된 다른 작업과 다른 형식의 TQV를 반환합니다. `value` 구조에는 요청의 각 `aggregateTypes`에 대한 필드가 포함되어 있습니다. `timestamp`에는 집계 발생 Unix 에포크 시간(초)이 포함됩니다.

자산 속성에 대한 집계 ()AWS CLI

자산 속성에 대한 집계를 쿼리하려면 ()AWS CLI

1. 다음 명령을 실행하여 자산 속성에 대한 집계를 가져옵니다. 이 명령은 특정한 1시간 간격에 대해 1시간 해상도로 평균과 합계를 쿼리합니다. *asset-id*를 자산의 ID로 바꾸고 *property-id*를 속성의 ID로 바꿉니다. 파라미터를 쿼리할 집계 및 간격으로 바꿉니다.

```
aws iotsitewise get-asset-property-aggregates \
  --asset-id asset-id \
  --property-id property-id \
  --start-date 1575216000 \
  --end-date 1575219600 \
  --aggregate-types AVERAGE SUM \
  --resolution 1h
```

이 작업은 다음 형식으로 속성의 과거 TQV를 포함하는 응답을 반환합니다. 응답에는 요청된 집계만 포함됩니다.

```
{
  "aggregatedValues": [
    {
      "timestamp": Number,
      "quality": "String",
      "value": {
        "average": Number,
        "count": Number,
        "maximum": Number,
        "minimum": Number,
        "standardDeviation": Number,
        "sum": Number
      }
    }
  ],
  "nextToken": "String"
}
```

2. 값 항목이 더 있는 경우 nextToken 필드의 페이지 매김 토큰을 후속 작업 호출에 전달할 수 있습니다. [GetAssetPropertyAggregates](#)

AWS IoT SiteWise 쿼리 언어

AWS IoT SiteWise 데이터 검색 [ExecuteQuery](#) API 작업을 사용하면 다음에서 선언적 구조 정의 및 이와 관련된 시계열 데이터에 대한 정보를 검색할 수 있습니다.

- 모델
- 자산
- 측정
- 지표
- 변형
- 골재

이는 쿼리 명령문과 같은 SQL을 사용하여 단일 API 요청으로 수행할 수 있습니다.

Note

이 기능은 AWS GovCloud (미국 서부) 를 제외하고, AWS IoT SiteWise 와 를 AWS IoT TwinMaker 모두 사용할 수 있는 모든 지역에서 사용할 수 있습니다.

주제

- [필수 조건](#)
- [쿼리 언어 참조](#)

필수 조건

AWS IoT SiteWise 산업 데이터를 구성하고 모델링할 수 AWS IoT TwinMaker 있으려면 통합 권한이 필요합니다.

모델, 자산, 측정, 지표, 변환 및 집계에 대한 정보를 검색하려면 먼저 다음 사전 요구 사항을 충족해야 합니다.

- 두 가지 모두에 대한 서비스 연계 역할 및 사용자 설정 AWS IoT SiteWise AWS IoT TwinMaker AWS 계정서비스 연결 역할에 대한 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 사용](#)을 참조하세요

- IAM 역할을 AWS IoT SiteWise 통합할 수 있게 되었습니다. 자세한 정보는 [AWS IoT SiteWise 및 AWS IoT TwinMaker 통합](#)을 참조하세요.
- 해당 IoTSiteWiseDefaultWorkspace 지역의 계정에 ID가 있는 AWS IoT TwinMaker 워크스페이스. 자세한 내용은 [사용 IoTSiteWiseDefaultWorkspaceAWS IoT TwinMaker 설명서의 사용](#)을 참조하십시오.
- 표준 또는 계층형 번들 가격 모드를 AWS IoT TwinMaker 활성화할 수 있습니다. 자세한 내용은 AWS IoT TwinMaker 사용 설명서의 [전환 AWS IoT TwinMaker 가격 책정 모드](#)를 참조하십시오.

쿼리 언어 참조

AWS IoT SiteWise 데이터 작업을 위한 풍부한 쿼리 언어를 지원합니다. 사용 가능한 데이터 유형, 연산자, 함수 및 구문은 다음 항목에 설명되어 있습니다.

AWS IoT SiteWise 쿼리 언어로 쿼리를 [쿼리 예제](#) 작성하려면 을 참조하십시오.

주제

- [뷰에 대한 이해](#)
- [지원되는 데이터 유형](#)
- [SELECT 문을 사용하여 데이터를 검색합니다.](#)
- [논리 연산자](#)
- [비교 연산자](#)
- [쿼리 예제](#)

뷰에 대한 이해

이 섹션에서는 프로세스 메타데이터 및 원격 분석 데이터와 같은 뷰를 이해하는 데 도움이 되는 정보를 제공합니다. AWS IoT SiteWise

다음 표에는 뷰 이름과 뷰 설명이 나와 있습니다.

데이터 모델

뷰 이름	보기 설명
asset	자산 및 모델 파생에 대한 정보가 들어 있습니다.

뷰 이름	보기 설명
자산_프로퍼티	에셋 프로퍼티의 구조에 대한 정보가 들어 있습니다.
원시_시리즈	시계열의 과거 데이터를 포함합니다.
최신_가치_시간_시리즈	시계열의 최신 값을 포함합니다.
미리 계산된 집계	자동으로 계산된 집계된 자산 속성 값을 포함합니다. 이는 여러 시간 간격으로 계산되는 기본 지표 집합입니다.

다음 뷰는 샘플 데이터와 함께 쿼리의 열 이름을 나열합니다.

뷰:에셋

에셋_ID	에셋_이름	에셋_설명	자산_모델_ID
88898498-0b8b-42b5-bf57-16180bc3d3a0	WindTurbine A	WindTurbine 에셋 A	17847250-5bf0-4f74-b775-cc03f05e7cb8
17847250-5bf0-4f74-b775-cc03f05e7cb8	풍력 터빈 자산 모델	풍력 발전 단지의 터빈을 나타냅니다.	

보기: 자산_재산

프로퍼티_ID	자산_ID	속성_이름	속성_데이터_유형	속성_별칭	자산_복합_모델_ID
b29be434-b000-4d74-b809-75287d83bcd6	88898498-0b8b-42b5-bf57-16180bc3d3a0	모터 온도	Double	Rochester 2/44///Line-5/Bus-2/Machine-5/Temperature	

프로퍼티_ID	자산_ID	속성_이름	속성_데이터_유형	속성_별칭	자산_복합_모델_ID
3b458f00-24e7-458a-b4e8-c6026eff654a	88898498-0b8b-42b5-bf57-16180bc3d3a0	바람의 방향	Double	/company/windfarm/3/turbine/7/winddirection	2f458n00-56e7-458h-b4e8-c6026eff985g

보기: 원시_타임_시리즈

에셋_ID	프로퍼티_ID	프로퍼티_앨리어스	이벤트_타임스탬프	품질	블리언_값	int_value	더블 밸류	문자열_값
88898498-0b8b-42b5-bf57-16180bc3d3a	b29be434-b000-4d7-b809-75287d83bcd	Rochester-5/Bus-2/Machine-5/Temperature	157521960	좋음			115.0	
88898498-0b8b-42b5-bf57-16180bc3d3a	3b458f00-24e7-458a-b4e8-c6026eff654a	/company/windfarm/3/turbine/7/winddirection	157521937	좋음			348.75	

Note

뷰를 쿼리하려면 event_timestamp 열에 필터 절을 포함해야 합니다. raw_time_series 이 필터는 필수 필터이며, 필터가 없으면 쿼리가 실패합니다.

Example 쿼리

```
SELECT event_timestamp, double_value FROM raw_time_series WHERE event_timestamp > 1234567890
```

보기: 최신_값_시간_시리즈

자산_ID	프로퍼티_ID	프로퍼티_앨리어스	이벤트_타임스탬프	품질	불리언_값	int_value	더블 밸류	문자열_값
888984980b8b-42b1-bf57-16180bc3d3a	3b458f00-24e7-458c-b4e8-c6026eff654a	/company, windfarr turbine /7/ winddirection	15752196	좋음			355.39	

보기: 미리 계산된 집계

에셋_id	프로퍼티_ID	프로퍼티_앨리어스	이벤트_타임스탬프	resolution	합계_값	개수_값	평균_값	최대_값	최소_값	표준_개발_값
888984980b8b-42b1-bf57-16180bc3d3a	b29be4b000-4c2-b809-75	Roches 2/44// 0 Li ne-5/	15752196	15m	1105.48	15	73.4	80.6	68	3.64

에셋_id	프로퍼티_ID	프로퍼티_앨리어스	이벤트_타임프	resolution	합계_값	개수_값	평균_값	최대_값	최소_값	표준_개발_값
80bc3d1	87d83b1	Bus-2/Machine-5/Temperature								

지원되는 데이터 유형

AWS IoT SiteWise 쿼리 언어는 다음 데이터 유형을 지원합니다.

뷰:에셋

데이터 유형	설명
STRING	최대 길이 1024바이트의 문자열입니다.
INTEGER	범위가 1인 부호있는 32비트 정수. -2,147,483,648 to 2,147,483,647
DOUBLE	범위가 0이고 배정밀도가 두 배인 부동 소수점 숫자입니다-10^100 to 10^100. IEEE 754
BOOLEAN	true 또는 false

Note

배정밀도 데이터는 정확하지 않습니다. 일부 값은 정확히 변환되지 않으며 제한된 정밀도로 인해 모든 실수를 나타내지는 않습니다. 쿼리의 부동 소수점 데이터는 내부적으로 표시되는 값과 같지 않을 수 있습니다. 입력 숫자의 정밀도가 너무 높으면 값이 반올림됩니다.

SELECT 문을 사용하여 데이터를 검색합니다.

SELECT 명령문은 하나 이상의 뷰에서 데이터를 검색하는 데 사용됩니다. AWS IoT SiteWise JOIN 암시적인 뷰를 지원합니다. 조인할 뷰를 쉼표로 구분하여 (SELECT 명령문 FROM 조항에) 나열할 수 있습니다.

Example

다음 SELECT 명령문을 사용하세요.

```
SELECT select_expr [, ...]
[ FROM from_item [, ...] ]
[ WHERE [LIKE condition ESCAPE condition] ]
```

이전 예제에서 LIKE 조항은 와일드카드를 사용하여 검색 및 필터링 조건을 지정합니다. AWS IoT SiteWise 는 와일드카드 percentage (%) 문자로 지원합니다.

Example %조건에서 사용하려면:

```
Prefix search: String%
Infix search: %String%
Suffix search: %String
```

Example 자산을 검색하려면:

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'Wind%'
```

Example ESCAPE 조건을 사용하여 자산을 검색하려면:

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'room\%' ESCAPE
'\'
```

논리 연산자

AWS IoT SiteWise 다음과 같은 논리 연산자를 지원합니다.

논리 연산자

연산자	설명	예
AND	TRUE 두 값이 모두 참인 경우	a AND b

a 또는 b가 FALSE 인 경우 이전 표현식은 false로 평가됩니다. AND연산자가 true로 평가되려면 a와 b가 모두 참이어야 합니다.

Example

```
SELECT a.asset_name
FROM asset as a, latest_value_time_series as t
WHERE t.int_value > 30 AND t.event_timestamp > 1234567890
```

비교 연산자

AWS IoT SiteWise 다음과 같은 비교 연산자를 지원합니다.

논리 연산자

운영자	설명
<	보다 작음
>	보다 큼
<=	작거나 같음
>=	크거나 같음
=	같음
!=	같지 않음

쿼리 예제

메타데이터 필터링

다음은 AWS IoT SiteWise 쿼리 언어를 사용하는 SELECT 명령문을 사용한 메타데이터 필터링의 예입니다.

```
SELECT a.asset_name, p.property_name
FROM asset a, asset_property p
WHERE a.asset_id = p.asset_id AND a.asset_name LIKE '%windmill%'
```

값 필터링

다음은 AWS IoT SiteWise 쿼리 언어가 포함된 SELECT 명령문을 사용한 값 필터링의 예입니다.

```
SELECT a.asset_name FROM asset a, raw_time_series r
WHERE a.asset_id = r.asset_id AND r.int_value > 30 AND r.event_timestamp > 1234567890
AND r.event_timestamp < 1234567891
```

다른 AWS 서비스와의 상호작용

AWS IoT SiteWise 자산 데이터를 AWS IoT MQTT 게시-구독 메시지 브로커에 게시할 수 있으므로 다른 서비스의 자산 데이터와 상호 작용할 수 있습니다. AWS IoT SiteWise 핵심 규칙을 사용하여 자산 데이터를 다른 AWS 서비스로 라우팅하는 데 사용할 수 있는 고유한 MQTT 주제를 각 자산 속성에 할당합니다. AWS IoT 예를 들어 다음 작업을 수행하도록 AWS IoT 핵심 규칙을 구성할 수 있습니다.

- 장비 결함을 식별하고 [AWS IoT Events](#)에 데이터를 전송하여 담당자에게 알립니다.
- 데이터를 [Amazon DynamoDB](#)로 전송하여 외부 소프트웨어 솔루션에서 사용할 특정 자산 데이터를 기록합니다.
- [AWS Lambda](#) 함수를 트리거하여 주간 보고서를 생성합니다.

DynamoDB에 속성 값을 저장하는 규칙의 설정 단계를 안내하는 자습서를 따를 수 있습니다. 자세한 내용은 [Amazon DynamoDB에 속성 값 업데이트 게시](#) 섹션을 참조하십시오.

규칙을 구성하는 방법에 대한 자세한 내용은 AWS IoT 개발자 가이드의 [규칙](#)을 참조하십시오.

다른 AWS 서비스의 데이터를 다시 사용할 수도 AWS IoT SiteWise 있습니다. AWS IoT SiteWise 규칙 작업을 통해 데이터를 수집하려면 [규칙을 사용한 AWS IoT Core 데이터 수집](#)을 참조하십시오.

주제

- [자산 속성의 MQTT 주제 이해](#)
- [자산 속성 알림 사용하기](#)
- [자산 속성 알림과 함께 Amazon S3로 데이터 내보내기](#)
- [Grafana와 통합](#)
- [AWS IoT SiteWise 및 AWS IoT TwinMaker 통합](#)
- [Amazon Lookout for Equipment를 사용하여 장비 이상을 감지합니다.](#)

자산 속성의 MQTT 주제 이해

모든 자산 속성에는 다음과 같은 형식의 고유한 MQTT 주제 경로가 있습니다.

```
$aws/sitewise/asset-models/assetModelId/assets/assetId/properties/propertyId
```

Note

AWS IoT SiteWise 핵심 규칙 엔진의 # (다단계) 주제 필터 와일드카드를 지원하지 않습니다. AWS IoT +(단일 레벨) 와일드카드를 사용할 수 있습니다. 예를 들어 다음 주제 필터를 사용하여 특정 자산 모델의 모든 업데이트를 일치시킬 수 있습니다.

```
$aws/sitewise/asset-models/assetModelId/assets/+/properties/+
```

주제 필터 와일드카드에 대한 자세한 내용은 AWS IoT Core 개발자 가이드의 [주제](#)를 참조하십시오.

자산 속성 알림 사용하기

속성 알림을 활성화하여 자산 데이터 업데이트를 게시한 다음 데이터에 대해 쿼리를 실행할 수 있습니다. AWS IoT Core 자산 속성 알림과 함께 Amazon S3로 AWS IoT SiteWise 데이터를 내보내는 데 사용할 수 있는 AWS CloudFormation 템플릿을 AWS IoT SiteWise 제공합니다.

Note

자산 데이터는 값이 변경되었는지 여부와 상관없이에서 AWS IoT SiteWise 수신할 AWS IoT Core 때마다 전송됩니다.

주제

- [자산 속성 알림 활성화\(콘솔\)](#)
- [자산 자산 알림 활성화 \(AWS CLI\)](#)
- [자산 속성 알림 메시지 쿼리](#)

자산 속성 알림 활성화(콘솔)

기본적으로 속성 값 업데이트를 게시하지 AWS IoT SiteWise 않습니다. AWS IoT SiteWise 콘솔을 사용하여 자산 속성에 대한 알림을 활성화할 수 있습니다.

자산 속성에 대한 알림을 활성화하거나 비활성화하려면(콘솔)

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.

2. 탐색 창에서 자산을 선택합니다.
3. 속성의 알림을 활성화할 자산을 선택합니다.

i Tip

화살표 아이콘을 선택하여 자산 계층 구조를 확장하고 자산을 찾을 수 있습니다.

4. 편집을 선택합니다.
5. 자산 속성의 알림 상태에서 ENABLED(활성)를 선택합니다.

The screenshot shows a form for editing an asset property. On the left, there is a text input field labeled '"Wind Speed"' with a placeholder 'Enter a property alias' and a note 'Must be less than 2048 characters.' On the right, there is a dropdown menu titled 'Notification status' which is currently set to 'ENABLED'. Below the dropdown, there is a text description: 'Notification will be published to topic \$aws/sitesite/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'. A red oval highlights the 'ENABLED' option in the dropdown menu.

DISABLED(비활성)를 선택하여 자산 속성에 대한 알림을 비활성화할 수도 있습니다.

6. 저장을 선택합니다.

자산 자산 알림 활성화 (AWS CLI)

기본적으로 속성 값 업데이트를 게시하지 AWS IoT SiteWise 않습니다. AWS Command Line Interface (AWS CLI) 를 사용하여 자산 속성에 대한 알림을 활성화하거나 비활성화할 수 있습니다.

이 절차를 완료하려면 자산의 `assetId` 및 속성의 `propertyId`를 알고 있어야 합니다. 외부 ID를 사용할 수도 있습니다. 자산을 생성했는데 모르는 경우 [ListAssetsAPI](#)를 사용하여 특정 모델의 모든 자산을 나열하십시오. `assetId` [DescribeAsset](#)작업을 사용하여 속성 ID를 포함한 자산의 속성을 볼 수 있습니다.

[UpdateAssetProperty](#)작업을 사용하여 자산 속성에 대한 알림을 활성화하거나 비활성화할 수 있습니다. 다음 파라미터를 지정합니다.

- `assetId` - 자산의 ID입니다.
- `propertyId` - 자산 속성의 ID입니다.
- `propertyNotificationState` - 속성 값 알림 상태는 ENABLED 또는 DISABLED입니다.
- `propertyAlias` - 속성의 별칭입니다. 알림 상태를 업데이트할 때 속성의 기존 별칭을 지정합니다. 이 파라미터를 생략하면 속성의 기존 별칭이 제거됩니다.

자산 속성에 대한 알리를 활성화하거나 비활성화하려면(CLI)

1. 다음 명령을 실행하여 자산 속성의 별칭을 검색합니다. *asset-id*를 자산의 ID로 바꾸고 *property-id*를 속성의 ID로 바꿉니다.

```
aws iotsitewise describe-asset-property \
  --asset-id asset-id \
  --property-id property-id
```

이 작업은 다음과 같은 형식으로 자산 속성의 세부 정보가 포함된 응답을 반환합니다. 속성 별칭은 JSON 객체에 있는 `assetProperty.alias`입니다.

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetName": "Wind Turbine 7",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetProperty": {
    "id": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "name": "Wind Speed",
    "alias": "/company/windfarm/3/turbine/7/windspeed",
    "notification": {
      "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE",
      "state": "DISABLED"
    },
    "dataType": "DOUBLE",
    "unit": "m/s",
    "type": {
      "measurement": {}
    }
  }
}
```

2. 다음 명령을 실행하여 자산 속성에 대한 알리를 활성화합니다. *property-alias*를 이전 명령의 응답에서 얻은 속성 별칭으로 바꾸거나, `--property-alias`를 생략하여 별칭 없이 속성을 업데이트합니다.

```
aws iotsitewise update-asset-property \
  --asset-id asset-id \
  --property-id property-id \
  --property-notification-state ENABLED \
```

```
--property-alias property-alias
```

--property-notification-state DISABLED를 전달하여 자산 속성에 대한 알림을 비활성화할 수도 있습니다.

자산 속성 알림 메시지 쿼리

자산 속성 알림을 쿼리하려면 SQL 문으로 구성된 AWS IoT Core 규칙을 생성하십시오.

AWS IoT SiteWise 자산 속성 데이터 업데이트를 다음 형식으로 AWS IoT Core에 게시합니다.

```
{
  "type": "PropertyValueUpdate",
  "payload": {
    "assetId": "String",
    "propertyId": "String",
    "values": [
      {
        "timestamp": {
          "timeInSeconds": Number,
          "offsetInNanos": Number
        },
        "quality": "String",
        "value": {
          "booleanValue": Boolean,
          "doubleValue": Number,
          "integerValue": Number,
          "stringValue": "String"
        }
      }
    ]
  }
}
```

values목록의 각 구조는 timestamp-quality-value (TQV) 구조입니다.

- timestamp는 현재 Unix epoch 시간(초)과 오프셋(나노초)을 포함합니다.
- quality는 데이터 요소의 품질을 나타내는 다음 문자열 중 하나를 포함합니다.
 - GOOD – 데이터가 어떤 문제의 영향도 받지 않습니다.
 - BAD – 데이터가 센서 고장과 같은 문제의 영향을 받습니다.

- UNCERTAIN – 데이터가 센서 부정확과 같은 문제의 영향을 받습니다.
- value는 속성 유형에 따라 다음 필드 중 하나를 포함합니다.
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue

values 배열의 값을 구문 분석하려면 규칙의 SQL 문에서 복잡한 중첩 객체 쿼리를 사용해야 합니다. 자세한 내용은 AWS IoT 개발자 가이드의 [중첩 객체 쿼리](#)를 참조하거나 [Amazon DynamoDB에 속성 값 업데이트 게시](#) 자습서에서 자산 속성 알림 메시지 구문 분석에 대한 구체적인 예를 참조하십시오.

Example 값 배열을 추출하는 쿼리 예제

다음 명령문은 해당 속성을 가진 모든 자산의 특정 이중 유형 속성을 확인하기 위해 업데이트된 속성 값 배열을 쿼리하는 방법을 보여 줍니다.

```
SELECT
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-1111EXAMPLE/assets/+/'
  properties/a1b2c3d4-5678-90ab-cdef-3333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

이전 규칙 쿼리 문은 다음과 같은 형식으로 데이터를 출력합니다.

```
{
  "windspeed": [
    26.32020195042838,
    26.282584572975477,
    26.352566977372508,
    26.283084346171442,
    26.571883739599322,
    26.60684140743005,
    26.628738636715045,
    26.273486932802125,
    26.436379105473964,
    26.600590095377303
  ]
}
```

}

Example 단일 값을 추출하는 쿼리 예제

다음 명령문은 해당 속성을 가진 모든 자산의 특정 이중 유형 속성을 확인하기 위해 속성 값 배열의 첫 번째 값을 쿼리하는 방법을 보여 줍니다.

```
SELECT
  get((SELECT VALUE (value.doubleValue) FROM payload.values), 0) AS windspeed
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-1111EXAMPLE/assets/+/'
  properties/a1b2c3d4-5678-90ab-cdef-3333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

이전 규칙 쿼리 문은 다음과 같은 형식으로 데이터를 출력합니다.

```
{
  "windspeed": 26.32020195042838
}
```

Important

이 규칙 쿼리 문은 각 배치에 있는 첫 번째 값 이외의 값 업데이트를 무시합니다. 각 배치에는 최대 10개의 값이 포함될 수 있습니다. 나머지 값을 포함해야 하는 경우 자산 속성 값을 다른 서비스로 출력하는 보다 복잡한 솔루션을 설정해야 합니다. 예를 들어 배열의 각 값을 다른 주제에 다시 게시하는 AWS Lambda 작업이 포함된 규칙을 설정하고, 해당 주제를 쿼리하여 각 값을 원하는 규칙 작업에 게시하는 다른 규칙을 설정할 수 있습니다.

자산 속성 알림과 함께 Amazon S3로 데이터 내보내기

수신 데이터를 계정의 Amazon S3 AWS IoT SiteWise 버킷으로 내보낼 수 있습니다. 기록 보고서를 생성하거나 복잡한 데이터 분석에 사용할 수 있는 형식으로 데이터를 백업할 수 있습니다.

Note

AWS IoT SiteWise 또한 고객 관리형 Amazon S3 버킷에 데이터를 저장할 수 있는 콜드 티어 스토리지도 지원합니다. 지원되는 스토리지 계층에 대한 자세한 내용은 [데이터 스토리지 관리 단원](#)을 참조하십시오.

AWS IoT SiteWise 이 기능을 템플릿으로 AWS CloudFormation 제공합니다. 템플릿에서 스택을 생성하면 수신 데이터를 S3 버킷으로 스트리밍하는 데 필요한 AWS 리소스가 AWS CloudFormation 생성됩니다. AWS IoT SiteWise

그러면 S3 버킷은 AWS IoT SiteWise 속성 값 업데이트 메시지에서 전송된 모든 자산 자산 데이터를 수신합니다. 자산 및 속성 이름과 기타 정보를 포함하는 자산 메타데이터도 S3 버킷으로 보냅니다.

Amazon S3로 내보낼 자산 속성에 대해 특성 값 업데이트 메시지를 사용할 수 있도록 하는 방법은 [다른 AWS 서비스와의 상호작용 단원](#)을 참조하십시오.

이 기능은 자산 속성 데이터 및 자산 메타데이터를 Amazon S3에 [Apache Parquet](#) 형식으로 저장합니다. Parquet란 JSON과 같은 행 지향 형식에 비해 쿼리 속도가 빠르고 공간이 절약되는 데이터 열 형식입니다.

Note

이 기능으로 자산 메타데이터를 검색하면 최대 약 1,500개의 자산을 사용할 수 있습니다. 이 제한은 자산 메타데이터에만 적용됩니다. 이 기능으로 자산 속성 데이터를 내보낼 때는 자산 수에 이러한 제한이 적용되지 않습니다.

각 리소스의 이름에는 스택을 만들 때 사용자 지정할 수 있는 접두사가 포함됩니다. 다음과 같은 리소스가 있습니다.

- Amazon S3 버킷
- AWS Lambda 함수
- AWS IoT Core 규칙
- AWS Identity and Access Management 역할
- Amazon Data Firehose 스트림
- 데이터베이스 AWS Glue

전체 목록은 [템플릿에서 생성된 리소스](#) 단원을 참조하십시오.

Important

이 AWS CloudFormation 템플릿이 생성하고 사용하는 리소스에 대한 요금이 부과됩니다. 이러한 요금에는 여러 AWS 서비스에 대한 데이터 저장 및 데이터 전송이 포함됩니다.

주제

- [AWS CloudFormation 스택 생성](#)
- [Amazon S3에서 데이터 보기](#)
- [Amazon Athena로 내보낸 데이터를 분석하십시오.](#)
- [템플릿에서 생성된 리소스](#)

AWS CloudFormation 스택 생성

자산 데이터를 Amazon AWS CloudFormation S3로 내보내려면 스택을 생성해야 합니다.

Amazon S3로 데이터 내보내기

1. [AWS CloudFormation 템플릿](#)을 열고 AWS Management Console에 로그인합니다.
2. 스택 생성 페이지 하단에서 다음을 선택합니다.
3. 스택 세부 정보 지정 페이지에서 이 템플릿이 자산 데이터를 수신하기 위해 생성하는 S3 버킷의 BucketNamea를 입력합니다. 이 버킷 이름은 전역적으로 고유해야 합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 [버킷 이름 지정 규칙](#)을 참조하십시오.
4. (선택 사항) 템플릿의 다른 파라미터를 변경합니다.
 - GlobalResourcePrefix – IAM 역할과 같이 이 템플릿에서 생성된 글로벌 리소스 이름에 대한 접두사입니다.
 - LocalResourcePrefix – 현재 리전의 이 템플릿에서 생성된 리소스 이름에 대한 접두사입니다.

Note

이 템플릿을 여러 번 생성하는 경우, 리소스 이름 충돌을 피하려면 버킷 이름과 리소스 접두사 파라미터를 변경해야 합니다.

5. 다음을 선택합니다.
6. Configure stack options(스택 옵션 구성) 페이지에서 Next(다음)를 선택합니다.
7. 페이지 하단에서 AWS CloudFormation 이 IAM 리소스를 생성할 수 있음에 동의합니다 확인란을 선택합니다.
8. 스택 생성을 선택합니다.

스택을 만드는 데 몇 분 정도 걸립니다. 스택이 생성되지 않는 경우, 계정에 권한이 부족하거나 이미 존재하는 버킷 이름을 입력했을 수 있습니다. 다음 단계에 따라 스택을 삭제하고 다시 시도하십시오.

- a. 오른쪽 상단에서 삭제를 선택합니다.

스택을 삭제하는 데 몇 분 정도 걸립니다.

Note

AWS CloudFormation S3 버킷 또는 CloudWatch 로그 그룹을 삭제하지 않습니다. 해당 서비스의 콘솔에서 이러한 리소스를 직접 삭제할 수 있습니다.

- b. 스택을 삭제할 수 없으면 [Delete(삭제)]를 다시 선택합니다.
 - c. 스택이 다시 삭제되지 않는 경우 AWS CloudFormation 콘솔의 단계에 따라 삭제에 실패한 리소스를 건너뛰고 다시 시도하십시오.
9. AWS CloudFormation 스택이 성공적으로 생성되면 다음 절차에 따라 Amazon S3의 자산 자산 자산 데이터를 탐색합니다.

Important

스택을 생성하고 나면 AWS 계정에서 새 리소스를 볼 수 있습니다. 이러한 리소스를 삭제하거나 수정하면 기능이 제대로 작동하지 않을 수 있습니다. 버킷에 대한 데이터 전송을 중지하거나 이 기능을 사용자 정의하려는 경우가 아니면 이러한 리소스를 수정하지 않는 것이 좋습니다.

Amazon S3에서 데이터 보기

기능을 생성한 후 Amazon S3에서 자산 속성 데이터 및 자산 메타데이터를 볼 수 있습니다.

Note

자산 메타데이터는 6시간마다 업데이트됩니다. S3 버킷에 표시된 자산 메타데이터를 보려면 최대 6시간을 기다려야 할 수 있습니다.

이 기능은 자산 특성 데이터를 다음 열에 저장하며, 각 행에 데이터 포인트가 하나씩 들어 있습니다.

- `type` - 속성 알림의 유형(PropertyValueUpdate)입니다.
- `asset_id` - 데이터 포인트를 받은 자산의 ID입니다.
- `asset_property_id` - 자산에 대한 데이터 포인트를 받은 속성의 ID입니다.
- `time_in_seconds` - 데이터를 수신한 시간으로, Unix Epoch 시간으로 초 단위로 표시됩니다.
- `offset_in_nanos` - `timeInSeconds`의 나노초 오프셋입니다.
- `asset_property_quality` - 데이터 포인트의 품질로, GOOD, UNCERTAIN 또는 BAD입니다.
- `asset_property_value` - 데이터 포인트의 값입니다.
- `asset_property_data_type` - 자산 속성의 데이터 유형으로, `boolean`, `double`, `integer` 또는 `string`입니다.

이 기능은 자산 메타데이터를 다음 열에 저장하며, 각 행에 자산 속성이 하나씩 들어 있습니다.

- `asset_id` - 자산의 ID입니다.
- `asset_name` - 자산의 이름입니다.
- `asset_model_id` - 자산 모델의 ID입니다.
- `asset_property_id` - 자산 속성의 ID입니다.
- `asset_property_name` - 자산 속성의 이름입니다.
- `asset_property_data_type` - 자산 속성의 데이터 유형으로, `BOOLEAN`, `DOUBLE`, `INTEGER` 또는 `STRING`입니다.
- `asset_property_unit` - 자산 속성의 단위입니다.
- `asset_property_alias` - 자산 속성의 별칭입니다.

Amazon S3에서 AWS IoT SiteWise 데이터를 보려면

1. [Amazon S3 콘솔](#)로 이동합니다.
2. 버킷 목록에서 템플릿을 생성할 때 선택한 이름의 버킷을 선택합니다.

3. 그 버킷에서 다음 폴더 중 하나를 선택합니다.
 - `asset-property-updates`— 이 폴더에는 내보낸 자산 속성 데이터가 들어 AWS IoT SiteWise 있습니다.
 - `asset-metadata`— 이 폴더에는 내보낸 자산 세부 정보가 들어 AWS IoT SiteWise 있습니다.
4. 보려는 객체를 선택합니다.
5. 객체 페이지에서 다음을 수행합니다.
 - a. 선택 위치 탭을 선택합니다.

이 패널에서 Parquet 파일의 레코드를 미리 볼 수 있습니다.
 - b. 파일 형식으로 Parquet를 선택합니다.
 - c. 파일 내용을 JSON 형식으로 표시하려면 파일 미리 보기 표시를 선택합니다.

Note

버킷에 새 데이터가 표시되지 않는 경우, 자산 속성에 대해 속성 값 업데이트 알림을 활성화했는지 확인합니다. 자세한 내용은 [다른 AWS 서비스와의 상호작용](#) 섹션을 참조하십시오.

S3 버킷에 저장된 자산 데이터를 분석하는 방법에 대한 자세한 내용은 [Amazon Athena로 내보낸 데이터를 분석하십시오](#) 단원을 참조하십시오.

Amazon Athena로 내보낸 데이터를 분석하십시오.

Amazon S3에 자산 자산 데이터가 있으면 여러 AWS 서비스를 사용하여 보고서를 생성하거나 데이터를 분석 및 쿼리할 수 있습니다.

- [Amazon Athena](#)를 사용하여 데이터에 대해 SQL 쿼리를 실행합니다.
- [Amazon EMR](#)을 사용하여 빅 데이터 분석을 수행합니다.
- [Amazon OpenSearch 서비스](#)를 사용하여 데이터를 검색하고 분석할 수 있습니다.

Amazon S3의 데이터와 상호 작용할 수 있는 다른 AWS 서비스는 의 Analytics에 나열되어 [AWS Management Console](#) 있습니다.

Note

스택은 자산 속성 데이터의 형식을 지정하는 AWS Glue 데이터베이스를 생성합니다. 이 데이터베이스에서 자산 데이터를 쿼리할 수 없습니다. 이 섹션의 단계에 따라 쿼리할 수 있는 AWS Glue 데이터베이스를 만드십시오.

이 자습서에서는 Amazon Athena를 사용하기 위한 사전 요구 사항을 구성하는 방법과 Athena를 사용하여 내보낸 자산 데이터에 대해 SQL 쿼리를 실행하는 방법을 알아봅니다. AWS IoT SiteWise Athena로 데이터를 쿼리하려면 먼저 자산 AWS Glue Data Catalog 데이터로 데이터를 채워야 합니다. 데이터 카탈로그에는 데이터베이스와 테이블이 포함되며 Athena는 데이터 카탈로그의 데이터에 액세스할 수 있습니다. 내보낸 자산 데이터로 데이터 카탈로그를 정기적으로 업데이트하는 AWS Glue 크롤러를 만들 수 있습니다.

주제

- [크롤러를 구성하여 AWS Glue Data Catalog채우기](#)
- [Athena를 사용하여 데이터 쿼리하기](#)

크롤러를 구성하여 AWS Glue Data Catalog채우기

AWS Glue 크롤러는 데이터 저장소를 크롤링하여 테이블에 값을 채웁니다. AWS Glue Data Catalog이 절차에서는 내보낸 자산 데이터가 포함된 S3 버킷용 AWS Glue 크롤러를 만들고 실행합니다. 크롤러는 자산 속성 업데이트에 대한 테이블과 자산 메타데이터에 대한 테이블을 생성합니다. 이후 Athena를 사용해 이러한 테이블에 SQL 쿼리를 수행할 수 있습니다. 자세한 내용은 AWS Glue 개발자 안내서의 [AWS Glue Data Catalog채우기](#) 및 [크롤러 정의](#)를 참조하십시오.

크롤러를 만들려면 AWS Glue

1. [AWS Glue 콘솔](#)로 이동합니다.
2. 탐색 창에서 크롤러를 선택합니다.
3. 크롤러 추가를 선택합니다.
4. 크롤러 추가 페이지에서 다음을 수행합니다.
 - a. **IoTSiteWiseDataCrawler**와 같이 크롤러 이름을 입력하고 다음을 선택합니다.
 - b. 크롤러 소스 유형에서 데이터 스토어를 선택하고 다음을 선택합니다.
 - c. 데이터 스토어 추가 페이지에서 다음을 수행합니다.

- i. 데이터 스토어 선택에서 S3을 선택합니다.
- ii. 경로 포함에서 **s3://DOC-EXAMPLE-BUCKET1**을 입력해 자산 데이터 버킷을 데이터 스토어로 추가합니다. DOC-EXAMPLE-BUCKET1은 스택을 생성할 때 선택한 버킷 이름으로 바꿉니다.
- iii. 다음을 선택합니다.

Add a data store

Choose a data store

S3

Connection

Select a connection

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

Add connection

Crawl data in

Specified path in my account

Specified path in another account

Include path

s3://AWSDOC-EXAMPLE-BUCKET1

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

▶ Exclude patterns (optional)

Back Next

- d. 다른 데이터 스토어 추가 페이지에서 아니요를 선택하고 다음을 선택합니다.
- e. IAM 역할 선택 페이지에서 다음을 수행합니다.
 - i. S3 버킷에 액세스할 수 있는 새 서비스 역할을 AWS Glue 생성하려면 IAM 역할 생성을 선택합니다.
 - ii. 역할 이름의 접미사를 입력합니다(예: **IoTSiteWiseDataCrawler**).
 - iii. 다음을 선택합니다.
- f. 빈도에서 시간별을 선택하고 다음을 선택합니다. 크롤러는 실행될 때마다 새 데이터로 테이블을 업데이트하므로 사용 사례에 맞는 빈도를 선택할 수 있습니다.
- g. 크롤러 출력 구성 페이지에서 다음을 수행합니다.
 - i. Add database (데이터베이스 추가) 를 선택하여 자산 데이터용 AWS Glue 데이터베이스를 생성합니다.

- ii. 데이터베이스의 이름을 입력합니다(예: **iot_sitewise_asset_database**).
 - iii. 생성을 선택합니다.
 - iv. 다음을 선택합니다.
- h. 크롤러 세부 정보를 검토한 다음 마침을 선택합니다.

Crawler info

Name IoTSiteWiseDataCrawler
Tags -

Data stores

Data store S3
Include path s3://AWSDOC-EXAMPLE-BUCKET1
Connection
Exclude patterns

IAM role

IAM role am:aws:iam::123456789012:role/service-role/AWSGlueServiceRole-IoTSiteWiseDataCrawler

Schedule

Schedule At 00 minutes past the hour

Output

Database iot_sitewise_asset_database
Prefix added to tables (optional)
Create a single schema for each S3 path false
▶ Configuration options

Back
Finish

기본적으로 새 크롤러는 즉시 실행되지 않습니다. 수동으로 실행하거나 구성된 일정에 따라 실행될 때까지 기다려야 합니다.

크롤러 실행

1. 크롤러 페이지에서 새 크롤러의 확인란을 선택하고 크롤러 실행을 선택합니다.

The screenshot shows the AWS Glue Crawlers console. On the left is a navigation menu with 'Crawlers' selected. The main area shows a description of crawlers and a 'Run crawler' button circled in red. Below is a table with one crawler entry:

<input checked="" type="checkbox"/>	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input checked="" type="checkbox"/>	IoTSiteWiseDataCrawler	At 00 minutes...	Ready		0 secs	0 secs	0	0

2. 크롤러가 완료되고 준비 상태가 될 때까지 기다립니다.

크롤러를 실행하는 데 몇 분 정도 걸릴 수 있으며 상태가 자동으로 업데이트됩니다.

3. 탐색 창에서 테이블을 선택합니다.

새 테이블 2개가 보여야 합니다(asset_metadata 및 asset_property_updates).

Athena를 사용하여 데이터 쿼리하기

Athena는 에서 자산 데이터 테이블을 자동으로 검색합니다. AWS Glue Data Catalog이러한 테이블의 교차점에 쿼리를 수행하려면 보기(논리 데이터 테이블)를 생성할 수 있습니다. 자세한 내용은 Amazon Athena 사용 설명서의 [뷰 작업](#)을 참조하십시오.

자산 속성 데이터와 메타데이터를 결합하는 보기를 생성한 후 자산 및 속성 이름이 연결된 속성 값을 출력하는 쿼리를 실행할 수 있습니다. 자세한 내용은 Amazon Athena 사용 설명서의 [Amazon Athena를 사용하여 SQL 쿼리 실행](#)을 참조하십시오.

Athena로 자산 데이터 쿼리

1. [Athena 콘솔](#)로 이동합니다.

시작하기 페이지가 나타나면 시작하기를 선택합니다.

2. Athena를 처음 사용하는 경우 다음 단계를 완료하여 쿼리 결과에 대한 S3 버킷을 구성합니다. Athena는 쿼리 결과를 이 버킷에 저장합니다.

⚠ Important

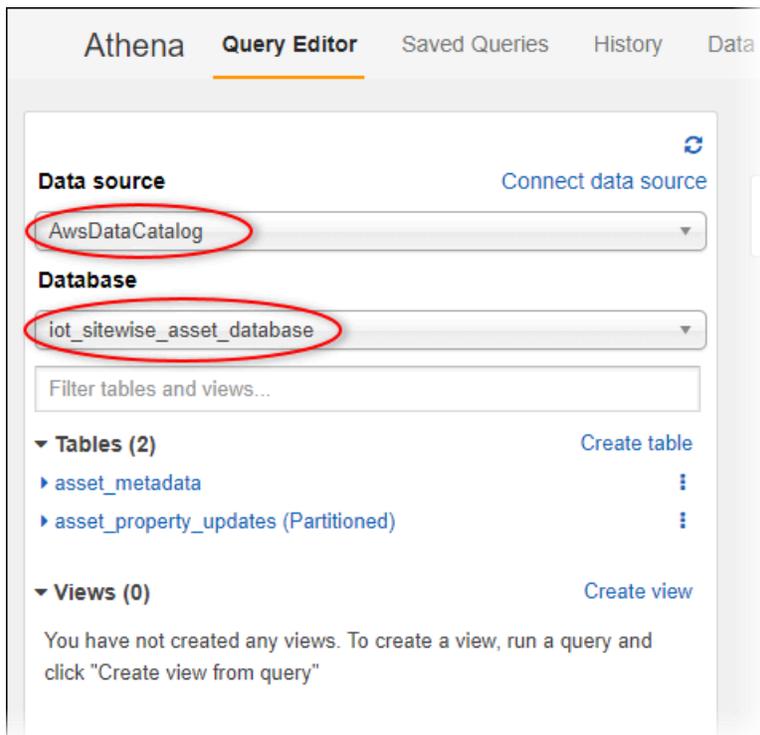
이전에 생성한 크롤러가 쿼리 결과를 크롤하지 않도록 자산 데이터 버킷과 다른 버킷을 사용하십시오. Athena 쿼리 결과에만 사용할 버킷을 생성하는 것이 좋습니다. 자세한 내용은

Amazon Simple Storage Service Console 사용 설명서의 [S3 버킷을 생성하려면 어떻게 해야 합니까?](#)를 참조하십시오.

- a. 설정을 선택합니다.
- b. 쿼리 결과 위치에 Athena 쿼리 결과를 위한 S3 버킷을 입력합니다. 버킷은 /로 끝나야 합니다.

The screenshot shows the 'Settings' dialog box for Athena. The 'Query result location' field is highlighted with a red oval and contains the text 's3://AWSDOC-EXAMPLE-BUCKET/'. Below it, an example path 's3://query-results-bucket/folder/' is shown. The 'Encrypt query results' and 'Autocomplete' checkboxes are unchecked. The 'Save' button is also highlighted with a red oval.

- c. 저장을 선택합니다.
3. 왼쪽 패널에는 쿼리할 데이터 원본이 포함되어 있습니다. 다음을 따릅니다.
 - a. 데이터 소스의 경우 사용을 선택합니다 AwsDataCatalog. AWS Glue Data Catalog
 - b. 데이터베이스의 경우 크롤러로 만든 AWS Glue 데이터베이스를 선택합니다.



테이블 2개가 보여야 합니다(asset_metadata 및 asset_property_updates).

4. 자산 속성 데이터와 메타데이터를 조합하여 보기를 생성하려면 다음 쿼리를 입력한 다음 쿼리 실행을 선택합니다.

```
CREATE
    OR REPLACE VIEW iot_sitewise_asset_data AS
SELECT "from_unixtime"("time_in_seconds" + ("offset_in_nanos" / 1000000000))
    "timestamp",
    "metadata"."asset_name",
    "metadata"."asset_property_name",
    "data"."asset_property_value",
    "metadata"."asset_property_unit",
    "metadata"."asset_property_alias"
FROM ( "iot_sitewise_asset_database".asset_property_updates data
INNER JOIN "iot_sitewise_asset_database".asset_metadata metadata
    ON ( ("data"."asset_id" = "metadata"."asset_id")
        AND ("data"."asset_property_id" = "metadata"."asset_property_id") ) );
```

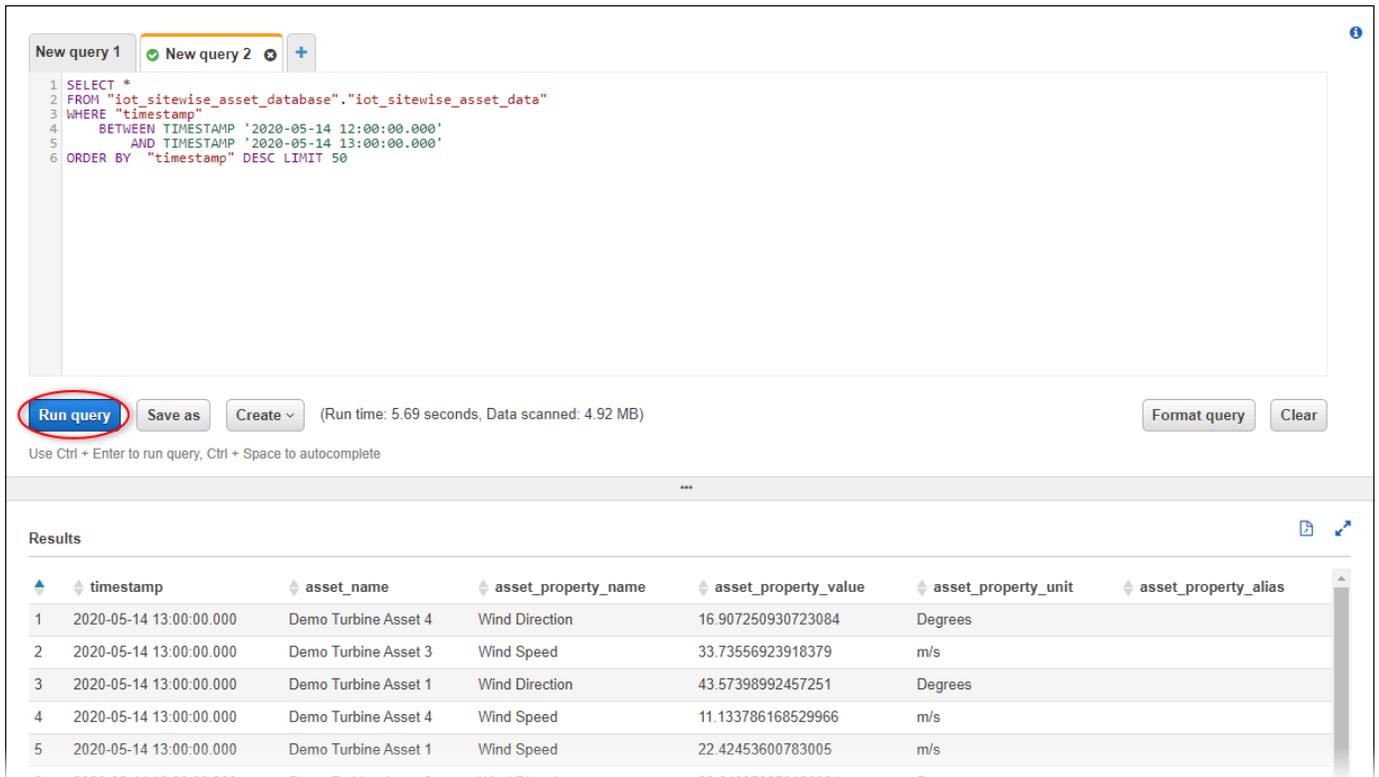
이 쿼리는 자산 ID 및 속성 ID의 자산 속성 데이터 및 메타데이터 테이블을 조인하여 보기를 생성합니다. 보기가 이미 존재하는 경우 기존 보기를 대체하기 때문에 이 쿼리는 여러 번 실행할 수 있습니다.

5. + 아이콘을 선택하여 새 쿼리를 추가합니다.
6. 자산 데이터 샘플을 보려면 다음 쿼리를 입력한 다음 쿼리 실행을 선택합니다. 타임스탬프는 버킷에 데이터가 있는 간격으로 바꿉니다.

```
SELECT *
FROM "iot_sitewise_asset_database"."iot_sitewise_asset_data"
WHERE "timestamp"
    BETWEEN TIMESTAMP '2020-05-14 12:00:00.000'
    AND TIMESTAMP '2020-05-14 13:00:00.000'
ORDER BY "timestamp" DESC LIMIT 50;
```

이 쿼리는 타임스탬프 2개 사이에 데이터 포인트 최대 50개를 출력하며 가장 최근 항목을 먼저 표시합니다.

쿼리 출력은 다음 결과와 비슷합니다.



The screenshot shows the AWS IoT SiteWise query editor interface. At the top, there are two tabs: 'New query 1' and 'New query 2'. The SQL query is entered in the main text area. Below the query, there are buttons for 'Run query', 'Save as', 'Create', 'Format query', and 'Clear'. The 'Run query' button is circled in red. Below the buttons, there is a status bar showing '(Run time: 5.69 seconds, Data scanned: 4.92 MB)'. Below the status bar, there is a 'Results' section with a table of data. The table has columns for 'timestamp', 'asset_name', 'asset_property_name', 'asset_property_value', 'asset_property_unit', and 'asset_property_alias'. The results show 5 rows of data for different assets and their properties.

timestamp	asset_name	asset_property_name	asset_property_value	asset_property_unit	asset_property_alias
2020-05-14 13:00:00.000	Demo Turbine Asset 4	Wind Direction	16.907250930723084	Degrees	
2020-05-14 13:00:00.000	Demo Turbine Asset 3	Wind Speed	33.73556923918379	m/s	
2020-05-14 13:00:00.000	Demo Turbine Asset 1	Wind Direction	43.57398992457251	Degrees	
2020-05-14 13:00:00.000	Demo Turbine Asset 4	Wind Speed	11.133786168529966	m/s	
2020-05-14 13:00:00.000	Demo Turbine Asset 1	Wind Speed	22.42453600783005	m/s	

이제 애플리케이션에 유용한 쿼리를 실행할 수 있습니다. AWS IoT SiteWise 자세한 내용은 Amazon Athena 사용 설명서에서 [Amazon Athena에 대한 SQL 참조](#)를 참조하십시오.

템플릿에서 생성된 리소스

템플릿에서 스택을 생성하면 는 다음과 같은 리소스를 AWS CloudFormation 생성합니다. 대부분의 리소스 이름에는 접두사가 포함되며, 스택을 생성할 때 이를 사용자 지정할 수 있습니다.

리소스 이름 파라미터

- **BucketName** – 이 템플릿에서 생성된 S3 버킷의 이름으로, 자산 데이터를 수신합니다.
- **GlobalResourcePrefix** – 이 템플릿에서 생성된 글로벌 리소스 이름에 대한 접두사입니다. 기본 값은 `sitewise-export-to-s3`입니다.
- **LocalResourcePrefix** – 현재 리전의 이 템플릿에서 생성된 리소스 이름에 대한 접두사입니다. 기본 값은 `sitewise_export_to_s3`입니다.

AWS CloudFormation 템플릿으로 생성된 리소스

Resource	설명	명칭
처리된 데이터용 S3 버킷	이 버킷에는 폴더가 두 개 있습니다. 한 폴더는 Firehose 전송 스트림에서 평면화되고 형식이 지정된 데이터를 수신하고 다른 폴더는 자산 메타데이터를 수신합니다.	<code>\${BucketName}</code>
AWS Glue 데이터베이스	이 데이터베이스에는 이 스택이 생성하는 AWS Glue 테이블이 포함되어 있습니다.	<code>\${LocalResourcePrefix}_firehose_glue_database</code>
AWS Glue 테이블	Firehose 전송 스트림은 이 테이블을 사용하여 데이터를 Parquet 형식으로 포맷합니다.	<code>\${LocalResourcePrefix}_firehose_glue_table</code>
데이터를 변환하는 AWS Lambda 함수	이 함수는 보낸 속성 값 알림 메시지의 값 배열을 평면화합니다. AWS IoT SiteWise	<code>\${LocalResourcePrefix}_lambda_transform_function</code>

Resource	설명	명칭
Lambda 변환 함수를 위한 IAM 역할	이 역할은 Lambda가 변환 함수에 대한 런타임 로그를 저장할 수 있도록 합니다.	<code>\${GlobalResourcePrefix}-lambda-transform-role</code>
Lambda 변환 함수 역할을 위한 IAM 정책	이 정책은 Lambda가 변환 함수에 대한 실행 로그를 저장할 수 있도록 허용합니다.	<code>\${GlobalResourcePrefix}-lambda-transform-policy</code>
CloudWatch 변환 함수의 로그 로그 그룹을 기록합니다.	이 로그 그룹에는 변환 함수에 대한 로그가 포함되어 있습니다.	<code>/aws/lambda/\${LocalResourcePrefix}_lambda_transform_function</code>
자산 메타데이터를 수집하는 Lambda 함수	이 함수는 자산에 AWS IoT SiteWise 대한 세부 정보를 검색하고 이 스택이 생성하는 Amazon S3 버킷에 세부 정보를 저장합니다.	<code>\${LocalResourcePrefix}_lambda_metadata_function</code>
메타데이터 함수를 위한 Lambda 계층	이 계층은 메타데이터 함수가 사용하는 AWS IoT SiteWise 작업이 포함된 AWS SDK를 제공합니다.	<code>\${LocalResourcePrefix}_lambda_metadata_layer</code>
메타데이터 Lambda 함수에 대한 IAM 역할	이 역할을 통해 Lambda는 의 자산에 대한 세부 정보를 검색할 수 있습니다. AWS IoT SiteWise	<code>\${GlobalResourcePrefix}-lambda-metadata-role</code>
메타데이터 Lambda 함수 역할에 대한 IAM 정책	이 정책을 통해 Lambda는 의 자산에 대한 세부 정보를 검색할 수 있습니다. AWS IoT SiteWise	<code>\${GlobalResourcePrefix}-lambda-metadata-policy</code>

Resource	설명	명칭
EventBridge 메타데이터 Lambda 함수에 대한 스케줄링된 이벤트	이 예약된 이벤트는 6시간마다 메타데이터 Lambda를 실행하여 자산 메타데이터 버킷을 업데이트합니다.	<code>\${LocalResourcePrefix}-metadata-event</code>
CloudWatch 메타데이터 함수의 로그 로그 그룹	이 로그 그룹에는 메타데이터 함수에 대한 로그가 포함되어 있습니다.	<code>/aws/lambda/\${LocalResourcePrefix}_lambda_metadata_function</code>
AWS IoT 규칙	이 규칙은 속성 값 알림 메시지를 쿼리하고 자산 데이터를 Amazon Data Firehose 전송 스트림으로 보냅니다.	<code>\${LocalResourcePrefix}_iot_topic_rule</code>
규칙의 IAM 역할 AWS IoT	이 역할을 통해 AWS IoT Firehose 전송 스트림으로 데이터를 전송할 수 있습니다.	<code>\${GlobalResourcePrefix}-core-firehose-role</code>
규칙 역할에 대한 AWS IoT IAM 정책	이 정책을 통해 AWS IoT Firehose 전송 스트림으로 데이터를 전송할 수 있습니다.	<code>\${GlobalResourcePrefix}-core-firehose-policy</code>
Firehose 전송 스트림	이 전송 스트림은 AWS IoT 규칙의 데이터를 소비하고, Lambda 함수를 사용하여 데이터를 평면화하고, 데이터를 Amazon S3로 전송합니다.	<code>\${LocalResourcePrefix}_firehose_delivery_stream</code>
전송 스트림에 대한 IAM 역할	이 역할을 통해 Firehose는 S3 버킷, AWS Glue 테이블, Lambda 함수 및 로그 로그 그룹에서 작업을 수행할 수 있습니다. CloudWatch	<code>\${GlobalResourcePrefix}-firehose-delivery-role</code>

Resource	설명	명칭
CloudWatch 전송 스트림의 로그 그룹	이 로그 그룹에는 Firehose 전송 스트림에 대한 로그를 수신하는 로그 스트림이 포함되어 있습니다. S3 Delivery	/aws/kinesisfirehose/\${LocalResourcePrefix}_firehose_delivery_stream

Grafana와 통합

Grafana는 대시보드에서 데이터를 시각화하고 모니터링하는 데 사용할 수 있는 데이터 시각화 플랫폼입니다. Grafana 버전 7.3.0 이상에서는 AWS IoT SiteWise 플러그인을 사용하여 Grafana 대시보드에서 AWS IoT SiteWise 자산 데이터를 시각화할 수 있습니다. 단일 Grafana 대시보드를 사용하여 여러 AWS 소스 (예: Amazon Timestream 및 CloudWatch Amazon) 및 기타 데이터 소스의 데이터를 시각화할 수 있습니다. AWS IoT SiteWise

AWS IoT SiteWise 플러그인에는 두 가지 옵션이 있습니다:

- 로컬 Grafana 서버

관리하는 Grafana 서버에서 AWS IoT SiteWise 플러그인을 설정할 수 있습니다. 플러그인 추가 및 사용 방법에 대한 자세한 내용은 웹 사이트의 [AWS IoT SiteWise 데이터 소스](#) README 파일을 참조하십시오. GitHub

- AWSGrafana용 관리형 서비스

AWS Grafana용 관리형 서비스(AMG)에서 AWS IoT SiteWise 플러그인을 사용할 수 있습니다. AMG는 사용자를 대신하여 Grafana 서버를 관리하므로 사용자는 하드웨어나 기타 Grafana 인프라를 구축, 패키징 또는 배포하지 않고도 데이터를 시각화할 수 있습니다. 자세한 내용은 AWSGrafana용 관리형 서비스 사용 설명서의 다음 주제를 참조하세요.

- [Amazon Grafana용 관리형 서비스\(AMG\)란?](#)
- [AWS IoT SiteWise 데이터 소스 사용](#)

Example Grafana 대시보드 예시

다음 Grafana 대시보드는 [데모 풍력 발전 단지](#)를 시각화합니다. [Grafana Play](#) 웹 사이트에서 이 데모 대시보드에 액세스할 수 있습니다.



AWS IoT SiteWise 및 AWS IoT TwinMaker 통합

통합하면 콘솔의 AWS IoT SiteWise 데이터 검색 ExecuteQuery API 및 고급 자산 검색과 같은 강력한 기능에 대한 액세스 권한을 AWS IoT TwinMaker 부여합니다 AWS IoT SiteWise. 서비스를 통합하고 이러한 기능을 사용하려면 먼저 통합을 활성화해야 합니다.

주제

- [통합 활성화](#)
- [AWS IoT SiteWise 및 AWS IoT TwinMaker 통합](#)

통합 활성화

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다. JSON 정책의 Action 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. AWS IoT SiteWise 지원 작업에 대한 자세한 정보는 서비스 승인 참조의 [AWS IoT SiteWise에 의해 정의된 작업](#)을 참조하십시오.

AWS IoT TwinMaker 서비스 연결 역할에 대한 자세한 내용은 사용 설명서의 [서비스 연결 역할](#)을 참조하십시오. AWS IoT TwinMaker AWS IoT TwinMaker

를 통합하려면 먼저 연결된 AWS IoT TwinMaker 작업 AWS IoT SiteWise 영역과 통합할 수 있는 다음 권한을 부여해야 합니다. AWS IoT SiteWise AWS IoT TwinMaker

- `iotsitewise:EnableSiteWiseIntegration`— 연결된 AWS IoT TwinMaker 작업 영역과 AWS IoT SiteWise 통합할 수 있습니다. 이 통합을 통해 AWS IoT TwinMaker AWS IoT TwinMaker 서비스 연계 역할을 AWS IoT SiteWise 통해 모든 모델링 정보를 읽을 수 있습니다. 이 권한을 활성화하려면 IAM 역할에 다음 정책을 추가하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:EnableSiteWiseIntegration"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS IoT SiteWise 및 AWS IoT TwinMaker 통합

AWS IoT SiteWise를 AWS IoT TwinMaker 통합하려면 다음이 있어야 합니다.

- AWS IoT SiteWise 계정에 서비스 연결 역할 설정
- AWS IoT TwinMaker 계정에 서비스 연결 역할 설정

- AWS IoT TwinMaker 해당 IoTSiteWiseDefaultWorkspace 지역의 사용자 계정에 ID가 있는 작업 공간

AWS IoT SiteWise 콘솔을 사용하여 통합하려면

콘솔에 AWS IoT TwinMaker 배너와의 통합이 표시되면 권한 부여를 선택합니다. 사전 요구 사항은 계정에서 생성됩니다.

를 사용하여 통합하려면 AWS CLI

를 AWS IoT SiteWise 통합하고 AWS IoT TwinMaker 사용하려면 AWS CLI 다음 명령을 입력합니다.

1. CreateServiceLinkedRole을 AWSServiceName 눌러 iotsitewise.amazonaws.com 호출하십시오.

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

2. CreateServiceLinkedRole 끄기로 AWSServiceName iottwinmaker.amazonaws.com 전화하세요.

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

3. CreateWorkspace 끄기로 ID IoTSiteWiseDefaultWorkspace 전화하세요.

```
aws iottwinmaker create-workspace --workspace-id IoTSiteWiseDefaultWorkspace
```

Amazon Lookout for Equipment를 사용하여 장비 이상을 감지합니다.

Note

예외 항목 탐지는 Amazon Lookout for Equipment를 사용할 수 있는 지역에서만 사용할 수 있습니다.

Amazon Lookout for AWS IoT SiteWise Equipment와 통합하면 산업용 장비의 이상 탐지 및 예측 유지 관리를 통해 산업용 장비에 대한 통찰력을 얻을 수 있습니다. Lookout for Equipment는 산업용 장비를

모니터링하기 위한 기계 학습 (ML) 서비스로, 비정상적인 장비 동작을 감지하고 잠재적 고장을 식별합니다. Lookout for Equipment를 사용하면 예측 유지보수 프로그램을 구현하고 최적이지 않은 장비 프로세스를 식별할 수 있습니다. 장비 특이성에 대한 자세한 내용은 [Amazon Lookout for Equipment 란 무엇입니까? 를 참조하십시오](#). Amazon Lookout for Equipment 사용 설명서에서 확인할 수 있습니다.

비정상적인 장비 동작을 탐지하도록 ML 모델을 훈련시키는 예측을 생성하면 자산 속성 값을 Lookout for Equipment로 AWS IoT SiteWise 전송하여 비정상적인 장비 동작을 탐지하도록 ML 모델을 훈련시킵니다. 자산 모델에서 예측 정의를 정의하려면 Lookout for Equipment가 데이터에 액세스하는 데 필요한 IAM 역할과 Lookout for Equipment로 전송하고 처리된 데이터를 Amazon S3로 전송할 속성을 지정합니다. 자세한 정보는 [자산 모델 생성](#)을 참조하세요.

Lookout for Equipment를 AWS IoT SiteWise 통합하고 Lookout을 수행하려면 다음과 같은 상위 단계를 수행해야 합니다.

- 추적하려는 속성을 설명하는 예측 정의를 자산 모델에 추가하십시오. 예측 정의는 해당 자산 모델을 기반으로 자산에 대한 예측을 생성하는 데 사용되는 측정, 변환 및 지표의 재사용 가능한 컬렉션입니다.
- 제공한 과거 데이터를 기반으로 예측을 학습시키십시오.
- 스케줄 추론은 특정 예측의 실행 빈도를 AWS IoT SiteWise 알려줍니다.

추론 일정이 잡히면 Lookout for Equipment 모델은 장비로부터 수신한 데이터를 모니터링하고 장비 동작의 이상을 찾습니다. AWS IoT SiteWise GET API 작업 또는 Lookout for Equipment 콘솔을 사용하여 SiteWise 모니터에서 결과를 보고 분석할 수 있습니다. 또한 자산 모델의 알람 감지기를 사용하여 경보를 생성하여 비정상적인 장비 동작에 대해 경고할 수 있습니다.

주제

- [예측 정의 추가 \(콘솔\)](#)
- [예측 학습 \(콘솔\)](#)
- [예측에 대한 추론 시작 또는 중지 \(콘솔\)](#)
- [예측 정의 추가 \(CLI\)](#)
- [예측 학습 및 시작 추론 \(CLI\)](#)
- [예측 학습 \(CLI\)](#)
- [예측에 대한 추론 시작 또는 중지 \(CLI\)](#)

예측 정의 추가 (콘솔)

에서 수집한 데이터를 Lookout for Equipment로 AWS IoT SiteWise 보내기 시작하려면 자산 모델에 예측 정의를 AWS IoT SiteWise 추가해야 합니다.

AWS IoT SiteWise 자산 모델에 예측 정의를 추가하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 모델을 선택하고 예측 정의를 추가할 자산 모델을 선택합니다.
3. 예측을 선택합니다.
4. 예측 정의 추가를 선택합니다.
5. 예측 정의에 대한 세부 정보를 정의합니다.
 - a. 예측 정의의 고유한 이름 및 설명을 입력합니다. 예측 정의를 생성한 후에는 이름을 변경할 수 없으므로 이름을 신중하게 선택하십시오.
 - b. Amazon Lookout for Equipment와 자산 데이터를 공유할 수 있는 IAM 권한 역할을 생성하거나 선택하십시오. AWS IoT SiteWise 역할에는 다음과 같은 IAM 및 신뢰 정책이 있어야 합니다. 역할 생성에 도움이 필요하면 [사용자 지정 신뢰 정책을 사용하여 역할 생성 \(콘솔\)](#) 을 참조하십시오.

IAM 정책

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "L4EPermissions",
    "Effect": "Allow",
    "Action": [
      "lookoutequipment:CreateDataset",
      "lookoutequipment:CreateModel",
      "lookoutequipment:CreateInferenceScheduler",
      "lookoutequipment:DescribeDataset",
      "lookoutequipment:DescribeModel",
      "lookoutequipment:DescribeInferenceScheduler",
      "lookoutequipment:ListInferenceExecutions",
      "lookoutequipment:StartDataIngestionJob",
      "lookoutequipment:StartInferenceScheduler",
      "lookoutequipment:UpdateInferenceScheduler",
      "lookoutequipment:StopInferenceScheduler"
    ]
  }],
}
```

```

    "Resource": [
      "arn:aws:lookoutequipment:Region:Account_ID:inference-
scheduler/IoTSiteWise_*",
      "arn:aws:lookoutequipment:Region:Account_ID:model/
IoTSiteWise_*",
      "arn:aws:lookoutequipment:Region:Account_ID:dataset/
IoTSiteWise_*"
    ]
  },
  {
    "Sid": "L4EPermissions2",
    "Effect": "Allow",
    "Action": [
      "lookoutequipment:DescribeDataIngestionJob"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3Permissions",
    "Effect": "Allow",
    "Action": [
      "s3:CreateBucket",
      "s3:ListBucket",
      "s3:PutObject",
      "s3:GetObject"
    ],
    "Resource": ["arn:aws:s3:::iotsitewise-*"]
  },
  {
    "Sid": "IAMPermissions",
    "Effect": "Allow",
    "Action": [
      "iam:GetRole",
      "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::Account_ID:role/Role_name"
  }
]
}

```

신뢰 정책

```
{
```

```

"Version": "2012-10-17",
"Statement": [{
  "Effect": "Allow",
  "Principal": {
    "Service": "iotsitewise.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "Account_ID"
    },
    "ArnEquals": {
      "aws:SourceArn":
"arn:aws:iotsitewise:Region:Account_ID:asset/*"
    }
  }
},
{
  "Effect": "Allow",
  "Principal": {
    "Service": "lookoutequipment.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "Account_ID"
    },
    "ArnEquals": {
      "aws:SourceArn":
"arn:aws:lookoutequipment:Region:Account_ID:*"
    }
  }
}
]
}

```

- c. 다음을 선택합니다.
6. Lookout for Equipment로 전송할 데이터 속성 (측정, 변환 및 지표) 을 선택합니다.
 - a. (선택 사항) 측정값을 선택합니다.
 - b. (선택 사항) 변환을 선택합니다.
 - c. (선택 사항) 지표를 선택합니다.

- d. 다음을 선택합니다.
7. 선택한 내용을 검토합니다. 자산 모델에 예측 정의를 추가하려면 요약 페이지에서 예측 정의 추가를 선택합니다.

활성 예측이 첨부된 기존 예측 정의를 편집하거나 삭제할 수도 있습니다.

예측 학습 (콘솔)

자산 모델에 예측 정의를 추가한 후 자산에 대한 예측을 훈련할 수 있습니다.

에서 예측을 훈련시키려면 AWS IoT SiteWise

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 [Assets] 를 선택하고 모니터링하려는 자산을 선택합니다.
3. 예측을 선택합니다.
4. 훈련하려는 예측을 선택합니다.
5. [액션] 에서 [훈련 시작] 을 선택하고 다음을 수행하십시오.
 - a. 예측 세부 정보에서 자산 데이터를 Lookout for AWS IoT SiteWise Equipment와 공유할 수 있는 IAM 권한 역할을 선택합니다. 새 역할을 생성해야 하는 경우 새 역할 생성을 선택합니다.
 - b. 훈련 데이터 설정의 경우 훈련 데이터 시간 범위를 입력하여 예측을 훈련하는 데 사용할 데이터를 선택합니다.
 - c. (선택 사항) 사후 처리 후 데이터의 샘플링 속도를 선택합니다.
 - d. (선택 사항) 데이터 레이블의 경우 레이블 지정 데이터를 보관하는 Amazon S3 버킷과 접두사를 제공합니다. 데이터 레이블 지정에 대한 자세한 내용은 Amazon Lookout for Equipment 사용 설명서의 [데이터 레이블](#) 지정을 참조하십시오.
 - e. 다음을 선택합니다.
6. (선택 사항) 훈련이 완료되는 즉시 예측이 활성화되도록 하려면 고급 설정에서 학습 후 예측 자동 활성화를 선택하고 다음을 수행하십시오.
 - a. 입력 데이터에서 데이터 업로드 빈도에 대해 데이터 업로드 빈도를 정의하고, 오프셋 지연 시간에서 사용할 버퍼의 양을 정의합니다.
 - b. 다음을 선택합니다.
7. 예측의 세부 정보를 검토하고 저장 후 시작을 선택합니다.

예측에 대한 추론 시작 또는 중지 (콘솔)

Note

Lookout for Equipment AWS IoT SiteWise 간에 데이터가 전송되는 예정된 추론에는 Lookout for Equipment 요금이 부과됩니다. 자세한 내용은 [Amazon Lookout for Equipment](#) 요금을 참조하십시오.

예측 “룩아웃 장비: CreateDataset “를 추가했지만 교육 후에 활성화하도록 선택하지 않은 경우, 이를 활성화해야 자산 모니터링을 시작할 수 있습니다.

예측에 대한 추론을 시작하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택하고 예측이 추가되는 자산을 선택합니다.
3. 예측을 선택합니다.
4. 활성화하려는 예측을 선택합니다.
5. 동작에서 추론 시작을 선택하고 다음을 수행하십시오.
 - a. 입력 데이터에서 데이터 업로드 빈도에 대해 데이터 업로드 빈도를 정의하고 오프셋 지연 시간에 대해 사용할 버퍼량을 정의합니다.
 - b. 저장 후 시작을 선택합니다.

예측에 대한 추론을 중지하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 자산을 선택하고 예측이 추가되는 자산을 선택합니다.
3. 예측을 선택합니다.
4. 중단하려는 예측을 선택합니다.
5. 동작에서 추론 중지를 선택합니다.

예측 정의 추가 (CLI)

새 자산 모델이나 기존 자산 모델에서 예측 정의를 정의하려면 AWS Command Line Interface (AWS CLI) 를 사용할 수 있습니다. 자산 모델에서 예측 정의를 정의한 후에는 Lookout for Equipment를 사용

하여 이상 탐지를 수행할 수 AWS IoT SiteWise 있도록 자산에 대한 예측을 훈련하고 추론 일정을 잡습니다.

사전 조건

이 단계를 완료하려면 자산 모델이 있어야 하고 자산이 하나 이상 생성되어 있어야 합니다. 자세한 내용은 [자산 모델 생성 \(AWS CLI\)](#) 및 [자산 만들기 \(\)AWS CLI](#) 섹션을 참조하세요.

를 처음 사용하는 경우 CreateBulkImportJob API 작업을 호출하여 AWS IoT SiteWise 자산 속성 값을 로 AWS IoT SiteWise 가져와야 합니다. 그러면 모델을 학습시키는 데 사용할 수 있습니다. 자세한 정보는 [대량 가져오기 작업 생성\(AWS CLI\)](#)을 참조하세요.

예측 정의를 추가하려면

- asset-model-payload.json이라는 파일을 생성합니다. 다른 섹션의 단계에 따라 자산 모델의 세부 정보를 파일에 추가하되, 자산 모델을 만들거나 업데이트하라는 요청을 제출하지 마세요.
 - 자산 모델을 만드는 방법에 대한 자세한 내용은 을 참조하십시오. [자산 모델 생성 \(AWS CLI\)](#)
 - 기존 자산 모델을 업데이트하는 방법에 대한 자세한 내용은 을 참조하십시오. [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#)
- 다음 코드를 추가하여 Lookout for Equipment 복합 모델 assetModelCompositeModels () 을 자산 모델에 추가합니다.
 - 포함하려는 속성의 *Property* ID로 바꾸십시오. 해당 ID를 받으려면 전화하세요 [DescribeAssetModel](#).
 - Lookout for Equipment가 데이터에 액세스할 수 있도록 허용하는 IAM 역할의 *RoleARN* ARN으로 대체하십시오. AWS IoT SiteWise

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "L4Epredictiondefinition",
      "type": "AWS/L4E_ANOMALY",
      "properties": [
        {
          "name": "AWS/L4E_ANOMALY_RESULT",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_RESULT",
          "unit": "none",
```

```

    "type": {
      "measurement": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INPUT",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INPUT",
    "type": {
      "attribute": {
        "defaultValue": "{\\"properties\\": [\\"Property1\\", \\"Property2\\"]}"
      }
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_PERMISSIONS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_PERMISSIONS",
    "type": {
      "attribute": {
        "defaultValue": "{\\"roleArn\\": \\"RoleARN\\"}"
      }
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_DATASET",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_DATASET",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_MODEL",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_MODEL",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INFERENCE",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE",

```

```

    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  }
]
}

```

3. 자산 모델을 만들거나 기존 자산 모델을 업데이트합니다. 다음 중 하나를 수행하십시오.

- 자산 모델을 생성하려면 다음 명령을 실행합니다.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- 다음 명령을 실행하여 기존 자산 모델을 업데이트합니다. 업데이트하려는 자산 모델의 *asset-model-id* ID로 바꾸십시오.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

명령을 실행한 후 응답에서 `assetModelId`를 기록합니다.

예측 학습 및 시작 추론 (CLI)

이제 예측 정의가 정의되었으므로 이를 기반으로 자산을 학습시키고 추론을 시작할 수 있습니다. 예측은 학습하되 추론은 시작하지 않으려는 경우에는 [예측 학습 \(CLI\)](#) 예측을 학습시키고 자산에 대한 추론을 시작하려면 대상 `assetId` 리소스의 정보가 필요합니다.

예측에 대한 학습 및 추론 시작

1. 다음 명령을 실행하여 `assetModelCompositeModelId` `assetModelCompositeModelSummaries` 언더를 구합니다. 에서 만든 에셋 모델의 `asset-model-id` ID로 [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#) 바꾸십시오.

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. 다음 명령을 실행하여 `TrainingWithInference` 액션을 찾습니다. `actionDefinitionId` 이전 단계에서 사용한 `asset-model-id` ID로 바꾸고 이전 단계에서 반환된 `asset-model-composite-model-id` ID로 바꾸십시오.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. 라는 `train-start-inference-prediction.json` 파일을 만들고 다음 코드를 추가하여 다음을 대체합니다.

- `asset-id` 대상 자산의 ID로
- `action-definition-id` `TrainingWithInference` 액션의 ID와 함께
- `StartTime` 교육 데이터 시작 시, 에포크 초 단위로 제공
- `EndTime` 교육 데이터의 끝 부분 포함, 에포크 초 단위로 제공
- `TargetSamplingRate` Lookout for Equipment에서 사후 처리를 수행한 후의 데이터 샘플링 속도와 함께 허용되는 값은 다음과 같습니다. PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
}
```

```

"actionDefinitionId": "action-definition-Id",
"actionPayload":{
  "stringValue": "{\\"14ETrainingWithInference\\":{\\"trainingWithInferenceMode
\\":\\"START\\",\\"trainingPayload\\":{\\"exportDataStartTime\\":StartTime,
\\"exportDataEndTime\\":EndTime},\\"targetSamplingRate\\":\\"TargetSamplingRate\\"},
\\"inferencePayload\\":{\\"dataDelayOffsetInMinutes\\":0,\\"dataUploadFrequency\\":\\"PT5M
\\}}}"
}
}

```

4. 다음 명령을 실행하여 학습 및 추론을 시작합니다.

```
aws iotsitewise execute-action --cli-input-json file://train-start-inference-prediction.json
```

예측 학습 (CLI)

이제 예측 정의가 정의되었으므로 이를 기반으로 자산을 학습할 수 있습니다. 자산에 대한 예측을 훈련하려면 대상 리소스의 예측이 필요합니다. `assetId`

예측을 학습시키려면

1. 다음 명령을 실행하여 `assetModelCompositeModelId` 언더를 `assetModelCompositeModelSummaries` 구합니다. 에서 만든 에셋 모델의 *asset-model-id*ID로 [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#) 바꾸십시오.

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. 다음 명령을 실행하여 Training 액션을 찾습니다. `actionDefinitionId` 이전 단계에서 사용한 *asset-model-id*ID로 바꾸고 이전 단계에서 반환된 *asset-model-composite-model-id*ID로 바꾸십시오.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. 라는 `train-prediction.json` 파일을 만들고 다음 코드를 추가하여 다음을 대체합니다.
 - *asset-id*대상 자산의 ID로

- *action-definition-id* 트레이닝 액션의 ID와 함께
- *StartTime* 훈련 데이터의 시작과 함께, 에포크 초 단위로 제공
- *EndTime* 교육 데이터의 끝 부분 포함, 에포크 초 단위로 제공
- (선택 사항) 레이블 데이터를 보관하는 Amazon S3 버킷의 이름 *BucketName* 포함
- (선택 사항) *Prefix* Amazon S3 버킷과 연결된 접두사를 포함합니다.
- *TargetSamplingRate* Lookout for Equipment에서 사후 처리를 수행한 후의 데이터 샘플링 속도와 함께 허용되는 값은 다음과 같습니다. PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H

 Note

버킷 이름과 접두사를 모두 포함하거나 둘 다 포함하지 마십시오.

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": { "stringValue": "{\"l4ETraining\": {\"trainingMode\":
  \"START\", \"exportDataStartTime\": StartTime, \"exportDataEndTime\": EndTime,
  \"targetSamplingRate\": \"TargetSamplingRate\", \"labelInputConfiguration\":
  {\"bucketName\": \"BucketName\", \"prefix\": \"Prefix\"}}}"
  }
}
```

4. 다음 명령을 실행하여 학습을 시작합니다.

```
aws iotsitewise execute-action --cli-input-json file://train-prediction.json
```

추론을 시작하려면 먼저 교육을 완료해야 합니다. 교육 상태를 확인하려면 다음 중 하나를 수행하십시오.

- 콘솔에서 예측이 적용되는 에셋으로 이동합니다.
- 에서 `AWS CLI propertyId trainingStatus` 프로퍼티의 `BatchGetAssetPropertyValue` 를 사용하여 호출합니다.

예측에 대한 추론 시작 또는 중지 (CLI)

예측이 학습되면 추론을 시작하여 Lookout for Equipment에서 자산 모니터링을 시작하도록 지시할 수 있습니다. 추론을 시작하거나 중지하려면 대상 `assetId` 리소스의 정보가 필요합니다.

추론을 시작하려면

1. 다음 명령을 실행하여 `assetModelCompositeModelId` `assetModelCompositeModelSummaries` 언더를 찾습니다. 에서 만든 에셋 모델의 `asset-model-id` ID로 [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#) 바꾸십시오.

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. 다음 명령을 실행하여 Inference 액션을 찾습니다. `actionDefinitionId` 이전 단계에서 사용한 `asset-model-id` ID로 바꾸고 이전 단계에서 반환된 `asset-model-composite-model-id` ID로 바꾸십시오.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. 라는 `start-inference.json` 파일을 만들고 다음 코드를 추가하여 다음을 대체합니다.

- `asset-id` 대상 자산의 ID로
- `action-definition-id` 추론 시작 액션의 ID와 함께
- `Offset` 사용할 버퍼의 양과 함께
- `Frequency` 데이터 업로드 빈도 포함

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": { "stringValue": "{ \"inferenceMode\": \"START \", \"dataDelayOffsetInMinutes\": Offset, \"dataUploadFrequency\": \"Frequency\" }" }
}
```

4. 다음 명령을 실행하여 추론을 시작합니다.

```
aws iotsitewise execute-action --cli-input-json file://start-inference.json
```

추론을 중지하려면

1. 다음 명령을 실행하여 `assetModelCompositeModelId` `assetModelCompositeModelSummaries` 언더를 찾으십시오. 에서 만든 에셋 모델의 `asset-model-id` ID로 [자산 또는 구성 요소 모델 업데이트 \(AWS CLI\)](#) 바꾸십시오.

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. 다음 명령을 실행하여 Inference 액션을 찾습니다. `actionDefinitionId` 이전 단계에서 사용한 `asset-model-id` ID로 바꾸고 이전 단계에서 반환된 `asset-model-composite-model-id` ID로 바꾸십시오.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. 라는 `stop-inference.json` 파일을 만들고 다음 코드를 추가하여 다음을 대체합니다.

- `asset-id` 대상 자산의 ID로
- `action-definition-id` 추론 시작 액션의 ID와 함께

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload":{ "stringValue": "{\"l4EInference\":{\"inferenceMode\":\"STOP\"}}"}
}
```

4. 다음 명령을 실행하여 추론을 중지합니다.

```
aws iotsitewise execute-action --cli-input-json file://stop-inference.json
```

데이터 스토리지 관리

다음 스토리지 계층에 데이터를 AWS IoT SiteWise 저장하도록 구성할 수 있습니다.

핫 티어

핫 스토리지 티어는 AWS IoT SiteWise 관리형 시계열 스토리지입니다. 핫 티어는 write-to-read 지연 시간이 짧고 자주 액세스하는 데이터에 가장 효과적입니다. 핫 티어에 저장된 데이터는 장비의 최신 측정값에 빠르게 액세스해야 하는 산업용 애플리케이션에서 사용됩니다. 여기에는 대화형 대시보드로 실시간 메트릭을 시각화하는 애플리케이션이나 운영을 모니터링하고 경보를 실행하여 성능 문제를 식별하는 애플리케이션이 포함됩니다.

기본적으로 수집된 데이터는 핫 AWS IoT SiteWise 티어에 저장됩니다. 핫 티어의 보존 기간을 정의한 후 구성에 따라 핫 티어의 데이터를 워م 티어 또는 콜드 티어 스토리지로 AWS IoT SiteWise 이동할 수 있습니다. 최상의 성능과 비용 효율성을 위해 핫 티어 보존 기간을 데이터를 자주 검색하는 데 걸리는 시간보다 길게 설정하십시오. 이는 실시간 지표, 경고 및 모니터링 시나리오에 사용됩니다. 보존 기간이 설정되지 않은 경우 데이터는 핫 티어에 무기한 저장됩니다.

웜 티어

웜 스토리지 티어는 기간별 데이터를 비용 효율적으로 저장하는 데 효과적인 AWS IoT SiteWise 관리형 계층입니다. write-to-read 지연 시간이 중간 정도인 대량의 데이터를 검색하는 데 가장 적합합니다. 웜 티어를 사용하면 대규모 워크로드에 필요한 기간별 데이터를 저장할 수 있습니다. 예를 들어 분석을 위한 데이터 검색, 비즈니스 인텔리전스 애플리케이션 (BI), 보고 도구, 기계 학습 (ML) 모델 교육에 사용됩니다. 콜드 스토리지 티어를 활성화하면 웜 티어 보존 기간을 정의할 수 있습니다. 보존 기간이 끝나면 웜 티어에서 데이터를 AWS IoT SiteWise 삭제합니다.

콜드 티어

콜드 스토리지 티어는 Amazon S3 버킷을 사용하여 거의 사용되지 않는 데이터를 저장합니다. 콜드 티어를 활성화하면 측정, 지표, 변환 및 집계, 자산 모델 정의를 포함한 시계열을 6시간마다 AWS IoT SiteWise 복제합니다. 콜드 티어는 기록 보고서 및 백업의 높은 읽기 지연 시간을 허용하는 데이터를 저장하는 데 사용됩니다.

주제

- [스토리지 설정 구성](#)
- [스토리지 설정 문제 해결](#)
- [콜드 티어에 저장된 데이터의 파일 경로 및 스키마](#)

스토리지 설정 구성

관리형 워م 티어 스토리지를 서비스하도록 선택하고 콜드 티어에 데이터를 복제하도록 스토리지 설정을 구성할 수 있습니다. 워م 티어와 핫 티어의 보존 기간에 대한 자세한 내용은 [데이터 보존에 미치는 영향](#)을 참조하십시오. 스토리지 설정을 구성하는 동안 다음을 수행하십시오.

- 핫 티어 보존 - 데이터가 삭제되고 스토리지 설정에 따라 서비스가 관리되는 워م 티어 스토리지 또는 콜드 티어 스토리지로 이동하기 전에 핫 티어에 데이터를 저장하는 보존 기간을 설정합니다. AWS IoT SiteWise 보존 기간이 끝나기 전에 존재했던 핫 티어의 모든 데이터를 삭제합니다. 보존 기간을 설정하지 않으면 데이터가 핫 티어에 무기한 저장됩니다.
- 워م 티어 보존 — 데이터를 AWS IoT SiteWise 스토리지에서 삭제하고 고객이 관리하는 콜드 티어 스토리지로 이동하기 전에 워م 티어에 데이터를 저장하는 기간을 설정합니다. AWS IoT SiteWise 보존 기간이 끝나기 전에 존재했던 모든 데이터를 워م 티어에서 삭제합니다. 보존 기간이 설정되지 않은 경우 데이터는 워م 티어에 무기한 저장됩니다.

Note

쿼리 성능을 개선하려면 워م 티어 스토리지로 핫 티어 보존 기간을 설정하십시오.

핫 티어 및 워م 티어 스토리지의 데이터 보존이 미치는 영향

- 핫 티어 스토리지의 보존 기간을 줄이면 데이터가 핫 티어에서 워م 티어 또는 콜드 티어로 영구적으로 이동합니다. 워م 티어의 보존 기간을 줄이면 데이터가 콜드 티어로 이동하고 워م 티어에서 영구적으로 삭제됩니다.
- 핫 티어 또는 워م 티어 스토리지의 보존 기간을 늘리면 그 AWS IoT SiteWise 이후부터 전송되는 데이터에도 변경 내용이 영향을 미칩니다. AWS IoT SiteWise 핫 티어를 채우기 위해 워م 스토리지나 콜드 스토리지에서 데이터를 검색하지 않습니다. 예를 들어 핫 티어 스토리지의 보존 기간을 처음에 30일로 설정했다가 60일로 늘린 경우 핫 티어 스토리지에 60일 분량의 데이터를 저장하는 데 30일이 걸립니다.

주제

- [웜 티어의 스토리지 설정을 구성합니다 \(콘솔\)](#).
- [웜 티어의 스토리지 설정을 구성합니다 \(AWS CLI\)](#)
- [콜드 티어의 스토리지 설정을 구성합니다 \(콘솔\)](#).

- [콜드 티어의 스토리지 설정을 구성합니다 \(AWS CLI\)](#).

웜 티어의 스토리지 설정을 구성합니다 (콘솔).

다음 절차는 AWS IoT SiteWise 콘솔의 웜 티어에 데이터를 복제하도록 스토리지 설정을 구성하는 방법을 보여줍니다.

콘솔을 사용하여 저장 설정을 구성하는 방법

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창의 설정 아래에서 저장을 선택합니다.
3. 오른쪽 상단 모서리에서 편집을 선택합니다.
4. 작업 편집 페이지에서 다음을 수행합니다.
5. 핫 티어 설정의 경우 다음을 수행하십시오.
 - 데이터를 삭제하고 서비스 관리형 웜 티어 스토리지로 이동하기 전에 핫 티어에 데이터를 저장하는 기간을 설정하려면 보존 기간 활성화를 선택합니다.
 - 보존 기간을 구성하려면 정수를 입력하고 단위를 선택합니다. 보존 기간은 30일 이상이어야 합니다.

AWS IoT SiteWise 핫 티어에서 보존 기간보다 오래된 모든 데이터를 삭제합니다. 보존 기간을 설정하지 않으면 데이터가 무기한 저장됩니다.
6. (권장) 웜 티어 설정의 경우 다음을 수행하십시오.
 - 웜 티어 스토리지를 선택하려면 웜 티어 스토리지 옵트인에 동의함을 선택하여 웜 티어 스토리지를 선택하도록 선택합니다.
 - (선택 사항) 보존 기간을 구성하려면 정수를 입력하고 단위를 선택합니다. 보존 기간은 365일 이상이어야 합니다.

AWS IoT SiteWise 보존 기간보다 이전에 존재했던 웜 티어의 데이터를 삭제합니다. 보존 기간을 설정하지 않으면 데이터가 무기한 저장됩니다.

Note

- 웜 티어를 선택하면 구성이 한 번만 표시됩니다.

- 핫 티어 보존을 설정하려면 워م 티어 또는 콜드 티어 스토리지가 있어야 합니다. 비용 효율성을 높이고 기간별 데이터를 검색하려면 AWS IoT SiteWise 워م 티어에 장기 데이터를 저장하는 것이 좋습니다.
- 워م 티어 보존을 설정하려면 콜드 티어 스토리지가 있어야 합니다.

7. 저장을 선택하여 스토리지 설정을 저장합니다.

AWS IoT SiteWise 스토리지 섹션에서 워م 티어 스토리지는 다음 상태 중 하나입니다.

- 활성화 — 데이터가 핫 티어 보존 기간 이전에 존재했다면 데이터를 워م 티어로 AWS IoT SiteWise 이 통합합니다.”
- 비활성화됨 - 워م 티어 스토리지가 비활성화됩니다.

웜 티어의 스토리지 설정을 구성합니다 (AWS CLI)

AWS CLI 및 다음 명령을 사용하여 데이터를 워م 티어로 이동하도록 스토리지 설정을 구성할 수 있습니다.

기존 구성을 재정의하지 않으려면 다음 명령을 실행하여 현재 스토리지 구성 정보를 검색하십시오.

```
aws iotsitewise describe-storage-configuration
```

Example 기존 콜드 티어 구성을 사용하지 않고 응답하십시오.

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-10-14T15:53:35-07:00",
  "warmTier": "DISABLED"
}
```

Example 기존 콜드 티어 구성을 통한 대응

```
{
  "storageType": "MULTI_LAYER_STORAGE",
```

```

    "multiLayerStorage": {
      "customerManagedS3Storage": {
        "s3ResourceArn": "arn:aws:s3:::bucket-name/prefix/",
        "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
      }
    },
    "disassociatedDataStorage": "ENABLED",
    "retentionPeriod": {
      "numberOfDays": retention-in-days
    },
    "configurationStatus": {
      "state": "ACTIVE"
    },
    "lastUpdateDate": "2023-10-25T15:59:46-07:00",
    "warmTier": "DISABLED"
  }

```

웜 티어의 스토리지 설정을 다음과 같이 구성합니다. AWS CLI

다음 명령을 실행하여 스토리지 설정을 구성합니다. AWS IoT SiteWise 스토리지 구성이 포함된 파일 이름으로 file-name 바꿉니다.

```
aws iotsitewise put-storage-configuration --cli-input-json file:///file-name.json
```

Example AWS IoT SiteWise 핫 티어와 웜 티어를 사용한 구성

```

{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "warmTier": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": hot-tier-retention-in-days
  }
}

```

hot-tier-retention-in-days 30일 이상의 정수여야 합니다.

Example 응답

```

{
  "storageType": "SITEWISE_DEFAULT_STORAGE",

```

```

    "configurationStatus": {
      "state": "UPDATE_IN_PROGRESS"
    }
  }
}

```

콜드 티어 스토리지를 활성화한 경우 을 참조하십시오 [기존 콜드 티어를 사용하여 AWS CLI 스토리지 설정을 구성합니다.](#)

기존 콜드 티어를 사용하여 AWS CLI 스토리지 설정을 구성합니다.

기존 콜드 티어 스토리지를 AWS CLI 사용하여 스토리지 설정을 구성합니다.

- 다음 명령을 실행하여 스토리지 설정을 구성합니다. *file-name*을 AWS IoT SiteWise 스토리지 구성에 포함된 파일의 이름으로 바꿉니다.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise 스토리지 구성

- *bucket-name*을 Amazon S3 버킷 이름으로 바꿉니다.
- *prefix*를 Amazon S3 접두사로 바꿉니다.
- AWS 계정 *aws-account-id*ID로 바꾸십시오.
- *## ###* Amazon S3로 데이터를 AWS IoT SiteWise 전송할 수 있는 Amazon S3 액세스 역할의 이름으로 대체하십시오.
- *hot-tier-retention-in-days# 30###* 크거나 같은 정수로 바꾸십시오.
- *warm-tier-retention-in-days# 365*일 이상의 정수로 바꾸십시오.

Note

AWS IoT SiteWise 콜드 티어의 보존 기간보다 오래된 워밍 티어의 모든 데이터를 삭제합니다. 보존 기간을 설정하지 않으면 데이터가 무기한 저장됩니다.

```

{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {

```

```

        "s3ResourceArn": "arn:aws:s3:::bucket-name/prefix/",
        "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    },
    "disassociatedDataStorage": "ENABLED",
    "retentionPeriod": {
        "numberOfDays": hot-tier-retention-in-days
    },
    "warmTier": "ENABLED",
    "warmTierRetentionPeriod": {
        "numberOfDays": warm-tier-retention-in-days
    }
}

```

Example 응답

```

{
    "storageType": "MULTI_LAYER_STORAGE",
    "configurationStatus": {
        "state": "UPDATE_IN_PROGRESS"
    }
}

```

콜드 티어의 스토리지 설정을 구성합니다 (콘솔).

다음 절차는 AWS IoT SiteWise 콘솔에서 콜드 티어에 데이터를 복제하도록 스토리지 설정을 구성하는 방법을 보여줍니다.

콘솔을 사용하여 저장 설정을 구성하는 방법

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창의 설정 아래에서 저장을 선택합니다.
3. 오른쪽 상단 모서리에서 편집을 선택합니다.
4. 작업 편집 페이지에서 다음을 수행합니다.
 - a. 스토리지 설정에서 콜드 티어 스토리지 활성화를 선택합니다. 기본적으로 콜드 티어 스토리지는 비활성화되어 있습니다.
 - b. S3 버킷 위치의 경우 기존 Amazon S3 버킷의 이름과 접두사를 입력합니다.

Note

- Amazon S3는 접두사를 Amazon S3 버킷의 폴더 이름으로 사용합니다. 접두사는 1~255자로 구성되어야 하며 슬래시(/)로 끝나야 합니다. AWS IoT SiteWise 데이터는 이 폴더에 저장됩니다.
- Amazon S3 버킷이 없는 경우에는 보기를 선택한 후 Amazon S3 콘솔에서 새 버킷을 생성합니다. 자세한 내용은 Amazon S3 사용 설명서에서 [첫 S3 버킷 생성](#)을 참조하세요.

c. S3 액세스 역할의 경우 다음 중 하나를 수행합니다.

- AWS 관리 템플릿에서 역할 생성을 선택하면 Amazon S3로 데이터를 AWS IoT SiteWise 전송할 수 있는 IAM 역할이 AWS 자동으로 생성됩니다.
- 기존 역할 사용을 선택하고 목록에서 생성한 에 역할을 선택합니다.

Note

- 이전 단계 및 IAM 정책에서 사용한 것과 동일한 Amazon S3 버킷 이름을 S3 버킷 위치에 사용해야 합니다.
- 역할에 다음 예에 표시된 것과 같은 권한이 있는지 확인합니다.

Example 권한 정책:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

```

    }
  ]
}

```

*bucket-name*을 Amazon S3 버킷 이름으로 바꿉니다.

- d. 핫 티어를 설정하려면 5단계를 참조하십시오 [웜 티어의 스토리지 설정을 구성합니다 \(콘솔\)](#).
- e. (선택 사항) AWS IoT Analytics 통합의 경우 다음을 수행합니다.
 - i. 데이터를 AWS IoT Analytics 쿼리하는 데 사용하려면 AWS IoT Analytics 데이터 저장소 활성화를 선택합니다.
 - ii. AWS IoT SiteWise 데이터 저장소의 이름을 생성하거나 다른 이름을 입력할 수 있습니다.

AWS IoT SiteWise 에서 데이터 저장소를 자동으로 AWS IoT Analytics 생성하여 데이터를 저장합니다. 를 사용하여 데이터셋을 AWS IoT Analytics 생성하여 데이터를 쿼리할 수 있습니다. 자세한 내용은 AWS IoT Analytics 사용 설명서의 AWS IoT SiteWise [데이터 작업을](#) 참조하세요.

- f. 저장을 선택합니다.

AWS IoT SiteWise 스토리지 섹션에서 콜드 티어 스토리지는 다음 값 중 하나일 수 있습니다.

- 활성화 — 데이터를 지정된 Amazon S3 버킷에 AWS IoT SiteWise 복제합니다.
- 활성화 - AWS IoT SiteWise 콜드 티어 스토리지를 활성화하기 위한 요청을 처리하는 것입니다. 이 프로세스는 완료하는 데 몇 분 정도 걸립니다.
- Enable_Failed — AWS IoT SiteWise 콜드 티어 스토리지 활성화 요청을 처리할 수 없습니다. Amazon CloudWatch Logs로 로그를 AWS IoT SiteWise 전송하도록 설정한 경우 이러한 로그를 사용하여 문제를 해결할 수 있습니다. 자세한 정보는 [Amazon CloudWatch 로그를 사용한 모니터링](#)을 참조하세요.
- Disabled- 콜드 티어 스토리지가 비활성화되었습니다.

콜드 티어의 스토리지 설정을 구성합니다 ()AWS CLI.

다음 절차에서는 AWS CLI를 사용하여 콜드 티어에 데이터를 복제하도록 스토리지 설정을 구성하는 방법을 보여줍니다.

를 사용하여 스토리지 설정을 구성하려면 AWS CLI

1. 계정의 Amazon S3 버킷으로 데이터를 내보내려면 다음 명령을 실행하여 스토리지 설정을 구성합니다. **## ###** AWS IoT SiteWise 스토리지 구성이 포함된 파일 이름으로 바꾸십시오.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise 스토리지 구성

- *bucket-name*을 Amazon S3 버킷 이름으로 바꿉니다.
- *prefix*를 Amazon S3 접두사로 바꿉니다.
- AWS 계정 *aws-account-id*ID로 바꾸십시오.
- **## ###** Amazon S3로 데이터를 AWS IoT SiteWise 전송할 수 있는 Amazon S3 액세스 역할의 이름으로 대체하십시오.
- 30일보다 크거나 같은 *retention-in-days*정수로 바꾸십시오.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3::bucket-name/prefix",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "retentionPeriod": {
    "numberOfDays": retention-in-days,
    "unlimited": false
  }
}
```

Note

- AWS IoT SiteWise 스토리지 구성 및 IAM 정책에서 동일한 Amazon S3 버킷 이름을 사용해야 합니다.
- 역할에 다음 예에 표시된 것과 같은 권한이 있는지 확인합니다.

Example 권한 정책:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

*bucket-name*을 Amazon S3 버킷 이름으로 바꿉니다.

Example 응답

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "retentionPeriod": {
    "numberOfDays": 100,
    "unlimited": false
  },
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

Note

스토리지 구성을 업데이트하는 데 몇 분 AWS IoT SiteWise 정도 걸릴 수 있습니다.

2. 스토리지 구성 정보를 검색하려면 다음 명령을 실행합니다.

```
aws iotsitewise describe-storage-configuration
```

Example 응답

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/torque/",
      "roleArn": "arn:aws:iam::123456789012:role/SWAccessS3Role"
    }
  },
  "retentionPeriod": {
    "numberOfDays": 100,
    "unlimited": false
  },
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:54:14-07:00"
}
```

3. Amazon S3 버킷으로 데이터 내보내기를 중지하려면 다음 명령을 실행하여 스토리지 설정을 구성합니다.

```
aws iotsitewise put-storage-configuration --storage-type SITEWISE_DEFAULT_STORAGE
```

Note

기본적으로 데이터는 핫 티어에만 저장됩니다 AWS IoT SiteWise.

Example 응답

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

4. 스토리지 구성 정보를 검색하려면 다음 명령을 실행합니다.

```
aws iotsitewise describe-storage-configuration
```

Example 응답

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:57:14-07:00"
}
```

(선택 사항) AWS IoT Analytics 데이터 저장소 생성 (AWS CLI)

AWS IoT Analytics 데이터 저장소는 데이터를 수신하고 저장하는 확장 가능하고 쿼리가 가능한 리포지토리입니다. AWS IoT SiteWise 콘솔 또는 AWS IoT Analytics API를 사용하여 데이터 저장소를 생성하여 AWS IoT Analytics 데이터를 저장할 수 있습니다. AWS IoT SiteWise 데이터를 쿼리하려면 를 사용하여 데이터세트를 생성합니다. AWS IoT Analytics 자세한 내용을 알아보려면 AWS IoT Analytics 사용 설명서의 [AWS IoT SiteWise 데이터를 사용하여 작업하기](#)를 참조하세요.

다음 단계는 에서 AWS IoT Analytics 데이터 저장소를 만드는 AWS CLI 데 사용됩니다.

다음 명령을 실행하여 데이터 스토어를 생성합니다. *file-name*을 데이터 스토어 구성이 포함된 파일의 이름으로 바꿉니다.

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

Note

- 기존 Amazon S3 버킷의 이름을 지정해야 합니다. Amazon S3 버킷이 없는 경우에는 먼저 하나를 만들 수 있습니다. 자세한 내용은 Amazon S3 사용 설명서에서 [첫 S3 버킷 생성](#)을 참조하세요.
- AWS IoT SiteWise 스토리지 구성, IAM 정책 및 AWS IoT Analytics 데이터 스토어 구성에서 동일한 Amazon S3 버킷 이름을 사용해야 합니다.

Example AWS IoT Analytics 데이터 스토어 구성

s3-## ### AWS IoT Analytics ### ## ## 및 Amazon S3 버킷 이름으로 *data-store-name* 바꾸십시오.

```
{
  "datastoreName": "data-store-name",
  "datastoreStorage": {
    "iotSiteWiseMultiLayerStorage": {
      "customerManagedS3Storage": {
        "bucket": "s3-bucket-name"
      }
    }
  },
  "retentionPeriod": {
    "numberOfDays": 90
  }
}
```

Example 응답

```
{
  "datastoreName": "datastore_IoTSiteWise_demo",
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/datastore_IoTSiteWise_demo",
  "retentionPeriod": {
    "numberOfDays": 90,
    "unlimited": false
  }
}
```

스토리지 설정 문제 해결

다음 정보를 사용하여 스토리지 구성 관련 문제를 해결하십시오.

문제

- [오류: 버킷이 존재하지 않습니다](#)
- [오류: Amazon S3 경로에 대한 액세스가 거부되었습니다.](#)
- [오류: 역할 ARN을 받을 수 없습니다](#)
- [오류: 리전 간 Amazon S3 버킷에 액세스하지 못했습니다.](#)

오류: 버킷이 존재하지 않습니다

해결 방법: AWS IoT SiteWise Amazon S3 버킷을 찾을 수 없습니다. 현재 리전의 기존 Amazon S3 버킷의 이름을 입력해야 합니다.

오류: Amazon S3 경로에 대한 액세스가 거부되었습니다.

해결 방법: AWS IoT SiteWise Amazon S3 버킷에 액세스할 수 없습니다. 다음을 따릅니다.

- IAM 정책에서 지정한 것과 동일한 Amazon S3 버킷을 사용해야 합니다.
- 역할에 다음 예에 표시된 것과 같은 권한이 있는지 확인합니다.

Example 권한 정책

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

*bucket-name*을 Amazon S3 버킷 이름으로 바꿉니다.

오류: 역할 ARN을 맡을 수 없습니다

해결 방법: AWS IoT SiteWise 사용자를 대신하여 IAM 역할을 맡을 수 없습니다. 역할이 다음 서비스를 신뢰하는지 확인합니다. [iotsitewise.amazonaws.com](#) 자세한 내용은 IAM 사용자 가이드의 [역할을 가정할 수 없습니다](#)를 참조하세요.

오류: 리전 간 Amazon S3 버킷에 액세스하지 못했습니다.

해결 방법: 지정한 Amazon S3 버킷이 다른 AWS 지역에 있습니다. Amazon S3 버킷과 AWS IoT SiteWise 자산이 같은 지역에 있는지 확인하십시오.

콜드 티어에 저장된 데이터의 파일 경로 및 스키마

AWS IoT SiteWise 측정, 지표, 변환 및 집계, 자산 및 자산 모델 정의를 포함한 시계열을 복제하여 데이터를 콜드 티어에 저장합니다. 다음은 콜드 티어로 전송되는 데이터의 파일 경로와 스키마를 설명합니다.

주제

- [장비 데이터\(측정값\)](#)
- [지표, 변환, 집계](#)
- [자산 메타데이터](#)
- [자산 계층 구조 메타데이터](#)
- [스토리지 데이터 인덱스 파일](#)

장비 데이터(측정값)

AWS IoT SiteWise 6시간마다 한 번씩 장비 데이터 (측정값) 를 콜드 티어로 내보냅니다. 원시 데이터는 [Apache AVRO\(.avro\)](#) 형식으로 콜드 티어에 저장됩니다.

파일 경로

AWS IoT SiteWise 다음 템플릿을 사용하여 장비 데이터 (측정값) 를 콜드 티어에 저장합니다.

```
{keyPrefix}/raw/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/
seriesBucket={seriesBucket}/raw_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Amazon S3의 원시 데이터에 대한 모든 파일 경로에는 다음과 같은 구성 요소가 포함되어 있습니다.

경로 구성 요소	설명
keyPrefix	AWS IoT SiteWise 스토리지 구성에서 지정한 Amazon S3 접두사. Amazon S3에서는 접두사를 버킷의 폴더 이름으로 사용합니다.
raw	장비의 시계열 데이터(측정값)를 저장하는 폴더. raw 폴더는 접두사 폴더에 저장됩니다.
seriesBucket	00~ff 사이의 16진수입니다. 이 숫자는 timeSeriesId 에서 파생되었습니다. 이 파티션은 콜드 티어에 AWS IoT SiteWise 쓸 때 처리량을 늘리는 데 사용됩니다. Amazon Athena를 사용하여 쿼리를 실행하면 세분화된 분할에 파티션을 사용하여 쿼리 성능을 개선할 수 있습니다. 자산 메타데이터의 seriesBucket 과 timeSeriesBucket 은 같은 숫자입니다.
startYear	시계열 데이터와 연결된 배타적 시작 시간의 연도.
startMonth	시계열 데이터와 연결된 배타적 시작 시간의 달.
startDay	시계열 데이터와 연결된 배타적 시작 시간의 달의 날짜.
fileName	파일 이름은 밑줄 (_)문자를 구분 기호로 사용하여 다음을 구분합니다.

경로 구성 요소	설명
	<ul style="list-style-type: none"> raw 접두사입니다. timeSeriesId 값입니다. 시계열 데이터와 관련된 배타적 시작 시간의 epoch 타임스탬프. 데이터 품질. 유효한 값: GOOD, BAD, 및 UNCERTAIN . 자세한 내용은 AWS IoT SiteWise API 참조의 AssetProperty 값을 참조하십시오. <p>파일은 Snappy 압축을 사용하여 .avro 형식으로 저장됩니다.</p>

Example 콜드 티어의 원시 데이터 파일 경로

```
keyPrefix/raw/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/
raw_7020c8e2-e6db-40fa-9845-ed0dd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1609577700_G00D.avro
```

필드

콜드 티어로 내보내는 원시 데이터의 스키마에는 다음 필드가 포함됩니다.

필드 이름	지원되는 유형	기본 형식	설명
seriesId	string	N/A	장비의 시계열 데이터 (측정값)를 식별하는 ID. 이 필드를 사용하여 쿼리에서 원시 데이터와 자산 메타데이터를 결합할 수 있습니다.
timeInSeconds	long	N/A	Unix epoch 형식의 타임스탬프 날짜(초)입니다. offsetInN

필드 이름	지원되는 유형	기본 형식	설명
			anos 에서 나노초 단위의 데이터를 제공합니다.
offsetInNanos	long	N/A	timeInSeconds 의 나노초 오프셋.
quality	string	N/A	시계열 값의 품질.
doubleValue	double 또는 null	null	더블 유형의 시계열 데이터(이동 소숫점 수).
stringValue	string 또는 null	null	문자열 유형의 시계열 데이터(문자 시퀀스).
integerValue	int 또는 null	null	정수 유형의 시계열 데이터(정수).
booleanValue	boolean 또는 null	null	부울 유형의 시계열 데이터(true 또는 false).
jsonValue	string 또는 null	null	JSON 유형의 시계열 데이터(문자열로 저장된 복잡한 데이터 유형).
recordVersion	long 또는 null	null	기록을 위한 버전 번호. 버전 번호를 사용하여 최신 레코드를 선택할 수 있습니다. 최신 레코드는 버전 번호가 더 큼니다.

Example 콜드 티어의 원시 데이터

```
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675887,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {
    "double": 0.75,
    "stringValue": null,
    "integerValue": null,
    "booleanValue": null,
    "jsonValue": null,
    "re": {
      "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
      "timeInSeconds": 1625675889,
      "offsetInNanos": 0,
      "quality": "GOOD",
      "doubleValue": {
        "double": 0.69,
        "stringValue": null,
        "integerValue": null,
        "booleanValue": null,
        "jsonValue": null,
        "re": {
          "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
          "timeInSeconds": 1625675890,
          "offsetInNanos": 0,
          "quality": "GOOD",
          "doubleValue": {
            "double": 0.66,
            "stringValue": null,
            "integerValue": null,
            "booleanValue": null,
            "jsonValue": null,
            "re": {
              "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
              "timeInSeconds": 1625675891,
              "offsetInNanos": 0,
              "quality": "GOOD",
              "doubleValue": {
                "double": 0.92,
                "stringValue": null,
                "integerValue": null,
                "booleanValue": null,
                "jsonValue": null,
                "re": {
                  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
                  "timeInSeconds": 1625675892,
                  "offsetInNanos": 0,
                  "quality": "GOOD",
                  "doubleValue": {
                    "double": 0.73,
                    "stringValue": null,
                    "integerValue": null,
                    "booleanValue": null,
                    "jsonValue": null,
                    "re": {

```

지표, 변환, 집계

AWS IoT SiteWise 6시간마다 한 번씩 메트릭을 내보내고, 변환하고, 집계를 콜드 티어로 내보냅니다. 지표, 변환 및 집계는 [Apache AVRO](#)(.avro) 형식으로 콜드 티어에 저장됩니다.

파일 경로

AWS IoT SiteWise 다음 템플릿을 사용하여 지표, 변환 및 집계를 콜드 티어에 저장합니다.

```
{keyPrefix}/agg/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/seriesBucket={seriesBucket}/agg_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Amazon S3의 지표, 변환 및 집계에 대한 모든 파일 경로에는 다음 구성 요소가 포함되어 있습니다.

경로 구성 요소	설명
keyPrefix	AWS IoT SiteWise 스토리지 구성에서 지정한 Amazon S3 접두사. Amazon S3에서는 접두사를 버킷의 폴더 이름으로 사용합니다.
agg	지표의 시계열 데이터를 저장하는 폴더. agg 폴더는 접두사 폴더에 저장됩니다.

경로 구성 요소	설명
seriesBucket	<p>00~ff 사이의 16진수입니다. 이 숫자는 <code>timeSeriesId</code> 에서 파생되었습니다. 이 파티션은 콜드 티어에 AWS IoT SiteWise 쓸 때 처리량을 늘리는 데 사용됩니다. Amazon Athena를 사용하여 쿼리를 실행하면 세분화된 분할에 파티션을 사용하여 쿼리 성능을 개선할 수 있습니다.</p> <p>자산 메타데이터의 <code>seriesBucket</code> 과 <code>timeSeriesBucket</code> 은 같은 숫자입니다.</p>
startYear	시계열 데이터와 연결된 배타적 시작 시간의 연도.
startMonth	시계열 데이터와 연결된 배타적 시작 시간의 달.
startDay	시계열 데이터와 연결된 배타적 시작 시간의 달의 날짜.
fileName	<p>파일 이름은 밑줄 (_)문자를 구분 기호로 사용하여 다음을 구분합니다.</p> <ul style="list-style-type: none"> raw 접두사입니다. <code>timeSeriesId</code> 값입니다. 시계열 데이터와 관련된 배타적 시작 시간의 epoch 타임스탬프. 데이터 품질. 유효한 값: GOOD, BAD, 및 UNCERTAIN . 자세한 내용은 AWS IoT SiteWise API 참조의 AssetProperty 값을 참조하십시오. <p>파일은 Snappy 압축을 사용하여 .avro 형식으로 저장됩니다.</p>

Example 콜드 티어의 지표에 대한 파일 경로

```
keyPrefix/agg/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/agg_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a_1609577700_G00D.avro
```

필드

콜드 티어로 내보내는 지표, 변환 및 집계기의 스키마에는 다음 필드가 포함됩니다.

필드 이름	지원되는 유형	기본 형식	설명
seriesId	string	N/A	장비, 지표 또는 변환의 시계열 데이터를 식별하는 ID. 이 필드를 사용하여 쿼리에서 원시 데이터와 자산 메타 데이터를 결합할 수 있습니다.
timeInSeconds	long	N/A	Unix epoch 형식의 타임스탬프 날짜(초)입니다. offsetInNanos 에서 나노초 단위의 데이터를 제공합니다.
offsetInNanos	long	N/A	timeInSeconds 의 나노초 오프셋.
quality	string	N/A	자산 데이터를 필터링하는 데 사용되는 품질.
resolution	string	N/A	데이터를 집계하는 데 걸리는 시간 간격.

필드 이름	지원되는 유형	기본 형식	설명
count	double 또는 null	null	현재 시간 간격 동안 해당 변수에 대한 총 데이터 포인트 수.
average	double 또는 null	null	현재 시간 간격 동안 해당 변수 값의 평균.
min	double 또는 null	null	현재 시간 간격 동안 해당 변수 값의 최소 값.
max	boolean 또는 null	null	현재 시간 간격 동안 해당 변수 값의 최대 값.
sum	string 또는 null	null	현재 시간 간격 동안 해당 변수 값의 합계.
recordVersion	long 또는 null	null	기록을 위한 버전 번호. 버전 번호를 사용하여 최신 레코드를 선택할 수 있습니다. 최신 레코드는 버전 번호가 더 큼니다.

Example 콜드 티어의 지표 데이터

```
{
  "seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334060,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "1s",
  "double": 16.0,
  "min": {
    "double": 1.0
  },
  "max": {
    "double": 31.0
  },
  "sum": {
    "double": 496.0
  },
  "recordVersion": null
}
{
  "seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334120,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "1s",
  "double": 46.0,
  "min": {
    "double": 32.0
  },
  "max": {
    "double": 60.0
  },
  "sum": {
    "double": 1334.0
  },
  "recordVersion": null
}
{
  "seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334540,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "1s",
  "double": 100.0,
  "min": {
    "double": 50.0
  },
  "max": {
    "double": 150.0
  },
  "sum": {
    "double": 3000.0
  },
  "recordVersion": null
}
```

```

{"double":16.0}, "min":{"double":1.0}, "max":{"double":31.0}, "sum":
{"double":496.0}, "recordVersion":null}
  {"seriesId":"f74c2828-5317-4df3-
ba16-6d41b5bcb531", "timeInSeconds":1637334600, "offsetInNanos":0, "quality":"GOOD", "resolution":
{"double":46.0}, "min":{"double":32.0}, "max":{"double":60.0}, "sum":
{"double":1334.0}, "recordVersion":null}
  {"seriesId":"f74c2828-5317-4df3-
ba16-6d41b5bcb531", "timeInSeconds":1637335020, "offsetInNanos":0, "quality":"GOOD", "resolution":
{"double":16.0}, "min":{"double":1.0}, "max":{"double":31.0}, "sum":
{"double":496.0}, "recordVersion":null}

```

자산 메타데이터

처음으로 콜드 티어로 데이터를 AWS IoT SiteWise 내보내는 것을 활성화하면 자산 메타데이터가 콜드 티어로 내보내집니다. 초기 구성 후에는 자산 모델 정의 또는 자산 정의를 변경하는 경우에만 자산 메타데이터를 계층으로 AWS IoT SiteWise 내보냅니다. 에셋 메타데이터는 줄바꿈으로 구분된 JSON () 형식으로 콜드 티어에 저장됩니다. .ndjson

파일 경로

AWS IoT SiteWise 다음 템플릿을 사용하여 자산 메타데이터를 콜드 티어에 저장합니다.

```
{keyPrefix}/asset_metadata/asset_{assetId}.ndjson
```

콜드 티어의 자산 메타데이터에 대한 모든 파일 경로에는 다음 구성 요소가 포함됩니다.

경로 구성 요소	설명
keyPrefix	스토리지 구성에서 지정한 Amazon S3 AWS IoT SiteWise 접두사입니다. Amazon S3에서는 접두사를 버킷의 폴더 이름으로 사용합니다.
asset_metadata	자산 메타데이터를 저장하는 폴더. asset_metadata 폴더는 접두사 폴더에 저장됩니다.
fileName	파일 이름은 밑줄 (_) 문자를 구분 기호로 사용하여 다음을 구분합니다. <ul style="list-style-type: none"> asset 접두사입니다. assetId 값입니다.

경로 구성 요소	설명
	파일은 .ndjson 형식으로 저장됩니다.

Example 더 콜드 티어의 자산 메타데이터에 대한 파일 경로

keyPrefix/asset_metadata/asset_35901915-d476-4dca-8637-d9ed4df939ed.ndjson

필드

콜드 티어로 내보내는 자산 메타데이터의 스키마에는 다음 필드가 포함됩니다.

필드 이름	설명
assetId	자산의 ID입니다.
assetName	자산 이름.
assetExternalId	자산의 외부 ID.
assetModelId	자산을 생성할 때 사용된 자산 모델의 ID입니다.
assetModelName	자산 모델의 이름입니다.
assetModelExternalId	에셋 모델의 외부 ID.
assetPropertyId	자산 속성의 ID입니다.
assetPropertyName	자산 특성의 이름입니다.
assetPropertyExternalId	자산 자산의 외부 ID.
assetPropertyDataType	자산 속성의 데이터 유형입니다.
assetPropertyUnit	자산 속성의 단위(예: Newtons 및 RPM).
assetPropertyAlias	OPC-UA 서버 데이터 스트림 경로(예: /company/windfarm/3/turbine/7/temperature)와 같이 속성을 식별하는 별칭입니다.

필드 이름	설명
timeSeriesId	장비, 지표 또는 변환의 시계열 데이터를 식별하는 ID. 이 필드를 사용하여 쿼리에서 원시 데이터와 자산 메타데이터를 결합할 수 있습니다.
timeSeriesBucket	00~ff 사이의 16진수입니다. 이 숫자는 timeSeriesId 에서 파생되었습니다. 이 파티션은 콜드 티어에 AWS IoT SiteWise 쓸 때 처리량을 높이는 데 사용됩니다. Amazon Athena를 사용하여 쿼리를 실행하면 세분화된 분할에 파티션을 사용하여 쿼리 성능을 개선할 수 있습니다. 원시 데이터에 대한 파일 경로의 timeSeriesBucket 와 seriesBucket 는 같은 숫자입니다.
assetCompositeModelId	복합 모델의 ID.
assetCompositeModelExternalId	복합 모델의 외부 ID.
assetCompositeModelDescription	복합 모델에 대한 설명입니다.
assetCompositeModelName	복합 모델의 이름입니다.
assetCompositeModelType	복합 모델의 유형입니다. 경보 복합 모델의 경우 이 유형은 AWS/ALARM 입니다.
assetCreationDate	UNIX epoch 시간 형식으로 표시된 자산이 생성된 날짜.
assetLastUpdateDate	Unix Epoch 시간으로 표시된 자산이 마지막으로 업데이트된 날짜.
assetStatusErrorCode	오류 코드입니다.
assetStatusErrorMessage	오류 메시지입니다.
assetStatusState	자산의 현재 상태입니다.

계층 메타데이터를 콜드 티어로 AWS IoT SiteWise 내보냅니다. 에셋 계층 메타데이터는 줄바꿈으로 구분된 JSON () 형식으로 콜드 티어에 저장됩니다. .ndjson

API를 호출하여 계층, 대상 자산 또는 소스 자산의 외부 식별자를 검색합니다. [DescribeAsset](#)

파일 경로

AWS IoT SiteWise 다음 템플릿을 사용하여 자산 계층 메타데이터를 콜드 티어에 저장합니다.

```
{keyPrefix}/asset_hierarchy_metadata/{parentAssetId}_{hierarchyId}.ndjson
```

콜드 티어의 자산 계층 구조 메타데이터에 대한 모든 파일 경로에는 다음 구성 요소가 포함됩니다.

경로 구성 요소	설명
keyPrefix	AWS IoT SiteWise 스토리지 구성에서 지정한 Amazon S3 접두사. Amazon S3에서는 접두사를 버킷의 폴더 이름으로 사용합니다.
asset_hierarchy_metadata	자산 계층 구조 메타데이터를 저장하는 폴더. asset_hierarchy_metadata 폴더는 접두사 폴더에 저장됩니다.
fileName	파일 이름은 밑줄 (_)문자를 구분 기호로 사용하여 다음을 구분합니다. <ul style="list-style-type: none"> parentAssetId 값입니다. hierarchyId 값입니다. <p>파일은 .ndjson 형식으로 저장됩니다.</p>

Example 콜드 티어의 자산 계층 구조 메타데이터에 대한 파일 경로

```
keyPrefix/asset_hierarchy_metadata/35901915-d476-4dca-8637-d9ed4df939ed_c5b3ced8-589a-48c7-9998-cdcccfc9747a0.ndjson
```

필드

콜드 티어로 내보내는 자산 계층 구조 메타데이터의 스키마에는 다음 필드가 포함됩니다.

필드 이름	설명
sourceAssetId	이 자산 관계에 있는 소스 자산의 ID.
targetAssetId	이 자산 관계에 있는 대상 자산의 ID.
hierarchyId	해당 계층 구조의 ID.
associationType	이 자산 관계의 연결 유형. 값은 CHILD여야 합니다. 대상 자산은 소스 자산의 하위 자산입니다.

Example 콜드 티어의 자산 계층 구조 메타데이터

```
{
  "sourceAssetId": "80388e72-2284-44fb-9c89-bfbaf0dfedd2",
  "targetAssetId": "2b866c25-0c74-4750-bdf5-b73683c8a2a2",
  "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee",
  "associationType": "CHILD"
}
{
  "sourceAssetId": "80388e72-2284-44fb-9c89-bfbaf0dfedd2",
  "targetAssetId": "6b51246e-984d-460d-bc0b-470ea47d1e31",
  "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee",
  "associationType": "CHILD"
}
```

콘솔 티어에서 데이터를 보려면

1. [Amazon S3 콘솔](#)로 이동합니다.
2. 탐색 패널에서, 버킷을 선택하고 Amazon S3 버킷을 선택합니다.
3. 원시 데이터, 자산 메타데이터 또는 자산 계층 구조 메타데이터가 포함된 폴더로 이동합니다.
4. 파일을 선택한 다음 작업에서 다운로드를 선택합니다.

스토리지 데이터 인덱스 파일

AWS IoT SiteWise 이러한 파일을 사용하여 데이터 쿼리 성능을 최적화합니다. Amazon S3 버킷에 표시되지만 사용할 필요는 없습니다.

파일 경로

AWS IoT SiteWise 다음 템플릿을 사용하여 콜드 티어에 데이터 인덱스 파일을 저장합니다.

```
keyPrefix/index/series=timeseriesId/startYear=startYear/startMonth=startMonth/  
startDay=startDay/index_timeseriesId_startTimestamp_quality
```

Example 데이터 스토리지 인덱스 파일의 파일 경로

```
keyPrefix/index/series=7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-  
d34e-43e1-bc6f-1b490154b07a/startYear=2022/startMonth=02/startDay=03/  
index_7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-  
bc6f-1b490154b07a_1643846400_G00D
```

보안 입력 AWS IoT SiteWise

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

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

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 적용되는 규정 준수 프로그램에 대해 자세히 알아보려면 규정 준수 [프로그램별 범위 내 AWS 서비스 규정 준수 프로그램별](#) 참조하십시오. AWS IoT SiteWise
- 클라우드에서의 보안 — 사용하는 AWS 서비스에 따라 책임이 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

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

주제

- [데이터 보호: AWS IoT SiteWise](#)
- [데이터 암호화](#)
- [ID 및 액세스 관리: AWS IoT SiteWise](#)
- [규정 준수 검증: AWS IoT SiteWise](#)
- [의 레질리언스 AWS IoT SiteWise](#)
- [의 인프라 보안 AWS IoT SiteWise](#)
- [구성 및 취약성 분석](#)
- [VPC 엔드포인트](#)
- [에 대한 보안 모범 사례 AWS IoT SiteWise](#)

데이터 보호: AWS IoT SiteWise

AWS [공동 책임 모델](#)의 AWS IoT SiteWise 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS는 모든 데이터를 실행하는 글로벌 인프라를 보호하는 역할을 AWS 클라우드합니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시 FAQ](#)를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하세요.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS IAM Identity Center OR AWS Identity and Access Management (IAM)을 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 리소스와 통신할 수 있습니다. AWS TLS 1.2는 필수이며 TLS 1.3를 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다. AWS CloudTrail
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 AWS 통해 액세스할 때 FIPS 140-2로 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용하십시오. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [FIPS\(Federal Information Processing Standard\) 140-2](#)를 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API AWS IoT SiteWise 또는 AWS 서비스 SDK를 사용하거나 다른 방법으로 작업하는 경우가 포함됩니다. AWS CLI AWS 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함해서는 안 됩니다.

주제

- [인터넷워크 트래픽 개인 정보](#)

인터넷워크 트래픽 개인 정보

SiteWise 엣지 게이트웨이와 같은 온-프레미스 애플리케이션 간 AWS IoT SiteWise 연결은 TLS (전송 계층 보안) 연결을 통해 보호됩니다. 자세한 정보는 [전송 중 암호화](#)를 참조하세요.

AWS IoT SiteWise AWS 지역 내 가용 영역 간 연결 또는 계정 간 연결은 지원하지 않습니다. AWS 한 번에 한 리전에서 IAM Identity Center를 구성할 수 있습니다. SiteWise 모니터는 IAM Identity Center용으로 구성된 지역에 연결됩니다. 즉, IAM Identity Center 액세스에는 하나의 리전을 사용하지만 어느 리전에서든 포털을 생성할 수 있습니다.

데이터 암호화

데이터 암호화란 전송 중 (SiteWise 엣지 게이트웨이와 서버 간 이동) 및 저장 중 (로컬 장치 또는 서비스에 저장되는 동안) 을 보호하는 것을 말합니다. AWS IoT SiteWise AWS 전송 계층 보안(TLS)을 사용하여 전송 중인 데이터를 보호하거나 클라이언트 측 암호화를 사용하여 유휴 상태인 데이터를 보호할 수 있습니다.

Note

AWS IoT SiteWise 엣지 프로세싱은 SiteWise 엣지 게이트웨이 내에 호스팅되고 로컬 네트워크를 통해 액세스할 수 있는 API를 노출합니다. 이러한 API는 엣지 커넥터가 소유한 서버 인증서를 기반으로 하는 TLS 연결을 통해 노출됩니다. AWS IoT SiteWise 클라이언트 인증의 경우 이러한 API는 액세스 제어 암호를 사용합니다. 서버 인증서 개인 키와 액세스 제어 암호는 모두 디스크에 저장됩니다. AWS IoT SiteWise 엣지 프로세싱은 저장된 이러한 자격 증명의 보안을 위해 파일 시스템 암호화를 사용합니다.

서버 측 암호화 및 클라이언트 측 암호화에 대한 자세한 내용은 아래 나열된 주제를 검토하십시오.

주제

- [저장 중 암호화](#)
- [전송 중 암호화](#)
- [키 관리](#)

저장 중 암호화

AWS IoT SiteWise 데이터를 AWS 클라우드와 AWS IoT SiteWise 엣지 게이트웨이에 저장합니다.

클라우드에 AWS 저장된 데이터

AWS IoT SiteWise 기본적으로 저장된 데이터를 암호화하는 다른 AWS 서비스에 데이터를 저장합니다. 저장 중 암호화는 자산 속성 값 및 집계 값을 암호화하는 데 사용되는 암호화 키를 관리하는 AWS Key Management Service (AWS KMS) 와 통합됩니다. AWS IoT SiteWise에서 자산 속성 값과 집계 값을 암호화하기 위해 고객 관리형 키를 사용하도록 선택할 수 있습니다. AWS KMS를 통해 KMS 키 암호화 키를 직접 생성하고 관리할 수 있습니다.

데이터를 암호화하는 AWS 소유 키 방법을 선택하거나 자산 자산 가치와 집계 값을 암호화하는 고객 관리 키를 선택할 수 있습니다.

작동 방식

저장 중 암호화는 데이터 암호화에 사용되는 암호화 키 관리를 AWS KMS 위한 기능과 통합됩니다.

- AWS 소유 키 — 기본 암호화 키. AWS IoT SiteWise 이 키를 소유합니다. AWS 계정에서는 이 키를 볼 수 없습니다. 또한 AWS CloudTrail 로그에서 키에 대한 작업을 볼 수 없습니다. 추가 비용 없이 이 키를 사용할 수 있습니다.
- 고객 관리형 키 - 키는 사용자가 생성, 소유 및 관리하는 계정에 저장됩니다. KMS 키를 완전히 제어할 수 있습니다. 추가 AWS KMS 요금이 적용됩니다.

AWS 소유 키

AWS 소유 키 계정에는 저장되지 않습니다. 여러 AWS 계정에서 사용할 수 있도록 AWS 소유하고 관리하는 KMS 키 컬렉션의 일부입니다. AWS 서비스는 데이터를 보호하는 AWS 소유 키 데 사용할 수 있습니다.

사용을 보거나, 관리하거나, 사용하거나 AWS 소유 키, 감사할 수 없습니다. 하지만 사용자는 데이터를 암호화하는 키를 보호하기 위해 어떤 작업을 수행하거나 어떤 프로그램을 변경할 필요가 없습니다.

사용할 AWS 소유 키경우 월별 요금이나 사용 요금이 청구되지 않으며 계정 AWS KMS 할당량에도 포함되지 않습니다.

고객 관리형 키

고객 관리형 키는 사용자가 생성, 소유 및 관리하는 계정의 KMS 키입니다. 다음과 같애 | KMS 키를 완전히 제어할 수 있습니다.

- 키 정책, IAM 정책 및 권한 부여 수립 및 유지
- 활성화 및 비활성화

- 암호화 자료 교체
- 태그 추가
- 이를 참조하는 별칭 생성
- KMS 키 삭제 예약

또한 Amazon CloudWatch Logs를 사용하여 CloudTrail 사용자를 대신하여 AWS IoT SiteWise 보내는 요청을 추적할 AWS KMS 수 있습니다.

고객 관리 키를 사용하는 경우 계정에 저장된 KMS 키에 AWS IoT SiteWise 대한 액세스 권한을 부여해야 합니다. AWS IoT SiteWise 봉투 암호화와 키 계층 구조를 사용하여 데이터를 암호화합니다. AWS KMS 암호화 키는 이 키 계층 구조의 루트 키를 암호화하는 데 사용됩니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 [봉투 암호화](#)를 참조하세요.

다음 예제 정책은 사용자 대신 고객 관리 키를 생성할 수 있는 AWS IoT SiteWise 권한을 부여합니다. 키를 생성할 때는 kms:CreateGrant 및 kms:DescribeKey 작업을 허용해야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

생성한 권한 부여의 암호화 컨텍스트는 사용자 aws:iotsitewise:subscriberId 및 계정 ID를 사용합니다.

SiteWise 에지 게이트웨이의 미사용 데이터

AWS IoT SiteWise 게이트웨이는 로컬 파일 시스템에 다음 데이터를 저장합니다.

- OPC-UA 소스 구성 정보

- 연결된 OPC-UA 소스에서 오는 OPC-UA 데이터 스트림 경로 집합
- SiteWise 에지 게이트웨이의 인터넷 연결이 끊어지면 산업 데이터가 캐시됩니다.

SiteWise 에지 게이트웨이는 계속 실행됩니다. AWS IoT Greengrass AWS IoT Greengrass Unix 파일 권한 및 전체 디스크 암호화 (활성화된 경우) 를 사용하여 코어에 저장된 데이터를 보호합니다. 파일 시스템 및 디바이스의 보안을 유지하는 것은 사용자의 책임입니다.

AWS IoT Greengrass 하지만 Secrets Manager에서 검색한 OPC-UA 서버 암호의 로컬 사본은 암호화합니다. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 [암호 암호화](#)를 참조하세요.

AWS IoT Greengrass 코어의 저장 중 암호화에 대한 자세한 내용은 개발자 안내서의 저장 중 [암호화](#)를 참조하십시오. AWS IoT Greengrass Version 1

전송 중 암호화

AWS IoT SiteWise 데이터가 전송되는 세 가지 통신 모드가 있습니다.

- [인터넷을 통한](#) — 로컬 장치 (SiteWise 에지 게이트웨이 포함) 간의 통신이 암호화되며 AWS IoT SiteWise 암호화됩니다.
- [로컬 네트워크를 통한](#) — SiteWise 애플리케이션과 SiteWise 에지 게이트웨이 간의 OpsHub 통신은 항상 암호화됩니다. 브라우저에서 실행되는 SiteWise 모니터 애플리케이션과 SiteWise Edge 게이트웨이 간의 통신은 항상 암호화됩니다. SiteWise 에지 게이트웨이와 OPC-UA 소스 간의 통신을 암호화할 수 있습니다.
- 에지 게이트웨이의 [구성 요소 간 — SiteWise 에지 게이트웨이의](#) AWS IoT Greengrass 구성 요소 간 통신은 SiteWise 암호화되지 않습니다.

주제

- [인터넷을 통해 전송 중인 데이터](#)
- [로컬 네트워크를 통해 전송 중인 데이터](#)
- [SiteWise 에지 게이트웨이의 로컬 구성 요소 간에 전송되는 데이터](#)

인터넷을 통해 전송 중인 데이터

AWS IoT SiteWise 전송 계층 보안 (TLS) 을 사용하여 인터넷을 통한 모든 통신을 암호화합니다. AWS 클라우드로 전송되는 모든 데이터는 MQTT 또는 HTTPS 프로토콜을 사용하는 TLS 연결을 통해 전송되므로 기본적으로 안전합니다. SiteWise 에서 AWS IoT Greengrass 실행되는 에지 게이트웨이와 속성

값 알림은 전송 보안 모델을 사용합니다. AWS IoT 자세한 정보는 AWS IoT 개발자 안내서의 [전송 보안](#)을 참조하세요.

로컬 네트워크를 통해 전송 중인 데이터

SiteWise 에지 게이트웨이는 로컬 OPC-UA 소스와의 통신을 위한 OPC-UA 사양을 따릅니다. 사용자는 메시지 보안 모드를 사용하여 전송 중인 데이터를 암호화하도록 소스를 구성할 책임이 있습니다.

서명 메시지 보안 모드를 선택하면 SiteWise 에지 게이트웨이와 소스 간에 전송되는 데이터가 서명되지만 암호화되지는 않습니다. 서명 및 암호화 메시지 보안 모드를 선택하면 SiteWise 에지 게이트웨이와 소스 간에 전송되는 데이터가 서명되고 암호화됩니다. 소스 구성에 대한 자세한 내용은 [데이터 소스 구성](#)을 참조하세요.

에지 콘솔 애플리케이션과 SiteWise 에지 게이트웨이 간의 통신은 항상 TLS로 암호화됩니다.

SiteWise 에지 게이트웨이의 SiteWise 에지 커넥터는 애플리케이션용 에지 콘솔과의 TLS 연결을 설정할 수 있도록 자체 서명된 인증서를 생성하고 저장합니다. AWS IoT SiteWise 애플리케이션을 에지 게이트웨이에 연결하기 전에 SiteWise 에지 게이트웨이에서 AWS IoT SiteWise 애플리케이션용 에지 콘솔로 이 인증서를 복사해야 합니다 SiteWise . 이렇게 하면 AWS IoT SiteWise 애플리케이션용 에지 콘솔이 신뢰할 수 있는 SiteWise 에지 게이트웨이에 연결되었는지 확인할 수 있습니다.

보안 및 서버 인증을 위한 TLS 외에도 SiteWise Edge는 SigV4 프로토콜을 사용하여 에지 콘솔 애플리케이션의 신뢰성을 설정합니다. SiteWise Edge Gateway의 SiteWise Edge 커넥터는 에지 콘솔 애플리케이션, 브라우저 내에서 실행되는 SiteWise 모니터링 애플리케이션 및 SDK를 기반으로 하는 기타 클라이언트에서 들어오는 연결을 확인할 수 있도록 비밀번호를 수락하고 저장합니다. AWS IoT SiteWise

비밀번호 및 서버 인증서 생성에 대한 자세한 내용은 [the section called “ SiteWise 에지 게이트웨이 관리”](#)을 참조하세요.

SiteWise 에지 게이트웨이의 로컬 구성 요소 간에 전송되는 데이터

SiteWise 에서 AWS IoT Greengrass 실행되는 엣지 게이트웨이는 데이터가 기기 외부로 유출되지 않기 때문에 AWS IoT Greengrass 코어에서 로컬로 교환되는 데이터를 암호화하지 않습니다. 여기에는 커넥터와 같은 AWS IoT Greengrass 구성 요소 간 통신이 포함됩니다. AWS IoT SiteWise 자세한 정보는 AWS IoT Greengrass Version 1 개발자 안내서의 [코어 디바이스의 데이터](#)를 참조하세요.

키 관리

AWS IoT SiteWise 클라우드 키 관리

기본적으로 AWS 클라우드의 데이터를 보호하는 AWS 관리형 키 데 AWS IoT SiteWise 사용합니다. 설정을 업데이트하여 고객 관리형 키를 사용하여 AWS IoT SiteWise에서 일부 데이터를 암호화할 수

있습니다. AWS Key Management Service (AWS KMS)을 통해 암호화 키를 만들고, 관리하고, 볼 수 있습니다.

AWS IoT SiteWise 다음 데이터를 암호화하기 AWS KMS 위해 고객 관리 키를 저장하여 서버 측 암호화를 지원합니다.

- 자산 속성 값
- 값 집계

Note

기타 데이터 및 리소스는 에서 관리하는 키를 사용한 기본 암호화를 사용하여 암호화됩니다. AWS IoT SiteWise이 키는 AWS IoT SiteWise 계정에 저장됩니다.

자세한 내용은 [What is AWS Key Management Service?](#) 를 참조하십시오. AWS Key Management Service 개발자 안내서에서

고객 관리형 키를 사용하여 암호화를 활성화합니다.

에서 고객 관리 키를 사용하려면 AWS IoT SiteWise 설정을 업데이트해야 합니다. AWS IoT SiteWise KMS 키를 사용하여 암호화를 활성화하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 계정 설정을 선택하고 편집을 선택하여 계정 설정 편집 페이지를 엽니다.
3. 암호화 키 유형에서 다른 AWS KMS 키 선택을 선택합니다. 이렇게 하면 고객 관리형 키를 저장하여 AWS KMS에 암호화할 수 있습니다.

Note

현재는 자산 속성 값 및 집계 값에 대해서만 고객이 관리하는 키 암호화를 사용할 수 있습니다.

4. 다음 옵션 중 하나로 KMS 키를 선택합니다.
 - 기존 KMS 키를 사용하려면 – 목록에서 KMS 키 별칭을 선택합니다.
 - 새 KMS 키를 만들려면 — 키 생성을 선택합니다. AWS KMS

Note

AWS KMS 대시보드가 열립니다. KMS 키 생성에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 [키 생성](#)을 참조하세요.

5. 설정을 저장하려면 저장을 선택합니다.

SiteWise 에지 게이트웨이 키 관리

SiteWise 에지 게이트웨이는 에서 실행되며 AWS IoT Greengrass, AWS IoT Greengrass 코어 디바이스는 퍼블릭 및 프라이빗 키를 사용하여 AWS 클라우드를 통해 인증하고 OPC-UA 인증 비밀과 같은 로컬 비밀을 암호화합니다. 자세한 정보는 AWS IoT Greengrass Version 1 개발자 설명서의 [키 관리](#)를 참조하세요.

ID 및 액세스 관리: AWS IoT SiteWise

AWS Identity and Access Management (IAM) 은 관리자가 리소스에 대한 액세스를 안전하게 제어하는데 도움이 되는 도구입니다. AWS IAM 관리자는 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. AWS IoT SiteWise IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

주제

- [고객](#)
- [ID를 통한 인증](#)
- [IAM의 AWS IoT SiteWise 작동 방식](#)
- [AWS 관리형 정책은 다음과 같습니다. AWS IoT SiteWise](#)
- [AWS IoT SiteWise의 서비스 링크 역할 사용](#)
- [AWS IoT Events 경보에 대한 권한 설정](#)
- [교차 서비스 혼동된 대리인 방지](#)
- [AWS IoT SiteWise ID 및 액세스 문제 해결](#)

고객

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

서비스 사용자 - AWS IoT SiteWise 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS IoT SiteWise 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AWS IoT SiteWise의 기능에 액세스할 수 없는 경우 [AWS IoT SiteWise ID 및 액세스 문제 해결](#)을 참조하세요.

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

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

ID를 통한 인증

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

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

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

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

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

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하세요. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 [루트 사용자 보안 인증이 필요한 태스크](#)를 참조하세요.

IAM 사용자 및 그룹

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

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

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

IAM 역할

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

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 연동 자격 증명에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 연동 자격 증명이 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 [타사 자격 증명 공급자의 역할 만들기](#)를 참조하세요.

조하십시오. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 아이덴티티가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관 짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [권한 세트](#)를 참조하십시오.

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

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

IAM의 AWS IoT SiteWise 작동 방식

AWS Identity and Access Management (IAM) 을 사용하여 액세스를 AWS IoT SiteWise관리하기 전에 사용할 수 있는 IAM 기능이 무엇인지 이해해야 합니다. AWS IoT SiteWise

IAM 특성	예 서 지 원 하 나 요? AWS IoT SiteWise
리소스 수준 권한이 있는 자격 증명 기반 정책	예
정책 작업	예
정책 리소스	예
정책 조건 키	예
리소스 기반 정책	아 니 요
액세스 제어 목록(ACL)	아 니 요
태그 기반 인증 (ABAC)	예
임시 보안 인증	예

IAM 특성

예
서
지
원
하
나
요?
AWS
IoT
SiteWise

[포워드 액세스 세션 \(FAS\)](#)

예

[서비스 연결 역할](#)

예

[서비스 역할](#)

예

기타 AWS 서비스가 AWS IoT SiteWise IAM과 연동되는 방식을 자세히 알아보려면 IAM 사용 설명서의 [IAM과 연동되는 AWS 서비스를](#) 참조하십시오.

목차

- [AWS IoT SiteWise IAM 역할](#)
 - [다음과 같은 임시 자격 증명 사용 AWS IoT SiteWise](#)
 - [순방향 액세스 세션 \(FAS\): AWS IoT SiteWise](#)
 - [서비스 연결 역할](#)
 - [서비스 역할](#)
 - [AWS IoT SiteWise에서 IAM 역할 선택](#)
- [AWS IoT SiteWise 태그 기반 인증](#)
- [AWS IoT SiteWise ID 기반 정책](#)
 - [정책 작업](#)
 - [BatchPutAssetPropertyValue 권한 부여](#)
 - [정책 리소스](#)
 - [정책 조건 키](#)
- [예제](#)

- [AWS IoT SiteWise ID 기반 정책 예제](#)
 - [정책 모범 사례](#)
 - [AWS IoT SiteWise 콘솔 사용](#)
 - [사용자가 자체 권한을 볼 수 있도록 허용](#)
 - [사용자가 하나의 계층에서 자산에 데이터를 수집할 수 있도록 허용](#)
 - [태그를 기반으로 AWS IoT SiteWise 자산 보기](#)
- [정책을 사용한 액세스 관리](#)
 - [ID 기반 정책](#)
 - [리소스 기반 정책](#)
 - [액세스 제어 목록\(ACLs\)](#)
 - [기타 정책 타입](#)
 - [여러 정책 타입](#)

AWS IoT SiteWise IAM 역할

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

다음과 같은 임시 자격 증명 사용 AWS IoT SiteWise

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

AWS IoT SiteWise 임시 자격 증명 사용을 지원합니다.

SiteWise 모니터는 페더레이션 사용자가 포털에 액세스할 수 있도록 지원합니다. 포털 사용자는 IAM Identity Center 또는 IAM 보안 인증으로 인증합니다.

Important

사용자 또는 역할에는 포털에 로그인할 수 있는 `iotsitewise:DescribePortal` 권한이 있어야 합니다.

사용자가 포털에 로그인하면 SiteWise Monitor는 다음 권한을 제공하는 세션 정책을 생성합니다.

- 포털 역할이 액세스를 제공하는 계정의 자산 및 자산 데이터에 대한 읽기 전용 접근 권한입니다.
AWS IoT SiteWise
- 사용자에게 관리자(프로젝트 소유자) 또는 읽기 전용(프로젝트 뷰어) 액세스 권한이 있는 포털의 프로젝트에 액세스합니다.

페더레이션 포털 사용자 권한에 대한 자세한 내용은 [서비스 역할 사용 대상 AWS IoT SiteWise Monitor](#)을 참조하세요.

순방향 액세스 세션 (FAS): AWS IoT SiteWise

전달 액세스 세션(FAS) 지원

IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS 경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

서비스 연결 역할

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

AWS IoT SiteWise 서비스 연결 역할을 지원합니다. AWS IoT SiteWise 서비스 연결 역할을 생성 또는 관리하는 방법에 대한 자세한 내용은 [AWS IoT SiteWise의 서비스 링크 역할 사용](#)을 참조하세요.

서비스 역할

이 기능을 사용하면 서비스가 사용자를 대신하여 [서비스 역할](#)을 수임할 수 있습니다. 이 역할을 사용하면 서비스가 다른 서비스의 리소스에 액세스해 사용자를 대신해 작업을 완료할 수 있습니다. 서비스 역할은 AWS 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 SiteWise 서비스 역할을 사용하여 SiteWise 모니터 포털 사용자가 사용자를 대신하여 일부 AWS IoT SiteWise 리소스에 접근할 수 있도록 합니다. 자세한 정보는 [서비스 역할 사용 대상 AWS IoT SiteWise Monitor](#)을 참조하세요.

에서 AWS IoT Events 알람 모델을 생성하려면 먼저 필요한 권한이 있어야 AWS IoT SiteWise합니다. 자세한 정보는 [AWS IoT Events 경보에 대한 권한 설정](#)을 참조하세요.

AWS IoT SiteWise에서 IAM 역할 선택

에서 portal AWS IoT SiteWise리소스를 생성할 때는 SiteWise 모니터 포털의 페더레이션 사용자가 자신을 대신하여 액세스할 AWS IoT SiteWise 수 있는 역할을 선택해야 합니다. 이전에 서비스 역할을 생성한 경우 에서는 선택할 수 있는 역할 목록을 AWS IoT SiteWise 제공합니다. 그렇지 않으면 포털을 만들 때 필요한 권한이 있는 역할을 만들 수 있습니다. 자산 및 자산 데이터에 액세스할 수 있는 역할을 선택하는 것이 중요합니다. 자세한 정보는 [서비스 역할 사용 대상 AWS IoT SiteWise Monitor](#)을 참조하세요.

AWS IoT SiteWise 태그 기반 인증

AWS IoT SiteWise 리소스에 태그를 첨부하거나 요청에 태그를 전달할 수 AWS IoT SiteWise있습니다. 태그를 기반으로 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다. AWS IoT SiteWise 리소스 태깅에 대한 자세한 내용은 [리소스에 태그 지정하기 AWS IoT SiteWise](#)을 참조하세요.

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 자격 증명 기반 정책의 예제는 [태그를 기반으로 AWS IoT SiteWise 자산 보기](#) 섹션에서 확인할 수 있습니다.

AWS IoT SiteWise ID 기반 정책

IAM 정책을 사용하면 누가 무엇을 할 수 있는지 제어할 수 있습니다. AWS IoT SiteWise어떤 작업을 허용할지 여부를 결정하고 이러한 작업에 대한 특정 조건을 설정할 수 있습니다. 예를 들어 정보를 보거나 변경할 수 있는 사람에 대한 규칙을 만들 수 AWS IoT SiteWise있습니다. AWS IoT SiteWise 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하세요.

정책 작업

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

JSON 정책의 Action요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

정책 조치는 조치 앞에 다음 접두사를 AWS IoT SiteWise 사용합니다. `iotsitewise`: 예를 들어 `BatchPutAssetPropertyValue` API 작업을 AWS IoT SiteWise 통해 자산 자산 데이터를 업로드할 권한을 다른 사람에게 부여하려면 해당 작업을 정책에 포함해야 합니다.

`iotsitewise:BatchPutAssetPropertyValue` 정책 설명에는 Action OR NotAction 요소가 포함되어야 합니다. AWS IoT SiteWise 이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 집합을 정의합니다.

명령문 하나에 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

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

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

```
"Action": "iotsitewise:Describe*"
```

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

BatchPutAssetPropertyValue 권한 부여

AWS IoT SiteWise 특이한 방식으로 [BatchPutAssetPropertyValue](#) 작업에 대한 액세스를 승인합니다. 대부분의 작업에서 액세스를 허용하거나 거부할 때 권한이 부여되지 않은 경우 해당 작업은 오류를 반환합니다. 를 사용하면 단일 API 요청으로 BatchPutAssetPropertyValue 여러 데이터 항목을 서로 다른 자산 및 자산 속성에 보낼 수 있습니다. AWS IoT SiteWise 각 데이터 입력을 독립적으로 승인합니다. 요청에서 권한 부여에 실패한 개별 항목의 경우 반환된 오류 목록에 를 AWS IoT SiteWise 포함하십시오. AccessDeniedException AWS IoT SiteWise 동일한 요청에서 다른 항목이 실패하더라도 권한을 부여하고 성공한 모든 항목의 데이터를 수신합니다.

Important

데이터를 데이터 스트림으로 인제스트하기 전에 다음을 수행하십시오.

- 속성 별칭을 사용하여 데이터 스트림을 식별하는 경우 time-series 리소스를 승인하십시오.

- 자산 ID를 사용하여 관련 자산 속성이 포함된 자산을 식별하는 경우 asset 리소스를 승인하십시오.

정책 리소스

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

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 개체를 지정합니다. 문장에는 Resource 또는 NotResource 요소가 반드시 추가되어야 합니다. 모범 사례에 따라 [Amazon 리소스 이름\(ARN\)](#)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 타입을 지원하는 작업에 대해 이 작업을 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

각 IAM 정책 명령문은 ARN을 사용하여 지정한 리소스에 적용됩니다. ARN의 일반적인 구문은 다음과 같습니다.

```
arn:${Partition}:${Service}:${Region}:${Account}:${ResourceType}/${ResourcePath}
```

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

예를 들어 명령문에서 ID가 a1b2c3d4-5678-90ab-cdef-2222EXAMPLE인 자산을 지정하려면 다음 ARN을 사용합니다.

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE"
```

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

```
"Resource": "arn:aws:iotsitewise:region:123456789012:time-series/*"
```

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

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/*"
```

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

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

단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

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

AWS IoT SiteWise 리소스 유형 및 해당 ARN 목록을 보려면 IAM 사용 설명서의 [리소스 정의](#) 기준을 참조하십시오. AWS IoT SiteWise 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 [AWS IoT SiteWise가 정의한 작업](#)을 참조하세요.

정책 조건 키

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

Condition 요소(또는 Condition 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같거나 작음과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition요소를 지정하거나 단일 Condition요소에서 여러 키를 지정하는 경우 AWS 는 논리적 AND태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 OR 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

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

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 [AWS 설명서의 글로벌 조건 컨텍스트 키](#)를 참조하십시오.

⚠ Important

여러 조건 키들이 하나의 리소스에 딸려 있고, 일부 API 작업은 다수의 리소스를 사용합니다. 조건 키로 정책 설명문을 작성하려면 설명문의 Resource 요소를 사용해 조건 키가 적용되는 리소스를 지정합니다. 그렇게 하지 않으면, 조건 키가 해당되지 않는 리소스에 대해서는 조건 검사가 실패하여 정책이 사용자로 하여금 작업을 전혀 수행하지 못하게 막을 수도 있습니다. 리소스를 지정하고 싶지 않거나 다수의 API 작업을 포함하도록 정책의 Action 요소를 작성했다면, 반드시 `...IfExists` 조건 유형을 사용해 조건 키가 그것을 사용하지 않는 리소스에 대해서는 무시되도록 해야 합니다. [자세한 내용은... 을 참조하십시오. IfExists IAM 사용 설명서의 조건.](#)

AWS IoT SiteWise 자체 조건 키 세트를 정의하며 일부 글로벌 조건 키 사용도 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 설명서의 [AWS 글로벌 조건 컨텍스트 키](#)를 참조하십시오.

AWS IoT SiteWise 조건 키

조건 키	설명	타입
<code>iotsitewise:isAssociatedWithAssetProperty</code>	데이터 스트림이 자산 속성과 연결되어 있는지 여부입니다. 이 조건 키를 사용하여 데이터 스트림에 대한 관련 자산 속성의 존재 여부에 따라 권한을 정의할 수 있습니다. 예제 값: <code>true</code>	String
<code>iotsitewise:assetHierarchyPath</code>	슬래시로 구분된 자산 ID의 문자열인 자산의 계층 구조 경로입니다. 이 조건 키를 사용하여 계정에 있는 모든 자산의 계층 구조 하위 집합을 기반으로 권한을 정의할 수 있습니다. 예제 값: <code>/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/a1b2c3d</code>	String

조건 키	설명	타입
	4-5678-90ab-cdef-6666EXAMPLE	
iotsitewise:propertyId	<p>자산 속성의 ID입니다. 자산 모델의 지정된 속성을 기반으로 사용 권한을 정의하려면 이 조건 키를 사용합니다. 이 조건 키는 해당 모델의 모든 자산에 적용됩니다.</p> <p>예제 값: a1b2c3d4-5678-90ab-cdef-3333EXAMPLE</p>	String
iotsitewise:childAssetId	<p>다른 자산에 대한 하위 자산으로 연결된 자산 ID입니다. 이 조건 키를 사용하여 하위 자산을 기반으로 사용 권한을 정의합니다. 상위 자산을 기반으로 권한을 정의하려면 정책 설명의 리소스 섹션을 사용합니다.</p> <p>예제 값: a1b2c3d4-5678-90ab-cdef-6666EXAMPLE</p>	String
iotsitewise:iam	<p>액세스 정책을 나열할 때 IAM ID의 ARN입니다. 이 조건 키를 사용하여 IAM ID에 대한 액세스 정책 권한을 정의합니다.</p> <p>예제 값: arn:aws:iam::123456789012:user/JohnDoe</p>	String, Null

조건 키	설명	타입
<code>iotsitewise:propertyAlias</code>	자산 속성 또는 데이터 스트림을 식별하는 별칭입니다. 이 조건 키를 사용하여 별칭을 기반으로 사용 권한을 정의합니다.	String
<code>iotsitewise:user</code>	<p>액세스 정책을 나열할 때 IAM Identity Center 사용자의 ID입니다. 이 조건 키를 사용하여 IAM Identity Center 사용자에 대한 액세스 정책 권한을 정의합니다.</p> <p>예제 값: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE</p>	String, Null
<code>iotsitewise:group</code>	<p>액세스 정책을 나열할 때 IAM Identity Center 그룹의 ID입니다. 이 조건 키를 사용하여 IAM Identity Center 그룹에 대한 액세스 정책 권한을 정의합니다.</p> <p>예제 값: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE</p>	String, Null
<code>iotsitewise:portal</code>	<p>액세스 정책의 포털 ID입니다. 포털을 기반으로 액세스 정책 권한을 정의하려면 이 조건 키를 사용합니다.</p> <p>예제 값: a1b2c3d4-5678-90ab-cdef-77777EXAMPLE</p>	String, Null

조건 키	설명	타입
iotsitewise:project	<p>액세스 정책의 프로젝트 ID 또는 대시보드에 대한 프로젝트 ID입니다. 이 조건 키를 사용하여 프로젝트를 기반으로 대시보드 또는 액세스 정책 권한을 정의할 수 있습니다.</p> <p>예제 값: a1b2c3d4-5678-90ab-cdef-88888EXAMPLE</p>	String, Null

조건 키를 사용할 수 있는 작업 및 리소스에 대해 알아보려면 [작업 정의 기준을](#) 참조하십시오 AWS IoT SiteWise.

예제

AWS IoT SiteWise ID 기반 정책의 예를 보려면 을 참조하십시오. [AWS IoT SiteWise ID 기반 정책 예제](#)

AWS IoT SiteWise ID 기반 정책 예제

기본적으로 엔티티 (사용자 및 역할) 는 AWS IoT SiteWise 리소스를 생성하거나 수정할 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 권한을 조정하려면 AWS Identity and Access Management (IAM) 관리자가 다음을 수행해야 합니다.

1. 사용자 및 역할에 필요한 리소스에서 특정 API 작업을 수행할 수 있는 권한을 부여하는 IAM 정책을 생성하십시오.
2. 해당 권한이 필요한 사용자 또는 그룹에 해당 정책을 연결하세요.

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

주제

- [정책 모범 사례](#)
- [AWS IoT SiteWise 콘솔 사용](#)
- [사용자가 자체 권한을 볼 수 있도록 허용](#)
- [사용자가 하나의 계층에서 자산에 데이터를 수집할 수 있도록 허용](#)
- [태그를 기반으로 AWS IoT SiteWise 자산 보기](#)

정책 모범 사례

ID 기반 정책은 누군가가 계정에서 AWS IoT SiteWise 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 따르십시오.

- AWS 관리형 정책으로 시작하여 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 [AWS 관리형 정책](#) 또는 [AWS 직무에 대한 관리형 정책](#)을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 [IAM의 정책 및 권한](#)을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 정보는 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 검증하여 안전하고 기능적인 권한 보장 – IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 검증합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 정보는 IAM 사용 설명서의 [IAM Access Analyzer 정책 검증](#)을 참조하세요.
- 멀티 팩터 인증 (MFA) 필요 - IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 AWS 계정 MFA를 활성화하십시오. API 작업을 직접 호출할 때 MFA가 필요하다면 정책에 MFA 조건을 추가합니다. 자세한 정보는 IAM 사용 설명서의 [MFA 보호 API 액세스 구성](#)을 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하세요.

AWS IoT SiteWise 콘솔 사용

AWS IoT SiteWise 콘솔에 액세스하려면 기본 권한 세트가 필요합니다. 이러한 권한을 통해 내 AWS IoT SiteWise 리소스에 대한 세부 정보를 보고 관리할 수 있습니다 AWS 계정.

정책을 너무 제한적으로 설정하면 해당 정책을 사용하는 사용자 또는 역할 (엔티티)에 대해 콘솔이 예상대로 작동하지 않을 수 있습니다. 해당 엔티티가 AWS IoT SiteWise 콘솔을 계속 사용할 수 있도록 하려면 [AWSIoTSiteWiseConsoleFullAccess](#) 관리형 정책을 엔티티에 연결하거나 해당 엔티티에 대해 동등한 권한을 정의하십시오. 자세한 내용은 IAM 사용 설명서의 [사용자에게 권한 추가](#)를 참조하세요.

엔티티가 콘솔이 아닌 AWS Command Line Interface (CLI) 또는 AWS IoT SiteWise API만 사용하는 경우 이러한 최소 권한은 필요하지 않습니다. 이 경우 API 작업에 필요한 특정 작업에 대한 액세스 권한을 부여하기만 하면 됩니다.

사용자가 자체 권한을 볼 수 있도록 허용

이 예시는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

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

```

```

        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

사용자가 하나의 계층에서 자산에 데이터를 수집할 수 있도록 허용

이 예시에서는 루트 자산부터 시작하여 특정 자산 계층 구조의 모든 자산 속성에 데이터를 쓸 수 있는 AWS 계정 액세스 권한을 사용자에게 부여하려고 합니다. a1b2c3d4-5678-90ab-cdef-22222EXAMPLE 정책은 사용자에게 `iotsitewise:BatchPutAssetPropertyValue` 권한을 부여합니다. 이 정책은 `iotsitewise:assetHierarchyPath` 조건 키를 사용하여 계층 경로가 자산 또는 하위 항목과 일치하는 자산에 대한 액세스를 제한합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PutAssetPropertyValuesForHierarchy",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/*"
          ]
        }
      }
    }
  ]
}

```

태그를 기반으로 AWS IoT SiteWise 자산 보기

ID 기반 정책의 조건을 사용하여 태그를 기반으로 AWS IoT SiteWise 리소스에 대한 액세스를 제어할 수 있습니다. 이 예제에서는 자산 보기를 허용하는 정책을 만드는 방법을 보여줍니다. 그러나 자산 태그 Owner에 해당 사용자의 사용자 이름 값이 있는 경우에만 권한이 부여됩니다. 또한 이 정책은 콘솔에서 이 작업을 완료할 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListAllAssets",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DescribeAssetIfOwner",
      "Effect": "Allow",
      "Action": "iotsitewise:DescribeAsset",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

계정의 사용자에게 이 정책을 첨부하십시오. 라는 사용자가 자산을 richard-roe 보려고 하는 경우 해당 AWS IoT SiteWise 자산에 Owner=richard-roe 또는 owner=richard-roe 태그를 지정해야 합니다. 그렇지 않으면 Richard는 액세스가 거부됩니다. 조건 태그 키 이름은 대소문자를 구분하지 않습니다. 따라서, Owner 과 둘 다 Owner 일치합니다. owner 자세한 정보는 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.

정책을 사용한 액세스 관리

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

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

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

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

ID 기반 정책

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

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

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. IAM의 AWS 관리형 정책은 리소스 기반 정책에 사용할 수 없습니다.

액세스 제어 목록(ACLs)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ACL을 지원하는 서비스의 예로는 아마존 S3와 아마존 VPC가 있습니다. AWS WAF ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 [액세스 제어 목록\(ACL\) 개요](#)를 참조하세요.

기타 정책 타입

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

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

여러 정책 타입

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

AWS 관리형 정책은 다음과 같습니다. AWS IoT SiteWise

정책을 직접 작성하는 대신 AWS 관리형 정책을 사용하여 사용자, 그룹, 역할에 권한을 추가하는 작업을 간소화할 수 있습니다. 팀에 정확한 권한을 제공하는 [IAM 고객 관리형 정책을 만들려면](#) 시간과 전문 지식이 필요합니다. 설정을 더 빠르게 하려면 일반적인 사용 사례에 대한 AWS 관리형 정책을 사용하는 것이 좋습니다. 내 페이지에서 AWS 관리형 정책을 찾아보세요 AWS 계정. AWS 관리형 정책에 대한 자세한 내용은 [IAM 사용 설명서](#)에서 AWS 관리형 정책을 참조하세요.

AWS 서비스는 AWS 관리형 정책의 업데이트 및 유지 관리를 담당하므로 이러한 정책의 권한은 사용자가 수정할 수 없습니다. 경우에 따라 새 기능을 수용하기 위한 권한이 추가되어 정책이 연결된 모든 ID에 영향을 미칠 AWS IoT SiteWise 수 있습니다. 이러한 업데이트는 새로운 서비스나 기능이 도입될 때 흔히 발생합니다. 그러나 권한은 절대 제거되지 않으므로 설정이 그대로 유지됩니다.

또한 여러 서비스에 걸친 작업 기능에 대한 관리형 정책을 AWS 지원합니다. 예를 들어 ReadOnly액세스 AWS 관리형 정책은 모든 AWS 서비스와 리소스에 대한 읽기 전용 액세스를 제공합니다. 서비스가 새 기능을 시작하면 새 작업 및 리소스에 대한 읽기 전용 권한이 AWS 추가됩니다. 직무 정책의 목록과 설명은 IAM 사용 설명서의 [직무에 관한AWS 관리형 정책](#)을 참조하세요.

AWS 관리형 정책: AWSIoTSiteWiseReadOnlyAccess

AWSIoTSiteWiseReadOnlyAccess AWS 관리형 정책을 사용하여 에 대한 읽기 전용 액세스를 허용할 수 있습니다. AWS IoT SiteWise

AWSIoTSiteWiseReadOnlyAccess 정책을 IAM 보안 인증에 연결할 수 있습니다.

서비스 수준 권한

이 정책은 에 대한 읽기 전용 액세스를 제공합니다. AWS IoT SiteWise이 정책에 다른 서비스 권한은 포함되지 않습니다.

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

```

        "iotsitewise:Describe*",
        "iotsitewise:List*",
        "iotsitewise:BatchGet*",
        "iotsitewise:Get*"
    ],
    "Resource": "*"
}
]
}

```

AWS 관리형 정책: AWSServiceRoleForIoTSiteWise

AWSServiceRoleForIoTSiteWise 역할은 다음 권한을 가진 AWSServiceRoleForIoTSiteWise 정책을 사용합니다. 이 정책:

- 에서 실행되는 SiteWise AWS IoT Greengrass 에지 게이트웨이를 AWS IoT SiteWise 배포할 수 있습니다.
- AWS IoT SiteWise 로깅을 수행할 수 있습니다.
- AWS IoT TwinMaker 데이터베이스에 대해 메타데이터 검색 AWS IoT SiteWise 쿼리를 실행할 수 있습니다.

단일 사용자 AWS IoT SiteWise 계정으로 사용하는 경우 AWSServiceRoleForIoTSiteWise 역할은 IAM 계정에 AWSServiceRoleForIoTSiteWise 정책을 생성하고 AWSServiceRoleForIoTSiteWise [서비스](#) 연결 역할에 정책을 연결합니다. AWS IoT SiteWise

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
  ],
}

```

```

{
  "Sid": "AllowSiteWiseAccessLogGroup",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:DescribeLogGroups"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
},
{
  "Sid": "AllowSiteWiseAccessLog",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogStream",
    "logs:DescribeLogStreams",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
  "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:GetWorkspace",
    "iottwinmaker:ExecuteQuery"
  ],
  "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "iottwinmaker:linkedServices": [
        "IOTSITWISE"
      ]
    }
  }
}
]
}

```

AWS IoT SiteWise AWS 관리형 정책 업데이트

이 서비스가 변경 내용 추적을 시작한 시점부터 시작하여 AWS 관리형 정책 업데이트에 대한 AWS IoT SiteWise 세부 정보를 볼 수 있습니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 AWS IoT SiteWise 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
AWSServiceRoleForIoTSiteWise - 기존 정책에 대한 업데이트	AWS IoT SiteWise 이제 AWS IoT TwinMaker 데이터베이스를 대상으로 메타데이터 검색 쿼리를 실행할 수 있습니다.	2023년 11월 6일
AWSIoTSiteWiseReadOnlyAccess -기존 정책 업데이트	AWS IoT SiteWise 일괄 읽기 작업을 수행할 수 있는 새 정책 접두사를 추가했습니다. BatchGet*	2022년 9월 16일
AWSIoTSiteWiseReadOnlyAccess - 새 정책	AWS IoT SiteWise 읽기 전용 액세스 권한을 부여하는 새 정책을 추가했습니다. AWS IoT SiteWise	2021년 11월 24일
AWS IoT SiteWise 변경 내용 추적 시작	AWS IoT SiteWise AWS 관리형 정책의 변경 사항 추적을 시작했습니다.	2021년 11월 24일

AWS IoT SiteWise의 서비스 링크 역할 사용

AWS IoT SiteWise AWS Identity and Access Management (IAM) [서비스](#) 연결 역할을 사용합니다. 서비스 연결 역할은 직접 연결되는 고유한 유형의 IAM 역할입니다. AWS IoT SiteWise 서비스 연결 역할은 사전 정의되며 서비스가 사용자를 AWS IoT SiteWise 대신하여 다른 서비스를 호출하는 데 필요한 모든 권한을 포함합니다. AWS

서비스 연결 역할은 필요한 모든 권한을 자동으로 포함하여 구성을 단순화합니다. AWS IoT SiteWise AWS IoT SiteWise 서비스 연결 역할의 권한을 정의하며, 달리 정의되지 않는 한 해당 역할만 맡을 AWS IoT SiteWise 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함됩니다. 그리고 이 권한 정책은 다른 IAM 엔티티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 하면 AWS IoT SiteWise 리소스 액세스 권한을 실수로 제거할 수 없으므로 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 [IAM으로 작업하는AWS 서비스](#)를 참조하고 서비스 연결 역할 열에 예가 표시된 서비스를 찾으십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 링크가 있는 예를 선택합니다.

주제

- [AWS IoT SiteWise에 대한 서비스 링크 역할 권한](#)
- [AWS IoT SiteWise에 대한 서비스 링크 역할 생성](#)
- [AWS IoT SiteWise에 대한 서비스 링크 역할 편집](#)
- [AWS IoT SiteWise에 대한 서비스 링크 역할 삭제](#)
- [서비스 연결 역할이 지원되는 지역 AWS IoT SiteWise](#)
- [서비스 역할 사용 대상 AWS IoT SiteWise Monitor](#)

AWS IoT SiteWise에 대한 서비스 링크 역할 권한

AWS IoT SiteWise 라는 서비스 연결 역할을 사용합니다. AWSServiceRoleForIoTSiteWise AWS IoT SiteWise 이 서비스 연결 역할을 사용하여 SiteWise Edge 게이트웨이 (에서 AWS IoT Greengrass 실행) 를 배포하고 로깅을 수행합니다.

AWSServiceRoleForIoTSiteWise 서비스 연결 역할은 다음 권한이 있는 AWSServiceRoleForIoTSiteWise 정책을 사용합니다. 이 정책은 다음과 같습니다.

- 에서 실행되는 SiteWise AWS IoT Greengrass 에지 게이트웨이를 AWS IoT SiteWise 배포할 수 있습니다.
- AWS IoT SiteWise 로깅을 수행할 수 있습니다.
- AWS IoT TwinMaker 데이터베이스에 대해 메타데이터 검색 AWS IoT SiteWise 쿼리를 실행할 수 있습니다.

에서 AWSServiceRoleForIoTSiteWise 허용된 작업에 대한 자세한 내용은 [AWS 관리형 정책을 참조하십시오](#) AWS IoT SiteWise.

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

```

    "greengrass:GetAssociatedRole",
    "greengrass:GetCoreDefinition",
    "greengrass:GetCoreDefinitionVersion",
    "greengrass:GetGroup",
    "greengrass:GetGroupVersion"
  ],
  "Resource": "*"
},
{
  "Sid": "AllowSiteWiseAccessLogGroup",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:DescribeLogGroups"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
},
{
  "Sid": "AllowSiteWiseAccessLog",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogStream",
    "logs:DescribeLogStreams",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
  "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:GetWorkspace",
    "iottwinmaker:ExecuteQuery"
  ],
  "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "iottwinmaker:linkedServices": [
        "IOTSITewise"
      ]
    }
  }
}
]

```

}

로그를 사용하여 SiteWise 에지 게이트웨이를 모니터링하고 문제를 해결할 수 있습니다. 자세한 정보는 [SiteWise 에지 게이트웨이 로그 모니터링](#)을 참조하세요.

IAM 엔티티(예: 사용자, 그룹 또는 역할)가 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 하려면 먼저 권한을 구성합니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 권한](#)을 참조하세요.

AWS IoT SiteWise에 대한 서비스 링크 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. AWS IoT SiteWise 콘솔에서 다음 작업을 수행하면 서비스 연결 AWS IoT SiteWise 역할이 자동으로 생성됩니다.

- 그린그래스 V1 게이트웨이를 생성합니다.
- 로깅 옵션을 구성합니다.
- 쿼리 실행 배너에서 옵트인 버튼 선택

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. AWS IoT SiteWise 콘솔에서 작업을 수행하면 서비스 연결 역할이 다시 AWS IoT SiteWise 생성됩니다.

또한 IAM 콘솔 또는 API를 사용하여 AWS IoT SiteWise를 위한 서비스 연결 역할을 만들 수도 있습니다.

- IAM 콘솔에서 이 작업을 수행하려면 AWSServiceRoleForIoTSiteWise 정책을 사용하여 역할을 생성하고 신뢰 관계를 생성해야 합니다. `iotsitewise.amazonaws.com`
- AWS CLI 또는 IAM API를 사용하여 이 작업을 수행하려면 `arn:aws:iam::aws:policy/aws-service-role/AWSServiceRoleForIoTSiteWise` 정책을 사용하여 역할을 생성하고 신뢰 관계를 생성하십시오. `iotsitewise.amazonaws.com`

자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 생성](#)을 참조하세요.

이 서비스 연결 역할을 삭제하면 동일한 프로세스를 사용하여 역할을 다시 생성할 수 있습니다.

AWS IoT SiteWise에 대한 서비스 링크 역할 편집

AWS IoT SiteWise AWSServiceRoleForIoTSiteWise 서비스 연결 역할을 편집할 수 없습니다. 서비스 연결 역할을 생성한 후에는 다양한 엔티티가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없

습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 편집](#)을 참조하세요.

AWS IoT SiteWise에 대한 서비스 링크 역할 삭제

서비스 연결 역할이 필요한 기능이나 서비스를 더 이상 사용하지 않는 경우 관련 역할을 삭제하는 것이 좋습니다. 이는 모니터링 또는 유지 관리되지 않는 비활성 엔티티가 발생하지 않도록 하기 위한 것입니다. 단, 서비스 링크 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

Note

AWS IoT SiteWise 서비스가 역할을 사용하고 있을 때 리소스를 삭제하려고 하면 삭제가 실패할 수 있습니다. 그 경우 몇 분 동안 기다렸다가 다시 시도하십시오.

에서 사용하는 AWS IoT SiteWise 리소스를 삭제하려면 `AWSServiceRoleForIoTSiteWise`

1. 에 대한 로깅을 AWS IoT SiteWise 비활성화합니다. 자세한 내용은 [로깅 수준 변경](#) 섹션을 참조하세요.
2. 활성 SiteWise 에지 게이트웨이를 모두 삭제합니다.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 `AWSServiceRoleForIoTSiteWise` 서비스 연결 역할을 삭제합니다. 자세한 내용은 [IAM 사용 설명서](#)의 서비스 연결 역할 삭제를 참조하세요.

서비스 연결 역할이 지원되는 지역 AWS IoT SiteWise

AWS IoT SiteWise 서비스를 사용할 수 있는 모든 지역에서 서비스 연결 역할을 사용할 수 있습니다. 자세한 내용은 [AWS IoT SiteWise 엔드포인트 및 할당량](#)을 참조하세요.

서비스 역할 사용 대상 AWS IoT SiteWise Monitor

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하세요.

페더레이션된 SiteWise 모니터 포털 사용자가 사용자 AWS IoT SiteWise 및 AWS IAM Identity Center 리소스에 접근할 수 있도록 하려면 생성한 각 포털에 서비스 역할을 연결해야 합니다. 서비스 역할은

SiteWise Monitor를 신뢰할 수 있는 엔티티로 지정하고 [AWSIoTSiteWiseMonitorPortalAccess](#) 관리형 정책을 포함하거나 [이에 상응하는 권한을](#) 정의해야 합니다. 이 정책은 SiteWise 모니터가 사용자 AWS IoT SiteWise 및 IAM Identity Center 리소스에 액세스하는 데 사용하는 권한 집합에 의해 유지 관리되며 이를 정의합니다.

SiteWise 모니터 포털을 생성할 때는 해당 포털의 사용자가 자신과 IAM Identity Center 리소스에 액세스할 수 있도록 허용하는 역할을 선택해야 합니다. AWS IoT SiteWise 콘솔에서 역할을 생성하고 구성할 수 있습니다. 나중에 IAM에서 역할을 편집할 수 있습니다. 역할에서 필요한 권한을 제거하거나 역할을 삭제하면 포털 사용자가 SiteWise 모니터 포털을 사용하는 데 문제가 발생합니다.

Note

2020년 4월 29일 이전에 생성된 포털에는 서비스 역할이 필요 없었습니다. 이 날짜 이전에 포털을 만든 경우, 계속 사용하려면 서비스 역할을 연결해야 합니다. 이렇게 하려면 [AWS IoT SiteWise 콘솔](#)의 포털 페이지로 이동한 다음 IAM 역할을 사용하도록 모든 포털 마이그레이션을 선택합니다.

다음 섹션에서는 AWS Management Console 또는 에서 SiteWise 모니터 서비스 역할을 생성하고 관리하는 방법을 설명합니다. AWS Command Line Interface

목차

- [SiteWise 모니터에 대한 서비스 역할 권한](#)
- [SiteWise 모니터 서비스 역할 관리 \(콘솔\)](#)
 - [포털의 서비스 역할 찾기\(콘솔\)](#)
 - [SiteWise 모니터 서비스 역할 생성 \(AWS IoT SiteWise 콘솔\)](#)
 - [SiteWise 모니터 서비스 역할 생성 \(IAM 콘솔\)](#)
 - [포털의 서비스 역할 변경\(콘솔\)](#)
- [SiteWise 모니터 서비스 역할 관리 \(CLI\)](#)
 - [포털의 서비스 역할 찾기\(CLI\)](#)
 - [SiteWise 모니터 서비스 역할 \(CLI\) 생성](#)
- [SiteWise 에 대한 업데이트 모니터링 AWSIoTSiteWiseMonitorServiceRole](#)

SiteWise 모니터에 대한 서비스 역할 권한

포털을 생성할 때 이름이 로 시작하는 역할을 생성할 수 `AWSIoTSiteWiseMonitorServiceRole` 있습니다. AWS IoT SiteWise 이 역할을 통해 페더레이션된 SiteWise 모니터 사용자는 포털 구성, 자산, 자산 데이터, IAM Identity Center 구성 및 데이터에 액세스할 수 있습니다.

이 역할은 역할을 맡기 위해 다음 서비스를 신뢰합니다.

- `monitor.iotsitewise.amazonaws.com`

역할은 이름이 로 시작하는 다음 권한 정책을 사용하여 SiteWise 모니터 사용자가 `AWSIoTSiteWiseMonitorServicePortalPolicy` 계정의 리소스에 대한 작업을 완료할 수 있도록 허용합니다. [AWSIoTSiteWiseMonitorPortalAccess](#) 관리형 정책은 동등한 권한을 정의합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",

```

```

        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",
        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iotevents:keyValue": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents>CreateAlarmModel",
        "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:RequestTag/iotsitewisemonitor": "false"
        }
    }
}

```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:UpdateAlarmModel",
        "iotevents:DeleteAlarmModel"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/iotsitewisemonitor": "false"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "iotevents.amazonaws.com"
          ]
        }
      }
    }
  ]
}

```

경보에 필요한 사용 권한에 대한 자세한 내용은 [AWS IoT Events 경보에 대한 권한 설정](#)을 참조하세요.

포털 사용자가 로그인하면 SiteWise Monitor는 서비스 역할과 해당 사용자의 접근 [정책의 교차점을 기반으로 세션](#) 정책을 생성합니다. 액세스 정책은 포털 및 프로젝트에 대한 ID의 액세스 수준을 정의합니다. 포털 권한 및 접근 정책에 대한 자세한 내용은 [SiteWise 모니터 포털 관리](#) 및 [CreateAccess정책](#)을 참조하십시오.

SiteWise 모니터 서비스 역할 관리 (콘솔)

포털의 SiteWise 모니터 서비스 역할을 AWS IoT SiteWise Console 쉽게 관리할 수 있습니다. 포털을 생성할 때 콘솔은 첨부 파일에 적합한 기존 역할을 확인합니다. 사용할 수 있는 서비스 역할이 없는 경

우 콘솔에서 자동으로 서비스 역할을 생성하고 구성할 수 있습니다. 자세한 정보는 [포털 생성](#)을 참조하세요.

주제

- [포털의 서비스 역할 찾기\(콘솔\)](#)
- [SiteWise 모니터 서비스 역할 생성 \(AWS IoT SiteWise 콘솔\)](#)
- [SiteWise 모니터 서비스 역할 생성 \(IAM 콘솔\)](#)
- [포털의 서비스 역할 변경\(콘솔\)](#)

포털의 서비스 역할 찾기(콘솔)

다음 단계를 사용하여 SiteWise 모니터 포털에 연결된 서비스 역할을 찾을 수 있습니다.

포털의 서비스 역할을 찾으려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 포털을 선택합니다.
3. 서비스 역할을 찾으려는 포털을 선택합니다.

포털에 연결된 역할이 권한, 서비스 역할 아래에 나타납니다.

SiteWise 모니터 서비스 역할 생성 (AWS IoT SiteWise 콘솔)

SiteWise 모니터 포털을 생성할 때 포털의 서비스 역할을 생성할 수 있습니다. 자세한 정보는 [포털 생성](#)을 참조하세요.

AWS IoT SiteWise 콘솔에서 기존 포털의 서비스 역할을 생성할 수도 있습니다. 이는 포털의 기존 서비스 역할을 대체합니다.

기존 포털에 대한 서비스 역할을 만들려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 포털을 선택합니다.
3. 새 서비스 역할을 만들려는 포털을 선택합니다.
4. 포털 세부 정보에서 편집을 선택합니다.
5. 권한 아래에서 새 서비스 역할 생성 및 사용을 선택합니다.
6. 새 역할의 이름을 입력합니다.

7. 저장을 선택합니다.

SiteWise 모니터 서비스 역할 생성 (IAM 콘솔)

IAM 콘솔의 서비스 역할 템플릿에서 서비스 역할을 생성할 수 있습니다. 이 역할 템플릿은 [AWSIoTSiteWiseMonitorPortalAccess](#) 관리형 정책을 포함하며 SiteWise 모니터를 신뢰할 수 있는 개체로 지정합니다.

포털 서비스 역할 템플릿에서 서비스 역할을 만들려면

1. [IAM 콘솔](#)로 이동합니다.
2. 탐색 창에서 역할을 선택합니다.
3. 역할 생성을 선택합니다.
4. 사용 사례 선택에서 IoT를 선택합니다 SiteWise.
5. 사용 사례 선택에서 IoT SiteWise Monitor - Portal를 선택합니다.
6. 다음: 권한을 선택합니다.
7. 다음: 태그를 선택합니다.
8. 다음: 검토를 선택합니다.
9. 역할 이름에 새 서비스 역할의 이름을 입력합니다.
10. 역할 생성을 선택합니다.

포털의 서비스 역할 변경(콘솔)

다음 절차를 사용하여 포털의 다른 SiteWise 모니터 서비스 역할을 선택합니다.

포털의 서비스 역할을 변경하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 포털을 선택합니다.
3. 서비스 역할을 변경할 포털을 선택합니다.
4. 포털 세부 정보에서 편집을 선택합니다.
5. 권한에서 기존 역할 사용을 선택합니다.
6. 이 포털에 연결할 기존 역할을 선택합니다.
7. 저장을 선택합니다.

SiteWise 모니터 서비스 역할 관리 (CLI)

를 다음 포털 서비스 역할 관리 작업에 사용할 수 있습니다. AWS CLI

주제

- [포털의 서비스 역할 찾기\(CLI\)](#)
- [SiteWise 모니터 서비스 역할 \(CLI\) 생성](#)

포털의 서비스 역할 찾기(CLI)

SiteWise 모니터 포털에 연결된 서비스 역할을 찾으려면 다음 명령어를 실행하여 현재 지역의 모든 포털을 나열하세요.

```
aws iotsitewise list-portals
```

이 작업은 포털 요약 정보가 포함된 응답을 다음 형식으로 반환합니다.

```
{
  "portalSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
      "name": "WindFarmPortal",
      "description": "A portal that contains wind farm projects for Example Corp.",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/role-name",
      "startUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
      "creationDate": "2020-02-04T23:01:52.90248068Z",
      "lastUpdateDate": "2020-02-04T23:01:52.90248078Z"
    }
  ]
}
```

포털의 ID를 알고 있는 경우 이 [DescribePortal](#) 작업을 사용하여 포털의 역할을 찾을 수도 있습니다.

SiteWise 모니터 서비스 역할 (CLI) 생성

다음 단계를 사용하여 새 SiteWise 모니터 서비스 역할을 생성합니다.

SiteWise 모니터 서비스 역할을 만들려면

1. SiteWise Monitor가 역할을 맡을 수 있도록 허용하는 신뢰 정책이 포함된 역할을 생성합니다. 이 예제에서는 JSON 문자열에 저장된 신뢰 정책으로 **MySiteWiseMonitorPortalRole**이라는 역할을 만듭니다.

Linux, macOS, or Unix

```
aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitor.iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

Windows command prompt

```
aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"monitor.iotsitewise.amazonaws.com\"},\"Action\":\"sts:AssumeRole\"}]}"
```

2. 출력의 역할 메타데이터에서 역할 ARN을 복사합니다. 포털을 생성할 때 이 ARN을 사용하여 역할을 포털에 연결합니다. 포털 생성에 대한 자세한 내용은 AWS IoT SiteWise API 참조를 참조하십시오 [CreatePortal](#).
3. **AWSIoTSiteWiseMonitorPortalAccess** 정책을 역할에 연결하거나 동등한 권한을 정의하는 정책을 연결합니다.

```
aws iam attach-role-policy --role-name MySiteWiseMonitorPortalRole --policy-arn arn:aws:iam::aws:policy/service-role/AWSIoTSiteWiseMonitorPortalAccess
```

서비스 역할을 기존 포털에 연결하려면

1. 포털의 기존 세부 정보를 검색하려면 다음 명령을 실행합니다. *portal-id*를 해당 포털의 ID로 바꿉니다.

```
aws iotsitewise describe-portal --portal-id portal-id
```

이 작업은 포털의 세부 정보가 포함된 응답을 다음 형식으로 반환합니다.

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:region:account-id:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp.",
  "portalClientId": "E-1a2b3c4d5e6f_sn6tbqHVzLWVEXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalContactEmail": "support@example.com",
  "portalStatus": {
    "state": "ACTIVE"
  },
  "portalCreationDate": "2020-04-29T23:01:52.90248068Z",
  "portalLastUpdateDate": "2020-04-29T00:28:26.103548287Z",
  "roleArn": "arn:aws:iam::123456789012:role/service-role/AWSIoTSiteWiseMonitorServiceRole_1aEXAMPLE"
}
```

2. 서비스 역할을 포털에 연결하려면 다음 명령을 실행합니다. *role-arn*을 서비스 역할 ARN으로 바꾸고 나머지 파라미터를 해당 포털의 기존 값으로 바꿉니다.

```
aws iotsitewise update-portal \
  --portal-id portal-id \
  --role-arn role-arn \
  --portal-name portal-name \
  --portal-description portal-description \
  --portal-contact-email portal-contact-email
```

SiteWise 에 대한 업데이트 모니터링 AWSIoTSiteWiseMonitorServiceRole

이 서비스가 변경 내용 추적을 시작한 시점부터 시작하여 AWSIoTSiteWiseMonitorServiceRolefor SiteWise Monitor의 업데이트에 대한 세부 정보를 볼 수 있습니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 AWS IoT SiteWise 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
AWSIoTSiteWiseMonitorPortalAccess - 정책 업데이트	AWS IoT SiteWise 알람 기능에 대한 AWSIoTSiteWiseMonitorPortalAccess 관리 정책을 업데이트했습니다.	2021년 5월 27일
AWS IoT SiteWise 변경 내용 추적 시작	AWS IoT SiteWise 서비스 역할에 대한 변경 사항 추적을 시작했습니다.	2020년 12월 15일

AWS IoT Events 경보에 대한 권한 설정

AWS IoT Events 알람 모델을 사용하여 AWS IoT SiteWise 자산 자산을 모니터링할 때는 다음과 같은 IAM 권한이 있어야 합니다.

- 데이터를 AWS IoT Events 전송할 AWS IoT SiteWise 수 있는 AWS IoT Events 서비스 역할. 자세한 내용은 AWS IoT Events 개발자 안내서에서 [AWS IoT Events의 자격 증명 및 액세스 관리](#)를 참조하세요.
- 다음과 같은 AWS IoT SiteWise 작업 권한이 있어야 합니다
`diotsitewise:UpdateAssetModelPropertyRouting`.
`iotsitewise:DescribeAssetModel` 및. 이러한 AWS IoT SiteWise 권한을 통해 자산 속성 값을 AWS IoT Events 알람 모델에 보낼 수 있습니다.

자세한 내용은 IAM 사용 설명서의 [리소스 기반 정책](#)을 참조하세요.

필요한 작업 권한

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다. JSON 정책의 Action요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다.

AWS IoT Events 경보 모델을 정의하기 전에 자산 속성 값을 경보 모델로 전송할 수 AWS IoT SiteWise 있는 다음과 같은 권한을 부여해야 합니다.

- `iotsitewise:DescribeAssetModel`— 자산 속성이 AWS IoT Events 존재하는지 확인할 수 있습니다.
- `iotsitewise:UpdateAssetModelPropertyRouting`— 데이터를 전송할 수 AWS IoT SiteWise 있는 구독을 자동으로 생성할 수 AWS IoT SiteWise 있습니다. AWS IoT Events

AWS IoT SiteWise 지원되는 작업에 대한 자세한 내용은 서비스 권한 부여 AWS IoT SiteWise참조에 [정의된 작업을](#) 참조하십시오.

Example 권한 정책 예 1

다음 정책은 자산 속성 값을 모든 AWS IoT Events 경보 모델에 AWS IoT SiteWise 전송할 수 있도록 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

Example 권한 정책 예 2

다음 정책은 지정된 자산 속성의 값을 지정된 AWS IoT Events 경보 모델로 AWS IoT SiteWise 보낼 수 있도록 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": [
        "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/12345678-90ab-cdef-1234-567890abcdef"
      ],
      "Condition": {
        "StringLike": {
          "iotsitewise:propertyId": "abcdef12-3456-7890-abcd-ef1234567890",
          "iotevents:alarmModelArn": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/MyAlarmModel"
        }
      }
    }
  ]
}
```

(선택 사항) ListInputRoutings 권한

자산 모델을 업데이트하거나 삭제할 때의 AWS IoT Events 알람 모델이 이 자산 모델과 관련된 자산 속성을 모니터링하고 있는지 확인할 AWS IoT SiteWise 수 있습니다. 이렇게 하면 AWS IoT Events 경보에서 현재 사용 중인 자산 속성을 삭제할 수 없습니다. 에서 AWS IoT SiteWise이 기능을 활성화하려면 `iotevents:ListInputRoutings` 권한이 있어야 합니다. 이 AWS IoT SiteWise 권한을 사용하면 에서 지원하는 [ListInput AWS IoT Events라우팅](#) API 작업을 호출할 수 있습니다.

Note

ListInputRoutings 권한을 추가하는 것이 좋습니다.

Example 권한 정책 예

다음 정책을 통해 자산 모델을 업데이트 및 삭제하고 ListInputRoutings API를 에서 AWS IoT SiteWise사용할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModel",
        "iotsitewise>DeleteAssetModel",
        "iotevents:ListInputRoutings"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

SiteWise 모니터에 필요한 권한

SiteWise 모니터 포털의 알람 기능을 사용하려면 [SiteWise 모니터 서비스 역할](#)을 다음 정책으로 업데이트해야 합니다.

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

```

{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:DescribePortal",
    "iotsitewise:CreateProject",
    "iotsitewise:DescribeProject",
    "iotsitewise:UpdateProject",
    "iotsitewise>DeleteProject",
    "iotsitewise:ListProjects",
    "iotsitewise:BatchAssociateProjectAssets",
    "iotsitewise:BatchDisassociateProjectAssets",
    "iotsitewise:ListProjectAssets",
    "iotsitewise:CreateDashboard",
    "iotsitewise:DescribeDashboard",
    "iotsitewise:UpdateDashboard",
    "iotsitewise>DeleteDashboard",
    "iotsitewise:ListDashboards",
    "iotsitewise:CreateAccessPolicy",
    "iotsitewise:DescribeAccessPolicy",
    "iotsitewise:UpdateAccessPolicy",
    "iotsitewise>DeleteAccessPolicy",
    "iotsitewise:ListAccessPolicies",
    "iotsitewise:DescribeAsset",
    "iotsitewise:ListAssets",
    "iotsitewise:ListAssociatedAssets",
    "iotsitewise:DescribeAssetProperty",
    "iotsitewise:GetAssetPropertyValue",
    "iotsitewise:GetAssetPropertyValueHistory",
    "iotsitewise:GetAssetPropertyAggregates",
    "iotsitewise:BatchPutAssetPropertyValue",
    "iotsitewise:ListAssetRelationships",
    "iotsitewise:DescribeAssetModel",
    "iotsitewise:ListAssetModels",
    "iotsitewise:UpdateAssetModel",
    "iotsitewise:UpdateAssetModelPropertyRouting",
    "sso-directory:DescribeUsers",
    "sso-directory:DescribeUser",
    "iotevents:DescribeAlarmModel",
    "iotevents:ListTagsForResource"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",

```

```

    "Action": [
      "iotevents:BatchAcknowledgeAlarm",
      "iotevents:BatchSnoozeAlarm",
      "iotevents:BatchEnableAlarm",
      "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "iotevents:keyValue": "false"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotevents:CreateAlarmModel",
      "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "aws:RequestTag/iotsitewisemonitor": "false"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotevents:UpdateAlarmModel",
      "iotevents>DeleteAlarmModel"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "aws:ResourceTag/iotsitewisemonitor": "false"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],

```

```

    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "iotevents.amazonaws.com"
        ]
      }
    }
  ]
}

```

교차 서비스 혼동된 대리인 방지

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

리소스 정책에 [aws:SourceArn](#) 및 [aws:SourceAccount](#) 글로벌 조건 컨텍스트 키를 사용하여 리소스에 다른 서비스에 AWS IoT SiteWise 부여하는 권한을 제한하는 것이 좋습니다. 만약 [aws:SourceArn](#) 값에 Amazon S3 버킷 Amazon 리소스 이름(ARN)과 같은 계정 ID가 포함되어 있지 않은 경우 권한을 제한하려면 두 전역 조건 컨텍스트 키를 모두 사용해야 합니다. 두 전역 조건 컨텍스트 키와 계정을 포함한 [aws:SourceArn](#) 값을 모두 사용하는 경우, [aws:SourceAccount](#) 값 및 [aws:SourceArn](#) 값의 계정은 동일한 정책 명령문에서 사용할 경우 반드시 동일한 계정 ID를 사용해야 합니다.

- 하나의 리소스만 교차 서비스 액세스와 연결되도록 허용하려는 경우 [aws:SourceArn](#)를 사용하십시오.
- 해당 계정의 모든 리소스가 교차 서비스 사용과 연결되도록 허용하려는 경우 [aws:SourceAccount](#)를 사용하세요.

의 값은 [sts:AssumeRole](#) 요청과 관련된 AWS IoT SiteWise 고객 [aws:SourceArn](#) 리소스여야 합니다.

혼동된 대리인 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn` 글로벌 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:service:*:123456789012:*`입니다.

Example — 혼란스러운 대리 예방

다음 예제는 `aws:SourceArn` 및 `aws:SourceAccount` 글로벌 조건 컨텍스트 키를 사용하여 혼동된 대리인 문제를 방지하는 AWS IoT SiteWise 방법을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Resource": [
      "arn:aws:iotsitewise:::ResourceName/*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:iotsitewise:*:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

AWS IoT SiteWise ID 및 액세스 문제 해결

다음 정보를 사용하면 AND AWS Identity and Access Management (IAM) 를 사용할 때 발생할 수 있는 일반적인 문제를 AWS IoT SiteWise 진단하고 해결하는 데 도움이 됩니다.

주제

- [저는 다음과 같은 조치를 취할 권한이 없습니다. AWS IoT SiteWise](#)

- [iam:PassRole 작업을 수행할 권한이 없음](#)
- [외부 사용자가 내 AWS IoT SiteWise 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.](#)

저는 다음과 같은 조치를 취할 권한이 없습니다. AWS IoT SiteWise

작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 도움을 요청해야 합니다. 관리자는 사용자 이름과 비밀번호를 제공한 사람입니다.

다음 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 자산의 세부 정보를 보려고 하지만 iotsitewise:DescribeAsset 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotsitewise:DescribeAsset on resource: a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

이 경우 Mateo는 iotsitewise:DescribeAsset 작업을 사용하여 a1b2c3d4-5678-90ab-cdef-22222EXAMPLE ID가 있는 자산 리소스에 액세스할 수 있도록 정책을 업데이트할 것을 관리자에게 요청합니다.

iam:PassRole 작업을 수행할 권한이 없음

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS IoT SiteWise에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 만드는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 AWS IoT SiteWise에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우 Mary가 iam:PassRole 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 AWS IoT SiteWise 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 이러한 기능의 AWS IoT SiteWise 지원 여부를 알아보려면 [IAM의 AWS IoT SiteWise 작동 방식](#).
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 IAM 사용 [설명서에서 자신이 소유한 다른 AWS 계정 IAM 사용자에게 액세스 권한 제공](#)을 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 [설명서의 타사 AWS 계정 AWS 계정 소유에 대한 액세스 제공](#)을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 [설명서의 외부에서 인증된 사용자에게 액세스 권한 제공\(ID 페더레이션\)](#)을 참조하세요.
- 크로스 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 [설명서의 IAM 역할과 리소스 기반 정책의 차이](#)를 참조하세요.

규정 준수 검증: AWS IoT SiteWise

AWS IoT SiteWise AWS 규정 준수 프로그램의 범위에 포함되지 않습니다.

특정 규정 준수 프로그램 범위 내 AWS 서비스 목록은 [규정 준수 프로그램별 범위 내 AWS 서비스 규정 준수](#) 참조하십시오. 일반 정보는 [AWS 규정 준수 프로그램 AWS 보증 프로그램 규정 AWS](#) 참조하십시오.

를 사용하여 AWS Artifact 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 보고서 <https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html> 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS IoT SiteWise 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- [보안 및 규정 준수 킷스타트 가이드](#) - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다. AWS

- [HIPAA 보안 및 규정 준수를 위한 설계 백서](#) — 이 백서는 기업이 HIPAA 준수 애플리케이션을 개발하는 데 사용할 수 있는 방법을 설명합니다. AWS
- [AWS 규정 준수 리소스 규정](#) — 이 통합 문서 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS Config 개발자 안내서의 [규칙을 사용하여 리소스 평가](#) — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- [AWS Security Hub](#) — 이 AWS 서비스는 보안 업계 표준 및 모범 사례를 준수하는지 확인하는 데 도움이 되는 내부 보안 상태를 종합적으로 보여줍니다.
- [산업용 IoT 솔루션을 위한 10가지 보안 항목](#) — 이 블로그 게시물에서는 산업용 제어 시스템(ICS), 산업용 사물 인터넷(IIoT) 및 클라우드 환경을 보호하는 데 도움이 되는 10가지 항목을 소개합니다.
- [제조 OT를 위한 보안 모범 사례](#) — 이 백서에서는 클라우드용 온프레미스 하이브리드 제조 워크로드를 설계, 배포 및 설계하기 위한 보안 모범 사례를 설명합니다. AWS

의 레질리언스 AWS IoT SiteWise

AWS 글로벌 인프라는 AWS 지역 및 가용 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며, 이러한 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹으로 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS IoT SiteWise 완전관리형이며 Amazon S3 및 Amazon EC2와 같은 가용성과 내구성이 뛰어난 AWS 서비스를 사용합니다. 가용 영역 장애 발생 시 가용성을 보장하기 위해 여러 가용 영역에서 AWS IoT SiteWise 운영합니다.

AWS 지역 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하십시오.

AWS 글로벌 인프라 외에도 데이터 복원력 및 백업 요구 사항을 지원하는 데 도움이 되는 몇 가지 기능을 AWS IoT SiteWise 제공합니다.

- MQTT 메시지를 AWS IoT Core 통해 속성 값 업데이트를 게시한 다음 해당 데이터에 적용되는 규칙을 구성할 수 있습니다. 이 기능을 사용하면 Amazon S3 및 Amazon DynamoDB와 같은 다른 AWS 서비스에 데이터를 백업할 수 있습니다. 자세한 내용은 [다른 AWS 서비스와의 상호작용](#) 및 [자산 속성 알림과 함께 Amazon S3로 데이터 내보내기](#) 섹션을 참조하세요.
- AWS IoT SiteWise Get*API를 사용하여 과거 자산 자산 자산 데이터를 검색하고 백업할 수 있습니다. 자세한 내용은 [과거 자산 속성 값 쿼리](#) 섹션을 참조하세요.

- AWS IoT SiteWise Describe*API를 사용하여 자산 및 모델과 같은 리소스에 대한 정의를 검색할 수 있습니다. 이러한 정의를 백업해 두었다가 나중에 이것으로 리소스를 다시 만들 수 있습니다. 자세한 내용은 [AWS IoT SiteWise API 참조](#)를 참조하세요.

의 인프라 보안 AWS IoT SiteWise

관리형 서비스로서 AWS 글로벌 네트워크 보안으로 AWS IoT SiteWise 보호됩니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 [AWS 클라우드 보안을](#) 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 [인프라 보호](#)를 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 AWS IoT SiteWise 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service](#)(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

SiteWise 에서 실행되는 에지 게이트웨이는 X.509 인증서와 암호화 키를 사용하여 클라우드에 AWS IoT Greengrass 연결하고 인증합니다. AWS 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 [AWS IoT Greengrass에 대한 디바이스 인증 및 권한 부여](#)를 참조하세요.

구성 및 취약성 분석

IoT 플릿은 다양한 기능을 수행하고 장기적으로 사용되며 지리적으로 분산된 다수의 디바이스로 구성될 수 있습니다. 이러한 특성으로 인해 플릿 설정이 복잡해지고 오류가 발생하기 쉬워집니다. 장치는 일반적으로 처리 능력, 메모리 및 스토리지가 제한되어 있기 때문에 암호화 및 기타 보안 조치를 항상 지원할 수는 없습니다. 또한 장치에서 알려진 취약성이 있는 소프트웨어를 사용하는 경우도 있습니다. 이러한 요소로 인해 IoT 플릿이 해커의 매력적인 대상이 되며, 디바이스 플릿을 지속적으로 보호하기 어렵게 됩니다.

AWS IoT Device Defender 보안 문제와 모범 사례와의 편차를 식별할 수 있는 도구를 제공하여 이러한 문제를 해결합니다. 연결된 장치를 분석, 감사 및 AWS IoT Device Defender 모니터링하여 비정상적인 동작을 탐지하고 보안 위험을 완화하는 데 사용합니다. AWS IoT Device Defender 디바이스 폴릿을 감사하여 보안 모범 사례를 준수하고 디바이스의 비정상 동작을 탐지하는지 확인할 수 있습니다. 이를 통해 디바이스 전체에 일관된 보안 정책을 적용하고 AWS IoT 디바이스가 손상될 경우 신속하게 대응할 수 있습니다. 자세한 내용은 AWS IoT 개발자 안내서의 [AWS IoT Device Defender](#)을 참조하세요.

SiteWise 에지 게이트웨이를 사용하여 서비스에 데이터를 수집하는 경우 SiteWise 에지 게이트웨이 환경을 구성하고 유지 관리하는 것은 사용자의 책임입니다. 이 책임에는 SiteWise Edge Gateway의 시스템 소프트웨어, AWS IoT Greengrass 소프트웨어 및 커넥터를 최신 버전으로 업그레이드하는 AWS IoT SiteWise 것이 포함됩니다. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서 및 [의 AWS IoT Greengrass 코어 구성을 참조하십시오 커넥터 업그레이드](#).

VPC 엔드포인트

인터페이스 VPC 엔드포인트는 가상 사설 클라우드 (VPC) 와 간에 사설 연결을 설정합니다. AWS IoT SiteWise [AWS PrivateLink](#) 인터페이스 엔드포인트에 전원을 공급하여 API 작업에 대한 프라이빗 액세스를 가능하게 합니다. AWS IoT SiteWise 인터넷 게이트웨이, NAT 디바이스, VPN 연결 등의 필요성을 우회할 수 있습니다. AWS Direct Connect VPC의 인스턴스는 AWS IoT SiteWise API 작업과 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다. VPC와 VPC 간의 트래픽이 AWS 네트워크를 AWS IoT SiteWise 벗어나지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 [탄력적 네트워크 인터페이스](#)로 표현됩니다.

에 대한 AWS IoT SiteWise 인터페이스 VPC 엔드포인트를 설정하기 전에 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트 속성 및 제한을](#) 검토하십시오.

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)를 참조하세요.

VPC 엔드포인트에 지원되는 API 작업

AWS IoT SiteWise VPC에서 다음 AWS IoT SiteWise API 작업에 대한 호출을 지원합니다.

- 모든 데이터 플레인 API 작업에 대해 다음 엔드포인트를 사용하십시오. 다음 *region* 엔드포인트로 바꾸십시오. AWS 리전

```
data.iotsitewise.region.amazonaws.com
```

데이터 영역 API 작업에는 다음이 포함됩니다.

- [BatchGetAssetProperty가치](#)
- [BatchGetAssetPropertyValueHistory](#)
- [BatchPutAssetProperty가치](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValue역사](#)
- [GetInterpolatedAssetProperty가치](#)
- 자산 모델, 자산, SiteWise 에지 게이트웨이, 태그 및 계정 구성을 관리하는 데 사용하는 컨트롤 플레인 API 작업의 경우 다음 엔드포인트를 사용하십시오. *region*를 사용자의 AWS 리전으로 교체합니다.

```
api.iotsitewise.region.amazonaws.com
```

지원되는 컨트롤 플레인 API 작업에는 다음이 포함됩니다.

- [AssociateAssets](#)
- [CreateAsset](#)
- [CreateAsset모델](#)
- [DeleteAsset](#)
- [DeleteAsset모델](#)
- [DeleteDashboard](#)
- [DescribeAsset](#)
- [DescribeAsset모델](#)
- [DescribeAsset재산](#)
- [DescribeDashboard](#)
- [DescribeLogging옵션](#)
- [DisassociateAssets](#)
- [ListAsset모델](#)
- [ListAsset관계](#)
- [ListAssets](#)

- [PutLogging옵션](#)
- [UpdateAsset](#)
- [UpdateAsset모델](#)
- [UpdateAsset재산](#)
- [CreateGateway](#)
- [DeleteGateway](#)
- [DescribeDefaultEncryptionConfiguration](#)
- [DescribeGateway](#)
- [DescribeGatewayCapabilityConfiguration](#)
- [DescribeStorage구성](#)
- [ListGateways](#)
- [ListTagsForResource](#)
- [UpdateGateway](#)
- [UpdateGatewayCapabilityConfiguration](#)
- [PutDefaultEncryptionConfiguration](#)
- [PutStorage구성](#)
- [TagResource](#)
- [UntagResource](#)

 Note

컨트롤 플레인 API 작업을 위한 인터페이스 VPC 엔드포인트는 현재 다음 SiteWise Monitor API 작업에 대한 호출을 지원하지 않습니다.

- [BatchAssociateProjectAssets](#)
- [BatchDisassociateProjectAssets](#)
- [CreateAccess정책](#)
- [CreateDashboard](#)
- [CreatePortal](#)
- [CreateProject](#)
- [DeleteAccess정책](#)

- [DeleteProject](#)
- [DescribeAccess](#) 정책
- [DescribePortal](#)
- [DescribeProject](#)
- [ListAccess](#) 정책
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjects](#)
- [ListProject](#) 자산
- [UpdateAccess](#) 정책
- [UpdateDashboard](#)
- [UpdatePortal](#)
- [UpdateProject](#)

AWS IoT SiteWise에 대한 인터페이스 VPC 엔드포인트 생성

AWS IoT SiteWise 서비스에 대한 VPC 엔드포인트를 생성하려면 Amazon VPC 콘솔 또는 () 를 사용하십시오. AWS Command Line Interface AWS CLI 자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트 생성](#)을 참조하세요.

다음 서비스 이름 중 하나를 AWS IoT SiteWise 사용하여 VPC 엔드포인트를 생성합니다.

- 데이터 플레인 API 작업의 경우 다음 서비스 이름을 사용하십시오.

```
com.amazonaws.region.iotsitewise.data
```

- 컨트롤 플레인 API 작업의 경우 다음 서비스 이름을 사용하십시오.

```
com.amazonaws.region.iotsitewise.api
```

인터페이스 VPC 엔드포인트를 AWS IoT SiteWise 통한 액세스

인터페이스 엔드포인트를 생성하면 통신에 사용할 수 있는 엔드포인트별 DNS 호스트 이름이 생성됩니다. AWS IoT SiteWise 프라이빗 DNS 옵션은 기본적으로 활성화되어 있습니다. 자세한 내용은 Amazon VPC 사용 설명서의 [프라이빗 호스팅 영역 사용](#)을 참조하세요.

엔드포인트에 프라이빗 DNS를 활성화하면 다음 VPC 엔드포인트 중 하나를 AWS IoT SiteWise 통해 API 요청을 보낼 수 있습니다.

- 데이터 플레인 API 작업의 경우 다음 엔드포인트를 사용하십시오. **###** 사용자 엔드포인트로 바꾸십시오. AWS 리전

```
data.iotsitewise.region.amazonaws.com
```

- 컨트롤 플레인 API 작업의 경우 다음 엔드포인트를 사용하십시오. **###** 사용자 것으로 바꾸십시오. AWS 리전.

```
api.iotsitewise.region.amazonaws.com
```

엔드포인트의 프라이빗 DNS를 비활성화한 경우 엔드포인트를 AWS IoT SiteWise 통해 액세스하려면 다음을 수행해야 합니다.

1. API 요청에서 VPC 엔드포인트 URL을 지정합니다.

- 데이터 플레인 API 작업의 경우 다음 엔드포인트 URL을 사용합니다. ***vpc-endpoint-id***와 **#**을 VPC 엔드포인트 ID 및 리전으로 바꾸십시오.

```
vpc-endpoint-id.data.iotsitewise.region.vpce.amazonaws.com
```

- 컨트롤 플레인 API 작업의 경우 다음 엔드포인트 URL을 사용합니다. ***vpc-endpoint-id***와 **#**을 VPC 엔드포인트 ID 및 리전으로 바꾸십시오.

```
vpc-endpoint-id.api.iotsitewise.region.vpce.amazonaws.com
```

2. 호스트 접두사 삽입을 비활성화합니다. AWS CLI 및 AWS SDK는 각 API 작업을 호출할 때 서비스 엔드포인트 앞에 다양한 호스트 접두사를 추가합니다. 이 기능을 사용하면 VPC 엔드포인트를 지정할 AWS IoT SiteWise 때 AWS CLI 및 AWS SDK가 유효하지 않은 URL을 생성합니다.

⚠ Important

또는 에서 호스트 접두사 삽입을 비활성화할 수 없습니다. AWS CLI AWS Tools for PowerShell 즉, 프라이빗 DNS를 비활성화하면 이러한 도구를 사용하여 VPC 엔드포인트를 AWS IoT SiteWise 통해 액세스할 수 없습니다. 프라이빗 DNS가 AWS CLI 를 사용하거나 를 사용하여 엔드포인트를 AWS IoT SiteWise 통해 액세스할 수 AWS Tools for PowerShell 있도록 합니다.

SDK에서 호스트 접두사 삽입을 비활성화하는 방법에 대한 자세한 내용은 각 AWS SDK의 다음 설명서 섹션을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto3\)](#)
- [AWS SDK for Ruby](#)

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트를 통해 서비스 액세스](#)를 참조하십시오.

에 대한 VPC 엔드포인트 정책 생성 AWS IoT SiteWise

AWS IoT SiteWise에 대한 액세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업.
- 작업을 수행할 있는 리소스.

자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하세요.

예: 작업에 대한 VPC 엔드포인트 정책 AWS IoT SiteWise

다음은 에 대한 AWS IoT SiteWise 엔드포인트 정책의 예입니다. 엔드포인트에 연결할 경우 이 정책은 지정된 *iotsitewiseadmin* 자산에서 AWS 계정 *123456789012#* 나열된 AWS IoT SiteWise 작업에 대한 액세스 권한을 사용자에게 부여합니다.

```
{
  "Statement": [
    {
      "Action": [
        "iotsitewise:CreateAsset",
        "iotsitewise:ListGateways",
        "iotsitewise:ListTagsForResource"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "Principal": {
        "AWS": [
          "123456789012:user/iotsitewiseadmin"
        ]
      }
    }
  ]
}
```

에 대한 보안 모범 사례 AWS IoT SiteWise

이 항목에는 에 대한 보안 모범 사례가 포함되어 AWS IoT SiteWise 있습니다.

OPC-UA 서버에서 인증 보안 인증 사용

OPC-UA 서버에 연결할 때 인증 보안 인증을 요구합니다. 방법은 서버 설명서를 참조하세요. 그런 다음 SiteWise 에지 게이트웨이가 OPC-UA 서버에 연결할 수 있도록 하려면 SiteWise 에지 게이트웨이 에 서버 인증 암호를 추가하십시오. 자세한 정보는 [원본 인증 구성](#)을 참조하세요.

OPC-UA 서버에 암호화된 통신 모드 사용

Edge 게이트웨이에 OPC-UA 소스를 구성할 때는 더 이상 사용되지 않는 암호화된 메시지 보안 모드를 선택하십시오. SiteWise 이는 OPC-UA 서버에서 Edge 게이트웨이로 이동하는 산업 데이터를 보호하는 데 도움이 됩니다. SiteWise 자세한 내용은 [로컬 네트워크를 통해 전송 중인 데이터 및 데이터 소스 구성](#) 섹션을 참조하세요.

구성 요소를 최신 상태로 유지

SiteWise 에지 게이트웨이를 사용하여 서비스에 데이터를 수집하는 경우 에지 게이트웨이 환경을 구성하고 유지 관리하는 것은 사용자의 책임입니다. SiteWise 이러한 책임에는 게이트웨이의 시스템 소프트웨어, AWS IoT Greengrass 소프트웨어 및 커넥터를 최신 버전으로 업그레이드하는 것이 포함됩니다.

Note

AWS IoT SiteWise 에지 커넥터는 파일 시스템에 비밀을 저장합니다. 이러한 비밀은 SiteWise 에지 게이트웨이 내에 캐시된 데이터를 볼 수 있는 사용자를 제어합니다. Edge 게이트웨이를 실행하는 시스템의 디스크 또는 파일 시스템 암호화를 활성화하는 것이 좋습니다. SiteWise

SiteWise 에지 게이트웨이의 파일 시스템을 암호화하세요.

에지 게이트웨이를 암호화하고 보호하여 SiteWise 에지 게이트웨이를 통해 이동하는 산업 데이터를 안전하게 보호할 수 있습니다. SiteWise SiteWise 에지 게이트웨이에 하드웨어 보안 모듈이 있는 경우 SiteWise 에지 게이트웨이를 AWS IoT Greengrass 보호하도록 구성할 수 있습니다. 자세한 정보는 AWS IoT Greengrass Version 1 개발자 안내서에서 [하드웨어 보안 통합](#)을 참조하세요. 그렇지 않으면 운영 체제 설명서를 참조하여 파일 시스템을 암호화하고 보호하는 방법을 알아보십시오.

엣지 구성에 대한 보안 액세스

에지 콘솔 애플리케이션 비밀번호나 SiteWise 모니터 애플리케이션 비밀번호를 공유하지 마세요. 이 암호를 다른 사람이 볼 수 있는 곳에 두지 마십시오. 암호에 적절한 만료 기간을 설정하여 건전한 암호 교체 정책을 구현하십시오.

SiteWise 모니터 사용자에게 가능한 최소 권한을 부여하십시오.

포털 사용자에 대한 최소 액세스 정책 권한 집합을 사용하여 최소 권한의 원칙을 따르십시오.

- 포털을 만들 때 해당 포털에 필요한 최소한의 자산 집합을 허용하는 역할을 정의합니다. 자세한 내용은 [서비스 역할 사용 대상 AWS IoT SiteWise Monitor](#) 섹션을 참조하세요.
- 사용자와 포털 관리자가 프로젝트를 만들고 공유할 때 해당 프로젝트에 필요한 최소한의 자산 집합을 사용합니다.
- 포털 또는 프로젝트에 더 이상 액세스할 필요가 없는 ID는 해당 리소스에서 제거합니다. 해당 ID를 조직에 더 이상 적용할 수 없는 경우 ID 저장소에서 해당 ID를 삭제하십시오.

최소 원칙의 모범 사례는 IAM 역할에도 적용됩니다. 자세한 내용은 [정책 모범 사례](#) 섹션을 참조하세요.

민감한 정보 노출 금지

보안 인증 및 개인 식별 정보(PII)와 같은 기타 민감한 정보의 로깅을 방지해야 합니다. SiteWise 에지 게이트웨이의 로컬 로그에 액세스하려면 루트 권한이 필요하고 로그에 액세스하려면 IAM 권한이 필요하지만 다음 보호 장치를 구현하는 것이 CloudWatch 좋습니다.

- 자산 또는 모델의 이름, 설명 또는 속성에 민감한 정보를 사용하지 마십시오.
- SiteWise 에지 게이트웨이 또는 소스 이름에는 민감한 정보를 사용하지 마십시오.
- 포털, 프로젝트 또는 대시보드의 이름이나 설명에 민감한 정보를 사용하지 마십시오.

AWS IoT Greengrass 보안 베스트 프랙티스를 따르세요.

SiteWise 에지 게이트웨이의 AWS IoT Greengrass 보안 모범 사례를 따르세요. 자세한 내용은 AWS IoT Greengrass Version 1 개발자 안내서의 [보안 모범 사례](#)를 참조하세요.

다음 사항도 참조하세요.

- AWS IoT 개발자 안내서의 [보안 모범 사례](#)
- [산업용 IoT 솔루션을 위한 10가지 보안 골든 룰](#)

로그인 및 모니터링 AWS IoT SiteWise

모니터링은 및 기타 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. AWS IoT SiteWise AWS IoT SiteWise 다음 모니터링 도구를 지원하여 서비스를 관찰하고, 문제 발생 시 보고하고, 적절한 경우 자동 조치를 취합니다.

- Amazon은 실행 중인 AWS 리소스와 애플리케이션을 AWS 실시간으로 CloudWatch 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 만들고, 지정된 지표가 특정 임계값에 도달하면 알리거나 조치를 취하는 경보를 설정합니다. 예를 들어 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 CloudWatch 추적하고 필요할 때 새 인스턴스를 자동으로 시작할 수 있습니다. 자세한 내용은 [Amazon CloudWatch 사용 설명서](#)를 참조하십시오.
- Amazon CloudWatch Logs는 SiteWise 에지 게이트웨이 및 기타 소스에서 로그 파일을 모니터링 CloudTrail, 저장 및 액세스합니다. CloudWatch 로그는 로그 파일의 정보를 모니터링하고 특정 임계값이 충족되면 알려줄 수 있습니다. 또한 매우 내구력 있는 스토리지에 로그 데이터를 저장할 수 있습니다. 자세한 내용은 [Amazon CloudWatch Logs 사용 설명서](#)를 참조하십시오.
- AWS CloudTrail계정에서 또는 AWS 계정을 대신하여 이루어진 API 호출 및 관련 이벤트를 캡처합니다. 그런 CloudTrail 다음 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 어떤 사용자와 계정이 전화를 걸었는지 AWS, 어떤 소스 IP 주소에서 호출이 이루어졌는지, 언제 호출이 발생했는지 확인할 수 있습니다. 자세한 내용은 [AWS CloudTrail 사용 설명서](#)를 참조하세요.

주제

- [Amazon CloudWatch 로그를 사용한 모니터링](#)
- [SiteWise 에지 게이트웨이 로그 모니터링](#)
- [Amazon AWS IoT SiteWise CloudWatch 메트릭을 사용한 모니터링](#)
- [를 AWS IoT SiteWise 사용하여 API 호출 로깅 AWS CloudTrail](#)

Amazon CloudWatch 로그를 사용한 모니터링

서비스를 모니터링하고 문제를 해결하기 위해 정보를 CloudWatch 로그에 기록하도록 구성합니다 AWS IoT SiteWise .

AWS IoT SiteWise 콘솔을 사용하면 서비스가 사용자를 대신하여 정보를 기록할 수 있는 서비스 연결 역할을 AWS IoT SiteWise 생성합니다. AWS IoT SiteWise 콘솔을 사용하지 않는 경우 로그를 받으려면 서비스 연결 역할을 수동으로 만들어야 합니다. 자세한 정보는 [AWS IoT SiteWise에 대한 서비스 링크 역할 생성](#)을 참조하세요.

로그 이벤트를 AWS IoT SiteWise 스트림에 넣을 수 있는 리소스 정책이 있어야 합니다 CloudWatch . CloudWatch 로그에 대한 리소스 정책을 만들고 업데이트하려면 다음 명령을 실행합니다. 생성할 정책 *logging-policy-name* 이름으로 바꿉니다.

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\": \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service\": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\": \"*\" } ] }"
```

CloudWatch 로그는 [aws: SourceArn](#) 및 [aws: SourceAccount](#) 조건 컨텍스트 키도 지원합니다. 이러한 조건 컨텍스트 키는 선택 사항입니다.

지정된 리소스와 관련된 로그만 AWS IoT SiteWise CloudWatch 스트림에 넣을 수 있는 AWS IoT SiteWise 리소스 정책을 만들거나 업데이트하려면 명령을 실행하고 다음을 수행하십시오.

- 생성할 정책 *logging-policy-name* 이름으로 바꿉니다.
- *Source-ARN#* AWS IoT SiteWise 리소스 (예: 자산 모델 또는 자산) 의 ARN으로 바꾸십시오. 각 AWS IoT SiteWise 리소스 유형의 ARN을 찾으려면 서비스 권한 부여 AWS IoT SiteWise [참조에 정의된 리소스 유형](#)을 참조하십시오.
- *## ID# ### #####* 연결된 AWS 계정 ID로 바꾸십시오. AWS IoT SiteWise

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\": \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service\": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\": \"*\", \"Condition\": { \"StringLike\": { \"aws:SourceArn\": [\"source-ARN\"], \"aws:SourceAccount\": [\"account-ID\"] } } } ] }"
```

기본적으로 로그에 CloudWatch 정보를 기록하지 AWS IoT SiteWise 않습니다. 로깅을 활성화하려면 Disabled (OFF) 이외의 다른 로깅 수준을 선택합니다. AWS IoT SiteWise 다음과 같은 로깅 수준을 지원합니다.

- OFF – 로깅이 해제되어 있습니다.
- ERROR – 오류를 기록합니다.
- INFO – 오류 및 정보 메시지를 기록합니다.

CloudWatch Logs를 통해 AWS IoT Greengrass정보를 기록하도록 SiteWise Edge 게이트웨이를 구성할 수 있습니다. 자세한 정보는 [SiteWise 에지 게이트웨이 로그 모니터링](#)을 참조하세요.

AWS IoT SiteWise 규칙 동작의 문제를 해결하는 경우 CloudWatch 로그에 정보를 AWS IoT Core 기록하도록 구성할 수도 있습니다. 자세한 정보는 [AWS IoT SiteWise 규칙 조치 문제 해결](#)을 참조하세요.

목차

- [로그인 관리 AWS IoT SiteWise](#)
 - [로그 수준 찾기](#)
 - [로그 수준 변경](#)
- [예: AWS IoT SiteWise 로그 파일 항목](#)

로그인 관리 AWS IoT SiteWise

AWS IoT SiteWise 콘솔을 사용하거나 AWS CLI 다음 로깅 구성 작업에 사용하십시오.

로그 수준 찾기

Console

AWS IoT SiteWise 콘솔에서 현재 로깅 수준을 찾아보려면 다음 절차를 따르십시오.

현재 AWS IoT SiteWise 로깅 수준을 확인하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 로깅 옵션을 선택합니다.

현재의 로깅 상태가 로깅 상태 아래에 나타납니다. 로깅을 활성화하면 현재 로깅 수준이 세부 사항 수준 아래에 나타납니다.

AWS CLI

다음 명령을 실행하여 에서 현재 AWS IoT SiteWise 로깅 수준을 찾으십시오 AWS CLI.

```
aws iotsitewise describe-logging-options
```

이 작업은 로깅 수준이 포함된 응답을 다음 형식으로 반환합니다.

```
{
```

```

    "loggingOptions": {
      "level": "String"
    }
  }
}

```

로깅 수준 변경

AWS IoT SiteWise 콘솔 또는 에서 다음 절차를 사용하여 로깅 수준을 변경합니다 AWS CLI.

Console

AWS IoT SiteWise 로깅 수준을 변경하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 로깅 옵션을 선택합니다.
3. 편집을 선택합니다.
4. 활성화할 세부 사항 수준을 선택합니다.
5. 저장을 선택합니다.

AWS CLI

다음 AWS CLI 명령을 실행하여 AWS IoT SiteWise 로깅 수준을 변경합니다. *logging-level*을 원하는 로깅 수준으로 바꿉니다.

```
aws iotsitewise put-logging-options --logging-options level=logging-level
```

예: AWS IoT SiteWise 로그 파일 항목

각 AWS IoT SiteWise 로그 항목에는 이벤트 정보와 해당 이벤트의 관련 리소스가 포함되므로 로그 데이터를 이해하고 분석할 수 있습니다.

다음 예제는 자산 모델을 성공적으로 생성할 때 AWS IoT SiteWise 기록되는 CloudWatch 로그 항목을 보여줍니다.

```

{
  "eventTime": "2020-05-05T00:10:22.902Z",
  "logLevel": "INFO",

```

```

"eventType": "AssetModelCreationSuccess",
"message": "Successfully created asset model.",
"resources": {
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
}

```

SiteWise 에지 게이트웨이 로그 모니터링

Amazon CloudWatch Logs 또는 로컬 파일 시스템에 정보를 기록하도록 AWS IoT SiteWise Edge 게이트웨이를 구성할 수 있습니다.

주제

- [아마존 CloudWatch 로그 사용](#)
- [서비스 로그 사용](#)
- [이벤트 로그 사용](#)

아마존 CloudWatch 로그 사용

로그를 Logs로 전송하도록 SiteWise 엣지 게이트웨이를 구성할 수 있습니다. CloudWatch 자세한 [내용은 AWS IoT Greengrass Version 2 개발자 안내서의 CloudWatch 로그 로깅 활성화를 참조하십시오.](#)

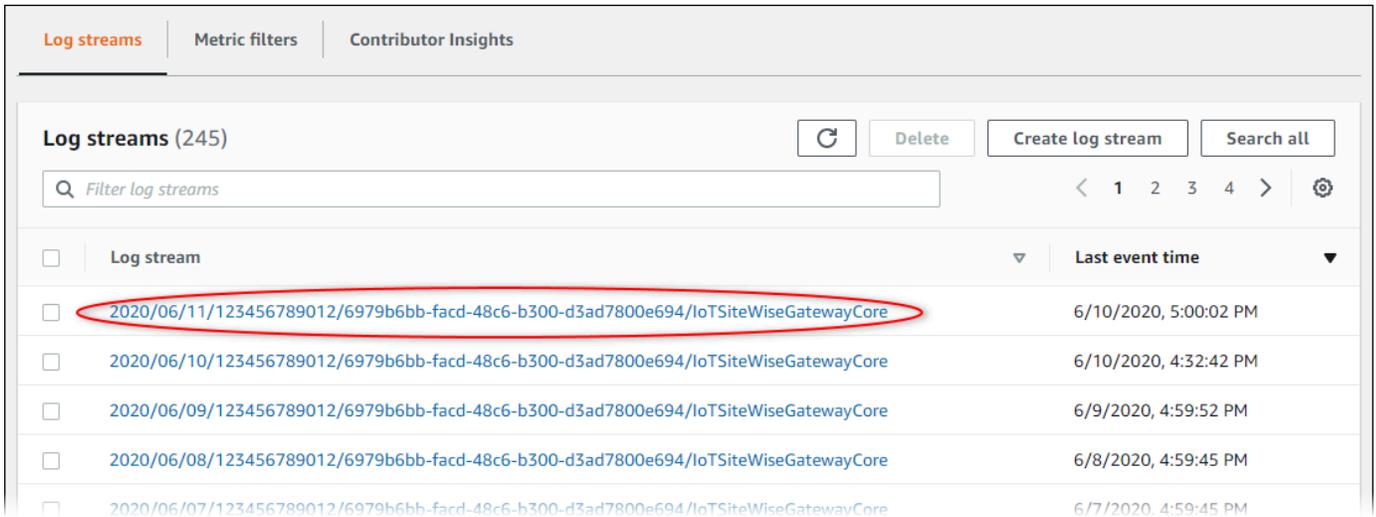
CloudWatch 로그 구성 및 액세스하기 (콘솔)

1. [CloudWatch 콘솔](#)로 이동합니다.
2. 탐색 창에서 로그 그룹을 선택합니다.
3. AWS IoT SiteWise 구성 요소 로그는 다음 로그 그룹에서 찾을 수 있습니다.
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgeCollector0pcua`— SiteWise 에지 게이트웨이의 OPC-UA 소스에서 데이터를 수집하는 SiteWise 에지 게이트웨이 구성 요소에 대한 로그입니다.
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgePublisher`— OPC-UA 데이터 스트림을 게시하는 SiteWise 에지 게이트웨이 구성 요소에 대한 로그입니다.

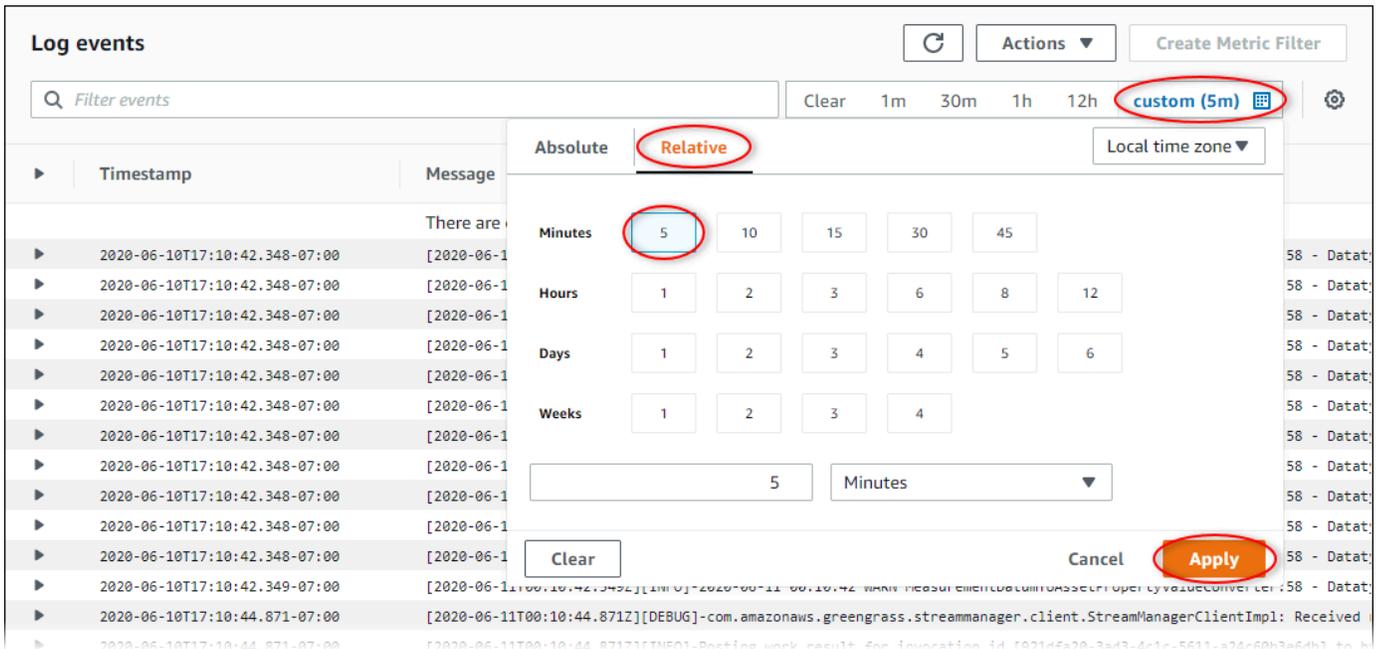
AWS IoT SiteWise

디버깅할 함수의 로그 그룹을 선택합니다.

4. 이름이 그룹 이름으로 끝나는 로그 스트림을 선택하십시오. AWS IoT Greengrass 기본적으로 가장 최근의 로그 스트림을 먼저 CloudWatch 표시합니다.



5. 최근 5분 동안의 로그를 표시하려면 다음을 수행합니다.
 - a. 오른쪽 상단 모서리에서 사용자 지정을 선택합니다.
 - b. 상대를 선택합니다.
 - c. 5분을 선택합니다.
 - d. 적용을 선택합니다.



6. (선택 사항) 더 적은 수의 로그를 보려면 오른쪽 상단 모서리에서 1분을 선택하면 됩니다.

7. 가장 최근 로그를 표시하려면 로그 항목의 맨 아래로 스크롤합니다.

서비스 로그 사용

SiteWise 에지 게이트웨이 디바이스에는 문제를 디버깅하는 데 도움이 되는 서비스 로그 파일이 포함되어 있습니다. 다음 섹션은 AWS IoT SiteWise OPC-UA Collector 및 AWS IoT SiteWise Publisher 구성 요소에 대한 서비스 로그 파일을 찾고 활용하는 데 도움이 됩니다.

AWS IoT SiteWise OPC-UA 컬렉터 서비스 로그 파일

AWS IoT SiteWise OPC-UA 컬렉터 구성요소는 다음 로그 파일을 사용합니다.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgeCollector0pcua.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollector0pcua.log
```

이 구성 요소의 로그를 보려면

- 코어 기기에서 다음 명령을 실행하여 이 구성 요소의 로그 파일을 실시간으로 확인합니다. */greengrass/v2* 또는 *C:\greengrass\v2* 를 AWS IoT Greengrass 루트 폴더 경로로 바꿉니다.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgeCollector0pcua.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollector0pcua.log -Tail 10 -Wait
```

AWS IoT SiteWise 퍼블리셔 서비스 로그 파일

AWS IoT SiteWise Publisher 구성 요소는 다음 로그 파일을 사용합니다.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log
```

이 구성 요소의 로그를 보려면

- 코어 기기에서 다음 명령을 실행하여 이 구성 요소의 로그 파일을 실시간으로 확인합니다. */greengrass/v2* 또는 *C:\greengrass\v2* 를 AWS IoT Greengrass 루트 폴더 경로로 바꿉니다.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log -Tail 10 -Wait
```

이벤트 로그 사용

SiteWise 에지 게이트웨이 디바이스에는 문제를 디버깅하는 데 도움이 되는 이벤트 로그 파일이 포함되어 있습니다. 다음 섹션은 AWS IoT SiteWise OPC-UA 컬렉터 및 AWS IoT SiteWise 게시자 구성 요소에 대한 이벤트 로그 파일을 찾고 활용하는 데 도움이 됩니다.

AWS IoT SiteWise OPC-UA 컬렉터 이벤트 로그

AWS IoT SiteWise OPC-UA Collector 구성요소에는 고객이 문제를 식별하고 해결하는 데 도움이 되는 이벤트 로그가 포함되어 있습니다. 로그 파일은 로컬 로그 파일과는 별개이며 다음 위치에서 찾을 수 있습니다. */greengrass/v2* 또는 *C:\greengrass\v2* 를 AWS IoT Greengrass 루트 폴더 경로로 바꿉니다.

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua/logs/
IotSiteWiseOpcUaCollectorEvents.log
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeCollectorOpcua\logs
\IotSiteWiseOpcUaCollectorEvents.log
```

이 로그에는 자세한 정보와 문제 해결 지침이 포함되어 있습니다. 문제 해결 정보는 문제 해결 방법에 대한 설명과 함께 진단과 함께 제공되며 경우에 따라 추가 정보로 연결되는 링크와 함께 제공됩니다. 진단 정보에는 다음이 포함됩니다.

- 심각도 수준
- Timestamp
- 추가 이벤트별 정보

Example 로그 예

```
dataSourceConnectionSuccess:
  Summary: Successfully connected to OpcUa server
  Level: INFO
  Timestamp: '2023-06-15T21:04:16.303Z'
  Description: Successfully connected to the data source.
  AssociatedMetrics:
  - Name: FetchedDataStreams
    Description: The number of fetched data streams for this data source
    Value: 1.0
    Namespace: IoTSiteWise
    Dimensions:
    - Name: SourceName
      Value: SourceName{value=OPC-UA Server}
    - Name: ThingName
      Value: test-core
  AssociatedData:
  - Name: DataSourceTrace
    Description: Name of the data source
    Data:
    - OPC-UA Server
```

```
- Name: EndpointUri
  Description: The endpoint to which the connection was attempted.
  Data:
  - '"opc.tcp://10.0.0.1:1234"'
```

AWS IoT SiteWise 퍼블리셔 이벤트 로그

AWS IoT SiteWise Publisher 구성 요소에는 고객이 문제를 식별하고 해결하는 데 도움이 되는 이벤트 로그가 포함되어 있습니다. 로그 파일은 로컬 로그 파일과는 별개이며 다음 위치에서 찾을 수 있습니다. `/greengrass/v2` 또는 `C:\greengrass\v2` 를 AWS IoT Greengrass 루트 폴더 경로로 바꿉니다.

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/logs/
IotSiteWisePublisherEvents.log
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgePublisher\logs
\IotSiteWisePublisherEvents.log
```

이 로그에는 자세한 정보와 문제 해결 지침이 포함되어 있습니다. 문제 해결 정보는 문제 해결 방법에 대한 설명과 함께 진단과 함께 제공되며 경우에 따라 추가 정보로 연결되는 링크와 함께 제공됩니다. 진단 정보에는 다음이 포함됩니다.

- 심각도 수준
- Timestamp
- 추가 이벤트별 정보

Example 로그 예

```
accountBeingThrottled:
  Summary: Data upload speed slowed due to quota limits
  Level: WARN
  Timestamp: '2023-06-09T21:30:24.654Z'
  Description: The IoT SiteWise Publisher is limited to the "Rate of data points
  ingested"
```

quota for a customer's account. See the associated documentation and associated metric for the number of requests that were limited for more information. Note that this may be temporary and not require any change, although if the issue continues

you may need to request an increase for the mentioned quota.

FurtherInformation:

- <https://docs.aws.amazon.com/iot-sitewise/latest/userguide/quotas.html>
- <https://docs.aws.amazon.com/iot-sitewise/latest/userguide/troubleshooting-gateway.html#gateway-issue-data-streams>

AssociatedMetrics:

- Name: TotalErrorCount

Description: The total number of errors of this type that occurred.

Value: 327724.0

AssociatedData:

- Name: AggregatePropertyAliases

Description: The aggregated property aliases of the throttled data.

FileLocation: /greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/./logs/data/AggregatePropertyAliases_1686346224654.log

Amazon AWS IoT SiteWise CloudWatch 메트릭을 사용한 모니터링

원시 데이터를 수집하여 읽을 수 있는 거의 실시간 지표로 처리하는 것을 AWS IoT SiteWise 사용하여 CloudWatch 모니터링할 수 있습니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임계값을 주시하다가 해당 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 [Amazon CloudWatch 사용 설명서](#)를 참조하십시오.

AWS IoT SiteWise 아래 섹션에 나열된 지표와 차원을 AWS/IoTSiteWise 네임스페이스에 게시합니다.

Tip

AWS IoT SiteWise 1분 간격으로 지표를 게시합니다. CloudWatch 콘솔에서 이러한 지표를 그래프로 볼 때는 기간을 1분으로 선택하는 것이 좋습니다. 이렇게 하면 지표 데이터를 가능한 최고 해상도로 볼 수 있습니다.

주제

- [AWS IoT Greengrass Version 2 게이트웨이 지표](#)
- [AWS IoT Greengrass Version 1 게이트웨이 메트릭](#)

AWS IoT Greengrass Version 2 게이트웨이 지표

AWS IoT SiteWise 다음 SiteWise 에지 게이트웨이 메트릭을 게시합니다. 모든 SiteWise Edge 게이트웨이 지표는 1분 간격으로 게시됩니다.

SiteWise 에지 게이트웨이 메트릭

지표	설명
Gateway.CpuUsage	SiteWise 에지 게이트웨이의 CPU 사용량. 단위: 퍼센트 측정 기준: 없음
Gateway.TotalDiskSpace	SiteWise 에지 게이트웨이의 총 디스크 공간. 단위: 바이트 측정 기준: 없음
Gateway.UsedDiskSpace	SiteWise 에지 게이트웨이의 사용된 디스크 공간. 단위: 바이트 측정 기준: 없음
Gateway.AvailableDiskSpace	SiteWise 에지 게이트웨이의 사용 가능한 디스크 공간. 단위: 바이트 측정 기준: 없음
Gateway.UsedPercentageDiskSpace	SiteWise 에지 게이트웨이의 디스크 공간 사용률. 단위: 바이트 측정 기준: 없음
Gateway.TotalMemory	SiteWise 에지 게이트웨이의 총 메모리.

지표	설명
	<p>단위: 바이트</p> <p>측정 기준: 없음</p>
Gateway.UsedMemory	<p>SiteWise 에지 게이트웨이의 사용된 메모리.</p> <p>단위: 바이트</p> <p>측정 기준: 없음</p>
Gateway.AvailableMemory	<p>SiteWise 에지 게이트웨이의 사용 가능한 메모리.</p> <p>단위: 바이트</p> <p>측정 기준: 없음</p>
Gateway.UsedPercentageMemory	<p>SiteWise 에지 게이트웨이의 메모리 사용률.</p> <p>단위: 바이트</p> <p>측정 기준: 없음</p>
Gateway.CloudConnectivity	<p>SiteWise 에지 게이트웨이의 클라우드 연결 상태.</p> <p>단위: 없음</p> <p>치수: GatewayId</p>
Gateway.SWE.Component.RunningStatus	<p>SiteWise 에지 게이트웨이의 구성 요소 실행 상태.</p> <p>단위: 없음</p> <p>치수: GatewayId</p>

OPC-UA 컬렉터 메트릭스

지표	설명
<code>OpcUaCollector.Heartbeat</code>	<p>SiteWise 에지 게이트웨이 () 에 연결된 각 OPC-UA 소스 (<code>sourceName</code>) 에 대해 1분마다 생성됩니다. <code>gatewayId</code></p> <p>단위: 개수 (1은 소스가 연결되었음을 나타내고 0은 소스가 연결되지 않았음을 나타냅니다.)</p> <p>크기: <code>GatewayId SourceName</code></p>
<code>OpcUaCollector.ActiveDataStreamCount</code>	<p>SiteWise 에지 게이트웨이 (<code>gatewayId</code>) 가 OPC-UA 소스에 대해 구독한 데이터 스트림 수 (). <code>sourceName</code></p> <p>단위: 수</p> <p>크기: <code>GatewayId SourceName PropertyGroup</code></p>
<code>OpcUaCollector.IncomingValuesCount</code>	<p>SiteWise Edge 게이트웨이 (<code>gatewayId</code>) 가 OPC-UA 소스 (<code>sourceName</code>) 에 대해 수신한 데이터 포인트의 수로, 1분마다 생성됩니다.</p> <p>단위: 수</p> <p>크기: <code>GatewayId SourceName PropertyGroup</code></p>
<code>OpcUaCollector.IncomingValuesError</code>	<p>SiteWise Edge 게이트웨이 (<code>gatewayId</code>) 가 OPC-UA 소스 (<code>sourceName</code>) 로부터 수신하지만 유효한 값이 아닌 데이터 포인트의 수. Collector는 이러한 데이터 포인트를 OpcUa 수집하지 않으며 1분마다 생성됩니다.</p> <p>단위: 수</p> <p>크기: <code>GatewayId, SourceName PropertyGroup</code></p>
<code>OpcUaCollector.ConversionErrors</code>	<p>SiteWise Edge Gateway (<code>gatewayId</code>) 가 OPC-UA 소스 (<code>sourceName</code>) 에 대해 수신하</p>

지표	설명
	<p>여 데이터를 보내는 동안 변환 오류가 발생한 데이터 포인트의 수입입니다. AWS IoT SiteWise Collector에서는 이러한 데이터 포인트를 수집하지 않습니다. OpcUa</p> <p>단위: 수</p> <p>크기: GatewayId SourceName</p>

AWS IoT SiteWise 프로세서 메트릭

지표	설명
Gateway.DataProcessor.IngestionSuccess	<p>성공적으로 수집되고 1분마다 생성되는 데이터 포인트의 수입입니다.</p> <p>단위: 수</p> <p>측정 기준: 없음</p>
Gateway.DataProcessor.IngestionThrottled	<p>병목 현상이 발생한 데이터 포인트의 수 (1분마다 생성됨).</p> <p>단위: 수</p> <p>치수: ThrottledAt</p>
Gateway.DataProcessor.MeasurementRejected	<p>거부된 측정 개수 (1분마다 생성됨)</p> <p>단위: 수</p> <p>치수: 이유</p>
Gateway.DataProcessor.MeasurementUnmodeled	<p>모델링되지 않고 1분마다 생성되는 측정값의 수입입니다.</p> <p>단위: 수</p> <p>치수: 이유</p>

지표	설명
<code>Gateway.DataProcessor.MessagesRemaining</code>	스트림에 남아 있는 메시지 수 (1분마다 생성됨) 단위: 수 크기: StreamName
<code>Gateway.DataProcessor.ProcessingError</code>	1분마다 생성되는 처리 오류 수입니다. 단위: 수 치수: 이유
<code>IoTSiteWiseProcessor.IsConnectedToMqttBroker</code>	SiteWise Edge 게이트웨이의 프로세서가 1분마다 생성합니다. 장치: 1 (프로세서를 나타내는 1은 MQTT 브로커에 연결됨) 크기: GatewayId
<code>IoTSiteWiseProcessor.NumberOfSubscriptionsToMqttBroker</code>	프로세서가 MQTT 브로커에 구독하는 토픽 수 (1분마다 생성됨) 다단계 와일드카드 주제는 1로 계산됩니다. 단위: 수 크기: GatewayId
<code>IoTSiteWiseProcessor.NumberOfUniqueMqttTopicsReceived</code>	프로세서가 MQTT 브로커로부터 수신한 고유 토픽 수 (1분마다 생성됨). 단위: 수 크기: GatewayId

지표	설명
<code>IoTSiteWiseProcessor.MqttMessageReceivedSuccessCount</code>	프로세서가 MQTT 브로커로부터 성공적으로 수신한 메시지 수 (1분마다 생성됨). 단위: 수 크기: GatewayId
<code>IoTSiteWiseProcessor.MqttReceivedSuccessBytes</code>	프로세서가 MQTT 브로커로부터 성공적으로 수신한 메시지 데이터의 바이트 수로, 1분마다 생성됩니다. 단위: 수 크기: GatewayId

AWS IoT SiteWise 퍼블리셔 지표

지표	설명
<code>IoTSiteWisePublisher.Heartbeat</code>	SiteWise Edge 게이트웨이에서 퍼블리셔가 1분마다 생성합니다. 단위: 1 (1은 게시자가 실행 중임을 나타내며 누락된 데이터 요소는 게시자가 실행되고 있지 않음을 나타냅니다.) 치수: GatewayId
<code>IoTSiteWisePublisher.PublishSuccessCount</code>	SiteWise Edge Gateway (GatewayId)가 클라우드에 성공적으로 게시하여 1분마다 생성한 데이터 포인트의 수입니다. 단위: 수 크기: GatewayId
<code>IoTSiteWisePublisher.PublishFailureCount</code>	SiteWise Edge Gateway (GatewayId)가 게시하지 못한 데이터 포인트의 수 (1분마다 생성).

지표	설명
	<p>단위: 수</p> <p>크기: GatewayId</p>
<code>IoTSiteWisePublisher.PublisherRejectedCount</code>	<p>SiteWise Edge Gateway (GatewayId) 가 클라우드 측에서 거부하여 1분마다 생성되는 데이터 포인트의 수입니다.</p> <p>단위: 수</p> <p>크기: GatewayId</p>
<code>IoTSiteWisePublisher.DroppedCount</code>	<p>SiteWise Edge Gateway (GatewayId) 에 의해 삭제되고 클라우드에 게시되지 않은 데이터 포인트의 수 (1분마다 생성됨).</p> <p>단위: 수</p> <p>크기: GatewayId</p>
<code>IoTSiteWisePublisher.IsConnectedToMqttBroker</code>	<p>SiteWise Edge 게이트웨이에서 퍼블리셔가 1분마다 생성합니다.</p> <p>유닛: 1 (퍼블리셔를 나타내는 1은 MQTT 브로커에 연결됨)</p> <p>크기: GatewayId</p>
<code>IoTSiteWisePublisher.NumberOfSubscriptionsToMqttBroker</code>	<p>퍼블리셔가 MQTT 브로커에 구독한 주제 수 (1분마다 생성됨). 다단계 와일드카드 주제는 1로 계산됩니다.</p> <p>단위: 수</p> <p>크기: GatewayId</p>

지표	설명
<code>IoTSiteWisePublisher.NumberOfUniqueMqttTopicsReceived</code>	<p>퍼블리셔가 MQTT 브로커로부터 받은 고유 주제 수 (1분마다 생성됨).</p> <p>단위: 수</p> <p>크기: GatewayId</p>
<code>IoTSiteWisePublisher.MqttMessageReceivedSuccessCount</code>	<p>퍼블리셔가 MQTT 브로커로부터 성공적으로 수신한 메시지 수 (1분마다 생성됨).</p> <p>단위: 수</p> <p>크기: GatewayId</p>
<code>IoTSiteWisePublisher.MqttReceivedSuccessBytes</code>	<p>게시자가 MQTT 브로커로부터 성공적으로 수신한 메시지 데이터의 바이트 수로, 1분마다 생성됩니다.</p> <p>단위: 수</p> <p>크기: GatewayId</p>

AWS IoT Greengrass Version 1 게이트웨이 메트릭

AWS IoT SiteWise 다음 SiteWise 에지 게이트웨이 메트릭을 게시합니다. 모든 SiteWise Edge 게이트웨이 지표는 1분 간격으로 게시됩니다.

Important

SiteWise 에지 게이트웨이 지표를 수신하려면 SiteWise 에지 게이트웨이의 AWS IoT SiteWise 커넥터 버전 6 이상을 사용해야 합니다. 자세한 내용을 알아보려면 AWS IoT Greengrass Version 1 개발자 가이드의 [AWS IoT SiteWise OPC-UA 수집기](#)를 참조하십시오.

SiteWise 에지 게이트웨이 메트릭

지표	설명
Gateway.Heartbeat	<p>연결된 각 SiteWise 에지 게이트웨이 (gatewayId)에 대해 1분마다 생성됩니다.</p> <p>단위: 1 (SiteWise 엣지 게이트웨이가 작동하고 에지 게이트웨이를 나타내는 데이터 포인트가 없음, SiteWise 클라우드와의 연결이 끊겼음을 나타내는 데이터 포인트 없음)</p> <p>치수: GatewayId</p>
Gateway.PublishSuccessCount	<p>SiteWise Edge Gateway (gatewayId)가 성공적으로 게시한 데이터 포인트의 수.</p> <p>단위: 수</p> <p>치수: GatewayId</p>
Gateway.PublishFailureCount	<p>SiteWise Edge gateway (gatewayId)가 게시하지 못한 데이터 포인트의 수입니다.</p> <p>이 지표는 BatchPutAssetPropertyValue작업에 대한 SiteWise Edge 게이트웨이 호출로 인해 발생한 오류를 계산합니다. SiteWise 에지 게이트웨이 문제 해결에 대한 자세한 내용은 을 참조하십시오. SiteWise 에지 게이트웨이 문제 해결</p> <p>단위: 수</p> <p>치수: GatewayId</p>
Gateway.ProcessFailureCount	<p>SiteWise Edge gateway (gatewayId)에서 처리하지 못한 데이터 포인트의 수입니다.</p> <p>이 지표는 소스에서 보고한 오류를 포함하여 SiteWise 에지 게이트웨이와 SiteWise 에지 게이트웨이 소스 간에 발생한 오류를 집계합니다. SiteWise 에지 게이트웨이 문제 해결에 대한 자</p>

지표	설명
	<p>세한 내용은 을 참조하십시오 SiteWise 에지 게이트웨이 문제 해결.</p> <p>단위: 수</p> <p>치수: GatewayId</p>
Gateway.PublishRejectedCount	<p>SiteWise Edge Gateway (gatewayId) 에서 거부된 데이터 포인트 수</p> <p>단위: 수</p> <p>치수: GatewayId</p>

PC-UA 관련 지표

지표	설명
OPCUACollector.Heartbeat	<p>SiteWise 에지 게이트웨이 () 에 연결된 각 OPC-UA 소스 (sourceName) 에 대해 1분마다 생성됩니다. gatewayId</p> <p>단위: 개수 (1은 소스가 연결되었음을 나타내고 0은 소스가 연결되지 않았음을 나타냅니다.)</p> <p>크기:, GatewayId SourceName</p>
OPCUACollector.ActiveDataStreamCount	<p>SiteWise 에지 게이트웨이 (gatewayId) 가 OPC-UA 소스에 대해 구독한 데이터 스트림 수 (). sourceName</p> <p>단위: 수</p> <p>크기:,, GatewayId SourceName PropertyGroup</p>
OpcUaCollector.IncomingValuesCount	<p>SiteWise Edge 게이트웨이 (gatewayId) 가 OPC-UA 소스 (sourceName) 에 대해 수신한 데이터 포인트의 수로, 1분마다 생성됩니다.</p>

지표	설명
	<p>단위: 수</p> <p>크기:,, GatewayId SourceName PropertyGroup</p>
OpcUaCollector.IncomingValuesError	<p>SiteWise Edge 게이트웨이 (gatewayId) 가 OPC-UA 소스 (sourceName) 로부터 수신했지만 유효한 값이 아닌 데이터 포인트의 수. 1분마다 생성되는 이러한 데이터 포인트는 OpcUa Collector에 의해 수집되지 않습니다.</p> <p>단위: 수</p> <p>크기: GatewayId,, SourceName PropertyGroup</p>
OpcUaCollector.ConversionErrors	<p>SiteWise Edge Gateway (gatewayId) 가 OPC-UA 소스 (sourceName) 에 대해 수신하여 데이터를 보내는 동안 변환 오류가 발생한 데이터 포인트의 수입니다. AWS IoT SiteWise Collector에서는 이러한 데이터 포인트를 수집하지 않습니다. OpcUa</p> <p>단위: 수</p> <p>크기:,, GatewayId SourceName</p>

EIP 관련 지표

지표	설명
EIPCollector.Heartbeat	<p>SiteWise 에지 게이트웨이 () 에 연결된 각 EIP 소스 (sourceName) 에 대해 1분마다 생성됩니다. gatewayId</p> <p>단위: 1(소스가 연결되었음을 나타내는 1이고 소스 연결이 끊겼음을 나타내는 데이터 포인트 누락)</p>

지표	설명
	크기:, GatewayId SourceName
EIPCollector.IncomingValuesCount	SiteWise Edge 게이트웨이 (gatewayId) 가 EIP 소스 () 에 대해 구독하는 데이터 스트림의 수 (sourceName). 단위: 수 크기:, GatewayId SourceName
EIPCollector.ActiveDataStreamCount	SiteWise Edge 게이트웨이 (gatewayId) 가 EIP 소스 (sourceName) 에 대해 수신한 데이터 포인트의 수. 단위: 수 크기:, GatewayId SourceName

Modbus 관련 지표

지표	설명
ModbusTCPCollector.Heartbeat	SiteWise 에지 게이트웨이 () 에 연결된 각 Modbus 소스 (sourceName) 에 대해 1분마다 생성됩니다. gatewayId 유닛: 1(Modbus 소스가 연결되어 있고 소스 연결이 끊겼음을 나타내는 데이터 포인트가 누락 됨) 크기:, GatewayId SourceName
ModbusTCPCollector.IncomingValuesCount	SiteWise 엣지 게이트웨이 (gatewayId) 가 Modbus 소스 () 에 구독하는 데이터 스트림 수. sourceName 단위: 수

지표	설명
	크기:, GatewayId SourceName
ModbusTCPCollector.ActiveDataStreamCount	SiteWise 엣지 게이트웨이 (gatewayId)가 Modbus 소스 (sourceName)에 대해 수신한 데이터 포인트의 수. 단위: 수 크기:, GatewayId SourceName

를 AWS IoT SiteWise 사용하여 API 호출 로깅 AWS CloudTrail

AWS IoT SiteWise 에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업의 기록을 제공하는 AWS 서비스와 통합됩니다 AWS IoT SiteWise. CloudTrail API 호출을 AWS IoT SiteWise 이벤트로 캡처합니다. 캡처된 호출에는 AWS IoT SiteWise 콘솔에서의 호출 및 AWS IoT SiteWise API 작업에 대한 코드 호출이 포함됩니다. 트레일을 생성하면 Amazon S3 버킷에 대한 이벤트 (에 대한 CloudTrail 이벤트 포함) 의 연속 전송을 활성화할 수 AWS IoT SiteWise 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 요청을 받은 사람 AWS IoT SiteWise, 요청한 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

에 대한 CloudTrail 자세한 내용은 [AWS CloudTrail 사용 설명서를](#) 참조하십시오.

AWS IoT SiteWise 에 대한 정보 CloudTrail

CloudTrail 계정을 만들 때 AWS 계정에서 활성화됩니다. 에서 AWS IoT SiteWise 지원되는 이벤트 활동이 발생하면 해당 활동이 CloudTrail 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다. 자세한 내용은 [이벤트 기록으로 CloudTrail 이벤트 보기를](#) 참조하십시오.

에 대한 이벤트를 포함하여 AWS 계정에서 진행 중인 이벤트의 기록을 보려면 AWS IoT SiteWise 트레일을 생성하세요. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 트레일을 생성하면 트레일이 모든 AWS 지역에 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원되는 서비스 및 통합](#)
- [에 대한 Amazon SNS 알림 구성 CloudTrail](#)
- [여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신](#)

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에게 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하십시오.

AWS IoT SiteWise 의 데이터 이벤트 CloudTrail

[데이터 이벤트](#)는 리소스 기반 또는 리소스에서 수행된 리소스 작업에 대한 정보를 제공합니다(예: Amazon S3 객체 읽기 또는 쓰기). 이를 데이터 영역 작업이라고도 합니다. 데이터 이벤트가 대량 활동인 경우도 있습니다. 기본적으로 데이터 이벤트를 기록하지 CloudTrail 않습니다. CloudTrail 이벤트 기록에는 데이터 이벤트가 기록되지 않습니다.

데이터 이벤트에는 추가 요금이 적용됩니다. CloudTrail 요금에 대한 자세한 내용은 [AWS CloudTrail 요금](#)을 참조하십시오.

CloudTrail 콘솔 또는 CloudTrail API 작업을 사용하여 AWS IoT SiteWise 리소스 유형에 대한 데이터 이벤트를 기록할 수 있습니다. AWS CLI 이 섹션의 [포에는](#) 사용 가능한 리소스 유형이 나와 AWS IoT SiteWise 있습니다.

- CloudTrail 콘솔을 사용하여 데이터 이벤트를 기록하려면 [트레일](#) 또는 [이벤트 데이터 저장소를 생성](#)하여 데이터 이벤트를 기록하거나 [기존 트레일 또는 이벤트 데이터 저장소를 업데이트하여](#) 데이터 이벤트를 기록하십시오.
 1. 데이터 이벤트를 선택하여 데이터 이벤트를 기록합니다.
 2. 데이터 이벤트 유형 목록에서 데이터 이벤트를 기록하려는 리소스 유형을 선택합니다.
 3. 사용하려는 로그 선택기 템플릿을 선택합니다. 리소스 유형에 대한 모든 데이터 이벤트를 기록하거나, 모든 이벤트를 기록하거나, 모든 readOnly 이벤트를 기록하거나

readOnlyeventName, 및 resources.ARN 필드를 기준으로 필터링할 사용자 지정 로그 선택기 템플릿을 만들 수 있습니다. writeOnly

- 를 사용하여 데이터 이벤트를 기록하려면 필드를 리소스 유형 값과 같게 설정하고 eventCategory 필드를 리소스 유형 값과 같게 Data 설정하도록 --advanced-event-selectors 매개 변수를 구성하십시오 ([표 참조](#)). AWS CLI resources.type 조건을 추가하여 readOnlyeventName, 및 resources.ARN 필드의 값을 기준으로 필터링할 수 있습니다.
- 데이터 이벤트를 기록하도록 트레일을 구성하려면 [AWS CloudTrail put-event-selectors](#) 명령을 실행합니다. 자세한 내용은 [를 사용한 트레일의 데이터 이벤트 로깅을 AWS CLI](#) 참조하십시오.
- 데이터 이벤트를 기록하도록 이벤트 데이터 저장소를 구성하려면 [AWS CloudTrail create-event-data-store](#) 명령을 실행하여 데이터 이벤트를 기록할 새 이벤트 데이터 저장소를 만들거나 [AWS CloudTrail update-event-data-store](#) 명령을 실행하여 기존 이벤트 데이터 저장소를 업데이트하십시오. 자세한 내용은 [를 사용한 이벤트 데이터 저장소의 데이터 이벤트 로깅을](#) 참조하십시오 AWS CLI.

다음 표에는 AWS IoT SiteWise 리소스 유형이 나열되어 있습니다. 데이터 이벤트 유형 (콘솔) 열에는 CloudTrail 콘솔의 데이터 이벤트 유형 목록에서 선택할 수 있는 값이 표시됩니다. resources.type 값 열에는 또는 resources.type API를 사용하여 고급 이벤트 선택기를 구성할 때 지정하는 값이 표시됩니다. AWS CLI CloudTrail 데이터 API 로깅 대상 CloudTrail 열에는 해당 리소스 유형에 대해 로깅된 API 호출이 표시됩니다. CloudTrail

데이터 이벤트 유형(콘솔)	resources.type 값	로깅된 데이터 API* CloudTrail
AWS IoT SiteWise 애셋	AWS::IoTSiteWise::Asset	<ul style="list-style-type: none"> • BatchPutAssetPropertyValues • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValues

데이터 이벤트 유형(콘솔)	resources.type 값	로깅된 데이터 API* CloudTrail
		<ul style="list-style-type: none"> • BatchGetAssetPropertyHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise 시계열	AWS::IoTSiteWise::TimeSeries	<ul style="list-style-type: none"> • BatchPutAssetPropertyValue • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValue • BatchGetAssetPropertyHistory • BatchGetAssetPropertyAggregates

Note

Cloudtrail 이벤트에 기록되는 resources.type은 API 요청에 사용된 식별자에 따라 달라집니다. 요청에 자산 ID가 지정되면 자산 resources.type이 기록되고, 그렇지 않으면 resources.type이 기록됩니다. TimeSeries

*고급 이벤트 선택터를 구성하여 eventName, readOnly, resources.ARN 필드를 기준으로 필터링하여 중요한 이벤트만 기록하도록 할 수 있습니다. 필드에 대한 자세한 내용은 [AdvancedFieldSelector](#) 섹션을 참조하세요.

AWS IoT SiteWise 의 관리 이벤트 CloudTrail

관리 이벤트는 AWS 계정의 리소스에서 수행되는 관리 작업에 대한 정보를 제공합니다. 이를 제어 영역 작업이라고도 합니다. 기본적으로 관리 이벤트를 CloudTrail 기록합니다.

AWS IoT SiteWise 모든 AWS IoT SiteWise 컨트롤 플레인 작업을 관리 이벤트로 기록합니다. AWS IoT SiteWise 로그되는 AWS IoT SiteWise 컨트롤 플레인 작업 목록은 [AWS IoT SiteWise API 참조를](#) 참조하십시오. CloudTrail

예: AWS IoT SiteWise 로그 파일 항목

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 CreateAsset 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Administrator",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Administrator",
    "sessionContext": {
      "sessionIssuer": {},
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-03-11T17:26:40Z"
      }
    }
  },
  "invokedBy": "signin.amazonaws.com"
},
{
  "eventTime": "2020-03-11T18:01:22Z",
  "eventSource": "iotsitewise.amazonaws.com",
  "eventName": "CreateAsset",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
```

```
"userAgent": "signin.amazonaws.com",
"requestParameters": {
  "assetName": "Wind Turbine 1",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-00000EXAMPLE"
},
"responseElements": {
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-east-1:123456789012:asset/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetStatus": {
    "state": "CREATING"
  }
},
"requestID": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
"eventID": "a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

리소스에 태그 지정하기 AWS IoT SiteWise

AWS IoT SiteWise 리소스에 태그를 지정하면 조직 자산을 효율적으로 분류, 관리 및 검색할 수 있는 강력한 방법이 제공됩니다. 키-값 쌍으로 구성된 태그를 할당하면 설명이 포함된 메타데이터를 리소스에 첨부할 수 있습니다. 태그의 메타데이터를 사용하여 작업을 간소화할 수 있습니다. 예를 들어 풍력 발전 시나리오에서 태그를 사용하면 위치, 용량, 운영 상태와 같은 특정 속성으로 터빈에 레이블을 지정할 수 있으므로 터빈에서 빠르게 식별하고 관리할 수 있습니다. AWS IoT SiteWise

태그를 AWS Identity and Access Management (IAM) 정책에 통합하면 조건부 액세스 규칙을 정의하여 보안 및 운영 제어를 강화할 수 있습니다. 즉, 특정 태그를 가진 사용자만 지정하도록 지정할 수 있습니다. 예를 들어 특정 역할이나 부서로 태그가 지정된 사용자만 특정 리소스에 액세스하거나 수정할 수 있습니다.

에서 태그 사용 AWS IoT SiteWise

태그를 사용하여 용도, 소유자, 환경 또는 사용 사례의 기타 분류에 따라 AWS IoT SiteWise 리소스를 분류할 수 있습니다. 동일한 유형의 리소스가 많은 경우, 할당한 태그를 기준으로 특정 리소스를 빠르게 식별할 수 있습니다.

각 태그는 사용자가 지정하는 키와 선택적 값으로 구성됩니다. 예를 들어 자산 모델에 일련의 태그를 설정하여 지원하는 산업 프로세스에 따라 자산 모델을 추적할 수 있습니다. 관리하는 각 리소스 유형에 맞는 맞춤형 태그 키 세트를 개발하는 것이 좋습니다. 일관된 태그 키 세트를 사용하면 리소스를 더 쉽게 관리할 수 있습니다.

를 사용하여 태그를 지정하십시오. AWS Management Console

이 태그 편집기는 모든 AWS 서비스의 리소스에 대한 태그를 만들고 관리할 수 있는 중앙의 통합 방법을 AWS Management Console 제공합니다. 자세한 내용을 알아보려면 AWS Resource Groups 사용 설명서의 [태그 편집기](#)를 참조하십시오.

API로 AWS IoT SiteWise 태그 지정

AWS IoT SiteWise API는 태그도 사용합니다. 태그를 생성하기 전에 태그 지정에 대한 제한 사항을 유의하십시오. 자세한 내용은 AWS 일반 참조의 [태그 이름 지정 및 사용 규칙](#)을 참조하십시오.

- 리소스를 만들 때 태그를 추가하려면 리소스의 tags 속성에서 태그를 정의합니다.
- 기존 리소스에 태그를 추가하거나 태그 값을 업데이트하려면 [TagResource](#) 작업을 사용하십시오.

- 리소스에서 태그를 제거하려면 [UntagResource](#) 작업을 사용합니다.
- 리소스와 연결된 태그를 검색하려면 [ListTagsForResource](#) 작업을 사용하거나 리소스를 설명하고 해당 tags 속성을 검사하십시오.

다음 표에는 AWS IoT SiteWise API를 사용하여 태그를 지정할 수 있는 리소스와 해당 Create 및 Describe 작업이 나열되어 있습니다.

태그가 가능한 리소스 AWS IoT SiteWise

Resource	작업 생성	작업 설명
자산 모델 또는 구성 요소 모델	CreateAssetModel	DescribeAssetModel
자산	CreateAsset	DescribeAsset
SiteWise 에지 게이트웨이	CreateGateway	DescribeGateway
Portal	CreatePortal	DescribePortal
프로젝트	CreateProject	DescribeProject
대시보드	CreateDashboard	DescribeDashboard
액세스 정책	CreateAccessPolicy	DescribeAccessPolicy
시계열	BatchPutAssetPropertyValue	DescribeTimeSeries

의 경우 [BatchPutAssetPropertyValue](#), 자산 모델 및 자산을 AWS IoT SiteWise 생성하기 전에 산업 데이터를 전송하도록 데이터 소스를 구성할 수 있습니다. AWS IoT SiteWise 장비로부터 원시 데이터 스트림을 수신하기 위해 데이터 스트림을 자동으로 생성합니다. 자세한 내용은 [데이터 수집 관리](#)를 참조하십시오.

다음 작업을 사용하여 태그 지정이 가능한 리소스의 태그를 보고 관리합니다.

- [TagResource](#)— 리소스에 태그를 추가하거나 기존 태그의 값을 업데이트합니다.
- [ListTagsForResource](#)— 리소스의 태그를 나열합니다.
- [UntagResource](#)— 리소스에서 태그를 제거합니다.

언제든지 리소스에 태그를 추가하거나 리소스에서 태그를 제거할 수 있습니다. 기존 태그 키의 값을 업데이트하려면 동일한 키와 원하는 새 값을 가진 새 태그를 리소스에 추가합니다. 이 작업은 이전 값을 새 값으로 대체합니다. 빈 문자열을 태그 값으로 할당할 수는 있지만 null 값을 할당할 수는 없습니다.

리소스를 삭제하면 해당 리소스에 연결된 태그도 모두 제거됩니다.

IAM 정책에 태그 사용

IAM 정책의 리소스 태그를 사용하여 사용자 액세스 및 권한을 제어하세요. 예를 들어 정책을 통해 사용자는 특정 태그가 첨부된 리소스만 생성할 수 있습니다. 또한 정책은 사용자가 특정 태그가 있는 리소스를 생성하거나 수정하는 것을 제한할 수 있습니다.

Note

태그를 사용하여 리소스에 대한 사용자의 액세스를 허용하거나 거부하는 경우, 동일한 리소스에 대한 사용자의 태그 추가 또는 제거 권한을 거부해야 합니다. 그렇지 않으면 사용자가 제한을 우회하고 태그를 수정하여 리소스에 액세스할 수 있습니다.

정책 설명의 Condition 요소(Condition 블록이라고도 함)에 다음 조건 컨텍스트 키와 값을 사용할 수 있습니다.

`aws:ResourceTag/tag-key: tag-value`

특정 태그가 있는 리소스에 대한 작업을 허용하거나 거부합니다.

`aws:RequestTag/tag-key: tag-value`

태그 지정이 가능한 리소스를 만들거나 수정할 때 특정 태그를 사용하거나 사용하지 않도록 요구합니다.

`aws:TagKeys: [tag-key, ...]`

태그 지정이 가능한 리소스를 만들거나 수정할 때 특정 태그 키 집합을 사용하거나 사용하지 않도록 요구합니다.

Note

IAM 정책의 조건 컨텍스트 키 및 값은 태그 지정 가능한 리소스가 필수 파라미터인 작업에만 적용됩니다. 예를 들어, 에 대한 태그 기반 조건부 액세스를 설정할 수 있습니다. [ListAssets](#) 요

청에서 태그 지정 가능한 리소스가 참조되지 [PutLoggingOptions](#) 않으므로 태그 기반 조건부 액세스를 켜도록 설정할 수 없습니다.

자세한 내용은 IAM 사용 설명서의 [리소스 태그 및 IAM JSON 정책 참조를 사용한 리소스 액세스 제어를 참조하십시오](#). AWS

태그를 사용하는 IAM 정책의 예

- [태그를 기반으로 AWS IoT SiteWise 자산 보기](#)

문제 해결 AWS IoT SiteWise

이 섹션의 정보를 사용하여 문제를 해결하고 해결하십시오 AWS IoT SiteWise.

주제

- [대량 가져오기 및 내보내기 작업 문제 해결](#)
- [AWS IoT SiteWise 포털 문제 해결](#)
- [SiteWise 에지 게이트웨이 문제 해결](#)
- [AWS IoT SiteWise 규칙 조치 문제 해결](#)

대량 가져오기 및 내보내기 작업 문제 해결

전송 작업 중에 발생한 오류를 처리하고 진단하려면 AWS IoT TwinMaker GetMetadataTransferJobAPI를 참조하십시오.

1. 전송 작업을 생성하고 실행한 후 GetMetadataTransferJobAPI를 호출합니다.

```
aws iottwinmaker get-metadata-transfer-job \
--metadata-transfer-job-id your_metadata_transfer_job_id \
--region us-east-1
```

2. 작업 상태가 다음 상태 중 하나로 변경됩니다.
 - 완료됨
 - 취소됨
 - ERROR
3. GetMetadataTransferJobAPI는 [MetadataTransferJobProgress](#) 객체를 반환합니다.
4. MetadataTransferJobProgress 객체에는 다음과 같은 매개변수가 포함됩니다.
 - FailedCount: 전송 프로세스 중에 실패한 자산의 수를 나타냅니다.
 - SkippedCount: 이전 프로세스 중에 건너뛰었던 자산의 수를 나타냅니다.
 - 성공 개수: 이전 프로세스 중에 성공한 자산 수를 나타냅니다.
 - 총계: 이전 프로세스에 관련된 자산의 총 수를 나타냅니다.

- 또한 미리 서명된 URL이 포함된 API 호출에서 ReportURL 요소가 반환됩니다. 전송 작업에 조사가 필요한 오류가 있는 경우 이 URL에서 전체 오류 보고서를 다운로드할 수 있습니다.

AWS IoT SiteWise 포털 문제 해결

AWS IoT SiteWise 포털과 관련된 일반적인 문제를 해결하세요.

사용자와 관리자가 AWS IoT SiteWise 포털에 접근할 수 없음

사용자 또는 관리자가 AWS IoT SiteWise 포털에 접근할 수 없는 경우, 첨부 AWS Identity and Access Management (IAM) 정책에 따라 로그인 성공을 방해하는 제한된 권한이 있을 수 있습니다.

로그인 실패로 이어지는 IAM 정책의 다음 예를 참조하십시오.

Note

"Condition" 요소가 포함된 IAM 정책을 첨부하면 로그인에 실패합니다.

예 1: 이 조건은 제한된 IP이며, 이로 인해 로그인에 실패합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "REPLACESAMPLEIP"
          ]
        }
      }
    }
  ]
}
```

예 2: 이 조건은 포함된 태그이며, 이로 인해 로그인에 실패합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/project": "*"
        }
      }
    }
  ]
}
```

포털에 사용자나 관리자를 추가할 때 제한된 IP와 같이 사용자 권한을 제한하는 IAM 정책을 생성하지 마십시오. 권한이 제한된 연결된 정책은 포털에 AWS IoT SiteWise 연결할 수 없습니다.

SiteWise 에지 게이트웨이 문제 해결

AWS IoT SiteWise 에지 게이트웨이는 구성 요소 세트를 실행합니다. AWS IoT Greengrass Amazon CloudWatch 및 SiteWise Edge 게이트웨이의 로컬 파일 시스템에 이벤트를 기록하도록 SiteWise Edge 게이트웨이를 구성할 수 있습니다. 그런 다음 로그 파일을 보고 SiteWise 에지 게이트웨이 문제를 해결할 수 있습니다.

SiteWise 에지 게이트웨이가 보고한 CloudWatch 메트릭을 보고 연결 또는 데이터 스트림 관련 문제를 해결할 수도 있습니다. 자세한 정보는 [Amazon AWS IoT SiteWise CloudWatch 메트릭을 사용한 모니터링](#)을 참조하세요.

주제

- [SiteWise 에지 게이트웨이 로그 구성 및 액세스](#)
- [SiteWise 에지 게이트웨이 문제 해결](#)
- [문제 AWS IoT Greengrass 해결](#)

SiteWise 에지 게이트웨이 로그 구성 및 액세스

SiteWise Edge 게이트웨이 로그를 보려면 먼저 Amazon Logs로 로그를 전송하거나 로컬 파일 시스템에 CloudWatch 로그를 저장하도록 SiteWise Edge 게이트웨이를 구성해야 합니다.

- 를 사용하여 SiteWise 에지 게이트웨이의 로그 파일을 AWS Management Console 보려면 CloudWatch 로그를 사용하십시오. 자세한 정보는 [아마존 CloudWatch 로그 사용](#)을 참조하세요.
- 명령줄 또는 로컬 소프트웨어를 사용하여 SiteWise Edge 게이트웨이의 로그 파일을 보려면 로컬 파일 시스템 로그를 사용하십시오. 자세한 정보는 [서비스 로그 사용](#)을 참조하세요.

SiteWise 에지 게이트웨이 문제 해결

다음 정보를 사용하여 SiteWise 에지 게이트웨이 문제를 해결하십시오.

문제

- [SiteWise 에지 게이트웨이에 팩을 배포할 수 없습니다.](#)
- [AWS IoT SiteWise OPC-UA 서버로부터 데이터를 받지 않음](#)
- [대시보드에 데이터가 표시되지 않았습니다.](#)
- [aws.iot에 “기본 클래스를 찾거나 로드할 수 없습니다”라는 메시지가 표시됩니다. SiteWiseEdgePublisher /greengrass/v2/logs의 로그 오류](#)

SiteWise 에지 게이트웨이에 팩을 배포할 수 없습니다.

AWS IoT Greengrass nucleus 구성 요소 (`aws.greengrass.Nucleus`) 가 최신 버전이 아닌 경우 SiteWise Edge 게이트웨이에 팩을 배포하지 못할 수 있습니다. AWS IoT Greengrass V2 콘솔을 사용하여 AWS IoT Greengrass nucleus 구성 요소를 업그레이드할 수 있습니다.

AWS IoT Greengrass 핵 구성 요소 업그레이드 (콘솔)

1. [AWS IoT Greengrass 콘솔](#)로 이동합니다.
2. 탐색 창의 AWS IoT Greengrass에서 배포를 선택합니다.
3. 배포 목록에서 수정하려는 배포를 선택합니다.
4. 수정을 선택합니다.
5. 대상 지정 페이지에서 다음을 선택합니다.
6. 구성 요소 선택 페이지의 퍼블릭 구성 요소에서 검색 상자에 `aws.greengrass.Nucleus`를 입력한 다음 `aws.greengrass.Nucleus`를 선택합니다.

7. 다음을 선택합니다.
8. 구성 요소 구성 페이지에서 다음을 선택합니다.
9. 고급 설정 구성 페이지에서 다음을 선택합니다.
10. 검토 페이지에서 배포를 선택합니다.

AWS IoT SiteWise OPC-UA 서버로부터 데이터를 받지 않음

AWS IoT SiteWise 자산이 OPC-UA 서버에서 전송한 데이터를 수신하지 않는 경우 SiteWise 에지 게이트웨이의 로그를 검색하여 문제를 해결할 수 있습니다. 다음 메시지가 포함된 정보 수준 `swPublisher` 로그를 찾습니다.

```
Emitting diagnostic name=PublishError.SomeException
```

로그 유형에 *SomeException* 따라 다음 예외 유형과 해당 문제를 사용하여 Edge 게이트웨이 문제를 해결하십시오. SiteWise

- `ResourceNotFoundException`— OPC-UA 서버가 자산의 속성 별칭과 일치하지 않는 데이터를 전송하고 있습니다. 이 예외는 두 가지 경우에 발생할 수 있습니다.
 - 속성 별칭은 정의한 원본 접두사를 포함해 OPC-UA 변수와 정확히 일치하지 않습니다. 속성 별칭 및 원본 접두사가 올바른지 확인합니다.
 - OPC-UA 변수를 자산 속성에 매핑하지 않았습니다. 자세한 정보는 [산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.

원하는 OPC-UA 변수를 모두 이미 매핑한 경우 Edge 게이트웨이가 보내는 OPC-UA 변수를 필터링할 수 있습니다. AWS IoT SiteWise SiteWise 자세한 정보는 [OPC-UA 노드 필터 사용](#)을 참조하세요.

- `InvalidRequestException`— OPC-UA 변수 데이터 유형이 자산 자산 자산 데이터 유형과 일치하지 않습니다. 예를 들어 OPC-UA 변수가 정수 데이터 형식인 경우, 해당 자산 속성은 정수 데이터 유형이어야 합니다. `double` 형식의 자산 속성은 OPC-UA 정수 값을 받을 수 없습니다. 이 문제를 해결하려면 새 속성을 올바른 데이터 형식으로 정의합니다.
- `TimestampOutOfRangeException`— SiteWise 에지 게이트웨이가 허용 범위를 벗어나는 데이터를 전송하고 있습니다. AWS IoT SiteWise AWS IoT SiteWise 과거 7일 이전 또는 5분 이후의 타임스탬프가 있는 모든 데이터 포인트를 거부합니다. SiteWise 에지 게이트웨이의 전원이 끊기거나 AWS 클라우드와의 연결이 끊긴 경우 SiteWise 에지 게이트웨이의 캐시를 지워야 할 수 있습니다.

- `ThrottlingException` 또는 `LimitExceededException`— 요청이 AWS IoT SiteWise 서비스 할당량을 초과했습니다 (예: 수집된 데이터 포인트 비율 또는 자산 자산 데이터 API 작업에 대한 요청 속도). 구성 이 [AWS IoT SiteWise 할당량](#)를 초과하지 않는지 확인합니다.

대시보드에 데이터가 표시되지 않았습니다.

대시보드에 데이터가 표시되지 않는 경우 게시자 구성과 SiteWise Edge Gateway의 데이터 원본이 동기화되지 않을 수 있습니다. 동기화되지 않은 경우 데이터 원본 이름을 업데이트하면 클라우드에서 옛 지로의 동기화가 빨라져 동기화 중단 오류가 해결될 수 있습니다.

데이터 소스 이름을 업데이트하려면

1. [AWS IoT SiteWise 콘솔](#)로 이동합니다.
2. 탐색 창에서 Edge Gateways를 선택합니다.
3. 대시보드에 연결된 SiteWise 에지 게이트웨이를 선택합니다.
4. 데이터 소스에서 편집을 선택합니다.
5. 새 소스 이름을 선택하고 저장을 선택하여 변경 내용을 확인합니다.
6. 데이터 원본 테이블에서 데이터 원본 이름이 업데이트되었는지 확인하여 변경 내용을 확인합니다.

aws.iot에 “기본 클래스를 찾거나 로드할 수 없습니다”라는 메시지가 표시됩니다.
SiteWiseEdgePublisher /greengrass/v2/logs의 로그 오류

이 오류가 표시되면 Edge 게이트웨이의 Java 버전을 업데이트해야 할 수 있습니다. SiteWise

- 터미널에서 다음 명령을 실행합니다.

```
java -version
```

SiteWise 에지 게이트웨이가 실행 중인 Java 버전이 아래에 표시됩니다 OpenJDK Runtime Environment. 다음과 같은 응답이 표시됩니다.

```
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment Corretto011.0.20.8.1 (build 11.0.20+8-LTS
OpenJDK 64-Bit Server VM Corretto-11.0.20.8.1 (build 11.0.20+8-LTS, mixed node)
```

자바 버전 11.0.20.8.1을 실행하는 경우 IoT SiteWise 퍼블리셔 팩을 버전 2.4.1 이상으로 업데이트해야 합니다. Java 버전 11.0.20.8.1만 영향을 받으며, 다른 Java 버전을 사용하는 환경에서는 이전 버전의 IoT SiteWise Publisher 구성 요소를 계속 사용할 수 있습니다. 구성 요소 팩 업데이트에 대한 자세한 내용은 [SiteWise Edge 게이트웨이 구성 요소 팩의 버전 변경](#) 단원을 참조하십시오.

문제 AWS IoT Greengrass 해결

SiteWise 에지 게이트웨이를 구성하거나 배포할 때 발생하는 여러 문제에 대한 AWS IoT Greengrass 해결책을 찾으려면 AWS IoT Greengrass 개발자 [안내서의 문제 해결을 AWS IoT Greengrass](#) 참조하십시오.

AWS IoT SiteWise 규칙 조치 문제 해결

에서 AWS IoT SiteWise 규칙 작업 문제를 AWS IoT Core 해결하려면 다음 절차 중 하나를 수행할 수 있습니다.

- Amazon CloudWatch 로그 구성
- 규칙에 대한 재게시 오류 작업 구성

그 다음 오류 메시지를 이 주제의 오류와 비교하여 문제를 해결합니다.

주제

- [AWS IoT Core 로그 구성](#)
- [재게시 오류 작업 구성](#)
- [문제 해결](#)
- [규칙 문제 해결](#)
- [규칙 문제 해결](#)

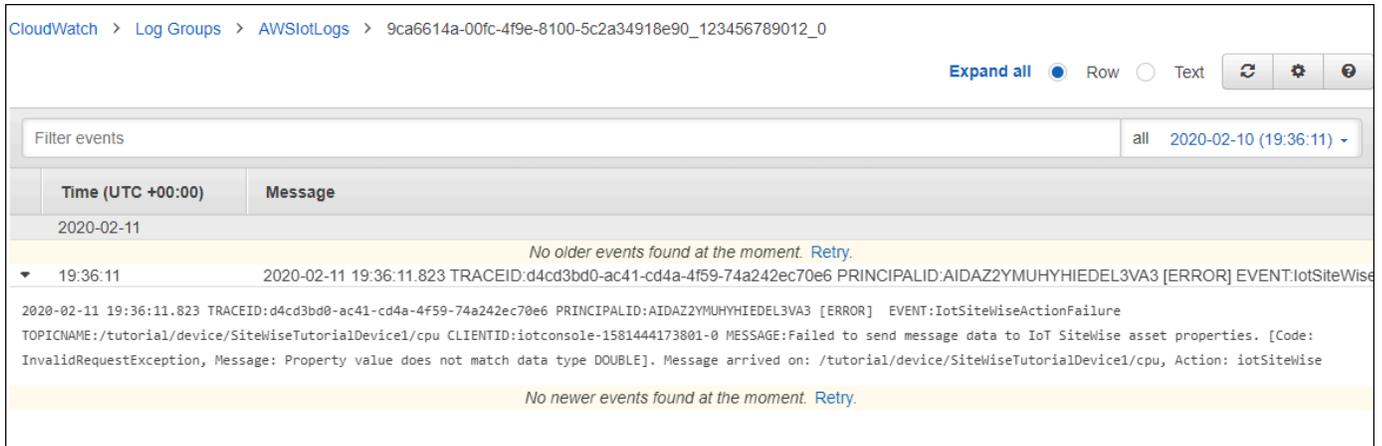
AWS IoT Core 로그 구성

다양한 수준의 정보를 AWS IoT 로그에 기록하도록 구성할 수 있습니다. CloudWatch

CloudWatch 로그를 구성하고 액세스하려면

1. 로깅을 구성하려면 AWS IoT 개발자 안내서의 CloudWatch [로그를 사용한 모니터링을](#) 참조하십시오. AWS IoT Core

2. [CloudWatch 콘솔](#)로 이동합니다.
3. 탐색 창에서 로그 그룹을 선택합니다.
4. AWSIoTLogs그룹을 선택합니다.
5. 최근 로그 스트림을 선택합니다. 기본적으로 가장 최근의 로그 스트림을 먼저 CloudWatch 표시합니다.
6. 로그 항목을 선택하여 로그 메시지를 확장합니다. 로그 항목은 다음 스크린샷처럼 보일 수 있습니다.



7. 오류 메시지를 이 주제의 오류와 비교하여 문제를 해결합니다.

재게시 오류 작업 구성

오류 메시지를 처리하도록 규칙에 대한 오류 작업을 구성할 수 있습니다. 이 절차에서는 MQTT 테스트 클라이언트에서 오류 메시지를 보기 위해 재게시 규칙 작업을 오류 작업으로 구성합니다.

Note

재게시 오류 작업은 ERROR 수준 로그와 동일한 로그만 출력합니다. 로그를 더 자세히 보려면 로그를 [CloudWatch 구성해야](#) 합니다.

규칙에 재게시 오류 작업 추가

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 작업을 선택한 다음 규칙을 선택합니다.
3. 규칙을 선택합니다.

- 오류 작업에서 작업 추가를 선택합니다.
- 주제에 메시지 다시 게시를 선택합니다. AWS IoT



- 페이지 하단에서 작업 구성을 선택합니다.
- 주제에 고유한 주제 (예:**sitewise/windfarm/rule/error**) 를 입력합니다. AWS IoT Core 이 주제에 오류 메시지를 다시 게시합니다.
- 선택을 선택하여 오류 조치를 수행할 AWS IoT Core 수 있는 액세스 권한을 부여합니다.
- 규칙에 대해 생성한 역할 옆에 있는 선택을 선택합니다.
- 역할 업데이트를 선택하여 역할에 권한을 더 추가합니다.
- 작업 추가를 선택합니다.

규칙의 오류 작업은 다음 스크린샷과 비슷해야 합니다.



- 콘솔 왼쪽 상단의 뒤쪽 화살표를 선택하여 AWS IoT 콘솔 홈으로 돌아가십시오.

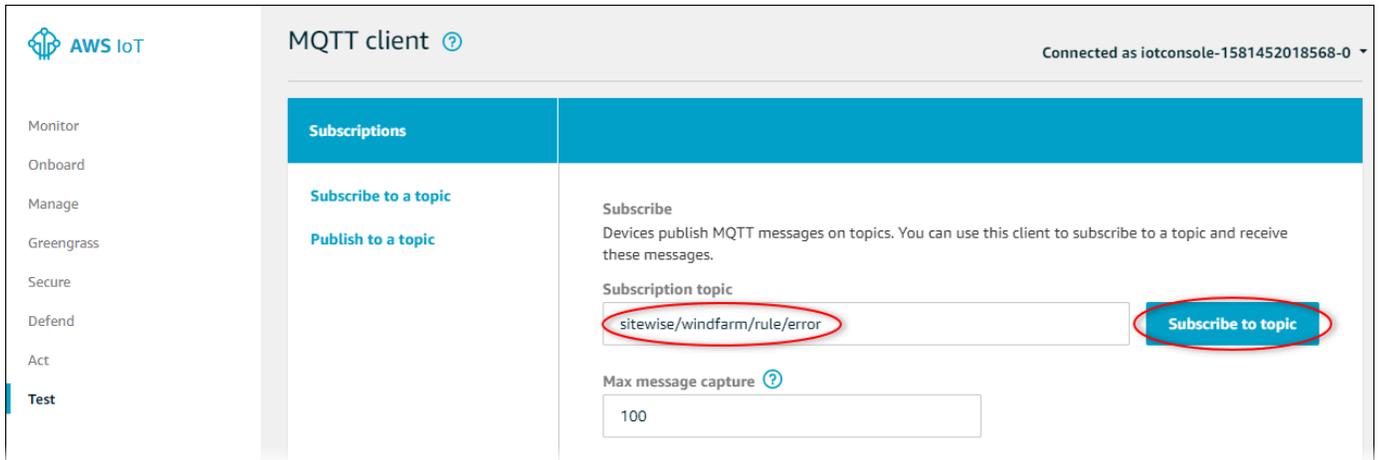
재게시 오류 작업을 설정한 후에는 AWS IoT Core의 MQTT 테스트 클라이언트에서 오류 메시지를 볼 수 있습니다.

다음 절차에서는 MQTT 테스트 클라이언트에서 오류 주제를 구독합니다. MQTT 테스트 클라이언트에서 규칙의 오류 메시지를 받아 문제를 해결할 수 있습니다.

오류 작업 주제 구독

- [AWS IoT 콘솔](#)로 이동합니다.
- 왼쪽 탐색 페이지에서 테스트를 선택하여 MQTT 테스트 클라이언트를 엽니다.

- 구독 주제 필드에 이전에 구성한 오류 주제(예: **sitewise/windfarm/rule/error**)를 입력하고 주제 구독을 선택합니다.



- 오류 메시지가 나타나는지 확인한 다음 오류 메시지에서 failures 배열을 확장합니다.
그런 다음 오류 메시지를 이 주제의 오류와 비교하여 문제를 해결합니다.

문제 해결

다음 정보를 사용하여 규칙 문제를 해결합니다.

문제

- 오류: 멤버는 현재 타임스탬프 전 604800초, 후 300초 이내에 있어야 합니다.
- 오류: 속성 값이 데이터 유형 <type>이 일치하지 않습니다.
- 오류: 사용자:: <role-arn>iotsitewise: 리소스에서 다음을 수행할 권한이 없습니다.
BatchPutAssetPropertyValue
- 오류: iot.amazonaws.com은 다음을 수행할 수 없습니다: sts: 리소스: AssumeRole <role-arn>
- 정보: 요청이 전송되지 않았습니다. PutAssetPropertyValueEntries 대체 템플릿을 수행한 후에는 비어 있었습니다.

오류: 멤버는 현재 타임스탬프 전 604800초, 후 300초 이내에 있어야 합니다.

타임스탬프는 현재 Unix 에포크 시간과 비교하여 7일 이전 또는 5분 이후입니다. 다음을 시도해 보십시오.

- 타임스탬프가 Unix 에포크 시간(UTC)인지 확인합니다. 다른 시간대로 타임스탬프를 제공하면 이 오류가 표시됩니다.

- 타임스탬프가 초 단위인지 확인하세요. AWS IoT SiteWise 타임스탬프가 초 단위의 시간 (Unix epoch 시간) 과 나노초 단위의 오프셋으로 분할될 것으로 예상됩니다.
- 지난 7일 이내로 타임스탬프가 지정된 데이터를 업로드하고 있는지 확인합니다.

오류: 속성 값이 데이터 유형 <type>이 일치하지 않습니다.

규칙 작업의 항목에는 대상 자산 속성이 아닌 다른 데이터 유형이 있습니다. 예를 들어 대상 자산 속성이 DOUBLE이고 선택한 데이터 유형이 정수이거나 integerValue에서 값을 전달했습니다. 다음을 시도해 보십시오.

- AWS IoT 콘솔에서 규칙을 구성하는 경우 각 항목에 대해 올바른 데이터 유형을 선택했는지 확인하십시오.
- API 또는 AWS Command Line Interface (AWS CLI) 에서 규칙을 구성하는 경우 value 객체가 올바른 유형 필드 (예: DOUBLE 속성) 를 사용하는지 확인하십시오. doubleValue

오류: 사용자:: <role-arn>iotsitewise: 리소스에서 다음을 수행할 권한이 없습니다.

BatchPutAssetPropertyValue

규칙이 대상 자산 속성에 액세스할 권한이 없거나 대상 자산 속성이 존재하지 않습니다. 다음을 시도해 보십시오.

- 속성 별칭이 올바른지와 지정된 속성 별칭이 있는 자산 속성이 있는지 확인합니다. 자세한 정보는 [산업 데이터 스트림을 자산 속성에 매핑](#)을 참조하세요.
- 규칙에 역할이 있는지와 역할이 대상 자산 속성(예: 대상 자산 계층 구조)에 대한 iotsitewise:BatchPutAssetPropertyValue 권한을 허용하는지 확인합니다. 자세한 정보는 [필요한 액세스 AWS IoT 권한 부여](#)을 참조하세요.

오류: iot.amazonaws.com은 다음을 수행할 수 없습니다: sts: 리소스: AssumeRole <role-arn>

사용자는 규칙 입력 (IAM) 에 따른 역할을 맡을 권한이 없습니다. AWS Identity and Access Management

사용자에게 규칙의 역할에 대한 iam:PassRole 권한이 허용되는지 확인합니다. 자세한 내용은 AWS IoT 개발자 가이드의 [역할 전달 권한](#) 단원을 참조하십시오.

정보: 요청이 전송되지 않았습니다. PutAssetPropertyValueEntries 대체 템플릿을 수행한 후에는 비어 있었습니다.

Note

이 메시지는 INFO 수준 로그입니다.

요청에 모든 필수 파라미터가 포함된 항목이 하나 이상 있어야 합니다.

대체 템플릿을 포함한 규칙의 파라미터가 비어 있지 않은 값을 생성하는지 확인합니다. 대체 템플릿은 규칙 쿼리 문의 AS 절에 정의된 값에 액세스할 수 없습니다. 자세한 내용은 AWS IoT 개발자 가이드의 [대체 템플릿](#)을 참조하십시오.

규칙 문제 해결

CPU 및 메모리 사용 데이터가 AWS IoT SiteWise 예상대로 표시되지 않는 경우 이 절차의 단계에 따라 규칙 문제를 해결하십시오. 이 절차에서는 MQTT 테스트 클라이언트에서 오류 메시지를 보기 위해 재게시 규칙 작업을 오류 작업으로 구성합니다. CloudWatch 로그에 로깅을 구성하여 문제를 해결할 수도 있습니다. 자세한 정보는 [AWS IoT SiteWise 규칙 조치 문제 해결](#)을 참조하세요.

규칙에 재게시 오류 작업 추가

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 메시지 라우팅을 선택한 다음 규칙을 선택합니다.
3. 이전에 생성한 규칙을 선택하고 편집을 선택합니다.
4. 오류 작업 - 선택 사항에서 오류 작업 추가를 선택합니다.
5. 주제에 메시지 다시 게시를 AWS IoT 선택합니다.
6. 주제에 오류 경로 (예:**sitewise/rule/tutorial/error**)를 입력합니다. AWS IoT Core 이 주제에 오류 메시지를 다시 게시합니다.
7. 이전에 만든 역할 (예: SiteWiseTutorialDeviceRuleRole)을 선택합니다.
8. 업데이트를 선택합니다.

재게시 오류 작업을 설정한 후에는 AWS IoT Core의 MQTT 테스트 클라이언트에서 오류 메시지를 볼 수 있습니다.

다음 절차에서는 MQTT 테스트 클라이언트에서 오류 주제를 구독합니다.

오류 작업 주제 구독

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 페이지에서 MQTT 테스트 클라이언트를 선택하여 MQTT 테스트 클라이언트를 엽니다.
3. 주제 필터 필드에서 **sitewise/rule/tutorial/error**를 입력한 다음 구독을 선택합니다.

오류 메시지가 나타나면 오류 메시지에서 failures 배열을 보고 문제를 진단합니다. 가능한 문제 및 해결 방법에 대한 자세한 내용은 [AWS IoT SiteWise 규칙 조치 문제 해결](#) 단원을 참조하십시오.

오류가 나타나지 않는 경우 규칙이 활성화되어 있고 오류 재게시 작업에서 구성한 것과 동일한 주제를 구독했는지 확인합니다. 그런 후에도 오류가 계속 나타나지 않으면 디바이스 스크립트가 실행 중이고 디바이스 새도우를 성공적으로 업데이트하는지 확인하십시오.

Note

디바이스의 새도우 업데이트 주제를 구독하여 AWS IoT SiteWise 액션이 파싱하는 페이로드를 볼 수도 있습니다. 이렇게 하려면 다음 주제를 구독하십시오.

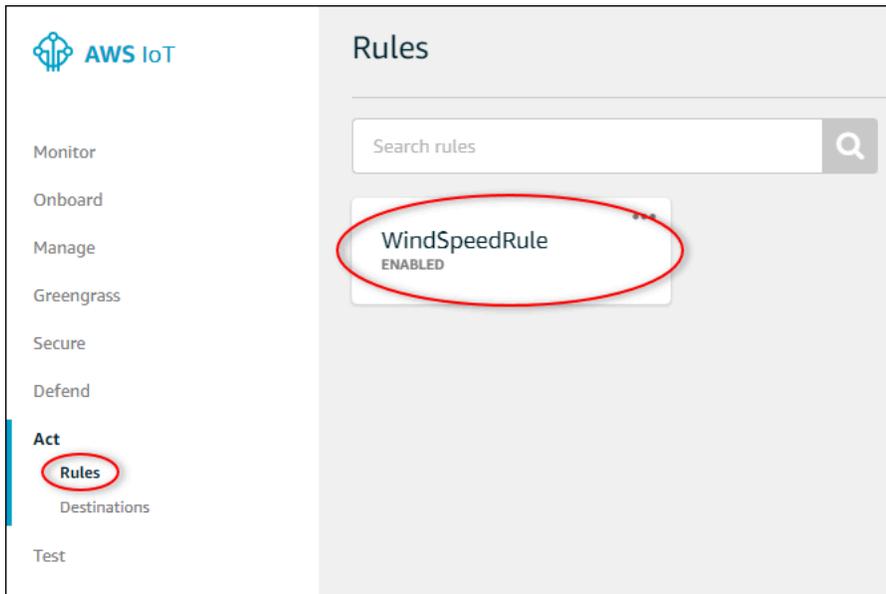
```
$aws/things/+shadow/update/accepted
```

규칙 문제 해결

데모 자산 데이터가 예상대로 DynamoDB 테이블에 표시되지 않는 경우 이 절차의 단계에 따라 규칙 문제를 해결합니다. 이 절차에서는 MQTT 테스트 클라이언트에서 오류 메시지를 보기 위해 재게시 규칙 작업을 오류 작업으로 구성합니다. 문제 해결을 위해 CloudWatch 로그에 로깅을 구성할 수도 있습니다. 자세한 내용은 AWS IoT 개발자 안내서의 CloudWatch [로그를 사용한 모니터링](#)을 참조하십시오.

규칙에 재게시 오류 작업 추가

1. [AWS IoT 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창에서 작업을 선택한 다음 규칙을 선택합니다.
3. 앞서 생성한 규칙을 선택합니다.



4. 오류 작업에서 작업 추가를 선택합니다.
5. AWS IoT 주제에 메시지 다시 게시를 선택합니다.



6. 페이지 하단에서 작업 구성을 선택합니다.
7. 주제에 **windspeed/error**를 입력합니다. AWS IoT Core는 이 주제에 대한 오류 메시지를 다시 게시합니다.

Configure action

 **Republish a message to an AWS IoT topic**
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

*Topic 

windspeed/error

Quality of Service 

0 - The message is delivered zero or more times.
 1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected Create Role Select

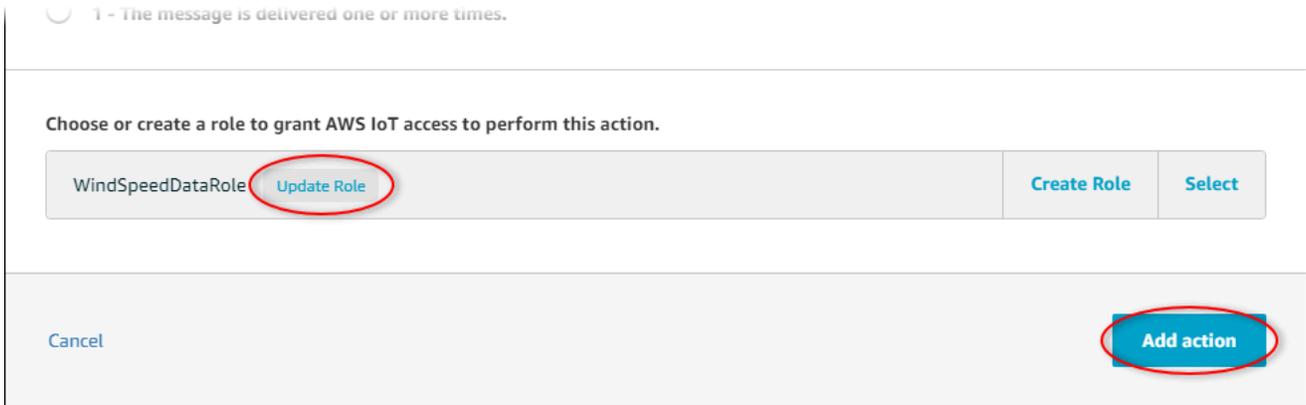
Cancel Add action

- 선택을 선택하면 이전에 만든 역할을 사용하여 오류 조치를 수행할 수 있는 AWS IoT 코어 액세스 권한을 부여할 수 있습니다.
- 해당 역할 옆에 있는 선택을 선택합니다.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected	Refresh	Create Role	Close
<input type="text" value="Search for IAM roles"/>			
WindSpeedDataRole			Select

- 역할 업데이트를 선택하여 역할에 권한을 더 추가합니다.



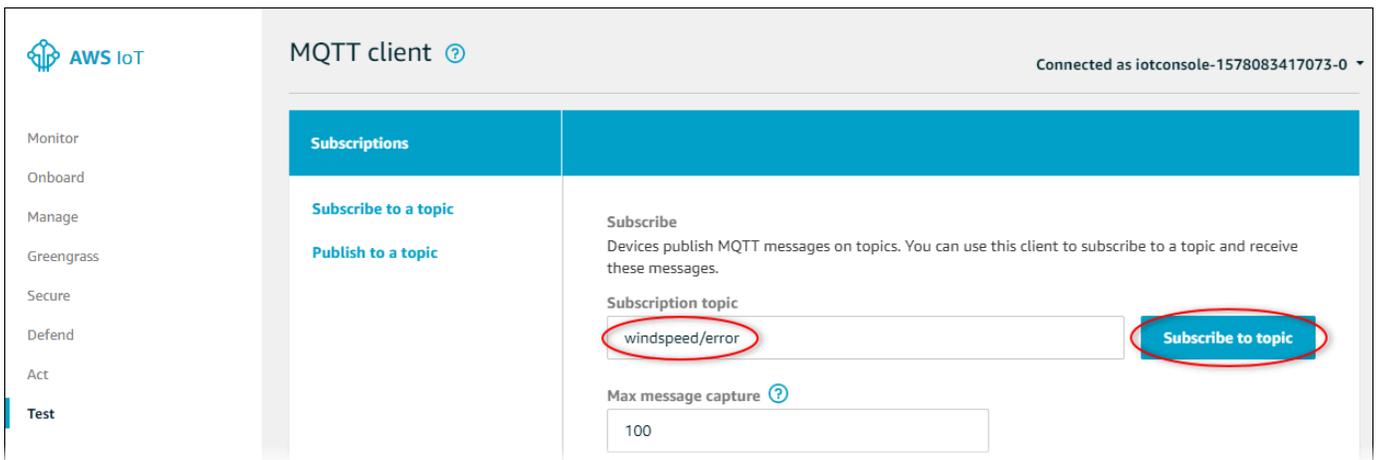
11. 작업 추가를 선택하여 오류 작업 추가를 완료합니다.
12. 콘솔 왼쪽 상단의 뒤쪽 화살표를 선택하여 AWS IoT Core 콘솔 홈으로 돌아가십시오.

재게시 오류 동작을 설정한 후 Core의 MQTT 테스트 클라이언트에서 AWS IoT 오류 메시지를 볼 수 있습니다.

다음 절차에서는 MQTT 테스트 클라이언트에서 오류 주제를 구독합니다.

오류 작업 주제 구독

1. AWS IoT Core 콘솔의 왼쪽 탐색 페이지에서 [Test] 를 선택합니다.
2. 구독 주제 필드에 **windspeed/error**를 입력하고 주제 구독을 선택합니다.



3. 오류 메시지가 나타나는지 확인하고 오류 메시지에서 failures 배열을 탐색하여 다음과 같은 일반적인 문제를 진단합니다.
 - 규칙 쿼리 문의 오타
 - 역할 권한 부족

오류가 나타나지 않는 경우 규칙이 활성화되어 있고 오류 재게시 작업에서 구성한 것과 동일한 주제를 구독했는지 확인합니다. 그래도 오류가 나타나지 않으면 데모 풍력 발전소 자산이 여전히 존재하고 풍속 속성에 대한 알림을 활성화했는지 확인합니다. 데모 자산이 만료되어 사라진 경우 새 데모를 만들고 업데이트된 자산 모델 및 속성 ID를 반영하도록 규칙 쿼리 문을 업데이트할 수 있습니다. AWS IoT SiteWise

AWS IoT SiteWise 엔드포인트 및 할당량

다음 섹션에서는 의 엔드포인트 및 할당량에 대해 설명합니다. AWS IoT SiteWise

내용

- [AWS IoT SiteWise 끝점](#)
- [AWS IoT SiteWise 할당량](#)

AWS IoT SiteWise 끝점

프로그래밍 방식으로 AWS IoT SiteWise 연결하려면 엔드포인트를 사용합니다. AWS SDK와 AWS Command Line Interface (AWS CLI) 는 지역의 기본 엔드포인트를 자동으로 사용합니다. AWS 사용 가능한 지역에 대한 자세한 내용은 AWS IoT SiteWise 의 [AWS IoT SiteWise 엔드포인트 및 할당량을](#) 참조하십시오. AWS 일반 참조

AWS IoT SiteWise 다음 엔드포인트를 지원합니다.

`data.iotsitewise.region.amazonaws.com`

이 엔드포인트를 사용하여 [BatchPutAssetPropertyValue](#), [GetAssetPropertyAggregates](#), [GetAssetPropertyValue](#), [GetAssetPropertyValueHistory](#) 및 [GetInterpolatedAssetPropertyValues](#) 데이터 플레인 API 작업에 액세스할 수 있습니다. AWS 해당 지역으로 *region* 바꾸십시오.

`api.iotsitewise.region.amazonaws.com`

AWS IoT SiteWise 자산 모델, 자산, SiteWise 에지 게이트웨이, 태그 및 계정 구성을 관리하는 데 사용하는 컨트롤 플레인 API 작업을 위한 이 통합 엔드포인트를 제공합니다. *region*을 해당 AWS 리전으로 바꿉니다.

Note

- 기본적으로 지원되는 컨트롤 플레인 API 작업을 호출할 때 통합 엔드포인트를 AWS IoT SiteWise 사용합니다.
- 지원되는 제어 플레인 API 작업에 통합 엔드포인트를 사용하는 것이 좋습니다.
- 통합 엔드포인트를 사용하여 SiteWise Monitor API 작업에 액세스할 수 없습니다.

지원되는 컨트롤 플레인 API 작업에는 [AssociateAssets](#),,, [CreateAsset](#), [CreateAssetModel](#), [DeleteAsset](#), [DeleteAssetModel](#),, [DeleteDashboard](#), [DescribeAsset](#), [DescribeAssetModel](#), [DescribeAssetProperty](#), [DescribeDashboard](#),, [DescribeLoggingOptions](#), [DisassociateAssets](#), [ListAssetModels](#), [ListAssetRelationships](#), [ListAssetsListAssociatedAssetsPutLoggingOptions](#), [UpdateAsset](#)등이 포함됩니다. [UpdateAssetModel](#),, [UpdateAssetProperty](#), [CreateGateway](#),, [DeleteGateway](#), [DescribeGateway](#),, [DescribeGatewayCapabilityConfiguration](#), [ListGateways](#),, [UpdateGateway](#), [UpdateGatewayCapabilityConfiguration](#), [DescribeStorageConfiguration](#),, [PutStorageConfiguration](#), [DescribeDefaultEncryptionConfiguration](#),, [ListTagsForResource](#), [PutDefaultEncryptionConfiguration](#),, [TagResource](#),,, [UntagResource](#).

제어 플레인 API 작업을 위한 인터페이스 VPC 엔드포인트는 통합 엔드포인트만 지원합니다. 자세한 내용은 [VPC 엔드포인트](#) 단원을 참조하십시오.

iotsitewise.region.amazonaws.com

이 엔드포인트를 사용하여 [DescribeStorageConfiguration](#),,, [PutStorageConfiguration](#), [DescribeDefaultEncryptionConfiguration](#), [ListTagsForResource](#), [PutDefaultEncryptionConfiguration](#) 및 같은 API 작업에 액세스할 수 [UntagResource](#) 있습니다. *region*을 해당 AWS 리전으로 바꿉니다.

model.iotsitewise.region.amazonaws.com

이 엔드포인트를 사용하여 [AssociateAssets](#),,,,,,, [CreateAsset](#), [CreateAssetModel](#), [DeleteAsset](#),, [DeleteAssetModel](#), [DeleteDashboard](#), [DescribeAsset](#), [DescribeAssetModel](#), [DescribeAssetProperty](#), [DescribeDashboard](#), [DescribeLoggingOptions](#), [DisassociateAssets](#), [ListAssetModels](#),, [ListAssetRelationships](#), [ListAssets](#), [ListAssociatedAssets](#), [PutLoggingOptions](#), [UpdateAsset](#), [UpdateAssetModel](#),, [UpdateAssetProperty](#). AWS 해당 지역으로 *region* 바꾸십시오.

edge.iotsitewise.region.amazonaws.com

이 엔드포인트를 사용하여 [CreateGateway](#),,, [DeleteGateway](#), [DescribeGateway](#), [DescribeGatewayCapabilityConfiguration](#), [ListGateways](#), [UpdateGateway](#), 및 같은 API 작업에 액세스할 수 [UpdateGatewayCapabilityConfiguration](#) 있습니다. AWS 해당 지역으로 *region* 바꾸십시오.

monitor.iotsitewise.region.amazonaws.com

이 엔드포인트를 사용하여 [BatchAssociateProjectAssets](#),,,,,,,, [BatchDisassociateProjectAssets](#), [CreateAccessPolicy](#), [CreateDashboard](#), [CreatePortal](#), [CreateProject](#),, [DeleteAccessPolicy](#),

[DeletePortal](#), [DeleteProject](#), [DescribeAccessPolicy](#), [DescribePortal](#), [DescribeProject](#), [ListAccessPolicies](#), [ListDashboards](#), [ListPortals](#), [ListProjectAssets](#), [ListProjects](#), [UpdateAccessPolicy](#), [UpdateDashboard](#), [UpdatePortal](#), [UpdateProject](#). *region*을 해당 AWS 리전으로 바꿉니다.

AWS IoT SiteWise 할당량

다음 표에는 의 할당량에 대한 설명이 나와 있습니다. AWS IoT SiteWise 할당량과 할당량 증가를 요청하는 방법에 대한 자세한 내용은 AWS 일반 참조의 [AWS Service Quotas](#)을 참조하십시오. 할당량에 대한 자세한 내용은 의 [AWS IoT SiteWise 서비스 AWS IoT SiteWise 할당량](#)을 참조하십시오. AWS 일반 참조

자산 및 자산 모델 할당량

Resource	할당량	조정 가능	참고
계정별 지역별 자산 모델 수 AWS	1000	예	
자산 모델당 자산 수	10,000개	예	
상위 자산당 하위 자산 수	2000	예	
자산 모델 계층 트리 깊이	30	예	
자산 모델당 계층 정의 수	30	예	
자산 모델별 루트 수준의 속성 수	500	예	assetMode lProperties 각 자산 모델의 최대 수입니다. 이 수에는 포함되지 않습니다 composite ModelProperties . 이 할당량은 이 자산 모델에서

Resource	할당량	조정 가능	참고
			생성된 모든 고유 자산에도 적용됩니다.
자산 모델당 속성 수	5000	예	ASSET_MODEL 또는 유형의 자산 모델의 최대 속성 COMPONENT_MODEL 수 이 수는 루트 자산 모델과 포함된 모델 component-model-based 또는 인라인 복합 모델의 속성을 결합하여 결정됩니다. 이 할당량은 이 자산 모델에서 생성된 모든 고유 자산에도 적용됩니다.
복합 모델당 속성 수	100	예	복합 모델에 허용되는 최대 속성 수입니다. 또한 특정 유형의 자산 모델에 허용되는 최대 속성 수입니다 COMPONENT_MODEL .
자산 모델당 속성 트리의 깊이	10	아니요	예를 들어 측정 속성 A를 사용하는 변환 속성 B를 사용하는 변환 속성 C가 있는 모델의 깊이는 3입니다.
계층 트리당 자산 모델 수	100	예	

Resource	할당량	조정 가능	참고
자산 모델당 직접 종속 속성의 수	20	아니요	이 할당량은 속성 공식 표현식에 정의된 대로 단일 속성에 직접 종속될 수 있는 속성의 수를 제한합니다. 자산 모델의 종속 속성 수는 자산 모델당 직접 종속 속성 수보다 커야 합니다. 자산 모델당 직접 종속 속성 수 제한이 자산 모델당 종속 속성 수 한도보다 클 경우 두 가지 모두에 대한 할당량 증가를 요청해야 합니다.
자산 모델당 종속 속성의 수	30	아니요	이 할당량은 속성 공식 표현식에 정의된 대로 단일 속성에 직접 또는 간접적으로 종속될 수 있는 속성의 수를 제한합니다.
자산 모델당 복합 모델 수	50	예	단일 자산 모델에 허용되는 최대 복합 모델 수.
복합 모델 깊이	2	예	인라인 및 복합 모델을 포함한 자산 모델별 component-model-based 복합 모델 트리의 최대 깊이.

Resource	할당량	조정 가능	참고
동일한 구성 요소 모델을 사용하는 고유 자산 모델 수	20	예	COMPONENT_MODEL 유형의 특정 자산 모델을 직접 참조하는 component-model-based 복합 모델이 하나 이상 있는 고유 자산 모델의 최대 수입니다.
속성 공식 표현식당 속성 변수의 수	10	아니요	예를 들어, avg(power) + max(temp) 표현식에는 power 및 temp의 두 개의 속성 변수가 있습니다. 이는 변환 계산 결과에도 적용됩니다.
속성 공식 표현식당 함수의 수	10	아니요	예를 들어, avg(power) + max(temp) 표현식에는 avg 및 max의 두 개의 함수가 있습니다.

자산 속성 데이터 할당량

Resource	할당량	조정 가능	참고
자산 속성 데이터 API 작업에 대한 요청 속도	계정당 지역당 초당 요청 1000개 AWS	예	이 할당량은 GetAssetProperty 및 BatchPutAssetProperty와 같은

Resource	할당량	조정 가능	참고
			API 작업에 적용됩니다.
자산 속성별 데이터 품질별 초당 데이터 포인트 수	10개의 데이터 포인트	아니요	이 할당량은 각 자산 속성에 대해 데이터 품질별로 동일한 타임스탬프 timestamp-quality-value (초) 를 가진 데이터 포인트의 최대 수 (TQV) 에 적용됩니다. 각 자산 속성에 대해 1초 동안 품질이 우수하거나 불확실한 데이터, 품질이 좋지 않은 데이터 포인트를 이 수까지 저장할 수 있습니다.
계정당 지역별 자산 속성당 초당 수집된 BatchPutAssetPropertyValues 항목 수. AWS	자산 자산당 항목 10개	아니요	이 할당량은 SiteWise Edge 게이트웨이, AWS IoT Core 규칙, API 호출을 비롯한 모든 소스의 BatchPutAssetPropertyValues 항목에 적용됩니다.
수집된 데이터 포인트의 비율	계정당 지역당 초당 5000개의 데이터 포인트 AWS	예	Timestamp-quality-value (TQV) 데이터 포인트.

Resource	할당량	조정 가능	참고
BatchGetAssetPropertyAggregates 에 대한 요청 속도	200	예	현재 리전의 이 계정에 서 수행할 수 있는 초 당 최대 BatchGetAssetPropertyAggregates 요청 수입니다.
BatchGetAssetPropertyValue 에 대한 요청 속도	500	예	현재 리전의 이 계정에 서 수행할 수 있는 초 당 최대 BatchGetAssetPropertyValue 요청 수입니다.
BatchGetAssetPropertyValueHistory 에 대한 요청 속도	200	예	현재 리전의 이 계정에 서 수행할 수 있는 초 당 최대 BatchGetAssetPropertyValueHistory 요청 수입니다.
계정별 지역별 자산 자산당 초당 수집된 BatchPutAssetPropertyValue 항목 수 AWS	자산 자산당 항목 10개	아니요	이 할당량은 SiteWise Edge 게이트웨이, AWS IoT Core 규칙, API 호출을 비롯한 모든 소스의 BatchPutAssetPropertyValue 항목에 적용됩니다.

Resource	할당량	조정 가능	참고
자산 자산별 GetAssetPropertyAggregates 요청 및 BatchGetAssetPropertyAggregates 항목 쿼리 비율	50	아니요	현재 리전에서 이 계정에 대한 초당 각 자산 속성에 대한 최대 GetAssetPropertyAggregates 요청 수와 BatchGetAssetPropertyAggregates 항목입니다.
자산 자산별 GetAssetPropertyValue 요청 및 BatchGetAssetPropertyValue 항목 쿼리 비율	500	아니요	현재 리전에서 이 계정에 대한 초당 각 자산 속성에 대한 최대 GetAssetPropertyValue 요청 수와 BatchGetAssetPropertyValue 항목입니다.
자산 자산별 GetAssetPropertyValueHistory 요청 및 BatchGetAssetPropertyValueHistory 항목 쿼리 비율	30	아니요	현재 리전에서 이 계정에 대한 초당 각 자산 속성에 대한 최대 GetAssetPropertyValueHistory 요청 수와 BatchGetAssetPropertyValueHistory 항목입니다.

Resource	할당량	조정 가능	참고
GetInterpolatedAssetPropertyValues 요청 비율	500	예	현재 리전의 이 계정에 수행할 수 있는 초당 최대 GetInterpolatedAssetPropertyValues 요청 수입입니다.
GetInterpolatedAssetPropertyValues 요청당 결과 수	10	예	페이지를 매긴 GetInterpolatedAssetPropertyValues 요청당 반환할 최대 결과 수입입니다.

Resource	할당량	조정 가능	참고
<p>GetAssetPropertyHistory 및 BatchGetAssetPropertyHistory 에서 검색된 데이터 포인트의 비율</p>	<p>계정당 지역별 초당 100MB의 읽기 응답 AWS</p>	<p>예</p>	<p>및 에 걸쳐 계정당 지역별로 초당 검색되는 데이터 포인트의 최대 바이트 속도 (MB/초) 입니다. AWS GetAssetPropertyHistory BatchGetAssetPropertyHistory 이 할당량에 대해 평가된 응답 페이로드는 각 데이터 포인트의 타임스탬프-품질-값(TQV) 필드를 사용하고 각 API 요청의 바이트 크기를 다음 4KB 증분으로 반올림합니다.</p> <p>초당 검색되는 Timestamp-quality-value (TQV) 데이터포인트는 데이터 유형에 따라 달라집니다.</p> <ul style="list-style-type: none"> • 정수 - 초당 최대 5 백만 TQV • 더블 - 초당 최대 4 백만 TQV • 부울 - 초당 최대 600만 TQV

Resource	할당량	조정 가능	참고
			<ul style="list-style-type: none"> 문자열 - 각 문자열 값 크기에 따라 달라 집니다.

에지 게이트웨이 할당량 SiteWise

Resource	할당량	조정 가능
계정별 지역별 SiteWise 엣지 게이트웨이 수 AWS	100	예
에지 게이트웨이당 OPC-UA 소스 수 SiteWise	100	아니요

할당량: AWS IoT SiteWise Monitor

Resource	할당량	조정 가능
계정별 지역별 포털 수 AWS	100	예
포털당 프로젝트 수	100	예
프로젝트당 대시보드 수	100	예
프로젝트당 루트 자산 수	1	아니요
대시보드당 시각화 수	10	예
대시보드 시각화당 지표 수	5	예
대시보드 시각화당 임계값 수	12	아니요

메타데이터 AWS IoT SiteWise 대량 가져오기 및 내보내기 할당량

Resource	설명	할당량	조정 가능
대기열에 있는 메타데이터 전송 작업 수	대기열에 있는 PENDING 메타데이터 전송 작업의 최대 수입니다.	10	예
메타데이터 전송 작업 가져오기 파일의 크기	가져온 파일의 최대 크기 (MB)	100MB	예
AWS IoT SiteWise 메타데이터 전송 작업의 리소스 할당량	단일 작업에서 가져오거나 내보낸 최대 리소스 수입니다. 리소스에는 에셋과 에셋 모델이 포함됩니다.	5000	아니요

데이터 AWS IoT SiteWise 대량 가져오기에 대한 할당량

Resource	할당량	조정 가능
실행 중인 대량 가져오기 작업 수	100	아니요
CSV 파일의 크기	10GB	아니요
압축되지 않은 쪽모이 세공 파일의 크기	256MB	아니요
버퍼 인제스트를 위한 CSV 파일 크기	256MB	아니요
압축되지 않은 파켓 행 그룹의 크기	64메가바이트	아니요
파켓 행 그룹별 고유 측정 횟수	2000	예

Resource	할당량	조정 가능
버퍼 설치의 과거 타임스탬프와 오늘 사이의 기간 (일)	30	예
각 계정의 CreateBulkImportJobs 지역별 요청 요금 AWS	10	예
각 AWS 계정의 ListBulkImportJobs 지역별 요청 요금	50	예
각 AWS 계정의 DescribeBulkImportJobs 지역별 요청 요금	50	예

예외 항목 탐지를 위한 할당량

이상 탐지를 위한 할당량은 AWS IoT SiteWise Amazon Lookout for Equipment 간에 공유됩니다. 자세한 내용은 [Lookout for Equipment 사용에 대한 할당량을](#) 참조하십시오.

AWS IoT SiteWise 사용 설명서의 문서 기록

다음 표에서는 이번 릴리스의 설명서를 설명합니다 AWS IoT SiteWise.

- API 버전: 2019-12-02

변경 사항	설명	날짜
지멘스 인더스트리얼 SiteWise 엣지에서 Edge를 실행하기 위한 지원이 추가되었습니다.	AWS IoT SiteWise 이제 지멘스 인더스트리얼 SiteWise 에지 디바이스에서 Edge를 실행할 수 있습니다.	2023년 11월 26일
웜 티어 스토리지에 대한 지원이 추가되었습니다.	AWS IoT SiteWise 이제 고객이 산업 데이터를 안전하게 저장하고 액세스할 수 있도록 하는 완전 관리형 스토리지 계층인 웜 스토리지가 지원됩니다.	2023년 11월 15일
사용자 정의 고유 식별자에 대한 지원이 추가되었습니다.	AWS IoT SiteWise 이제 자산, 자산 모델, 속성 및 계층에 대한 사용자 정의 고유 식별자 사용을 지원합니다.	2023년 11월 15일
산업 자산의 다변량 이상 탐지에 대한 지원이 추가되었습니다.	AWS IoT SiteWise 이제 Amazon Lookout for Equipment와 과거 및 실시간 장비 데이터를 통합하여 산업 자산의 다변량 이상 탐지를 지원합니다.	2023년 11월 15일
비용 효율적이고 확장 가능한 시계열 데이터 수집에 대한 지원이 추가되었습니다. AWS IoT SiteWise	AWS IoT SiteWise 이제 분석 사용 사례에 필요한 비용 효율적이고 확장 가능한 시계열 데이터 수집을 지원합니다.	2023년 11월 15일

대량 가져오기, 내보내기 및 업데이트에 대한 지원이 추가되었습니다.	AWS IoT SiteWise 이제 산업 장비 메타데이터의 대량 가져오기, 내보내기 및 업데이트를 지원합니다.	2023년 11월 15일
자산 모델 구성 요소에 대한 지원이 추가되었습니다.	AWS IoT SiteWise 이제 산업 고객이 재사용 가능한 구성 요소를 만들 수 있도록 자산 모델 구성 요소를 지원합니다.	2023년 11월 15일
IoT 대시보드 애플리케이션 지원 추가	AWS IoT SiteWise 이제 운영 데이터를 시각화하고 상호 작용할 수 있는 오픈 소스 대시보드 애플리케이션을 지원합니다.	2023년 11월 15일
에 대한 서비스 연결 역할을 업데이트했습니다. AWS IoT SiteWise	AWS IoT SiteWise 새로운 서비스 연결 역할이 있으며 데이터베이스에 대해 메타데이터 검색 쿼리를 실행할 수 있습니다. AWS IoT TwinMaker	2023년 11월 6일
데이터 스트림 리소스에 대한 AWS IoT SiteWise 태깅이 업데이트되었습니다.	데이터 스트림 리소스 태깅에 대한 지원이 추가되었습니다.	2022년 8월 18일
업데이트된 SiteWise 에지 게이트웨이	이제 엣지에서 클라우드로 전송되는 데이터와 클라우드로 전송되는 순서를 제어하도록 게시자를 구성할 수 있습니다.	2022년 1월 12일
데모를 업데이트했습니다. AWS IoT SiteWise	이제 데모를 사용하여 SiteWise 모니터 포털을 만들 수 있습니다.	2022년 1월 10일
업데이트된 스토리지 관리	이제 보존 기간을 정의하여 핫 tier에 데이터가 보관되는 기간을 제어할 수 있습니다.	2021년 11월 29일

추가된 데이터 스트림 관리에 대한 지원	이제 자산 모델 및 자산을 생성하기 AWS IoT SiteWise 전에 데이터를 수집할 수 있습니다.	2021년 11월 24일
업데이트된 자산 모델 계층 구조	하위 자산 모델을 여러 상위 자산 모델과 연결할 수 있습니다.	2021년 10월 28일
리전 출시	AWS GovCloud (미국 서부) AWS IoT SiteWise 에서 출시.	2021년 9월 29일
업데이트된 함수	<p>추가된 기능</p> <ul style="list-style-type: none"> 지표에서는 집계 함수와 임시 함수에 중첩 표현식을 사용할 수 있습니다. 변환에서는 pretrigger() 함수를 사용하여 현재 변환 계산을 트리거한 속성 업데이트 이전에 변수 값을 검색할 수 있습니다. 	2021년 8월 10일
사용자 지정 지표 시간 간격	지표의 사용자 지정 시간 간격 및 오프셋에 대한 지원이 추가되었습니다.	2021년 8월 3일
AWS IoT SiteWise 엣지에서의 사용	이제 엣지 프로세싱 기능이 일반적으로 이용 가능합니다.	2021년 7월 29일
Amazon S3로 데이터 내보내기	AWS IoT SiteWise 이제 Amazon S3로 데이터를 내보낼 수 있습니다.	2021년 7월 27일
VPC 엔드포인트(AWS PrivateLink)	이제 컨트롤 플레인 API 작업을 위한 인터페이스 VPC 엔드포인트를 정식으로 사용할 수 있습니다.	2021년 7월 15일
변형	이제 변환에서 여러 자산 속성 변수를 입력할 수 있습니다.	2021년 7월 8일

업데이트된 timestamp() 함수	이제 변환에서 변수를 함수에 대한 timestamp() 인수로 제공할 수 있습니다.	2021년 6월 16일
경보의 일반 사용 가능	이제 경보 기능이 일반적으로 이용 가능합니다.	2021년 5월 27일
Modbus-TCP 프로토콜 어댑터 버전 2 릴리스	Modbus-TCP 프로토콜 어댑터 커넥터 버전 2를 사용할 수 있습니다. 이 릴리스에서는 ASCII, UTF8 및 ISO8859 인코딩 소스 문자열에 대한 지원이 추가되었습니다.	2021년 5월 24일
업데이트된 Service Quotas	GetInterpolatedAssetPropertyValues 요청 비율, 요청당 GetInterpolatedAssetPropertyValues 결과 수, 시작 날짜의 과거와 오늘 사이의 일 수 등의 GetInterpolatedAssetPropertyValues API 할당량을 추가했습니다. GetInterpolatedAssetPropertyValues	2021년 4월 29일
업데이트된 공식 표현식	다음 연산자 및 함수가 추가되었습니다. <ul style="list-style-type: none"> 연산자 <, >, <=, >=, ==, !=, !, and, or, not이 추가되었습니다. 비교 함수 neq(x, y)가 추가되었습니다. 문자열 함수 join(), format(), f' '가 추가되었습니다. 	2021년 4월 22일

VPC 엔드포인트(AWS PrivateLink)	인터페이스 VPC 엔드포인트를 생성하여 가상 프라이빗 클라우드 (VPC) 와 AWS IoT SiteWise 컨트롤 플레인 API 간에 프라이빗 연결을 설정하는 방법에 대한 정보가 추가되었습니다.	2021년 3월 16일
IAM 페더레이션	이제 SiteWise 모니터 포털 관리자와 사용자는 IAM 자격 증명을 사용하여 할당된 포털에 로그인할 수 있습니다.	2021년 3월 16일
리전 출시	중국 (베이징) AWS IoT SiteWise 에서 출시되었습니다.	2021년 2월 3일
IoT SiteWise 커넥터 버전 10 출시	IoT SiteWise 커넥터 버전 10을 사용할 수 있습니다. 이 릴리스에서는 소스 연결이 끊겼다가 다시 설정되었을 때 처리를 개선하도록 StreamManager 를 구성하였습니다. 또한 이 버전은 SourceTimestamp 를 사용할 수 없을 때 ServerTimestamp 와 함께 OPC-UA 값을 허용합니다.	2021년 1월 22일
날짜 및 시간 함수	AWS IoT SiteWise 이제 날짜 및 시간 함수를 지원합니다.	2021년 1월 21일
함수 구문	이제 함수에 UFCS (통일 함수 호출 구문) 를 사용할 수 있습니다. AWS IoT SiteWise	2021년 1월 11일

Grafana와 통합

Grafana 대시보드에서 AWS IoT SiteWise 데이터를 시각화하는 방법에 대한 정보가 추가되었습니다.

2020년 12월 15일

[AWS IoT SiteWise 기능 출시](#)

이제 알람으로 데이터를 모니터링하고, 엣지에서 산업 데이터를 처리하고, Modbus TCP 및 이더넷/IP 소스를 SiteWise Edge 게이트웨이에 사용하고, 수신 데이터를 데드밴드로 필터링하는 등의 작업을 수행할 수 있습니다.

2020년 12월 15일

- AWS IoT SiteWise에서 경보를 정의, 구성 및 응답하는 데 사용할 수 있는 [경보를 통한 데이터 모니터링](#) 섹션이 추가되었습니다.
- [엣지 디바이스](#)의 산업 데이터 처리를 구성하는 데 사용할 수 있는 엣지 프로세싱 섹션이 추가되었습니다.
- 엣지 게이트웨이 소스 문서에 [Modbus TCP](#) 및 이더넷/IP 섹션을 추가했습니다. SiteWise
- 수신되는 산업 데이터를 보내는 위치를 사용자 지정하는 데 사용할 수 있는 [소스 대상](#) 섹션을 추가했습니다.
- 산업용 로컬 서버에서 Edge 게이트웨이로 전송되는 데이터의 빈도와 유형을 제어하는 데 사용할 수 있는 [OPC-UA 필터링](#) 섹션이 추가되었습니다. SiteWise

[AWS IoT SiteWise 이제 고객 관리형 CMK를 지원합니다.](#)

AWS IoT SiteWise 이제 고객 관리형 CMK를 통한 암호화가 지원됩니다.

2020년 11월 24일

<u>IoT SiteWise 커넥터 버전 8 출시</u>	IoT SiteWise 커넥터 버전 8을 사용할 수 있습니다. 이 릴리스에서는 커넥터에 간헐적인 네트워크 연결이 발생할 때의 안정성이 개선되었습니다.	2020년 11월 19일
<u>공식 표현식에 문자열 및 조건문 사용</u>	변환 및 지표용 공식 표현식에서 문자열 및 조건부 함수를 사용하는 방법에 대한 정보가 추가되었습니다.	2020년 11월 16일
<u>AWS IoT Greengrass 스트림 관리자를 사용한 데이터 수집</u>	AWS IoT Greengrass 에지 디바이스를 사용하여 로컬 데이터 소스에서 대용량 IoT 데이터를 수집하는 방법에 대한 정보가 추가되었습니다.	2020년 9월 16일
<u>VPC 엔드포인트(AWS PrivateLink)</u>	인터페이스 VPC 엔드포인트를 생성하여 가상 사설 클라우드(VPC)와 AWS IoT SiteWise 데이터 API 간에 사설 연결을 설정하는 방법에 대한 정보가 추가되었습니다.	2020년 9월 4일
<u>IoT SiteWise 커넥터 버전 7 출시</u>	IoT SiteWise 커넥터 버전 7을 사용할 수 있습니다. 이 릴리스에서는 SiteWise Edge 게이트웨이 메트릭과 관련된 문제가 수정되었습니다.	2020년 8월 14일

[콘솔에서 IAM ID 센터 사용자 생성 AWS IoT SiteWise](#)

콘솔에서 IAM ID 센터 사용자를 생성하는 방법에 대한 정보가 추가되었습니다. AWS IoT SiteWise 이제 새 포털 또는 기존 포털에 사용자를 할당할 때 IAM Identity Center 사용자를 생성할 수 있습니다. 이 기능을 사용하도록 [풍력 발전 단지 데이터 시각화 및 공유](#) 자습서를 업데이트했습니다. 이 변경으로 자습서의 단계 수가 줄어듭니다.

2020년 8월 4일

[SiteWise 에지 게이트웨이 문제 해결 개선](#)

SiteWise 에지 게이트웨이 문제를 해결하는 방법 및 원본의 [OPC-UA 클라이언트 인증서를 내보내는 방법](#)에 대한 추가 정보가 추가되었습니다.

2020년 6월 18일

[콘솔 작업 설명서](#)

[산업 자산 모델링, 자산 속성 데이터 쿼리 및 기타 서비스와 상호 작용](#)에 대한 콘솔 작업 설명서가 추가되었습니다. 다음 지침에 따라 AWS IoT SiteWise 콘솔에서 작업을 완료할 수 있습니다.

2020년 6월 11일

[내보낸 데이터 분석 자습서](#)

[내보내기 기능 AWS CloudFormation](#) 템플릿으로 S3로 내보낸 자산 데이터를 Amazon Athena를 사용하여 분석하는 방법을 배울 수 있는 자습서가 추가되었습니다.

2020년 5월 27일

공식 표현식 사용 개선

AWS IoT SiteWise 수식 속성의 동작에 대한 세부 정보를 추가하고 필터링된 데이터 포인트를 계산하는 방법에 대한 예제를 추가했습니다.

2020년 5월 18일

IoT SiteWise 커넥터 버전 6 출시

IoT SiteWise 커넥터 버전 6을 사용할 수 있습니다. 이번 릴리스에는 새로운 OPC-UA 태그의 CloudWatch 메트릭 및 자동 검색에 대한 지원이 추가되었습니다. 즉, OPC-UA 소스의 태그가 변경될 때 SiteWise 에지 게이트웨이를 다시 시작할 필요가 없습니다. 이 버전의 커넥터에는 스트림 관리자 및 AWS IoT Greengrass Core 소프트웨어 v1.10.0 이상이 필요합니다.

2020년 4월 29일

[AWS IoT SiteWise 기능 출시](#)

AWS IoT SiteWise 기능 출시. 이제 API를 사용하여 SiteWise Edge 게이트웨이를 관리하고, 포털에 로고를 추가하고, SiteWise Edge Gateway 메트릭을 보는 등의 작업을 수행할 수 있습니다.

2020년 4월 29일

- 새 [데이터 값을 S3 버킷으로 내보내는 데 사용할 수 있는 AWS CloudFormation 템플릿과 함께 Amazon S3로 데이터 내보내기](#) 섹션을 추가했습니다.
- SiteWise 에지 게이트웨이 소스 설명서를 개선하고 새로운 SiteWise Edge 게이트웨이 API를 포함하는 [데이터 소스 구성](#) 섹션을 추가했습니다.
- [SiteWise 에지 게이트웨이가 게시하는 CloudWatch 메트릭을 설명하는 SiteWise 에지 게이트웨이 지표](#) 섹션이 추가되었습니다.
- Amazon EC2 인스턴스에서 SiteWise 에지 게이트웨이 종속성을 빠르게 구성하는 데 사용할 수 있는 AWS CloudFormation 템플릿과 함께 Amazon EC2에서 SiteWise 에지 게이트웨이 구성 섹션을 추가했습니다.
- 모니터 [포털의 새로운 권한 기능을 설명하는 포털 서비](#)

[스 역할](#) 섹션이 추가되었습니다. SiteWise

- 포털 서비스 역할 및 포털 로고에 대한 [포털 설명서](#)를 업데이트했습니다.
- [AWS IoT SiteWise 리소스 태그 지정 섹션](#)이 추가되었습니다.
- 새 대시보드 정의 구조에 대한 [대시보드 생성\(CLI\)](#) 섹션을 업데이트했습니다.
- [보안](#) 섹션을 추가했습니다.

[데이터 수집: AWS IoT Events](#)

이벤트 발생 AWS IoT Events 시점의 데이터를 수집하는 방법에 대한 정보가 추가되었습니다.

2020년 4월 20일

[모니터 튜토리얼에서 SiteWise 풍력 발전 단지 데이터 시각화 및 공유](#)

자산 데이터를 시각화하고 공유하는 방법을 배우기 위해 따라할 수 있는 자습서가 추가되었습니다. AWS IoT SiteWise Monitor

2020년 3월 12일

[AWS IoT SiteWise 개념](#)

서비스 및 일반 용어에 대해 알아볼 수 있는 AWS IoT SiteWise 개념 용어집이 추가되었습니다.

2020년 3월 5일

<u>AWS IoT Greengrass 설치 지침이 제거되었습니다.</u>	사용 AWS IoT SiteWise 설명서에서 AWS IoT Greengrass Core 소프트웨어 설치 지침을 제거했습니다. <u>AWS IoT Greengrass 개발자 안내서는 Amazon EC2 및 Docker와 같은 다른 AWS IoT Greengrass 플랫폼에서 설정하는 데 필요한 장치 설정 스크립트와 지침을 제공합니다.</u>	2020년 2월 14일
<u>규칙을 사용한 데이터 수집을 개선했습니다. AWS IoT Core</u>	MQTT <u>메시지에서 데이터를 수집하는 데 사용할 수 있는 사용 방법 및 AWS IoT SiteWise 규칙 작업 문제 해결</u> 방법에 대한 자세한 정보가 추가되었습니다. AWS IoT Core	2020년 2월 14일
<u>IoT SiteWise 커넥터 버전 5 출시</u>	IoT SiteWise 커넥터 버전 5를 사용할 수 있습니다. 이 릴리스는 AWS IoT Greengrass Core 소프트웨어 v1.9.4와의 호환성 문제를 수정합니다.	2020년 2월 12일
<u>IoT SiteWise 커넥터 버전 4 출시</u>	IoT SiteWise 커넥터 버전 4를 사용할 수 있습니다. 이 릴리스는 OPC-UA 서버 재연결 문제를 해결합니다.	2020년 2월 7일

[모델링 산업 자산 재구성](#)

자산 및 모델 업데이트 섹션을 산업 자산 모델링 내의 여러 주제로 재구성했습니다.

2020년 2월 4일

- [자산 및 모델 상태](#)
- [산업 데이터 스트림을 자산 속성에 매핑](#)
- [속성 값 업데이트](#)
- [자산 연결 및 연결 해제](#)
- [자산 및 모델 업데이트](#)
- [자산 및 모델 삭제](#)

[AWS IoT 사물 튜토리얼의 데이터 수집](#)

신규 또는 기존 사물로부터 데이터를 수집하도록 AWS IoT SiteWise 규칙 작업을 구성하는 방법을 배울 수 있는 자습서가 추가되었습니다. AWS IoT

2020년 2월 4일

[에서 데이터를 검색하는 방식을 재구성했습니다. AWS IoT SiteWise](#)

[데이터 검색 섹션을 자산 자산 가치 및 집계 쿼리, 다른 서비스와의 상호 작용이라는 두 개의 최상위 섹션으로 재구성했습니다. AWS](#)

2020년 1월 21일

[Amazon DynamoDB에 속성 값 업데이트 게시 자습서](#)

속성 값 알림을 사용하여 DynamoDB에 자산 데이터를 저장하는 방법을 알아보기 위해 따를 수 있는 자습서를 추가했습니다.

2020년 1월 8일

[공식 표현식 사용](#)

변환 및 미터법 속성에 사용할 수 있는 상수와 함수를 정리하기 위해 공식 표현식 참조를 추가했습니다. [자산 속성](#)을 각 속성 유형에 대한 별도의 주제로 재구성했습니다.

2020년 1월 7일

<u>OPC-UA 노드 필터 사용</u>	Edge 게이트웨이 소스를 추가할 때 OPC-UA 노드 필터를 사용하여 Edge 게이트웨이 성능을 향상시키는 SiteWise 방법에 대한 정보가 추가되었습니다. SiteWise	2020년 1월 3일
<u>커넥터 업그레이드</u>	새 커넥터 버전이 출시될 때 SiteWise 에지 게이트웨이를 업그레이드하는 방법에 대한 정보가 추가되었습니다.	2019년 12월 30일
<u>IoT SiteWise 커넥터 버전 3 출시</u>	IoT SiteWise 커넥터 버전 3을 사용할 수 있습니다. 이 릴리스에서는 IoT: * 권한 요구 사항을 제거했습니다.	2019년 12월 17일
<u>IoT SiteWise 커넥터 버전 2 출시</u>	IoT SiteWise 커넥터 버전 2를 사용할 수 있습니다. 이 릴리스에는 여러 OPC-UA 암호 리소스에 대한 지원이 추가되었습니다.	2019년 12월 10일
<u>대시보드 생성(AWS CLI)</u>	를 AWS IoT SiteWise Monitor 사용하여 대시보드를 만드는 방법에 대한 정보가 추가되었습니다 AWS CLI.	2019년 12월 6일

[AWS IoT SiteWise 버전 2 출시](#)

버전 2의 미리보기가 AWS IoT SiteWise 출시되었습니다. 이제 OPC-UA, MQTT 및 HTTP를 통해 데이터를 수집하고, 데이터를 자산 계층 구조로 모델링하고, Monitor를 사용하여 데이터를 시각화할 수 있습니다. SiteWise

2019년 12월 2일

- 자산, 자산 모델 및 자산 계층 구조가 변경되어 [자산 모델링](#) 섹션을 다시 작성했습니다.
- 커넥터 단계 및 비게이트웨이 데이터 통합 섹션을 포함하도록 [데이터 통합](#) 섹션을 업데이트했습니다. AWS IoT Greengrass
- Monitor 웹 [애플리케이션 사용 방법을 보여주는 AWS IoT SiteWise Monitor](#) 섹션과 별도의 [애플리케이션 가이드](#)가 추가되었습니다. SiteWise

- [에서 데이터 쿼리 AWS IoT SiteWise](#) 및 [다른 AWS 서비스와의 상호작용](#) 섹션을 추가했습니다.
- 업데이트된 데모 환경에 맞게 [시작하기](#) 섹션을 다시 작성했습니다.

[AWS IoT SiteWise 버전 1이 출시되었습니다.](#)

버전 1의 초기 미리보기를 AWS IoT SiteWise 출시했습니다.

2019년 2월 25일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 [AWS 용어집](#)을 참조하세요.

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