



사용 설명서

AWS Ground Station



AWS Ground Station: 사용 설명서

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 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

AWS Ground Station란 무엇인가요?	1
일반 사용 사례	1
다음 단계	2
AWS Ground Station 작동 방식	3
위성 온보딩	3
미션 프로파일 구성	3
고객 응대 예약	5
고객 응대 실행	6
디지털 트윈	9
AWS Ground Station 핵심 구성 요소 이해	9
미션 프로파일	11
구성	14
데이터 흐름 엔드포인트 그룹	23
AWS Ground Station 에이전트	30
시작하기	32
에 가입 AWS 계정	32
AWS 계정에 AWS Ground Station 권한 추가	32
위성 온보딩	34
고객 온보딩 프로세스 개요	34
(선택 사항) 위성 이름 지정	34
퍼블릭 브로드캐스트 위성	37
데이터 흐름 통신 경로 계획	38
비동기 데이터 전송	38
동기식 데이터 전송	39
원격 측정 계획	40
구성 생성	40
데이터 전송 구성	41
원격 측정 구성(선택 사항)	41
위성 구성	41
미션 프로파일 생성	41
다음 단계 이해	42
AWS Ground Station 위치	44
지상국 위치의 AWS 리전 찾기	44
AWS Ground Station 지원되는 AWS 리전	45

디지털 트윈 가용성	46
전용 안테나	46
지상국에서 안테나 보기	46
예: 지상국의 안테나 나열	46
지상국 예약 보기	47
예약 나열	47
예약 유형	48
코드 예제	48
AWS Ground Station 사이트 마스크	50
고객별 마스크	50
사이트 마스크가 사용 가능한 연락 시간에 미치는 영향	51
AWS Ground Station 사이트 기능	51
가 에페메리스를 AWS Ground Station 사용하는 방법 이해	54
기본 에페메리스 데이터	54
사용자 지정 에페메리스 데이터 제공	55
개요	55
예:에서 고객 제공 에페메리스 사용 AWS Ground Station	55
TLE 에페메리스 데이터 제공	56
OEM 에페메리스 데이터 제공	63
방위각 고도 에페메리스 데이터 제공	71
사용자 지정 에페메리스를 사용하여 연락처 예약	82
개요	82
고객 응대 예약 워크플로	82
워크플로 1: 사용 가능한 연락처를 나열한 다음 예약	83
워크플로 2: 직접 연락처 예약	87
고객 응대 상태 변경 모니터링	90
모범 사례 및 고려 사항	92
사용되는 에페메리스 이해	93
TLE 및 OEM 에페메리스	93
방위각 고도 에페메리스	94
새 에페메리스가 이전에 예약된 연락처에 미치는 영향	94
위성의 현재 에페메리스 가져오기	95
기본 에페메리스를 사용하는 위성에 대한 GetSatellite 반환 예제	96
사용자 지정 에페메리스를 사용하는 위성에 대한 GetSatellite 예제	96
방위각 고도 에페메리스 나열	97
기본 에페메리스 데이터로 되돌리기	98

TLE 및 OEM 에페메리스 되돌리기	98
방위각 고도 에페메리스 관리	98
데이터 흐름 작업	100
AWS Ground Station 데이터 영역 인터페이스	100
리전 간 데이터 전송 사용	101
Amazon S3 설정 및 구성	101
Amazon VPC 설정 및 구성	101
AWS Ground Station 에이전트를 사용한 VPC 구성	102
데이터 흐름 엔드포인트를 사용한 VPC 구성	104
Amazon EC2 설정 및 구성	107
제공 공통 소프트웨어	107
AWS Ground Station Amazon Machine Image(AMIs)	108
원격 측정 작업	109
원격 측정 작동 방식	109
사용 가능한 원격 측정 유형	110
리전별 가용성	110
원격 측정 설정	110
1단계: 사전 조건 AWS 리소스 생성	111
2단계: TelemetrySinkConfig 생성	112
3단계: 미션 프로파일에 원격 측정 추가	113
4단계: 고객 응대 예약	113
다음 단계	114
원격 측정 데이터 이해	114
데이터 형식 개요	114
텔레메트리 가리키기	115
원격 측정 추적	116
Kinesis Data Streams 스트림에서 데이터 읽기	118
스키마 버전 관리 및 진화	119
연락처 작업	120
고객 응대 수명 주기 이해	120
AWS Ground Station 고객 응대 상태	123
고객 응대 데이터 보존	124
고객 응대 결제 이해	124
대역폭 정의	125
예약 모드	125
CancelContact	125

시나리오 1: 단일 고객 응대	126
시나리오 2: 중지된 단일 고객 응대	126
시나리오 3: 단일 중복	127
시나리오 4: 짧은 중복	128
시나리오 5: 여러 중복	129
시나리오 6: 다중 중지	131
시나리오 7: 중복이 없는 다중 안테나 지상국	132
시나리오 8: 중복 접촉이 있는 다중 안테나 지상국	133
연락처 및 연락처 버전 관리 업데이트	134
고객 응대 버전 관리 작동 방식	134
연락처 업데이트	134
고객 응대 버전 상태	136
코드 예제	136
고려 사항	141
AWS Ground Station 디지털 트윈	142
AWS Ground Station 전용 안테나	143
전용 안테나란?	143
예약 가시성 향상	144
관련 리소스	144
모니터링	146
이벤트로 자동화	147
AWS Ground Station 이벤트 유형	147
연락 이벤트 타임라인	148
에페메리스 이벤트	150
CloudTrail을 사용하여 API 호출 로깅	151
AWS Ground Station CloudTrail의 정보	151
AWS Ground Station 로그 파일 항목 이해	152
Amazon CloudWatch를 사용하여 지표 보기	154
AWS Ground Station 지표 및 차원	154
지표 보기	158
보안	165
자격 증명 및 액세스 관리	165
대상	166
ID를 통한 인증	166
정책을 사용하여 액세스 관리	167
AWS Ground Station 에서 IAM을 사용하는 방법	169

자격 증명 기반 정책 예시	174
문제 해결	176
AWS 관리형 정책	178
AWSGroundStationAgentInstancePolicy	178
AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy	179
정책 업데이트	180
서비스 연결 역할 사용	181
Ground Station에 대한 서비스 연결 역할 권한	181
Ground Station에 대한 서비스 연결 역할 생성	182
Ground Station에 대한 서비스 연결 역할 편집	182
Ground Station에 대한 서비스 연결 역할 삭제	182
Ground Station 서비스 연결 역할이 지원되는 리전	183
문제 해결	183
에 대한 저장 데이터 암호화 AWS Ground Station	183
고객 관리형 키 생성	185
에 대한 고객 관리형 키 지정 AWS Ground Station	186
AWS Ground Station 암호화 컨텍스트	186
TLE 및 OEM 에페메리스 데이터에 대한 저장 데이터 암호화	187
방위각 고도 에페메리스에 대한 유희 시 암호화	195
에 대한 전송 중 데이터 암호화 AWS Ground Station	204
AWS Ground Station 에이전트 스트림	204
데이터 흐름 엔드포인트 스트림	204
미션 프로파일 구성 예	205
JPSS-1 - 퍼블릭 브로드캐스트 위성(PBS) - 평가	205
Amazon S3 데이터 전송을 활용하는 퍼블릭 브로드캐스트 위성	206
통신 경로	206
AWS Ground Station 구성	208
AWS Ground Station 미션 프로파일	210
함께 넣기	210
데이터 흐름 엔드포인트(협대역)를 사용하는 퍼블릭 브로드캐스트 위성	211
통신 경로	211
AWS Ground Station 구성	218
AWS Ground Station 미션 프로파일	220
함께 넣기	220
데이터 흐름 엔드포인트를 사용하는 퍼블릭 브로드캐스트 위성(변조 및 디코딩됨)	222
통신 경로	222

AWS Ground Station 구성	230
AWS Ground Station 미션 프로파일	233
함께 넣기	234
AWS Ground Station 에이전트(와이드밴드)를 활용하는 퍼블릭 브로드캐스트 위성	236
통신 경로	236
AWS Ground Station 구성	247
AWS Ground Station 미션 프로파일	248
함께 넣기	249
문제 해결	252
Amazon EC2로 데이터를 전송하는 고객 응대 문제 해결	252
1단계: EC2 인스턴스가 실행 중인지 확인	253
2단계: 사용되는 데이터 흐름 애플리케이션 유형 결정	253
3단계: 데이터 흐름 애플리케이션이 실행 중인지 확인	253
4단계: 데이터 흐름 애플리케이션 스트림이 구성되어 있는지 확인	255
5단계: 수신기 인스턴스(들) 서브넷에 사용 가능한 IP 주소가 충분한지 확인	257
실패 고객 응대 문제 해결	257
데이터 흐름 엔드포인트 FAILED 사용 사례	258
AWS Ground Station 에이전트 FAILED 사용 사례	259
실패한 고객 응대 업데이트 문제 해결	259
동기 검증 오류	259
비동기 장애 코드	261
업데이트 상태 확인	263
FAILED_TO_SCHEDULE 연락처 문제 해결	264
안테나 다운로드 데모 디코딩 구성에 지정된 설정은 지원되지 않습니다.	265
일반 문제 해결 단계	265
정상 상태가 아닌 DataflowEndpointGroups 문제 해결	265
잘못된 에페메리스 문제 해결	266
에페메리스 검증 오류 이해	266
TLE 에페메리스에 대한 일반적인 검증 오류	267
OEM 에페메리스의 일반적인 검증 오류	267
방위각 고도 에페메리스에 대한 일반적인 검증 오류	268
문제 해결 단계	269
오류 코드 참조 완료	269
데이터를 수신하지 못한 고객 응대 문제 해결	274
잘못된 다운로드 구성	274
위성 조작	274

AWS Ground Station 중단	274
원격 측정 문제 해결	275
일반적인 설정 문제	275
원격 측정 전송 문제	278
데이터 형식 문제	279
도움말 가져오기	280
할당량 및 제한	282
서비스 약관	283
문서 기록	284
AWS 용어집	290
.....	CCXci

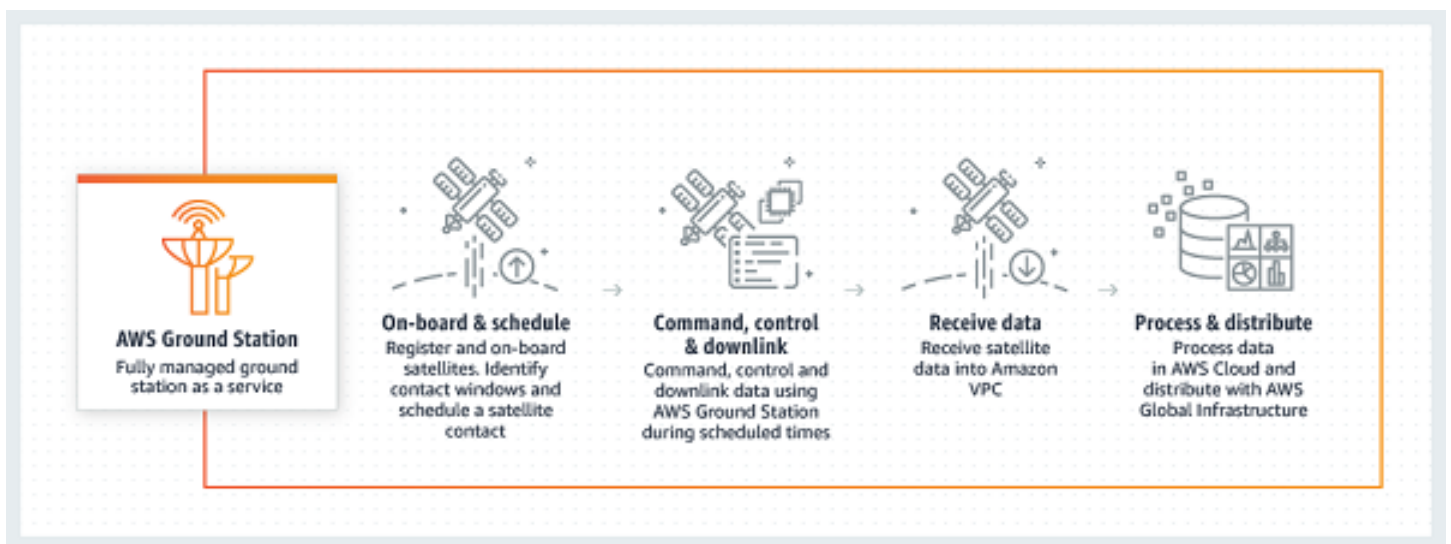
AWS Ground Station란 무엇인가요?

AWS Ground Station 는 글로벌 인프라에서 안전하고 빠르고 예측 가능한 위성 통신을 제공하는 완전 관리형 서비스입니다. 를 사용하면 더 이상 자체 지상국 인프라를 구축, 관리 또는 확장할 필요가 AWS Ground Station 없습니다. AWS Ground Station 를 사용하면 자체 지상국을 구축, 운영 및 확장하는 데 리소스를 소비하는 대신 위성 데이터를 수집하는 새로운 애플리케이션을 혁신하고 신속하게 실험하는 데 집중할 수 있습니다.

AWS의 지연 시간이 짧은 고대역폭 글로벌 광섬유 네트워크를 사용하면 안테나 시스템에서 수신한 후 몇 초 이내에 위성 데이터 처리를 시작할 수 있습니다. 이를 통해 원시 데이터를 몇 초 안에 처리된 정보 또는 분석된 지식으로 변환할 수 있습니다.

특수 요구 사항이 있는 조직의 경우 AWS Ground Station 는 사용자를 대신하여를 AWS 관리하는 [AWS Ground Station 전용 안테나](#) 사용자 지정 구축 안테나 시스템을 제공하여 사양에 맞게 구축된 안테나에 대한 전용 액세스를 제공합니다.

일반 사용 사례



AWS Ground Station 를 사용하면 위성과 양방향으로 통신할 수 있으며 다음 사용 사례를 지원합니다.

- 다운링크 데이터 - 위성에서 데이터를 수신하여 실시간(VITA-49 형식) 또는 계정의 Amazon S3 버킷([PCAP 형식](#))으로 Amazon EC2 인스턴스로 전송된 X 대역 및 S 대역 빈도를 전송합니다. Amazon S3 또한 지원되는 변조 및 인코딩 체계를 사용하는 위성의 경우 복조 및 디코딩된 데이터 수신 또는 원시 디지털 중간 주파수(DigIF) 샘플(VITA-49 형식) 중에서 선택할 수 있습니다.

- 업링크 데이터 - 전송할 DigIF 데이터(VITA-49 형식)를 전송하여 S 대역 주파수를 수신하는 위성에 데이터 및 명령을 전송합니다 AWS Ground Station.
- 업링크 에코 - 물리적으로 함께 배치된 안테나에서 전송된 신호를 수신하여 우주선으로 전송된 명령을 검증하고 기타 고급 작업을 수행합니다.
- 소프트웨어 정의 라디오(SDR) / 프런트 엔드 프로세서(FEP) - Amazon EC2 인스턴스에서 실행할 수 있는 기존 SDR 및/또는 FEP를 사용하여 데이터를 실시간으로 처리하여 기존 파형을 전송/수신하고 데이터 제품을 생성합니다.
- 원격 측정, 추적 및 명령(TT&C) - 이전에 나열된 사용 사례를 조합하여 TT&C를 수행하여 위성 플릿을 관리합니다.
- 리전 간 데이터 전송 - 단일 AWS 리전에서 AWS Ground Station의 글로벌 안테나 네트워크를 사용하여 여러 동시 고객 응대를 운영합니다.
- 디지털 트윈 - 프로덕션 안테나 용량을 사용하지 않고도 절감된 비용으로 예약, 구성 확인 및 적절한 오류 처리를 테스트합니다.

다음 단계

다음 단원을 읽고 시작하면 도움이 됩니다.

- 필수 AWS Ground Station 개념을 알아보려면 섹션을 참조하세요 [AWS Ground Station 작동 방식](#).
- 사용할 계정 및 리소스를 설정하는 방법은 섹션을 AWS Ground Station참조하세요 [시작하기](#).
- 프로그래밍 방식으로를 사용하려면 [AWS Ground Station API 참조](#)를 참조 AWS Ground Station하세요. API 참조는에 대한 모든 API 작업을 AWS Ground Station 자세히 설명합니다. 또한 지원되는 웹 서비스 프로토콜에 대한 샘플 요청, 응답 및 오류를 제공합니다. 원하는 언어로 [AWS CLI](#) 또는 [AWS SDK](#)를 사용하여와 상호 작용하는 코드를 작성할 수 있습니다 AWS Ground Station.

AWS Ground Station 작동 방식

AWS Ground Station 는 지상 기반 안테나를 작동하여 위성과의 통신을 용이하게 합니다. 안테나가 수행할 수 있는 작업의 물리적 특성을 추상화하고 이를 기능이라고 합니다. 안테나의 물리적 위치와 현재 기능은 [AWS Ground Station 위치](#) 섹션에서 참조할 수 있습니다. 사용 사례에 추가 기능, 추가 위치 제공 또는 보다 정확한 안테나 위치가 필요한 [AWS Support Center Console](#) 경우를 통해 문의하세요.

AWS Ground Station 안테나 중 하나를 사용하려면 특정 위치에서 시간을 예약해야 합니다. 이 예약을 고객 응대라고 합니다. 고객 응대를 성공적으로 예약하려면 성공하려면 추가 데이터가 AWS Ground Station 필요합니다.

- 위성을 하나 이상의 위치에 온보딩해야 합니다. 이렇게 하면 요청된 위치에서 다양한 기능을 운영할 수 있는 승인을 받을 수 있습니다.
- 위성에는 유효한 에페메리스가 있어야 합니다. 이렇게 하면 안테나에 가시선이 있고 접촉 중에 위성을 정확하게 가리킬 수 있습니다.
- 유효한 미션 프로파일이 있어야 합니다. 이를 통해 위성에 데이터를 수신하고 전송하는 방법을 포함하여 고객 응대의 작동 방식을 사용자 지정할 수 있습니다. 동일한 차량에 대해 여러 미션 프로파일을 활용하여 서로 다른 운영 태세 또는 시나리오에 맞게 서로 다른 접촉을 생성할 수 있습니다.

위성 온보딩

위성들에 온보딩하는 AWS Ground Station 것은 통합 및 테스트와 함께 데이터 수집, 기술 검증, 스펙트럼 라이선스와 관련된 다단계 프로세스입니다. 가이드의 [위성 온보딩](#) 섹션에서는이 프로세스를 안내합니다.

미션 프로파일 구성

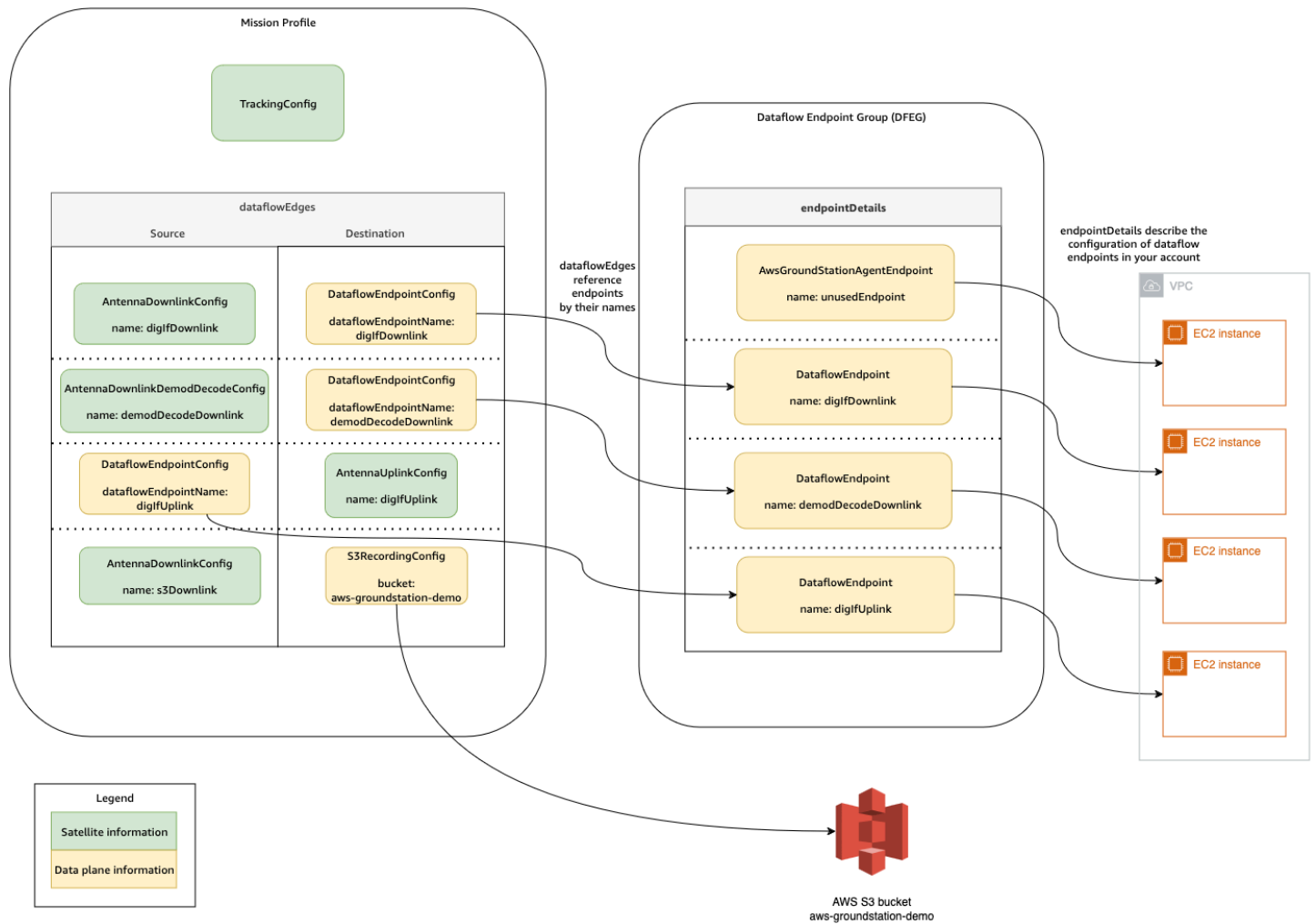
위성 주파수 정보, [데이터 영역](#) 정보 및 기타 세부 정보는 미션 프로파일로 캡슐화됩니다. 미션 프로파일은 구성 구성 요소의 모음입니다. 이를 통해 사용 사례에 맞게 다양한 미션 프로파일에서 구성 구성 요소를 재사용할 수 있습니다. 미션 프로파일은 개별 위성을 직접 참조하지 않고 대신 기술적 기능에 대한 정보만 제공하므로 동일한 구성을 가진 여러 위성에서 미션 프로파일을 재사용할 수도 있습니다.

유효한 미션 프로파일에는 추적 구성과 하나 이상의 데이터 흐름이 있습니다. 추적 구성은 고객 응대 중 추적에 대한 기본 설정을 지정합니다. 데이터 흐름 내의 각 구성 페어는 소스와 대상을 설정합니다. 위성 및 운영 모드에 따라 정확한 데이터 흐름 수는 업링크 및 다운링크 통신 경로와 데이터 처리 측면을 나타내는 미션 프로파일에서 달라집니다.

- 고객 응대 중에 사용할 Amazon VPC, Amazon S3 및 Amazon EC2 리소스를 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [데이터 흐름 작업](#).
- 각 구성의 작동 방식에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station Configs 사용](#).
- 예상되는 모든 파라미터에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station 미션 프로파일 사용](#).
- 사용 사례를 지원하기 위해 다양한 미션 프로파일을 생성하는 방법에 대한 예는 섹션을 참조하세요 [미션 프로파일 구성 예](#).

다음 다이어그램은 예제 미션 프로파일과 필요한 추가 리소스를 보여줍니다. 이 예제에서는 유연성을 보여주기 위해 unusedEndpoint라는 이 미션 프로파일에 필요하지 않은 데이터 흐름 엔드포인트를 보여줍니다. 이 예제는 다음 데이터 흐름을 지원합니다.

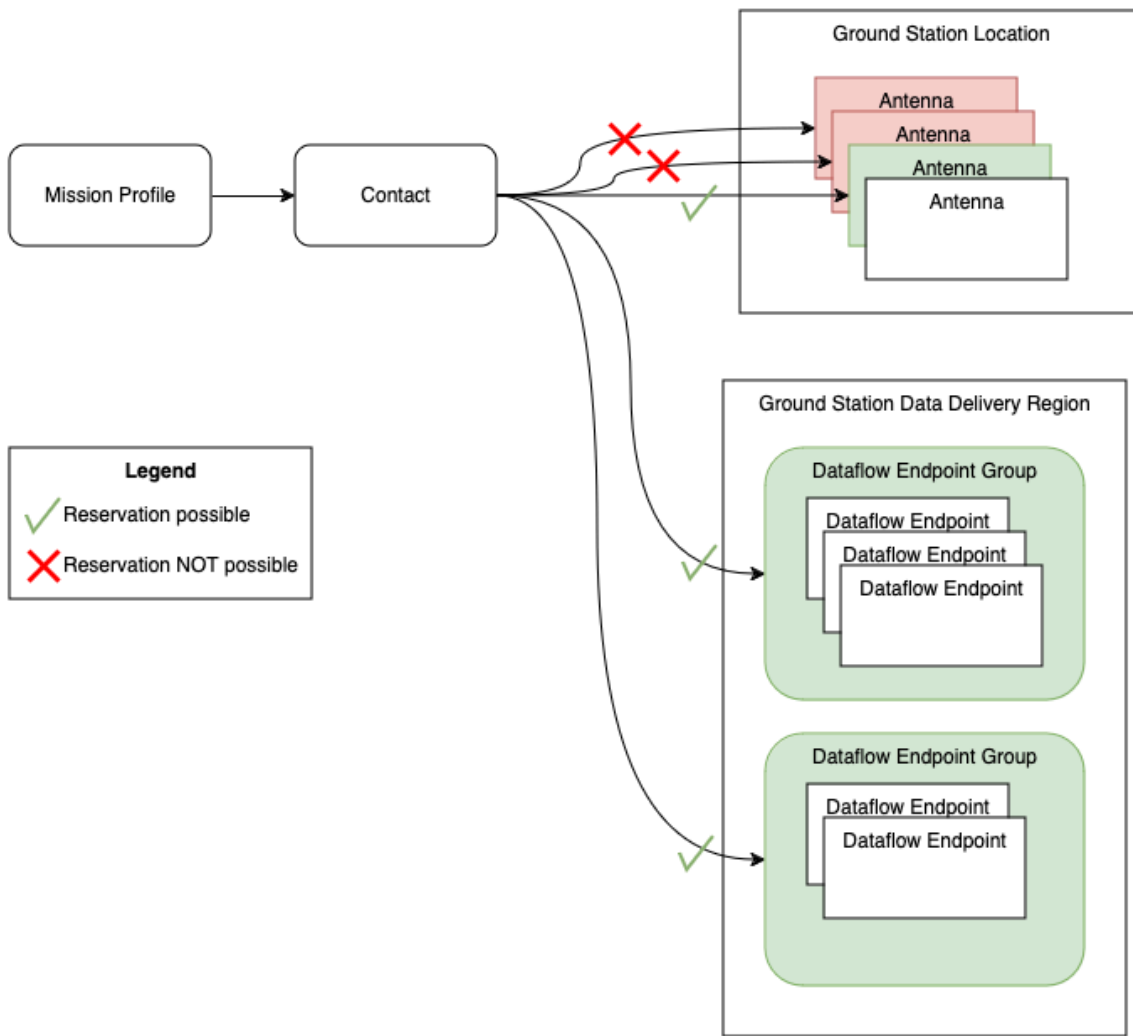
- 관리하는 Amazon EC2 인스턴스에 대한 디지털 중간 주파수 데이터의 동기식 다운링크입니다. digIfDownlink 이름으로 표시됩니다.
- Amazon S3 버킷에 대한 디지털 중간 주파수 데이터의 비동기 다운링크입니다. 버킷 이름 aws-groundstation-demo로 표시됩니다.
- 복조 및 디코딩된 데이터를 관리하는 Amazon EC2 인스턴스로 동기식 다운링크. demodDecodeDownlink 이름으로 표시됩니다.
- 관리하는 Amazon EC2 인스턴스에서 AWS Ground Station 관리형 안테나로 데이터를 동기식으로 업링크합니다. digIfUplink 이름으로 표시됩니다.



고객 응대 예약

유효한 미션 프로파일을 사용하면 온보딩된 위성과의 연락을 요청할 수 있습니다. 고객 응대 예약 요청은 비동기식이므로 글로벌 안테나 서비스가 관련된 모든 AWS 리전에서 일관된 일정을 달성할 수 있습니다. 이 프로세스 중에 요청된 지상국 위치의 다양한 안테나를 평가하여 고객 응대를 사용할 수 있고 처리할 수 있는지 확인합니다. 이 프로세스 중에 구성된 데이터 흐름 엔드포인트도 평가되어 가용성을 결정합니다. 이 평가가 진행되는 동안 고객 응대 상태는 예약 중이 됩니다.

이 비동기식 예약 프로세스는 요청 후 5분 이내에 완료되지만 일반적으로 1분 이내에 완료됩니다. 예약 시간 동안 [이벤트 AWS Ground Station 로 자동화](#) 이벤트 기반 모니터링을 검토하세요.



수행할 수 있고 가용성이 있는 고객 응대는 예약 고객 응대로 이어집니다. 예약된 고객 응대를 사용하면 고객 응대를 수행하는 데 필요한 리소스가 미션 프로파일에 정의된 대로 필요한 AWS 리전에 예약되었습니다. 수행할 수 없거나 사용할 수 없는 부분이 있는 고객 응대는 **FAILED_TO_SCHEDULE** 고객 응대로 이어집니다. 디버깅 세부 정보는 [FAILED_TO_SCHEDULE 연락처 문제 해결](#) 섹션을 참조하세요.

고객 응대 실행

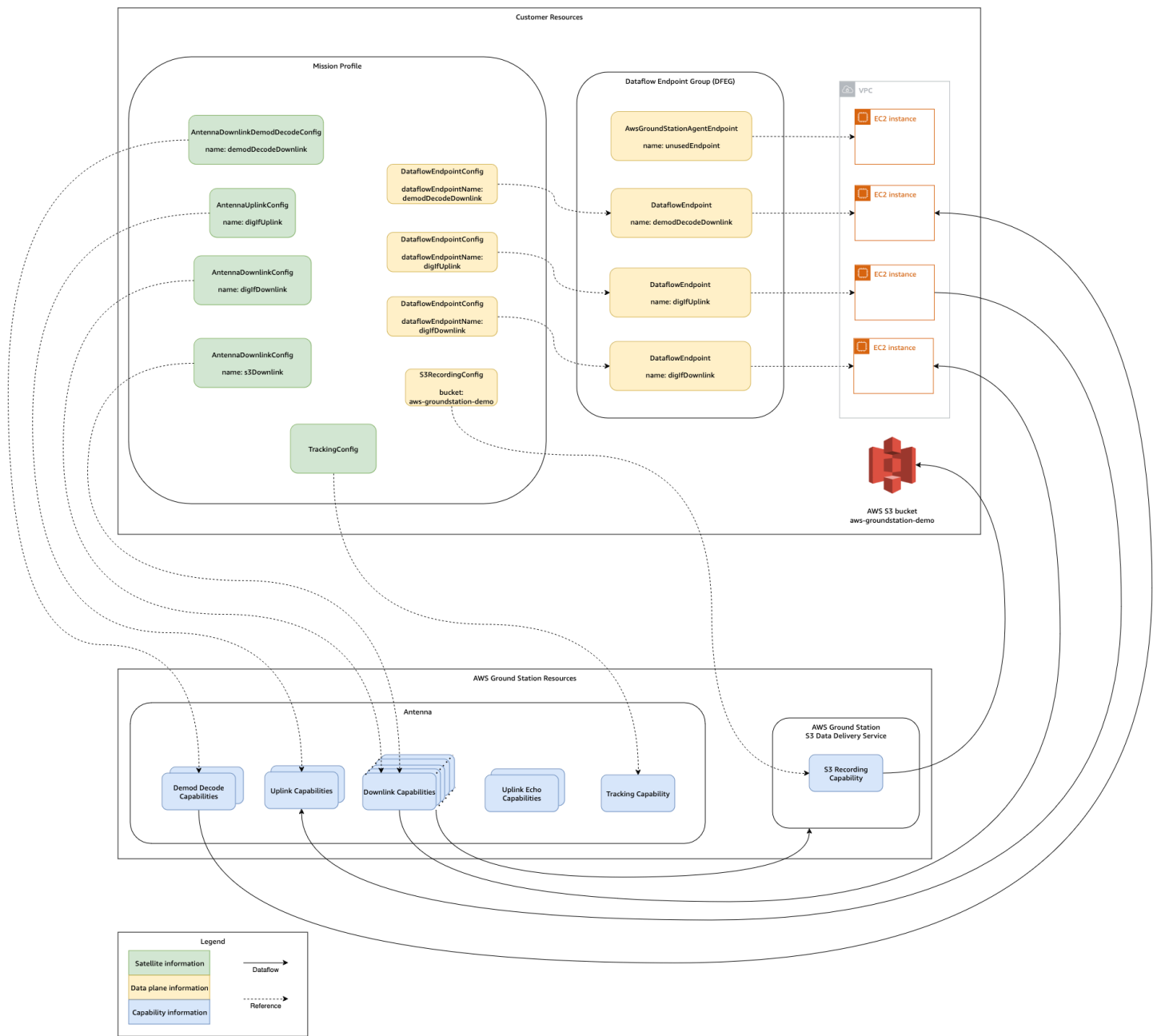
AWS Ground Station 는 고객 응대 예약 중에 AWS 관리형 리소스를 자동으로 오케스트레이션합니다. 해당하는 경우 미션 프로파일에서 데이터 흐름 엔드포인트로 정의한 EC2 리소스를 오케스트레이션할 책임이 있습니다. 리소스 오케스트레이션을 자동화하여 비용을 절감하기 위한 [AWS EventBridge 이벤트를](#) AWS Ground Station 제공합니다. 자세한 내용은 [이벤트 AWS Ground Station 로 자동화](#) 섹션을 참조하세요.

고객 응대 중에 고객 응대 성능에 대한 원격 측정이 AWS CloudWatch로 전달됩니다. 실행 중에 고객 응대를 모니터링하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [를 사용한 모니터링 이해 AWS Ground Station](#).

다음 다이어그램은 고객 응대 중에 오케스트레이션된 동일한 리소스를 표시하여 이전 예제를 계속합니다.

Note

이 예제에서 모든 안테나 기능이 사용된 것은 아닙니다. 예를 들어, 각 안테나에는 여러 주파수 및 편광을 지원하는 12개 이상의 안테나 다운링크 기능이 있습니다. AWS Ground Station 안테나에서 사용할 수 있는 각 기능 유형의 수와 지원되는 주파수 및 편광에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station 사이트 기능](#).



고객 응대가 끝나면 AWS Ground Station 는 고객 응대의 성과를 평가하고 최종 고객 응대 상태를 결정합니다. 오류가 감지되지 않는 연락처는 COMPLETED 연락처 상태가 됩니다. 서비스 오류로 인해 고객 응대 중에 데이터 전송 문제가 발생한 고객 응대는 AWS_FAILED 상태가 됩니다. 클라이언트 또는 사용자 오류가 고객 응대 중에 데이터 전송 문제를 일으킨 고객 응대는 실패 상태가 됩니다. 사전 통과 또는 사후 통과 중에 발생하는 연락 시간 이외의 오류는 심사 중에 고려되지 않습니다.

자세한 내용은 [고객 응대 수명 주기 이해](#) 섹션을 참조하세요.

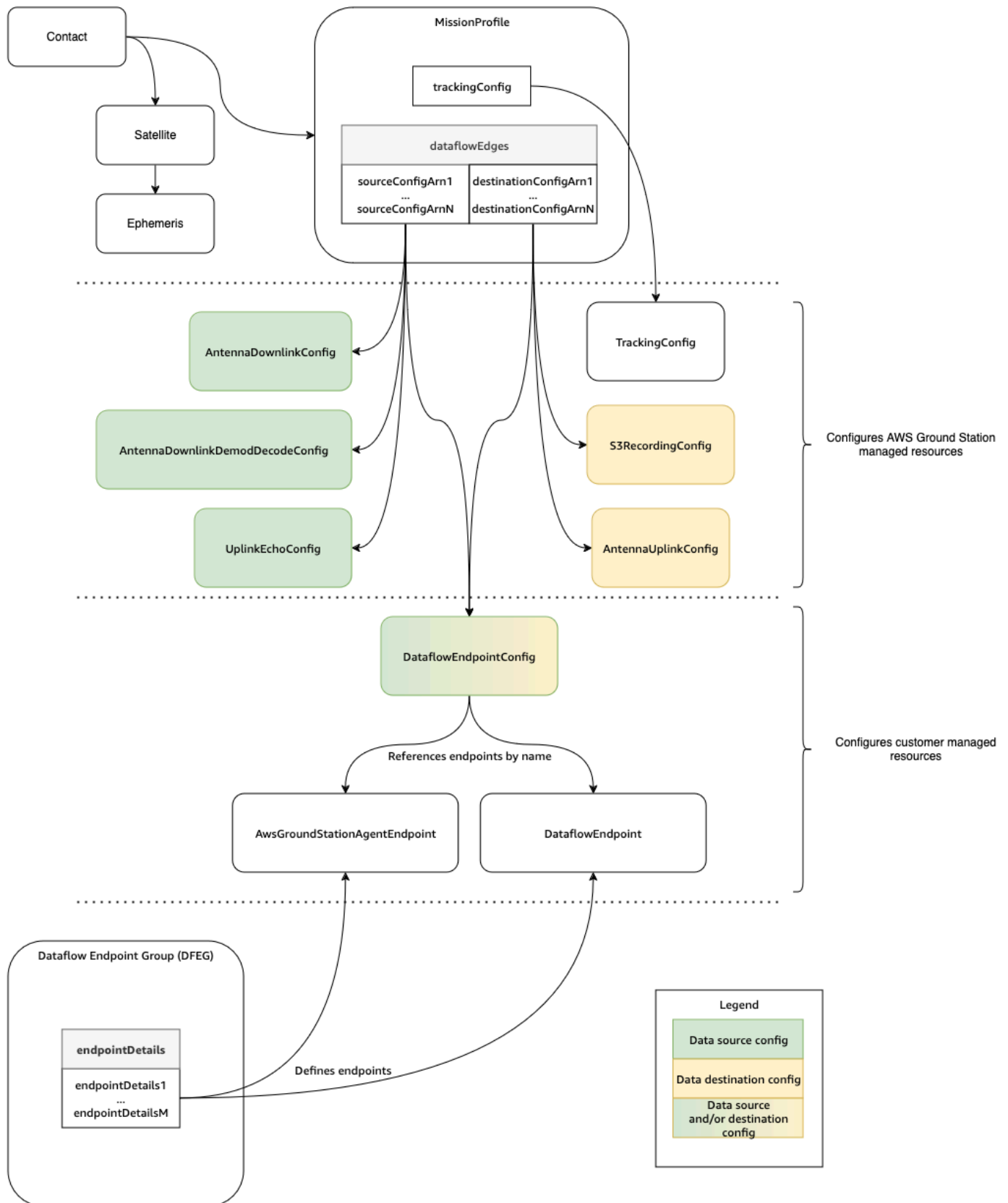
디지털 트윈

의 디지털 트윈 기능을 AWS Ground Station 사용하면 가상 지상국 위치에 대한 고객 응대를 예약할 수 있습니다. 이러한 가상 지상국은 안테나 기능, 사이트 마스크 및 실제 GPS 좌표를 포함하여 프로덕션 지상국의 정확한 복제본입니다. 디지털 트윈 기능을 사용하면 프로덕션 지상국에 비해 적은 비용으로 고객 응대 오케스트레이션 워크플로를 테스트할 수 있습니다. 자세한 내용은 [AWS Ground Station 디지털 트윈 기능 사용](#)을 참조하세요.

AWS Ground Station 핵심 구성 요소 이해

이 섹션에서는 AWS Ground Station의 핵심 구성 요소에 대한 자세한 정의를 제공합니다.

다음 다이어그램은의 핵심 구성 요소와 이러한 구성 요소가 서로 AWS Ground Station 어떻게 관련되어 있는지 보여줍니다. 화살표는 각 구성 요소가 해당 종속성을 가리키는 구성 요소 간의 종속성 방향을 나타냅니다.



다음 주제에서는 AWS Ground Station 핵심 구성 요소에 대해 자세히 설명합니다.

주제

- [AWS Ground Station 미션 프로파일 사용](#)
- [AWS Ground Station Configs 사용](#)
- [AWS Ground Station Dataflow 엔드포인트 그룹 사용](#)
- [AWS Ground Station 에이전트 사용](#)

AWS Ground Station 미션 프로파일 사용

미션 프로파일에는 접촉이 실행되는 방법에 대한 구성과 파라미터가 포함되어 있습니다. 접촉을 예약하거나 이용 가능한 접촉을 검색할 때 사용하려는 미션 프로파일을 제공합니다. 미션 프로파일은 모든 구성을 종합하며 접촉 중에 안테나가 구성되는 방식과 데이터가 이동하는 위치를 정의합니다.

미션 프로파일은 동일한 라디오 특성을 공유하는 위성 간에 공유할 수 있습니다. 추가 데이터 흐름 엔드포인트 그룹을 생성하여 별자리에 대해 수행하려는 최대 동시 연락처를 제한할 수 있습니다.

추적 구성은 미션 프로파일 내에서 고유한 필드로 지정됩니다. 추적 구성은 고객 응대 중에 프로그램 추적 및 자동 추적을 사용하기 위한 기본 설정을 지정하는 데 사용됩니다. 자세한 내용은 [추적 구성](#) 섹션을 참조하세요.

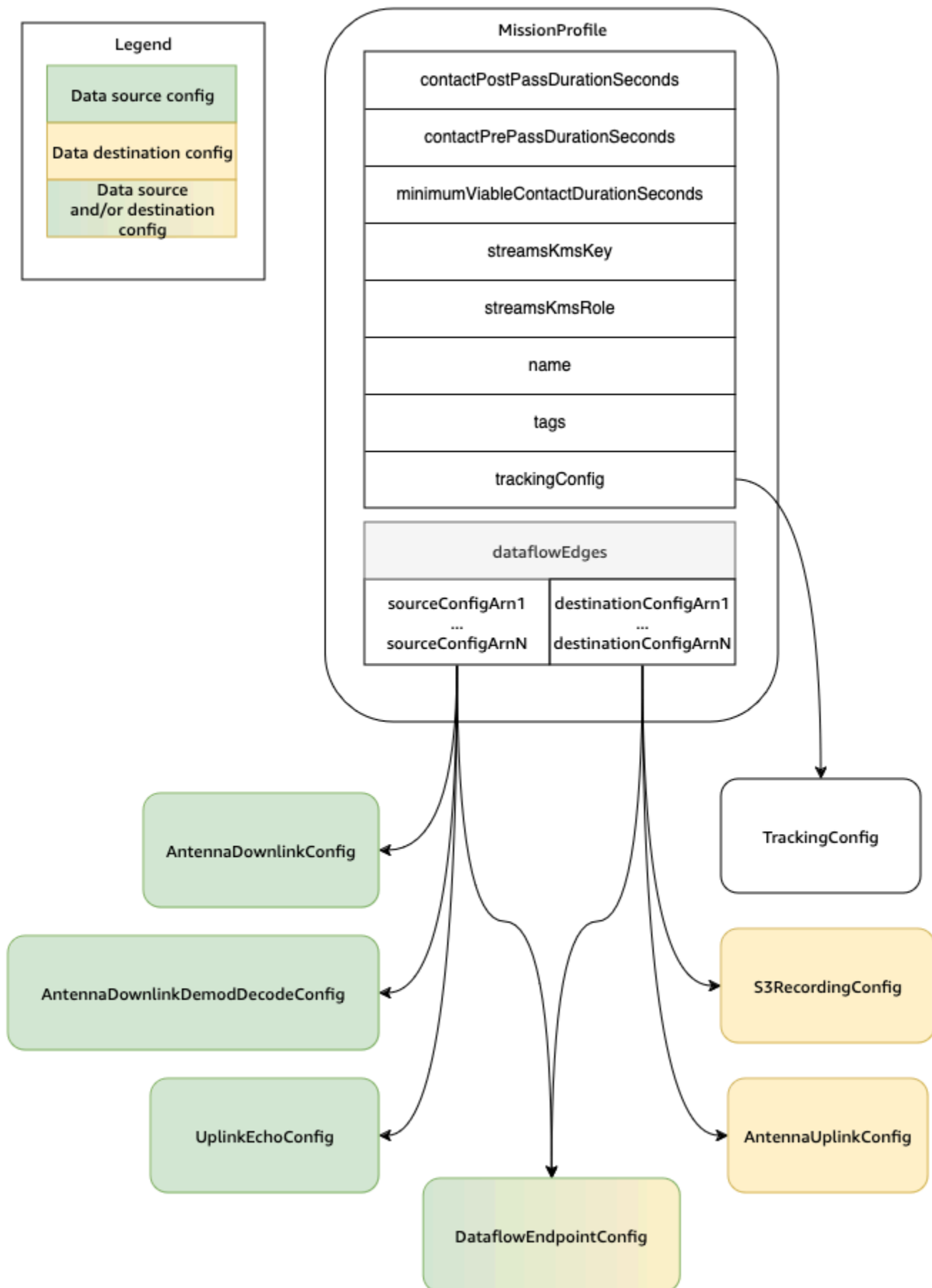
다른 모든 구성은 미션 프로파일의 dataflowEdges 필드에 포함됩니다. 이러한 구성은 각각 데이터 및 관련 구성을 전송하거나 수신할 수 있는 AWS Ground Station 관리형 리소스를 나타내는 데이터 흐름 노드로 생각할 수 있습니다. dataflowEdges 필드는 필요한 소스 및 대상 데이터 흐름 노드(구성)를 정의합니다. 단일 데이터 흐름 엣지는 두 개의 구성 [Amazon 리소스 이름\(ARNs\) 목록입니다](#). 첫 번째는 소스 구성이고 두 번째는 대상 구성입니다. 두 구성 간에 데이터 흐름 엣지를 지정하면 고객 응대 중에 데이터가 AWS Ground Station 흐르는 위치와 위치를 알 수 있습니다. 자세한 내용은 [AWS Ground Station Configs 사용](#) 섹션을 참조하세요.

contactPrePassDurationSeconds 및 contactPostPassDurationSeconds 사용하면 CloudWatch 이벤트 알림을 받을 연락처를 기준으로 시간을 지정할 수 있습니다. 연락처와 관련된 이벤트 타임라인은 섹션을 참조하세요 [고객 응대 수명 주기 이해](#).

미션 프로파일의 name 필드는 생성하는 미션 프로파일 간을 구별하는 데 도움이 됩니다.

streamsKmsRole 및 streamsKmsKey는 AWS Ground Station 에이전트를 통한 데이터 전송 AWS Ground Station 에에서 사용하는 암호화를 정의하는 데 사용됩니다. [에 대한 전송 중 데이터 암호화 AWS Ground Station](#) 섹션을 참조하세요.

telemetrySinkConfigArn 필드는 선택 사항이며 연락 중에 AWS Ground Station 원격 측정을 활성화할 수 있습니다. 지정된 경우는 고객 응대를 실행하는 동안 실시간에 가까운 원격 측정 데이터를 계정으로 AWS Ground Station 스트리밍합니다. 원격 측정 구성 및 사용에 대한 자세한 내용은 섹션을 참조하세요 [원격 측정 작업](#).



파라미터 및 예제의 전체 목록은 다음 설명서에 포함되어 있습니다.

- [AWS::GroundStation::MissionProfile CloudFormation 리소스 유형](#)

AWS Ground Station Configs 사용

Config는 각 연락처의 각 측면에 대한 파라미터를 정의하는 데 AWS Ground Station 사용하는 리소스입니다. 원하는 구성을 미션 프로파일에 추가하면 접착을 실행할 때 해당 미션 프로파일이 사용됩니다. 여러 가지 유형의 구성을 정의할 수 있습니다. 구성은 세 가지 범주로 그룹화할 수 있습니다.

- 구성 추적
- 데이터 흐름 구성
- 원격 측정 구성

TrackingConfig는 추적 구성의 유일한 유형입니다. 접착 중에 안테나의 자동 추적 설정을 구성하는 데 사용되며 미션 프로파일에 필요합니다.

미션 프로파일 데이터 흐름에 사용할 수 있는 구성은 각각 데이터를 보내거나 받을 수 있는 AWS Ground Station 관리형 리소스를 나타내는 데이터 흐름 노드로 생각할 수 있습니다. 미션 프로파일에는 이러한 구성 쌍이 하나 이상 필요합니다. 하나는 데이터 소스를 나타내고 다른 하나는 대상을 나타냅니다. 이러한 구성은 다음 표에 요약되어 있습니다.

구성 이름	데이터 흐름 소스/대상
AntennaDownlinkConfig	소스
AntennaDownlinkDemodDecodeConfig	소스
UplinkEchoConfig	소스
S3RecordingConfig	Destination
AntennaUplinkConfig	Destination
DataflowEndpointConfig	소스 및/또는 대상

TelemetrySinkConfig는 원격 측정 구성의 유일한 유형입니다. 고객 응대 중에 원격 측정 데이터가 전달되는 위치를 구성하는 데 사용되며 미션 프로파일에서 선택 사항입니다. 포함된 경우는 고객 응대를 실행하는 동안 계정에 실시간에 가까운 원격 측정을 AWS Ground Station 스트리밍합니다.

CloudFormation AWS Command Line Interface, 또는 AWS Ground Station API를 사용하여 구성에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요. 특정 구성 유형에 대한 설명서 링크도 아래에 제공됩니다.

- [AWS::GroundStation::Config CloudFormation 리소스 유형](#)
- [Config AWS CLI 참조](#)
- [Config API 참조](#)

추적 구성

미션 프로파일에서 추적 구성을 사용하여 접촉 중에 자동 추적을 활성화할지 여부를 결정할 수 있습니다. 이 구성에는 autotrack이라는 단일 파라미터가 있습니다. autotrack 파라미터에는 다음과 같은 값이 있습니다.

- REQUIRED - 자동 추적이 접촉에 필수입니다.
- PREFERRED - 자동 추적이 접촉에 선호되지만, 자동 추적이 없더라도 접촉을 실행할 수 있습니다.
- REMOVED - 자동 추적을 접촉에 사용하지 않습니다.

AWS Ground Station 는 자동 추적을 사용하지 않을 때 에페메리스를 기준으로 가리키는 프로그래밍 방식 추적을 활용합니다. 에페메리스 구성 방법에 대한 [가 에페메리스를 AWS Ground Station 사용하는 방법 이해](#) 자세한 내용은 섹션을 참조하세요.

자동 추적은 예상 신호가 발견될 때까지 프로그램 추적을 사용합니다. 이 경우 신호의 강도에 따라 계속 추적됩니다.

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 구성을 추적하는 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config TrackingConfig CloudFormation 속성](#)
- [Config AWS CLI 참조](#)(trackingConfig -> (structure)섹션 참조)
- [TrackingConfig API 참조](#)

안테나 다운링크 구성

안테나 다운링크 구성을 사용하여 접착 중에 다운링크를 위한 안테나를 구성할 수 있습니다. 이 구성은 다운링크 접착 중에 사용해야 하는 주파수, 대역폭 및 편광을 지정하는 스펙트럼 구성으로 이루어집니다.

이 구성은 데이터 흐름의 소스 노드를 나타냅니다. 무선 주파수 데이터의 디지털화를 담당합니다. 이 노드에서 스트리밍된 데이터는 신호 데이터/IP 형식을 따릅니다. 이 구성으로 데이터 흐름을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요. [데이터 흐름 작업](#)

다운링크 사용 사례에 복조나 디코드가 필요한 경우 [안테나 다운링크 복조 디코드 구성](#) 단원을 참조하세요.

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 안테나 다운링크 구성에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config AntennaDownlinkConfig CloudFormation 속성](#)
- [구성 AWS CLI 참조](#)(antennaDownlinkConfig -> (structure)섹션 참조)
- [AntennaDownlinkConfig API 참조](#)

안테나 다운링크 복조 디코드 구성

안테나 다운링크 복조 디코딩 구성은 복조 및/또는 디코딩으로 다운링크 접착을 실행하는 데 사용할 수 있는 보다 복잡하고 사용자 지정 가능한 구성 유형입니다. 이러한 유형의 고객 응대를 실행하려면 AWS Support 티켓을 개설하세요. [AWS Support Center Console](#). 사용 사례에 적합한 구성 및 임무 프로필을 정의할 수 있도록 도와드리겠습니다.

이 구성은 데이터 흐름의 소스 노드를 나타냅니다. 무선 주파수 데이터를 디지털화하고 지정된 대로 복조 및 디코딩을 수행할 책임이 있습니다. 이 노드에서 스트리밍되는 데이터는 복조/디코딩된 데이터/IP 형식을 따릅니다. 이 구성으로 데이터 흐름을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요. [데이터 흐름 작업](#)

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 안테나 다운링크 데모 디코딩 구성에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config AntennaDownlinkDemodDecodeConfig CloudFormation 속성](#)
- [구성 AWS CLI 참조](#)(antennaDownlinkDemodDecodeConfig -> (structure)섹션 참조)
- [AntennaDownlinkDemodDecodeConfig API 참조](#)

안테나 업링크 구성

안테나 업링크 구성을 사용하여 업링크 접속 중에 안테나를 구성할 수 있습니다. 주파수, 편광 및 목표 유효 등방성 복사 전력(EIRP)이 포함된 스펙트럼 구성으로 구성됩니다. 업링크 루프백을 구성하는 방법에 대한 자세한 내용은 [안테나 업링크 에코 구성](#) 단원을 참조하세요.

이 구성은 데이터 흐름의 대상 노드를 나타냅니다. 제공된 디지털화된 무선 주파수 데이터 신호를 아날로그 신호로 변환하고 위성이 수신할 수 있도록 내보냅니다. 이 노드로 스트리밍되는 데이터는 신호 데이터/IP 형식을 충족할 것으로 예상됩니다. 이 구성으로 데이터 흐름을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요. [데이터 흐름 작업](#)

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 안테나 업링크 구성에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config AntennaUplinkConfig CloudFormation 속성](#)
- [Config AWS CLI 참조](#)(antennaUplinkConfig -> (structure)섹션 참조)
- [AntennaUplinkConfig API 참조](#)

안테나 업링크 에코 구성

업링크 에코 구성은 업링크 에코를 실행하는 방법을 안테나에 알립니다. 업링크 에코를 사용하여 우주선으로 전송된 명령을 검증하고 다른 고급 작업을 수행할 수 있습니다. 이는 AWS Ground Station 안테나에서 전송한 실제 신호(예: 업링크)를 기록하여 달성할 수 있습니다. 이렇게 하면 안테나가 데이터 흐름 엔드포인트로 다시 보낸 신호가 에코되고 전송된 신호와 일치해야 합니다. 업링크 에코 구성에는 업링크 구성의 ARN이 포함됩니다. 안테나는 업링크 에코를 실행할 때 ARN이 가리키는 업링크 구성의 파라미터를 사용합니다.

이 구성은 데이터 흐름의 소스 노드를 나타냅니다. 이 노드에서 스트리밍된 데이터는 신호 데이터/IP 형식을 충족합니다. 이 구성으로 데이터 흐름을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요. [데이터 흐름 작업](#)

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 업링크 에코 구성에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config UplinkEchoConfig CloudFormation 속성](#)
- [Config AWS CLI 참조](#)(uplinkEchoConfig -> (structure)섹션 참조)
- [UplinkEchoConfig API 참조](#)

데이터 흐름 엔드포인트 구성

Note

데이터 흐름 엔드포인트 구성은 Amazon EC2로의 데이터 전송에만 사용되며 Amazon S3로의 데이터 전송에는 사용되지 않습니다.

데이터 흐름 엔드포인트 구성을 사용하여 [데이터 흐름 엔드포인트 그룹](#) 내의 어느 데이터 흐름 엔드포인트를 접속 중에 어느 곳으로 전송할지 또는 어느 곳으로 데이터를 전송할지 지정할 수 있습니다. 데이터 흐름 엔드포인트 구성의 두 파라미터는 데이터 흐름 엔드포인트의 이름과 영역을 지정합니다. 연락처를 예약할 때 AWS Ground Station 는 지정한 [미션 프로파일을](#) 분석하고 미션 프로파일에 포함된 데이터 흐름 엔드포인트 구성에 의해 지정된 모든 데이터 흐름 엔드포인트가 포함된 AWS 리전 내의 데이터 흐름 엔드포인트 그룹을 찾으려고 시도합니다. 적절한 데이터 흐름 엔드포인트 그룹이 발견되면 고객 응대 상태가 예약됨이 되고, 그렇지 않으면 FAILED_TO_SCHEDULE이 됩니다. 연락처의 가능한 상태에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station 고객 응대 상태](#).

데이터 흐름 엔드포인트 구성의 dataflowEndpointName 속성은 데이터 흐름 엔드포인트 그룹의 어느 데이터 흐름 엔드포인트가 접속 중에 어느 데이터로 또는 어느 데이터로부터 전달되는지를 지정합니다.

dataflowEndpointRegion 속성은 데이터플로우 엔드포인트가 위치한 리전을 지정합니다. 데이터 흐름 엔드포인트 구성에 리전이 지정된 경우는 지정된 리전의 데이터 흐름 엔드포인트를 AWS Ground Station 찾습니다. 리전을 지정하지 않으면 AWS Ground Station 기본적으로 연락처의 지상국 리전으로 설정됩니다. 데이터 흐름 엔드포인트의 리전이 접속의 Ground Station 리전과 동일하지 않은 경우 접속은 리전 간 데이터 전송 접속으로 간주됩니다. 리전 간 데이터 흐름에 대한 [데이터 흐름 작업](#) 자세한 내용은 섹션을 참조하세요.

데이터 흐름의 다양한 이름 지정 체계가 사용 사례에 어떻게 도움이 될 수 있는지 [AWS Ground Station Dataflow 엔드포인트 그룹 사용](#)에 대한 팁은 섹션을 참조하세요.

이 구성으로 데이터 흐름을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요. [데이터 흐름 작업](#)

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 데이터 흐름 엔드포인트 구성에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config DataflowEndpointConfig CloudFormation 속성](#)
- [Config AWS CLI 참조](#)(dataflowEndpointConfig -> (structure)섹션 참조)
- [DataflowEndpointConfig API 참조](#)

Amazon S3 레코딩 구성

Note

Amazon S3 레코딩 구성은 Amazon S3로의 데이터 전송에만 사용되며 Amazon EC2로의 데이터 전송에는 사용되지 않습니다.

이 구성은 데이터 흐름의 대상 노드를 나타냅니다. 이 노드는 데이터 흐름의 소스 노드에서 들어오는 데이터를 pcap 데이터로 캡슐화합니다. 이 구성으로 데이터 흐름을 구성하는 방법에 대한 자세한 내용은 섹션을 참조하세요. [데이터 흐름 작업](#)

S3 레코딩 구성을 사용하여 사용된 이름 지정 규칙과 함께 다운로드된 데이터를 전송할 Amazon S3 버킷을 지정할 수 있습니다. 다음은 이러한 파라미터에 대한 제한 및 세부 정보를 지정합니다.

- Amazon S3 버킷 이름은 aws-groundstation로 시작해야 합니다.
- IAM 역할에는 groundstation.amazonaws.com 서비스 보안 주체가 역할을 수임하도록 허용하는 신뢰 정책이 있어야 합니다. 예제는 아래 [신뢰 정책 예제](#)를 참조하세요. 구성 생성 중에는 구성 리소스 ID가 존재하지 않으므로 신뢰 정책은 *your-config-id* 대신 별표(*)를 사용해야 하며, 신뢰 정책은 생성 후 구성 리소스 ID를 사용하여 업데이트할 수 있습니다.

신뢰 정책 예제

역할의 신뢰 정책을 업데이트하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 관리](#)를 참조하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
```

```

        "aws:SourceAccount": "999999999999"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:groundstation:us-
east-1:999999999999:config/s3-recording/your-config-id"
      }
    }
  ]
}

```

- IAM 역할에는 역할이 버킷에서 작업을 수행하고 버킷 객체에 대한 `s3:GetBucketLocation` 작업을 수행할 수 있도록 허용하는 `s3:PutObject` 정책이 있어야 합니다. Amazon S3 버킷에 버킷 정책이 있는 경우, 버킷 정책은 IAM 역할이 이러한 작업을 수행하도록 허용해야 합니다. 예제는 아래 [역할 정책 예제](#)를 참조하세요.

역할 정책 예제

IAM 정책을 업데이트하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 정책 관리](#)를 참조하세요.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::your-bucket-name"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ]
    }
  ]
}

```

```

    ],
    "Resource": [
        "arn:aws:s3:::your-bucket-name/*"
    ]
}
]
}

```

- 접두사는 S3 데이터 객체의 이름을 지정할 때 사용됩니다. 대체할 선택적 키를 지정할 수 있습니다. 이러한 값은 연락처 세부 정보의 해당 정보로 대체됩니다. 예를 들어 접두사가 {satellite_id}/{year}/{month}/{day} 대체되고 다음과 같은 출력이 발생합니다.
fake_satellite_id/2021/01/10

대체를 위한 선택적 키: {satellite_id} | {config-name} | {config-id} | {year} | {month} | {day} |

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 S3 레코딩 구성에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config S3RecordingConfig CloudFormation 속성](#)
- [Config AWS CLI 참조](#)(s3RecordingConfig -> (structure)섹션 참조)
- [S3RecordingConfig API 참조](#)

원격 측정 싱크 구성

원격 측정 싱크 구성을 사용하여 위성 접촉 중에 원격 측정 데이터를 전송할 위치를 지정할 수 있습니다. 원격 측정 싱크 구성은 선택 사항이며 원격 측정 지원 연락처를 예약하기 위해 미션 프로파일에 추가됩니다. 다음은 이러한 파라미터에 대한 제한 및 세부 정보를 지정합니다.

- IAM 역할에는 groundstation.amazonaws.com 서비스 보안 주체가 역할을 수임하도록 허용하는 신뢰 정책이 있어야 합니다. 예제는 아래 [신뢰 정책 예제](#)를 참조하세요.

신뢰 정책 예제

역할의 신뢰 정책을 업데이트하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 관리](#)를 참조하세요.

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

- IAM 역할에는 역할이 스트림에서 `kinesis:DescribeStream` `kinesis:PutRecord` 및 `kinesis:PutRecords` 작업을 수행하도록 허용하는 IAM 정책이 있어야 합니다. 예제는 아래 [역할 정책 예제](#)를 참조하세요.

역할 정책 예제

IAM 정책을 업데이트하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 정책 관리](#)를 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesis:DescribeStream",
        "kinesis:PutRecord",
        "kinesis:PutRecords"
      ],
      "Resource": "arn:aws:kinesis:us-east-2:999999999999:stream/your-stream-name"
    }
  ]
}
```

미션 프로파일에 원격 측정 싱크 구성을 포함하면 AWS Ground Station 는 연락 중에 원격 측정 데이터를 계정으로 스트리밍합니다. 원격 측정 유형, 데이터 형식 및 필요한 AWS 리소스 설정에 대한 자세한 내용은 섹션을 참조하세요 [원격 측정 작업](#).

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 원격 측정 싱크 구성에서 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::GroundStation::Config TelemetrySinkConfig CloudFormation 속성](#)
- [Config AWS CLI 참조](#)(telemetrySinkConfig -> (structure)섹션 참조)
- [TelemetrySinkConfig API 참조](#)

AWS Ground Station Dataflow 엔드포인트 그룹 사용

데이터 흐름 엔드포인트는 고객 응대 중에 데이터를 동기식으로 스트리밍할 위치를 정의합니다. 데이터 흐름 엔드포인트는 항상 데이터 흐름 엔드포인트 그룹의 일부로 생성됩니다. 여러 데이터 흐름 엔드포인트를 한 그룹에 포함시키면 지정된 엔드포인트를 단일 접속 중에 모두 함께 사용할 수 있다고 선언하는 것입니다. 예를 들어, 접속에서 세 개의 개별 데이터 흐름 엔드포인트에 데이터를 전송해야 하는 경우 미션 프로파일의 데이터 흐름 엔드포인트 구성과 일치하는 세 개의 엔드포인트가 단일 데이터 흐름 엔드포인트 그룹에 있어야 합니다.

Dataflow 엔드포인트 그룹 버전

AWS Ground Station 는 두 가지 버전의 데이터 흐름 엔드포인트 그룹을 지원합니다.

- DataflowEndpointGroup - [데이터 흐름 엔드포인트](#)를 사용하는 업링크 및 다운링크와 [AWS Ground Station 에이전트 엔드포인트](#)에 대한 다운링크 전용을 지원하는 원래 구현
- DataflowEndpointGroupV2 - 향상된 명확성과 기능으로 AWS Ground Station 에이전트 엔드포인트에 대한 업링크 및 다운링크 데이터 흐름을 모두 지원하는 업데이트된 버전

데이터 흐름 엔드포인트 그룹 비교

기능	DataflowEndpointGroup	DataflowEndpointGroupV2
지원되는 엔드포인트 유형	DataflowEndpoint, AwsGroundStationAgentEndpoint	DownlinkAwsGroundStationAgentEndpoint, UplinkAwsGroundStationAgentEndpoint
업링크를 지원하는 엔드포인트	DataflowEndpoint	UplinkAwsGroundStationAgentEndpoint

기능	DataflowEndpointGroup	DataflowEndpointGroupV2
다운링크를 지원하는 엔드포인트	DataflowEndpoint, AwsGroundStationAgentEndpoint	DownlinkAwsGroundStationAgentEndpoint

DataflowEndpointGroupV2는 업링크 데이터 흐름을 지원하고 데이터 흐름 엔드포인트 그룹을 둘러싼 언어를 더 명확하게 만들기 위해 생성되었습니다. 모든 새 사용 사례에 대해 DataflowEndpointGroupV2와 함께 [UplinkAwsGroundStationAgentEndpoint](#) 및 [DownlinkAwsGroundStationAgentEndpoint](#) 엔드포인트를 사용하는 것이 좋습니다. [DataflowEndpointGroupV2](#) DataflowEndpointGroup은 이전 버전과의 호환성을 위해 계속 지원되지만 DataflowEndpointGroupV2는 향상된 기능과 더 명확한 구성 옵션을 제공합니다.

Tip

데이터 흐름 엔드포인트는 고객 응대를 실행할 때 선택한 이름으로 식별됩니다. 이러한 이름은 계정 전체에서 고유할 필요는 없습니다. 이렇게 하면 동일한 미션 프로파일을 사용하여 서로 다른 위성과 안테나의 여러 접촉을 동시에 실행할 수 있습니다. 이는 운영 특성이 동일한 위성 집합이 있는 경우에 유용할 수 있습니다. 위성 집합에 필요한 최대 동시 연락 수에 맞게 데이터 흐름 엔드포인트 그룹 수를 확장할 수 있습니다.

데이터 흐름 엔드포인트 그룹에 있는 하나 이상의 리소스가 접촉에 사용 중이면 전체 그룹이 해당 접촉의 기간에 예약됩니다. 여러 접촉을 동시에 실행할 수 있지만, 이러한 접촉을 서로 다른 데이터 흐름 엔드포인트 그룹에서 실행해야 합니다.

Important

데이터 흐름 엔드포인트 그룹은 이를 사용하여 연락을 예약할 수 있는 HEALTHY 상태여야 합니다. HEALTHY 상태가 아닌 데이터 흐름 엔드포인트 그룹의 문제를 해결하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [정상 상태가 아닌 DataflowEndpointGroups 문제 해결](#).

CloudFormation AWS Command Line Interface 또는 AWS Ground Station API를 사용하여 데이터 흐름 엔드포인트 그룹에 대한 작업을 수행하는 방법에 대한 자세한 내용은 다음 설명서를 참조하세요.

- [AWS::CloudFormation::DataflowEndpointGroup CloudFormation 리소스 유형](#)
- [데이터 흐름 엔드포인트 그룹 AWS CLI 참조](#)

- [데이터플로우 엔드포인트 그룹 API 참조](#)

데이터 흐름 엔드포인트

데이터 흐름 엔드포인트 그룹의 멤버는 데이터 흐름 엔드포인트입니다. 지원되는 엔드포인트 유형은 사용하는 데이터 흐름 엔드포인트 그룹 버전에 따라 다릅니다.

DataflowEndpointGroup 엔드포인트

DataflowEndpointGroup은 [데이터 흐름 엔드포인트](#)를 사용하는 업링크 및 다운링크와 [AWS Ground Station 에이전트 엔드포인트](#)에 대한 다운링크 전용을 지원합니다. 두 유형의 엔드포인트 모두에 대해 데이터 흐름 엔드포인트 그룹을 생성하기 전에 지원 구문(예: IP 주소)을 생성합니다. 사용할 데이터 흐름 엔드포인트 유형과 지원 구문을 설정하는 방법에 [데이터 흐름 작업](#) 대한 권장 사항은 섹션을 참조하세요.

다음 섹션에서는 지원되는 두 엔드포인트 유형을 모두 설명합니다.

Important

단일 데이터 흐름 엔드포인트 그룹 내의 모든 데이터 흐름 엔드포인트는 동일한 유형이어야 합니다. [AWS Ground Station 에이전트 엔드포인트](#)를 동일한 그룹의 [Dataflow 엔드포인트](#)와 혼합할 수 없습니다. 사용 사례에 두 가지 유형의 엔드포인트가 모두 필요한 경우 각 유형에 대해 별도의 데이터 흐름 엔드포인트 그룹을 생성해야 합니다.

DataflowEndpointGroupV2의 경우 동일한 그룹에서 [UplinkAwsGroundStationAgentEndpoint](#)와 [DownlinkAwsGroundStationAgentEndpoint](#)를 혼합할 수 있습니다.

AWS Ground Station 에이전트 엔드포인트

AWS Ground Station 에이전트 엔드포인트는 AWS Ground Station 에이전트를 소프트웨어 구성 요소로 활용하여 연결을 종료합니다. AWS Ground Station 에이전트 엔드포인트를 구성하려면 EndpointDetails의 AwsGroundStationAgentEndpoint 필드만 채웁니다. AWS Ground Station 에이전트에 대한 자세한 내용은 전체 [AWS Ground Station 에이전트 사용 설명서](#)를 참조하세요.

AwsGroundStationAgentEndpoint는 다음 구성 요소로 이루어져 있습니다.

- Name - 데이터 흐름 엔드포인트 이름입니다. 연락처가이 데이터 흐름 엔드포인트를 사용하려면이 이름이 데이터 흐름 엔드포인트 구성에 사용된 이름과 일치해야 합니다.

- EgressAddress - 에이전트에서 데이터를 내보내는 데 사용되는 IP 및 포트 주소입니다.
- IngressAddress - 에이전트로 데이터를 수신하는 데 사용되는 IP 및 포트 주소입니다.

데이터 흐름 엔드포인트

Dataflow 엔드포인트는 네트워킹 애플리케이션을 소프트웨어 구성 요소로 활용하여 연결을 종료합니다. 디지털 신호 데이터를 업링크하거나, 50MHz 미만의 디지털 신호 데이터를 다운링크하거나, 복조/디코딩된 신호 데이터를 다운링크하려는 경우 데이터 흐름 엔드포인트를 사용합니다. 데이터 흐름 엔드포인트를 구성하려면 EndpointDetails의 Endpoint 및 Security Details 필드를 채웁니다.

Endpoint는 다음 구성 요소로 이루어져 있습니다.

- Name - 데이터 흐름 엔드포인트 이름입니다. 연락처가이 데이터 흐름 엔드포인트를 사용하려면이 이름이 데이터 흐름 엔드포인트 구성에 사용된 이름과 일치해야 합니다.
- Address - 사용된 IP 및 포트 주소입니다.

SecurityDetails는 다음 구성 요소로 이루어져 있습니다.

- roleArn - VPC에서 탄력적 네트워크 인터페이스(ENI)를 생성하기 위해 수입 AWS Ground Station 할 역할의 Amazon 리소스 이름(ARN)입니다. ENIs 이러한 ENI는 접속 중에 스트리밍되는 데이터의 수신 및 송신 지점 역할을 합니다.
- securityGroupIds - 탄력적 네트워크 인터페이스에 연결할 보안 그룹입니다.
- subnetIds -가 탄력적 네트워크 인터페이스를 AWS Ground Station 배치하여 인스턴스로 스트림을 전송할 수 있는 서브넷 목록입니다. 여러 서브넷이 지정된 경우 서로 라우팅할 수 있어야 합니다. 서브넷이 서로 다른 가용 영역(AZs)에 있는 경우 AZ 간 데이터 전송 요금이 발생할 수 있습니다.

roleArn으로 전달된 IAM 역할에는 groundstation.amazonaws.com 서비스 보안 주체가 역할을 수입하도록 허용하는 신뢰 정책이 있어야 합니다. 예제는 아래 [신뢰 정책 예제](#)를 참조하세요. 엔드포인트 생성 중에는 엔드포인트 리소스 ID가 존재하지 않으므로 신뢰 정책은 *your-endpoint-id*에 별표(*)를 사용해야 합니다. 특정 데이터 흐름 엔드포인트 그룹에 대한 신뢰 정책의 범위를 지정하기 위해 엔드포인트 리소스 ID를 사용하도록 생성 후 업데이트할 수 있습니다.

IAM 역할에는가 ENIs를 AWS Ground Station 설정하도록 허용하는 IAM 정책이 있어야 합니다. 예제는 아래 [역할 정책 예제](#)를 참조하세요.

신뢰 정책 예제

역할의 신뢰 정책을 업데이트하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 관리](#)를 참조하세요.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "999999999999"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:groundstation:us-east-1:999999999999:dataflow-endpoint-group/your-endpoint-id"
        }
      }
    }
  ]
}
```

역할 정책 예제

IAM 정책을 업데이트하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 정책 관리](#)를 참조하세요.

JSON

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

```
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:DeleteNetworkInterface",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:DescribeSubnets",
    "ec2:DescribeVpcs",
    "ec2:DescribeSecurityGroups"
  ],
  "Resource": "*"
}
```

DataflowEndpointGroupV2 엔드포인트

DataflowEndpointGroupV2에는 보다 명확한 구성과 향상된 기능을 제공하는 특수 엔드포인트 유형이 도입되었습니다.

- [UplinkAwsGroundStationAgentEndpoint](#) - 업링크 데이터 흐름에 최적화됨
- [DownlinkAwsGroundStationAgentEndpoint](#) - 다운링크 데이터 흐름에 최적화됨

이러한 특수 엔드포인트는 일반 [AwsGroundStationAgentEndpoint](#)를 데이터 흐름을 더 쉽게 설정하고 관리할 수 있는 방향별 구성으로 대체합니다.

업링크 AWS Ground Station 에이전트 엔드포인트

[UplinkAwsGroundStationAgentEndpoint](#)는 업링크 데이터 흐름을 위해 특별히 설계되었으며 더 명확한 구성 옵션을 제공합니다. 위성으로 업링크 AWS Ground Station 하기 위해에 데이터를 제공해야 하는 경우 이 엔드포인트 유형을 사용합니다.

UplinkAwsGroundStationAgentEndpoint는 다음 구성 요소로 이루어져 있습니다.

- Name - 데이터 흐름 엔드포인트 이름입니다. 연락처가 이 데이터 흐름 엔드포인트를 사용하려면 이 이름이 데이터 흐름 엔드포인트 구성에 사용된 이름과 일치해야 합니다.
- IngressAddressAndPort - 에이전트에 데이터를 입력하기 위한 단일 IP 및 포트 주소
- AgentIpAndPortAddress - 에이전트 통신을 위한 포트 범위

다운링크 AWS Ground Station 에이전트 엔드포인트

[DownlinkAwsGroundStationAgentEndpoint](#)는 협대역 다운링크, 광대역 복조/디코딩, 업링크 에코 시나리오를 포함한 다운링크 데이터 흐름에 최적화되어 있습니다.

[DownlinkAwsGroundStationAgentEndpoint](#)는 다음 구성 요소로 이루어져 있습니다.

- Name - 데이터 흐름 엔드포인트 이름입니다. 연락처가이 데이터 흐름 엔드포인트를 사용하려면이 이름이 데이터 흐름 엔드포인트 구성에 사용된 이름과 일치해야 합니다.
- EgressAddressAndPort - 에이전트의 데이터 출력을 위한 단일 IP 및 포트 주소
- AgentIpAndPortAddress - 에이전트 통신을 위한 포트 범위

데이터 흐름 엔드포인트 그룹 생성

다음 버전 중 하나를 사용하여 데이터 흐름 엔드포인트 그룹을 생성할 수 있습니다.

CreateDataflowEndpointGroup

이전 버전과의 호환성을 위해 또는 일반 [AwsGroundStationAgentEndpoint](#) 또는 [DataflowEndpoint](#) 유형을 사용해야 하는 경우 [CreateDataflowEndpointGroup](#)을 사용합니다.

CreateDataflowEndpointGroupV2

새 구현에 [CreateDataflowEndpointGroupV2](#)를 사용하면 업링크 및 다운링크 데이터 흐름을 모두 지원하는 특수 엔드포인트 유형을 활용할 수 있습니다. 이 API는 [UplinkAwsGroundStationAgentEndpoint](#) 및 [DownlinkAwsGroundStationAgentEndpoint](#)만 지원합니다.

마이그레이션 고려 사항

현재 [DataflowEndpointGroup](#)을 사용하는 경우 변경 없이 기존 구성을 계속 사용할 수 있습니다.는 완전한 이전 버전과의 호환성을 AWS Ground Station 유지합니다.

새 [DataflowEndpointGroupV2](#)를 사용하도록 마이그레이션하고 현재 [Dataflow](#) 엔드포인트 애플리케이션과 함께 [DataflowEndpoint](#)를 사용하여 데이터를 수신하는 경우 AWS Ground Station 에이전트를 사용하도록 마이그레이션해야 합니다. 다운링크에 AWS Ground Station 에이전트를 이미 사용하고 있는 경우 업링크에도 동일한 에이전트 인스턴스를 사용할 수 있습니다. 추가 에이전트 인스턴스는 필요하지 않습니다.

DataflowEndpointGroupV2로 마이그레이션하려면

1. DataflowEndpoint에서 마이그레이션하는 경우 AWS Ground Station 에이전트 [AWS Ground Station 사용 설명서에 따라 에이전트](#)를 설정합니다.
2. 데이터 흐름 방향을 식별하고 적절한 엔드포인트 유형([UplinkAwsGroundStationAgentEndpoint](#) 또는 [DownlinkAwsGroundStationAgentEndpoint](#))을 생성합니다.
3. 해당 엔드포인트를 참조하는 [DataflowEndpointGroupV2](#) 생성
4. 이름으로 새 [DataflowEndpointGroupV2](#)를 참조하는 새 데이터 흐름 엔드포인트 구성 생성
DataflowEndpointGroupV2
5. 데이터 흐름 엔드포인트 구성을 데이터 흐름 엣지로 참조하는 새 미션 프로파일 생성
6. 새 미션 프로파일을 사용하여 고객 응대 예약
7. 프로덕션에 배포하기 전에 구성 테스트

전체 워크플로에 대한 자세한 내용은 [AWS Ground Station 핵심 구성 요소 이해](#) 및 섹션을 참조하세요 [요구성 생성](#).

AWS Ground Station 에이전트 사용

AWS Ground Station 에이전트를 사용하면 AWS Ground Station 고객 응대 중에 동기 광대역 디지털 중간 주파수(DigIF) 데이터 흐름을 수신(다운링크)할 수 있습니다.

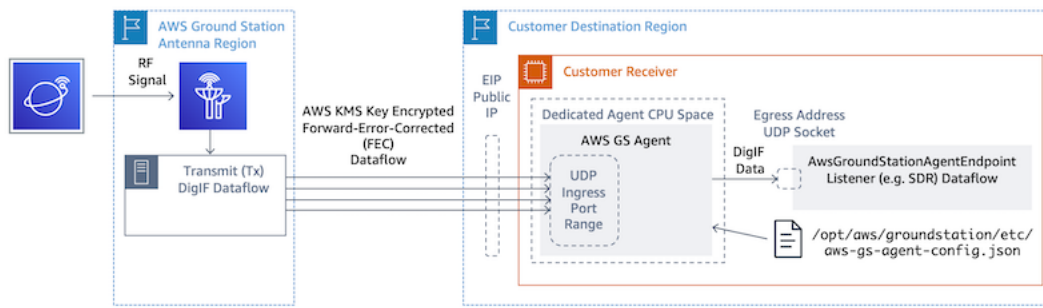
작동 방식

데이터 전송을 위한 두 가지 옵션을 선택할 수 있습니다.

1. EC2 인스턴스로 데이터 전송 - 소유한 EC2 인스턴스로 데이터 전송. AWS Ground Station 에이전트를 관리합니다. 이 옵션은 실시간에 가까운 데이터 처리가 필요한 경우에 가장 적합할 수 있습니다. EC2 데이터 전송에 대한 자세한 내용은 [데이터 흐름 작업](#) 섹션을 참조하세요.
2. S3 버킷으로 데이터 전송 - AWS S3 버킷으로의 데이터 전송은에서 완벽하게 관리합니다 AWS Ground Station. S3 데이터 전송에 대한 자세한 내용은 [시작하기](#) 안내서를 참조하세요.

두 데이터 전송 모드 모두 AWS 리소스 세트를 생성해야 합니다. 신뢰성, 정확성 및 지원 가능성을 보장하려면 CloudFormation을 사용하여 AWS 리소스를 생성하는 것이 좋습니다. 각 접촉은 EC2 또는 S3에만 데이터를 전송할 수 있으며, 두 접촉 모두에 동시에 데이터를 전송할 수는 없습니다.

다음 다이어그램은 소프트웨어 정의 라디오(SDR) 또는 유사한 리스너를 사용하여 AWS Ground Station 안테나 리전에서 EC2 인스턴스로의 DigIF 데이터 흐름을 보여줍니다.



추가 정보

자세한 내용은 전체 [AWS Ground Station 에이전트 사용 설명서](#)를 참조하세요.

시작하기

시작하기 전의 기본 개념을 숙지해야 합니다 AWS Ground Station. 자세한 내용은 [AWS Ground Station 작동 방식](#) 섹션을 참조하세요.

에 가입 AWS 계정

를 시작하려면이 AWS필요합니다 AWS 계정. 생성에 대한 자세한 AWS 계정내용은 AWS Account Management 참조 안내서의 [시작하기 AWS 계정](#)를 참조하세요.

AWS 계정에 AWS Ground Station 권한 추가

관리 사용자 AWS Ground Station 없이를 사용하려면 새 정책을 생성하여 AWS 계정에 연결해야 합니다.

1. 에 로그인 AWS Management Console 하고 [IAM 콘솔](#)을 엽니다.
2. 새 정책 생성. 다음 단계를 사용합니다.
 - a. 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
 - b. JSON 탭에서 다음 값 중 하나를 사용해 JSON을 편집합니다. 애플리케이션에 가장 적합한 JSON을 사용합니다.
 - Ground Station 관리자 권한의 경우 Action을 다음과 같이 groundstation: *으로 설정합니다.

JSON

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

- 읽기 전용 권한의 경우 Action을 다음과 같이 `groundstation:Get*`, `groundstation:List*` 및 `groundstation:Describe*`로 설정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "groundstation:Get*",
        "groundstation:List*",
        "groundstation:Describe*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

- 멀티 팩터 인증을 통한 추가 보안의 경우 Action을 다음과 같이 `groundstation:*`로, Condition/Bool을 다음과 같이 `aws:MultiFactorAuthPresent:true`로 설정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "groundstation:*",
      "Resource": "*",
      "Condition": {
        "Bool": {
          "aws:MultiFactorAuthPresent": true
        }
      }
    }
  ]
}
```

}

3. IAM 콘솔에서, 생성한 정책을 원하는 사용자에게 연결합니다.

IAM 사용자 및 정책 연결에 대한 자세한 내용은 [IAM 사용 설명서](#)를 참조하세요.

위성 온보딩

위성 온보딩하는 AWS Ground Station 것은 통합 및 테스트와 함께 데이터 수집, 기술 검증, 스펙트럼 라이선스와 관련된 다단계 프로세스입니다. 또한 비공개 계약(NDAs)이 필요합니다.

고객 온보딩 프로세스 개요

위성 온보딩은 AWS Ground Station 콘솔 페이지의 [위성 및 리소스](#) 섹션에서 찾을 수 있는 수동 프로세스입니다. 다음은 전체 프로세스에 대한 설명입니다.

1. [AWS Ground Station 위치](#) 섹션을 검토하여 위성이 지리적 및 무선 주파수 특성을 충족하는지 확인합니다.
2. 위성 온보딩을 시작하려면 AWS Ground Station 콘솔 페이지의 위성 [및 리소스 섹션에서 위성 온보딩](#) 설문지를 제출 AWS Ground Station하세요. 조직 이름, 필요한 빈도, 위성이 시작되거나 시작된 시기, 위성의 궤도 유형, 사용 계획 여부 등 미션 및 위성 요구 사항에 대한 간략한 요약 포함 하세요 [AWS Ground Station 디지털 트윈 기능 사용](#).
3. 요청이 검토 및 승인되면 AWS Ground Station 는 사용하려는 특정 위치에서 규제 라이선스를 신청합니다. 이 단계의 기간은 위치 및 기존 규정에 따라 달라집니다.
4. 이 승인을 받으면 위성을 사용할 수 있습니다. AWS Ground Station 에서 업데이트 성공 알림을 보냅니다.

(선택 사항) 위성 이름 지정

온보딩 후 위성 레코드를 더 쉽게 인식하기 위해 위성 레코드에 이름을 추가할 수 있습니다. AWS Ground Station 콘솔에는 연락처 페이지를 사용할 때 Norad ID와 함께 위성의 사용자 정의 이름을 표시할 수 있습니다. 위성 이름을 표시하면 일정을 잡을 때 올바른 위성을 훨씬 쉽게 선택할 수 있습니다. 이를 위해 [태그](#)를 사용할 수 있습니다.

AWS Ground Station 위성에 태그 지정은 AWS CLI 또는 AWS SDK 중 하나를 사용하여 [태그 리소스](#) API를 통해 수행할 수 있습니다. SDKs 이 가이드에서는 AWS Ground Station CLI를 사용하여 퍼블릭 브로드캐스트 위성 Aqua(Norad ID 27424)에 태그를 지정하는 방법을 다룹니다 `us-west-2`.

AWS Ground Station CLI

를 사용하여와 상호 작용할 AWS CLI 수 있습니다 AWS Ground Station. AWS CLI 를 사용하여 위성에 태그를 지정하기 전에 다음 AWS CLI 사전 조건을 충족해야 합니다.

- AWS CLI 가 설치되어 있는지 확인합니다. 설치에 대한 자세한 내용은 AWS CLI 버전 2 설치를 AWS CLI참조하세요. <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>
- AWS CLI 가 구성되어 있는지 확인합니다. 구성에 대한 자세한 내용은 AWS CLI 버전 2 구성을 AWS CLI참조하세요. <https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html>
- AWS CLI에서 유지 관리되는 파일에 자주 사용되는 구성 설정과 보안 인증을 저장할 수 있습니다. AWS Ground Station 연락처를 예약하고 관리하려면 이러한 설정과 자격 증명이 필요합니다 AWS CLI. 구성 및 자격 증명 설정 저장에 대한 자세한 내용은 [구성 및 자격 증명 파일 설정 단원을](#) 참조하십시오.

AWS CLI 가 구성되고 사용할 준비가 되면 [AWS Ground Station CLI 명령 참조](#) 페이지를 검토하여 사용 가능한 명령을 숙지하세요. 이 서비스를 사용할 때는 AWS CLI 명령 구조를 따르고 명령줄에 접두사groundstation로 붙여를 사용하려는 서비스로 지정합니다 AWS Ground Station . AWS CLI 명령 구조에 대한 자세한 내용은 [AWS CLI 페이지의 명령 구조를 참조하세요](#). 예제 명령 구조는 다음과 같습니다.

```
aws groundstation <command> <subcommand> [options and parameters]
```

위성 이름 지정하기

먼저 태그하려는 위성의 ARN을 가져와야 합니다. AWS CLI의 [list-satellites](#) API를 통해이 작업을 수행할 수 있습니다.

```
aws groundstation list-satellites --region us-west-2
```

위의 CLI 명령을 실행하면 다음과 비슷한 출력 결과가 반환됩니다.

```
{
  "satellites": [
    {
      "groundStations": [
        "Ohio 1",
        "Oregon 1"
      ]
    }
  ]
}
```

```

    ],
    "noradSatelliteID": 27424,
    "satelliteArn":
"arn:aws:groundstation::111111111111:satellite/11111111-2222-3333-4444-555555555555",
    "satelliteId": "11111111-2222-3333-4444-555555555555"
  }
]
}

```

태그를 지정하려는 위성을 찾아 `satelliteArn`를 메모해 둡니다. 태그 지정에 대한 한 가지 중요한 주의 사항은 [태그 리소스](#) API에 리전별 ARN이 필요하고 [list-satellites](#)에서 반환하는 ARN이 전역적이라는 것입니다. 다음 단계에서는 태그를 삽입하려는 리전(예약하려는 리전 등)으로 ARN을 확장해야 합니다. 이 예제에서는 `us-west-2`를 사용합니다. 이번 변경으로 ARN은 다음과 같이 변경됩니다. 기존:

```
arn:aws:groundstation::111111111111:satellite/11111111-2222-3333-4444-555555555555
```

변경 후:

```
arn:aws:groundstation:us-west-2:111111111111:satellite/11111111-2222-3333-4444-555555555555
```

콘솔에 위성 이름을 표시하려면 위성에 키로 사용되는 `"Name"`과 있는 태그가 있어야 합니다. 또한를 사용하므로 따옴표 AWS CLI는 백슬래시로 이스케이프 처리해야 합니다. 태그는 다음과 같이 표시됩니다.

```
{\"Name\": \"AQUA\"}
```

다음으로 [tag-resource](#) API를 호출하여 위성에 태그를 지정합니다. 이 작업은 다음과 AWS CLI 같이 수행할 수 있습니다.

```
aws groundstation tag-resource --region us-west-2 --resource-arn
arn:aws:groundstation:us-west-2:111111111111:satellite/11111111-2222-3333-4444-555555555555 --tags
'{"Name": "AQUA"}'
```

이렇게 하면 설정한 위성 이름을 AWS Ground Station 콘솔에서 확인할 수 있습니다.

위성 이름 변경

위성의 이름을 변경하려면 동일한 “Name” 키를 사용하지만 [태그의 값이 다른 위성 ARN을 사용하여 tag-resource](#)를 다시 호출하면 됩니다. 그러면 기존 태그가 업데이트되고 콘솔에 새 이름이 표시됩니다. 다음은 이에 대한 예제 직접적 호출입니다.

```
aws groundstation tag-resource --region us-west-2 --resource-arn
arn:aws:groundstation:us-
west-2:111111111111:satellite/11111111-2222-3333-4444-555555555555 --tags
'{"Name": "NewName"}'
```

위성 이름 제거

[untag-resource](#) API를 사용하여 위성에 설정된 이름을 제거할 수 있습니다. 이 API에는 태그가 속한 리전과 태그 키 목록이 포함된 위성 ARN이 필요합니다. 이름의 경우 태그 키는 “Name”입니다. AWS CLI를 사용하여 이 API를 직접적으로 호출하는 예제는 다음과 같습니다.

```
aws groundstation untag-resource --region us-west-2 --resource-arn
arn:aws:groundstation:us-
west-2:111111111111:satellite/11111111-2222-3333-4444-555555555555 --tag-keys Name
```

퍼블릭 브로드캐스트 위성

자체 위성을 온보딩하는 것 외에도 공개적으로 액세스할 수 있는 다운링크 통신 경로를 제공하는 지원되는 퍼블릭 브로드캐스트 위성을 사용하여 온보딩을 요청할 수 있습니다. 이를 통해 사용하여 이러한 위성에서 데이터를 다운링크 AWS Ground Station 할 수 있습니다.

Note

이러한 위성은 업링크할 수 없습니다. 공개적으로 액세스할 수 있는 다운링크 통신 경로만 사용할 수 있습니다.

AWS Ground Station 는 다음 위성의 온보딩을 지원하여 다이렉트 브로드캐스트 데이터를 다운링크합니다.

- Aqua
- SNPP

- JPSS-1/NOAA-20
- Terra

온보딩되면 이러한 위성에 즉시 액세스할 수 있습니다.는 서비스를 더 쉽게 시작할 수 있도록 미리 구성된 여러 CloudFormation 템플릿을 AWS Ground Station 유지합니다. 를 사용하는 방법에 [미션 프로파일 구성 예](#) 대한 예는 단원 AWS Ground Station 을 참조하십시오.

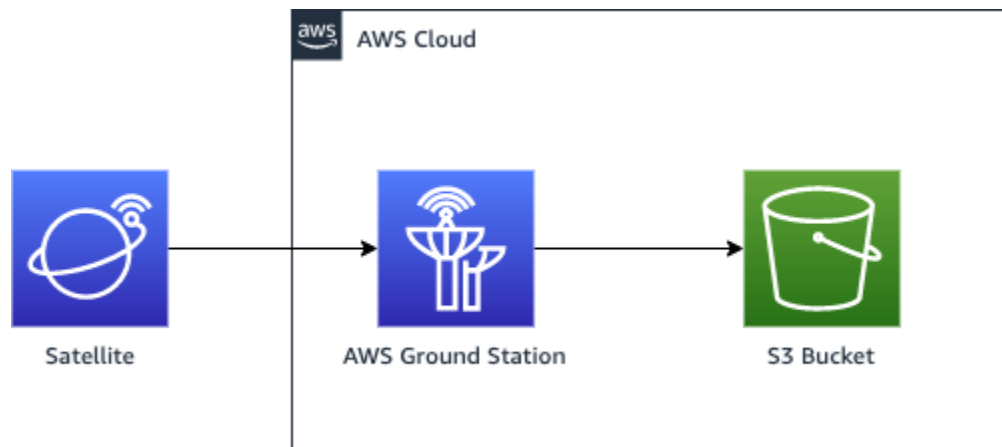
이러한 위성과 이들이 전송하는 데이터의 종류에 대한 자세한 내용은 [Aqua](#), [JPSS-1/NOAA-20](#) 및 [SNPP](#), [Terra](#)를 참조하세요.

데이터 흐름 통신 경로 계획

위성의 각 통신 경로에 대해 동기식 통신과 비동기식 통신 중에서 선택할 수 있습니다. 위성 및 사용 사례에 따라 하나 또는 두 가지 유형이 모두 필요할 수 있습니다. 동기식 통신 경로를 사용하면 거의 실시간 업링크는 물론 협대역 및 광대역 다운로드 작업을 수행할 수 있습니다. 비동기 통신 경로는 협대역 및 광대역 다운로드 작업만 지원합니다.

비동기 데이터 전송

Amazon S3로 데이터 전송 시 접촉 데이터가 계정의 Amazon S3 버킷에 비동기적으로 전송됩니다. 접촉 데이터는 패킷 캡처(pcap) 파일로 전송되므로 접촉 데이터를 소프트웨어 정의 라디오(SDR)로 재생하거나 pcap 파일에서 페이로드 데이터를 추출하여 처리할 수 있습니다. 안테나 하드웨어가 접촉 데이터를 수신할 때마다 pcap 파일이 30초마다 Amazon S3 버킷으로 전송되어 필요한 경우 연락 중에 접촉 데이터를 처리할 수 있습니다. 수신되면 자체 사후 처리 소프트웨어를 사용하여 데이터를 처리하거나 Amazon SageMaker AI 또는 Amazon Rekognition과 같은 다른 AWS 서비스를 사용할 수 있습니다. Amazon S3로 데이터를 전송하는 것은 위성 데이터를 다운로드하는 경우에만 사용할 수 있습니다. Amazon S3에서 위성으로 데이터를 업링크하는 것은 불가능합니다.



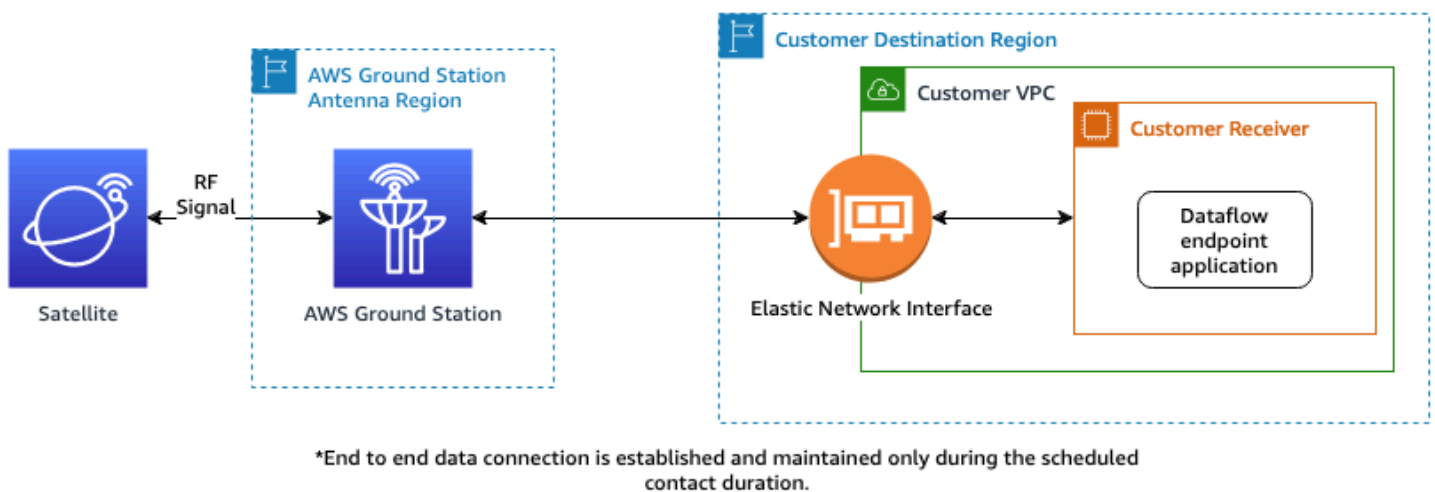
이 경로를 활용하려면 사용하여 데이터를 전송할 Amazon S3 버킷 AWS Ground Station 을 생성해야 합니다. 다음 단계에서는 다음 단계에서 S3 레코딩 구성도 생성해야 합니다. 버킷 이름 지정에 대한 [Amazon S3 레코딩 구성](#) 제한과 파일에 사용되는 이름 지정 규칙을 지정하는 방법은 섹션을 참조하세요.

동기식 데이터 전송

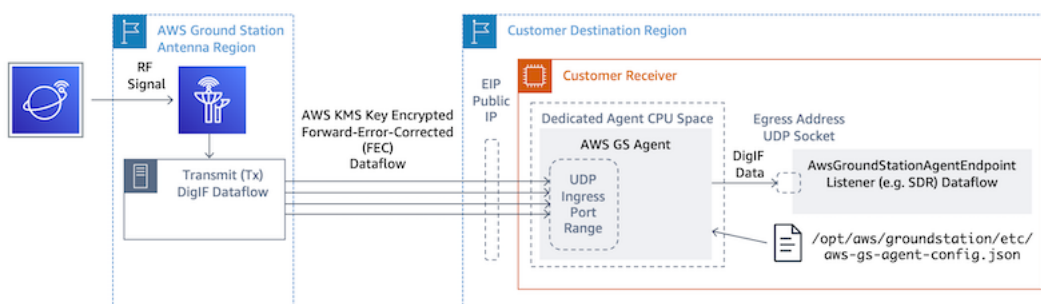
Amazon EC2로 데이터를 전송하면 접촉 데이터가 Amazon EC2 인스턴스 간에 스트리밍됩니다. Amazon EC2 인스턴스에서 실시간으로 데이터를 처리하거나 사후 처리를 위해 데이터를 전달할 수 있습니다.

동기 경로를 활용하려면 사용하여 Amazon EC2 인스턴스를 설정 및 구성하고 하나 이상의 Dataflow 엔드포인트 그룹을 생성해야 합니다. Amazon EC2 인스턴스를 구성하려면 참조하세요 [Amazon EC2 설정 및 구성](#). Dataflow 엔드포인트 그룹을 생성하려면 참조하세요 [AWS Ground Station Dataflow 엔드포인트 그룹 사용](#).

다음은 데이터 흐름 엔드포인트 구성을 사용하는 경우의 통신 경로를 보여줍니다.



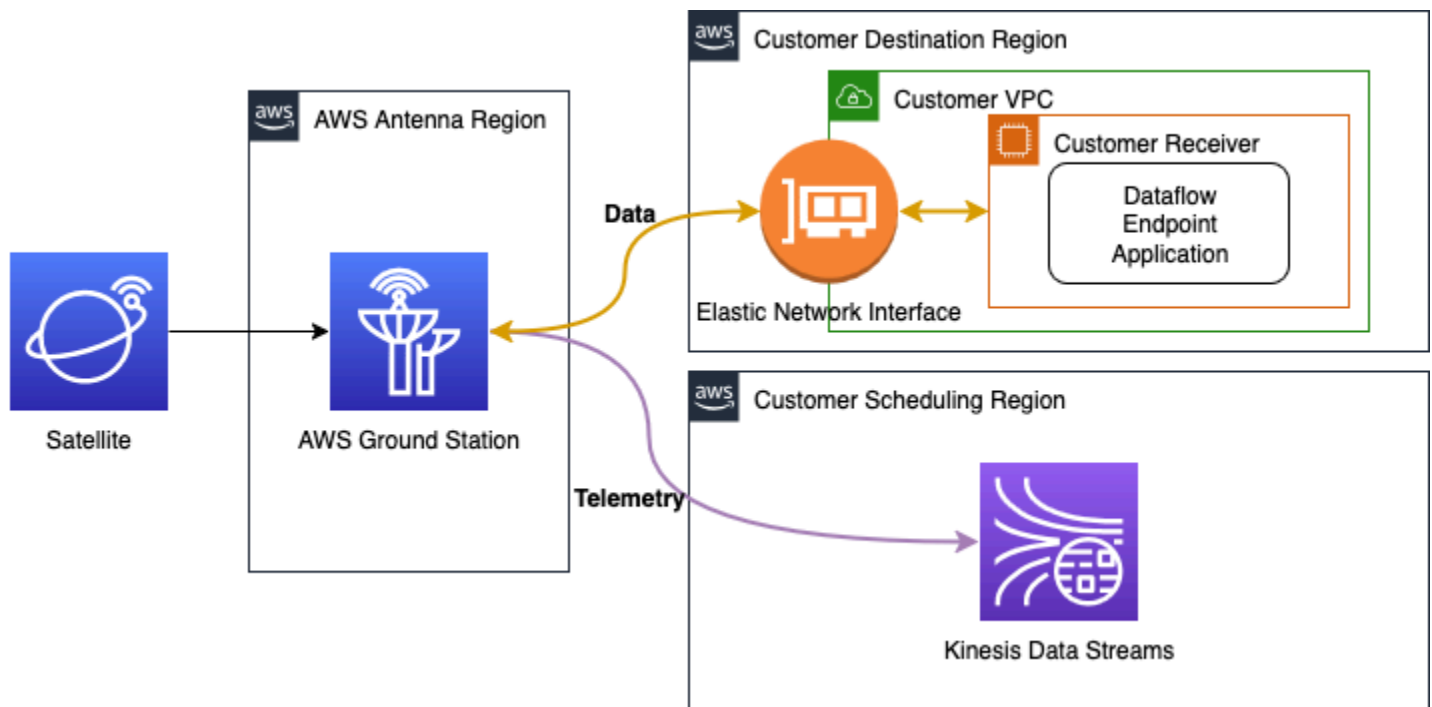
다음은 AWS Ground Station 에이전트 구성을 사용하는 경우의 통신 경로를 보여줍니다.



원격 측정 계획

AWS Ground Station 원격 측정은 위성 접촉 중에 안테나에서 AWS Ground Station AWS 계정으로 지표를 스트리밍하는 선택적 기능입니다. 이를 통해 고객 응대 성능을 거의 실시간으로 모니터링하고 사용자 지정 모니터링 솔루션을 구축할 수 있습니다.

AWS Ground Station 원격 측정을 사용하면 AWS Ground Station 안테나의 지표가 계정으로 직접 스트리밍됩니다. 원격 측정 데이터는 고객 응대 시작 시 스트리밍을 시작하고 고객 응대 기간 내내 계속됩니다. 원격 측정 데이터는 안테나 하드웨어에서 샘플링될 때 거의 실시간으로 계정에 전송됩니다. 수신되면 자체 사후 처리 소프트웨어를 사용하여 데이터를 처리하거나 Amazon Data Firehose 또는 같은 다른 AWS 서비스를 사용할 수 있습니다 AWS Lambda.



다음 단계에서는 미션 프로파일에 필요한 구성을 생성합니다. 원격 측정을 활성화하려면 추적 구성 및 데이터 흐름 구성 외에도 원격 측정 싱크 구성을 생성합니다. 자세한 설정 지침은 [원격 측정 설정](#) 섹션을 참조하세요.

TelemetrySinkConfig에 대한 자세한 내용은 섹션을 참조하세요 [원격 측정 싱크 구성](#).

구성 생성

이 단계에서는 필요에 따라 위성, 통신 경로 및 IAM, Amazon EC2 및 Amazon S3 리소스를 식별했습니다. 이 단계에서는 해당 파라미터를 저장하는 구성을 생성합니다 AWS Ground Station .

데이터 전송 구성

생성할 첫 번째 구성은 데이터를 전송할 위치 및 방법과 관련이 있습니다. 이전 단계의 정보를 사용하여 다음 구성 유형 중 많은 유형을 구성합니다.

- [Amazon S3 레코딩 구성](#) - Amazon S3 버킷에 데이터를 전송합니다.
- [데이터 흐름 엔드포인트 구성](#) - Amazon EC2 인스턴스로 데이터를 전송합니다.

원격 측정 구성(선택 사항)

고객 응대 중에 거의 실시간에 가까운 원격 측정을 받으려면 TelemetrySinkConfig를 생성할 수 있습니다. 이 구성은 선택 사항이며 AWS Ground Station 가 원격 측정 데이터를 전송하는 위치를 지정합니다.

- [원격 측정 싱크 구성](#) - 원격 측정 데이터를 계정에 전송합니다.

자세한 설정 지침은 섹션을 참조하세요 [원격 측정 설정](#).

위성 구성

위성 구성은 위성과의 통신 AWS Ground Station 하는 방법과 관련이 있습니다. 에서 수집한 정보를 참조합니다 [위성 온보딩](#).

- [추적 구성](#) - 고객 응대 중에 차량이 물리적으로 추적되는 방식에 대한 기본 설정을 지정합니다. 이는 미션 프로파일 구성에 필요합니다.
- [안테나 다운링크 구성](#) - 디지털화된 무선 주파수 데이터를 제공합니다.
- [안테나 다운링크 복조 디코드 구성](#) - 복조 및 디코드된 무선 주파수 데이터를 제공합니다.
- [안테나 업링크 구성](#) - 위성에 데이터를 연결합니다.
- [안테나 업링크 에코 구성](#) - 업링크 신호 데이터의 에코를 전달합니다.

미션 프로파일 생성

이전 단계에서 구성된 구성을 통해 위성을 추적하는 방법, 위성과의 통신하는 가능한 방법, 고객 응대 실행 중에 거의 실시간에 가까운 원격 측정을 활성화하는 방법을 식별했습니다. 이 단계에서는 하나 이상의 미션 프로파일을 구성합니다. 미션 프로파일은 가능한 구성을 예상 동작으로 집계한 다음 예약 및 운영할 수 있음을 나타냅니다.

최신 파라미터는 [AWS::GroundStation::MissionProfile CloudFormation 리소스 유형](#)을 참조하세요.

1. 미션 프로파일의 이름을 지정합니다. 이를 통해 시스템 내 사용량을 빠르게 파악할 수 있습니다. 예를 들어 긴급 satellite-wideband-narrowband-nominal-operations 연산과 satellite-narrowband-emergency-operations이 있을 수 있습니다.
2. 추적 구성을 설정합니다.
3. 실행 가능한 최소 연락 기간을 설정합니다. 이를 통해 미션 요구 사항에 맞게 잠재적 연락처를 필터링할 수 있습니다.
4. 전송 중 데이터를 암호화하는 데 사용되는 streamsKmsKey 및 streamsKmsRole을 설정합니다. 이는 모든 AWS Ground Station 에이전트 데이터 흐름에 사용됩니다.
5. 데이터 흐름을 설정합니다. 이전 단계에서 생성한 구성을 사용하여 통신 사업자 신호와 일치하도록 데이터 흐름을 생성합니다.
6. [선택 사항] 사전 통과 및 사후 통과 고객 응대 지속 시간 초를 설정합니다. 이는 접촉 전과 후에 각 접촉당 이벤트를 내보내는 데 사용됩니다. 자세한 정보는 [이벤트 AWS Ground Station 로 자동화](#)을 참조하세요.
7. [선택 사항] 고객 응대 중에 원격 측정을 활성화하도록 telemetrySinkConfigArn을 설정합니다. 이를 통해 모니터링 및 분석을 위해 계정에서 직접 실시간에 가까운 원격 측정을 수신할 수 있습니다. 자세한 정보는 [원격 측정 작업](#)을 참조하세요.
8. [선택 사항] 태그를 미션 프로파일에 연결할 수 있습니다. 이를 사용하여 미션 프로파일을 프로그래밍 방식으로 구분할 수 있습니다.

를 참조하여 몇 가지 잠재적 구성만 [미션 프로파일 구성 예](#)볼 수 있습니다.

다음 단계 이해

이제 온보딩된 위성과 유효한 미션 프로파일이 있으므로 연락처를 예약하고 위성과 통신할 준비가 되었습니다 AWS Ground Station.

다음 방법 중 하나로 고객 응대를 예약할 수 있습니다.

- [AWS Ground Station 콘솔](#)입니다.
- AWS CLI [reserve-contact](#) 명령입니다.
- AWS SDK. [ReserveContact](#) API.

가 위성의 궤적을 AWS Ground Station 추적하는 방법과 해당 정보가 사용되는 방법에 대한 자세한 내용은 섹션을 참조하세요 [가 에페메리스를 AWS Ground Station 사용하는 방법 이해](#).

AWS Ground Station 는 서비스를 더 쉽게 시작할 수 있도록 미리 구성된 여러 CloudFormation 템플릿을 유지 관리합니다. 를 사용하는 방법에 [미션 프로파일 구성 예](#) 대한 예는 단원 AWS Ground Station 을 참조하십시오.

디지털 중간 주파수 데이터 또는에서 제공된 복조 및 디코딩된 데이터를 처리하는 AWS Ground Station 것은 특정 사용 사례에 따라 달라집니다. 다음 블로그 게시물은 사용 가능한 몇 가지 옵션을 이해하는 데 도움이 될 수 있습니다.

- [AWS Ground Station Amazon S3 데이터 전송을 사용한 자동 지구 관측](#)(및 연결된 GitHub 리포지토리 [awslabs/aws-groundstation-eos-pipeline](#))
- [를 사용하여 위성 지상 세그먼트 가상화 AWS](#)
- [를 사용한 지구 관측치 AWS Ground Station: 안내 방법](#)
- [AWS Ground Station WideBand DigIF 및 Amphinicy Blink SDR\(및 연결된 GitHub 리포지토리 \[aws-samples/aws-groundstation-wbdigif-snpp\]\(#\)\)을 사용하여 고처리량 위성 데이터 다운로드 아키텍처 구축](#) GitHub [aws-samples/aws-groundstation-wbdigif-snpp](#)

AWS Ground Station 위치

AWS Ground Station 는 AWS 인프라 리전의 글로벌 네트워크와 매우 가까운 글로벌 지상국 네트워크를 제공합니다. 지원되는 AWS 리전에서 이러한 위치의 사용을 구성할 수 있습니다. 여기에는 데이터가 전송되는 AWS 리전이 포함됩니다.



지상국 위치의 AWS 리전 찾기

AWS Ground Station 글로벌 네트워크에는 연결된 [AWS 리전](#)에 물리적으로 위치하지 않은 지상국 위치가 포함됩니다. 액세스 권한이 있는 지상국 목록은 AWS SDK [ListGroundStation](#) 응답을 통해 검색할 수 있습니다. 2026년 6월 16일부터는 더 이상 서울에서 안테나 리소스를 지원하지 AWS Ground Station 않습니다. 지상국 위치의 전체 목록은 아래에 나와 있습니다. 위성에 대한 사이트 승인을 추가하거나 수정하려면 온보딩 가이드를 참조하세요.

Ground Station 이름	Ground Station 위치	AWS 리전 이름	AWS 리전 코드	참고
알래스카 1	알래스카, 미국	미국 서부(오리곤)	us-west-2	AWS 리전에 물리적으로 위치하지 않음
바레인 1	바레인	중동(바레인)	me-south-1	

Ground Station 이름	Ground Station 위치	AWS 리전 이름	AWS 리전 코드	참고
케이프타운 1	남아프리카 공화국 케이프타운	Africa (Cape Town)	af-south-1	
더보 1	호주 더보	아시아 태평양(시드니)	ap-southeast-2	AWS 리전에 물리적으로 위치하지 않음
하와이 1	하와이, 미국	미국 서부(오리곤)	us-west-2	AWS 리전에 물리적으로 위치하지 않음
아일랜드 1	아일랜드	유럽(아일랜드)	eu-west-1	
오하이오 1	미국 오하이오	미국 동부(오하이오)	us-east-2	
오레곤 1	미국 오레곤	미국 서부(오리곤)	us-west-2	
폰타 아레나 1	폰타 아레나, 칠레	남아메리카(상파울루)	sa-east-1	AWS 리전에 물리적으로 위치하지 않음
싱가포르 1	싱가포르	아시아 태평양(싱가포르)	ap-southeast-1	
스톡홀름 1	스톡홀름, 스웨덴	Europe (Stockholm)	eu-north-1	

AWS Ground Station 지원되는 AWS 리전

지원되는 AWS 리전에서 AWS SDK 또는 AWS Ground Station 콘솔을 통해 데이터를 전송하고 연락처를 구성할 수 있습니다. 엔드포인트 및 [AWS Ground Station 할당량에서 지원되는 리전 및 관련 엔드포인트를 볼 수 있습니다.](#)

디지털 트윈 가용성

[AWS Ground Station 디지털 트윈 기능 사용](#)를 사용할 수 있는 모든 [AWS 리전](#)에서 사용할 수 AWS Ground Station 있습니다. 디지털 트윈 지상국은 Ground Station 이름에 대한 수정 접두사가 있는 프로덕션 지상국의 정확한 사본입니다. 예를 들어 "Digital Twin Ohio 1"은 "Ohio 1" 프로덕션 지상국의 정확한 사본인 디지털 트윈 지상국입니다.

전용 안테나

는 위에 나열된 공개적으로 사용 가능한 지상국 위치 외에도 전용 안테나를 AWS Ground Station 제공합니다. 전용 안테나는 사용자를 대신하여 AWS 관리하는 사용자 지정 안테나 시스템입니다. 전용 안테나는 기존 AWS Ground Station 지상국 위치로 제한되지 않으며에 설명된 대로 퍼블릭 지상국 이외의 기능으로 빌드할 수 있습니다 [AWS Ground Station 사이트 기능](#). 전용 안테나의 위치 및 기능은 공개적으로 공개되지 않습니다.

전용 안테나에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station 전용 안테나](#). 자세한 내용을 알아보거나 전용 안테나를 시작하려면 AWS Support 통해 문의하세요 [AWS Support Center Console](#).

지상국에서 안테나 보기

각 지상국 위치에는 하나 이상의 안테나가 있습니다. [ListAntennas](#) API를 사용하여 지상국에서 안테나를 볼 수 있습니다. 이 API는 각 안테나의 이름을 포함하여 지정된 지상국에서 안테나를 반환합니다.

안테나 정보는 [ListGroundStationReservations](#) API와 결합하여 지상국의 용량과 가용성을 파악할 때 유용합니다. 예약 보기에 대한 자세한 내용은 섹션을 참조하세요 [지상국 예약 보기](#).

를 호출하려면 위성이 지상국에 온보딩되어 있거나 지상국에 대한 방위각 고도 에페메리스 권한이 [ListAntennas](#) 있어야 합니다. 자세한 내용은 [방위각 고도 에페메리스 데이터 제공](#) 섹션을 참조하세요.

예: 지상국의 안테나 나열

다음 예제에서는 AWS SDK for Python(Boto3)을 사용하여 지상국의 모든 안테나를 나열합니다.

```
import boto3

# Create AWS Ground Station client
```

```

ground_station_client = boto3.client("groundstation")

# The ground station ID to list antennas for.
# Use the ListGroundStations API to find available ground station IDs.
ground_station_id = "Ohio 1"

# List all antennas at a ground station.
# This is useful for understanding the capacity of a ground station
# and for planning multi-antenna operations.
print(f"Listing antennas for ground station '{ground_station_id}'...")

paginator = ground_station_client.get_paginator("list_antennas")
page_iterator = paginator.paginate(
    groundStationId=ground_station_id,
    PaginationConfig={
        "MaxItems": 100,
        "PageSize": 20,
    },
)

for page in page_iterator:
    for antenna in page["antennaList"]:
        print(f"    Antenna: {antenna['antennaName']}")
        print(f"    Ground Station: {antenna['groundStationName']}")
        print(f"    Region: {antenna['region']}")
        print()

```

지상국 예약 보기

[ListGroundStationReservations](#) API를 사용하여 지상국의 안테나 간 예약을 볼 수 있습니다. 예약은 예약된 고객 응대를 포함하여 안테나의 시간 블록을 나타냅니다. [AWS Ground Station 전용 안테나](#) 고객은 유지 관리 기간도 볼 수 있습니다.

이 정보는 고객 응대 일정을 계획할 때 안테나 가용성을 이해하는 데 도움이 되며 지상국의 안테나에서 발생하는 상황에 대한 가시성을 제공합니다.

예약 나열

예약을 나열하려면 지상국 식별자와 시간 범위를 사용하여 [ListGroundStationReservations](#)를 호출합니다. API는 지정된 기간 내에 지상국의 모든 안테나에 대한 예약을 반환합니다.

표시되는 예약은 액세스 수준에 따라 다릅니다.

- 공개 AWS Ground Station 고객 - 자체 고객 응대 예약만 볼 수 있습니다. 유지 관리 기간 및 다른 계정이 소유한 연락처는 포함되지 않습니다.
- AWS Ground Station 전용 안테나 고객 - 유지 관리 기간 및 다른 계정에서 예약한 연락처를 포함하여 전용 안테나에 대한 모든 예약을 볼 수 있습니다. 연락처 식별자는 소유한 연락처에 대해서만 포함됩니다. 자세한 내용은 [AWS Ground Station 전용 안테나](#) 단원을 참조하십시오.

예약 유형

각 예약에는 안테나 시간이 사용되는 대상을 나타내는 유형이 있습니다.

- 연락처 - 연락처 예약은 위성 통신용으로 예약된 안테나 시간을 나타냅니다. 예약 시작 및 종료 시간은 위성 패스 기간뿐만 아니라 사전 패스 및 사후 패스 시간을 포함한 전체 안테나 예약을 반영합니다.
- 유지 관리 - 유지 관리 예약은 유지 관리로 인해 안테나를 사용할 수 없는 기간을 나타냅니다. 유지 관리 예약에는 유지 관리의 계획 여부를 maintenanceType 나타내는가 포함됩니다.

코드 예제

다음 예시에서는 예약 유형별 필터링을 포함하여 Python용 AWS SDK(Boto3)를 사용하여 향후 7일 동안 지상국에서의 예약을 나열합니다.

```
import boto3
from datetime import datetime, timezone, timedelta

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# The ground station ID to list reservations for
ground_station_id = "Ohio 1"

# Define the time range to query. Reservations include both your
# scheduled contacts and maintenance windows at the ground station.
start_time = datetime.now(timezone.utc)
end_time = start_time + timedelta(days=7)

# List all reservations at a ground station for the next 7 days.
# You can filter by reservation type to see only contacts or
# only maintenance windows.
print(f"Listing reservations for ground station '{ground_station_id}'...")
```

```
print(f"Time range: {start_time} to {end_time}")

paginator = ground_station_client.get_paginator("list_ground_station_reservations")
page_iterator = paginator.paginate(
    groundStationId=ground_station_id,
    startTime=start_time,
    endTime=end_time,
    PaginationConfig={
        "MaxItems": 100,
        "PageSize": 20,
    },
)

for page in page_iterator:
    for reservation in page["reservationList"]:
        reservation_type = reservation["reservationType"]
        antenna_name = reservation["antennaName"]
        res_start = reservation["startTime"]
        res_end = reservation["endTime"]

        print(f"  Type: {reservation_type}")
        print(f"    Antenna: {antenna_name}")
        print(f"    Start: {res_start}")
        print(f"    End: {res_end}")

        details = reservation["reservationDetails"]
        if "contact" in details:
            contact_id = details["contact"].get("contactId", "N/A")
            print(f"    Contact ID: {contact_id}")
        elif "maintenance" in details:
            maintenance_type = details["maintenance"]["maintenanceType"]
            print(f"    Maintenance Type: {maintenance_type}")

    print()

# For Dedicated Antenna customers, you can also filter to show only maintenance windows
print("Listing only maintenance reservations...")

page_iterator = paginator.paginate(
    groundStationId=ground_station_id,
    startTime=start_time,
    endTime=end_time,
    reservationTypes=["MAINTENANCE"],
    PaginationConfig={
```

```

        "MaxItems": 100,
        "PageSize": 20,
    },
)

for page in page_iterator:
    for reservation in page["reservationList"]:
        maintenance_type = reservation["reservationDetails"]["maintenance"][
            "maintenanceType"
        ]
        print(
            f" {maintenance_type} maintenance on {reservation['antennaName']}: "
            f"{reservation['startTime']} to {reservation['endTime']}"
        )

```

AWS Ground Station 사이트 마스크

각 AWS Ground Station [antenna 위치](#)에 연결된 사이트 마스크가 있습니다. 이 마스크는 특정 방향 (일반적으로 수평선 근처)을 가리킬 때 해당 위치의 안테나가 전송하거나 수신하지 못하도록 차단합니다. 마스크에는 다음 사항이 고려될 수 있습니다.

- 안테나를 둘러싼 지리적 지형의 기능 - 예를 들어 여기에는 무선 주파수(RF) 신호를 차단하거나 전송을 방지하는 산이나 건물과 같은 것이 포함됩니다.
- 무선 주파수 간섭(RFI) - 이는 수신(AWS Ground Station 안테나로의 다운링크 신호에 영향을 미치는 외부 RFI 소스) 및 전송(AWS Ground Station 안테나가 전송하는 RF 신호가 외부 수신기에 부정적인 영향을 미치는 경우) 기능에 모두 영향을 미칩니다.
- 법적 권한 - 각 리전에서 AWS Ground Station을 운영하기 위한 로컬 사이트 권한 부여에는 전송을 위한 최소 고도 각도와 같은 특정 제한이 포함될 수 있습니다.

이러한 사이트 마스크는 시간이 지남에 따라 변경될 수 있습니다. 예를 들어 안테나 위치 근처에 새 건물을 짓거나, RFI 소스가 변경되거나, 다른 제한으로 법적 승인을 갱신할 수 있습니다. AWS Ground Station 사이트 마스크는 비밀 유지 계약(NDA)에 따라 사용할 수 있습니다.

고객별 마스크

각 사이트의 AWS Ground Station 사이트 마스크 외에도 특정 리전의 위성과 통신할 수 있는 법적 권한에 대한 제한으로 인해 추가 마스크가 있을 수 있습니다. AWS Ground Station을 사용하여 인공위성과 통신할 때 규정 준수를 보장하기 위해 AWS Ground Station에서 사례별로 이러한 마스크를 구성할 수 있습니다. 자세한 내용은 AWS Ground Station 팀에 문의하세요.

사이트 마스크가 사용 가능한 연락 시간에 미치는 영향

사이트 마스크에는 업링크(전송) 사이트 마스크와 다운링크(수신) 사이트 마스크의 두 종류가 있습니다.

ListContacts 작업을 사용하여 사용 가능한 연락 시간을 나열할 때 AWS Ground Station은 위성이 다운링크 마스크 위로 올라가서 아래로 설정된 시기에 따라 가시성 시간을 반환합니다. 사용 가능한 연락 시간은이 다운링크 마스크 가시성 기간을 기반으로 합니다. 이렇게 하면 위성이 다운링크 마스크 아래에 있을 때 시간을 예약하지 않아도 됩니다.

미션 프로파일에 데이터 흐름 엣지에 [안테나 업링크 구성](#)이 포함되어 있더라도 업링크 사이트 마스크는 사용 가능한 연락 시간에 적용되지 않습니다. 이렇게 하면 업링크 사이트 마스크로 인해 해당 시간 동안 업링크를 사용할 수 없는 경우에도 다운링크에 사용 가능한 모든 연락 시간을 사용할 수 있습니다. 그러나 위성 연락처용으로 예약된 일부 또는 전체 시간 동안에는 업링크 신호가 전송되지 않을 수 있습니다. 업링크 전송을 예약할 때 제공된 업링크 마스크를 고려해야 합니다.

접점에서 업링크를 사용할 수 없는 부분은 안테나 위치의 업링크 사이트 마스크를 기준으로 접촉 중의 위성 궤적에 따라 달라집니다. 업링크 사이트 마스크와 다운링크 사이트 마스크가 비슷한 리전에서는 일반적으로 이 지속 시간이 짧습니다. 업링크 마스크가 다운링크 사이트 마스크보다 상당히 높을 수 있는 다른 리전에서는 이로 인해 접속 기간의 상당 부분 또는 전체가 업링크에 사용할 수 없게 될 수 있습니다. 예약된 시간의 일부를 업링크에 사용할 수 없는 경우에도 전체 연락 시간이 청구됩니다.

AWS Ground Station 사이트 기능

경험을 단순화하기 위해 안테나 유형에 대한 공통 기능 세트를 AWS Ground Station 결정한 다음 여러 안테나를 지상국 위치에 배포합니다. 온보딩 단계의 일부는 위성이 특정 위치의 안테나 유형과 호환되도록 합니다. 연락처를 예약할 때 사용되는 안테나 유형을 간접적으로 결정합니다. 이렇게 하면 사용 중인 안테나에 관계없이 특정 지상국 위치에서의 경험을 시간이 지남에 따라 동일하게 유지할 수 있습니다. 고객 응대의 특정 성능은 사이트의 날씨와 같은 다양한 환경 문제로 인해 달라집니다.

현재 모든 사이트는 다음 기능을 지원합니다.

Note

다음 표의 각 행은 달리 명시되지 않는 한 독립적인 통신 경로를 나타냅니다. 여러 통신 경로를 동시에 사용할 수 있는 다중 채널 기능을 반영하기 위해 중복 행이 존재합니다.

기능 유형	주파수 범위	대역폭 범위	편광	일반 이름	Notes
안테나 다운 링크	7750~8500 MHz	50~400MHz	RHCP	X-대역 광대역 다운링크	이 기능을 사용하려면 AWS Ground Station 에이전트 를 사용해야 합니다. 알래스카 1 또는 폰타 아레나 1에서는 이 기능이 지원되지 않습니다.
안테나 다운 링크	7750~8500 MHz	50~400MHz	RHCP		
안테나 다운 링크	7750~8500 MHz	50~400MHz	RHCP		
안테나 다운 링크	7750~8500 MHz	50~400MHz	RHCP		
안테나 다운 링크	7750~8500 MHz	50~400MHz	RHCP		
안테나 다운 링크	7750~8500 MHz	50~400MHz	LHCP		집계 대역폭은 각 위치에서 편광당 400MHz를 초과해서는 안 됩니다. 사용된 모든 주파수 범위는 겹치지 않아야 합니다.
안테나 다운 링크	7750~8500 MHz	50~400MHz	LHCP		
안테나 다운 링크	7750~8500 MHz	50~400MHz	LHCP		
안테나 다운 링크	7750~8500 MHz	50~400MHz	LHCP		
안테나 다운 링크	7750~8500 MHz	50~400MHz	LHCP		
안테나 다운 링크	2200~2290 MHz	최대 40MHz	RHCP	S 밴드 다운링크	한 번에 하나의 편광만 사용할 수 있습니다.
안테나 다운 링크	2200~2290 MHz	최대 40MHz	LHCP		
안테나 다운 링크	7750~8500 MHz	최대 40MHz	RHCP	X-대역 협대역 다운링크	한 번에 하나의 편광만 사

기능 유형	주파수 범위	대역폭 범위	편광	일반 이름	Notes
안테나 다운 링크	7750~8500 MHz	최대 40MHz	LHCP		용할 수 있습니다.
안테나 링크	2025~2110 MHz	최대 40MHz	RHCP	S 대역 업링크	한 번에 하나의 편광만 사용할 수 있습니다.
안테나 링크	2025~2110 MHz	최대 40MHz	LHCP		EIRP 20~50dBW
antenna-uplink-echo	2025~2110 MHz	2MHz	RHCP	업링크 에코	안테나 링크 제한과 일치
antenna-uplink-echo	2025~2110 MHz	2MHz	LHCP		
antenna-downlink-demod-decode	7750~8500 MHz	최대 500MHz	RHCP	X 대역 복조 및 디코딩된 다운링크	
antenna-downlink-demod-decode	7750~8500 MHz	최대 500MHz	LHCP		
추적	N/A	N/A	N/A	N/A	자동 추적 및 프로그램 추적 지원

* RHCP = 오른손 원형 편광, LHCP = 왼손 원형 편광. 편광에 대한 자세한 내용은 [원형 편광을 참조하세요](#).

가 에페메리스를 AWS Ground Station 사용하는 방법 이해

복수 에페메리스인 [에페메리스](#)는 천체의 궤적을 제공하는 파일 또는 데이터 구조입니다. 과거에는 이 파일이 표 형식의 데이터만 참조했지만 점차 우주선 궤적을 나타내는 다양한 데이터 파일로 이동하게 되었습니다.

Ephemeris API를 사용하면 위성과 함께 사용하기 위해 사용자 지정 에페메리스를 AWS Ground Station에 업로드할 수 있습니다. 이러한 에페메리스는 [Space-Track](#)의 기본 에페메리스를 재정의합니다(참조: [기본 에페메리스 데이터](#)).는 Orbit Ephemeris Message(OEM), 2줄 요소(TLE) 및 방위각 고도 형식의 에페메리스 데이터를 AWS Ground Station 지원합니다. 최고의 정확도와 가장 유연한 시간 범위를 위해 OEM 형식으로 에페메리스 데이터를 제공합니다(참조 [OEM 에페메리스 데이터 제공](#)).

AWS Ground Station는 에페메리스 데이터를 사용하여 제공된 에페메리스를 기반으로 연락처를 사용할 수 있는 시기를 결정하고 AWS Ground Station 네트워크의 안테나를 올바르게 명령합니다. 위성에 할당된 [NORAD ID](#)가 AWS Ground Station 있는 경우 기본적으로 에페메리스를 제공하는 데 필요한 작업은 없습니다.

사용자 지정 에페메리스를 업로드하면 추적 품질을 개선하고, [스페이스 트랙](#) 에페메리스를 사용할 수 없는 초기 작업을 처리하고 AWS Ground Station, 조작을 고려할 수 있습니다.

또는 위성 궤도 정보를 제공하지 않고도 안테나 방향 지정을 직접 지정할 수 있는 방위각 고도 형식을 AWS Ground Station 지원합니다. 위성 궤적 정보가 부정확하거나 알 수 없기 때문에 정확한 안테나 포인팅이 필요한 시나리오에 유용합니다.

주제

- [기본 에페메리스 데이터](#)
- [사용자 지정 에페메리스 데이터 제공](#)
- [사용자 지정 에페메리스를 사용하여 연락처 예약](#)
- [사용되는 에페메리스 이해](#)
- [위성의 현재 에페메리스 가져오기](#)
- [기본 에페메리스 데이터로 되돌리기](#)

기본 에페메리스 데이터

기본적으로는 [Space-Track](#)에서 공개적으로 사용 가능한 데이터를 AWS Ground Station 사용하며 이러한 기본 에페메리스를 제공하는 AWS Ground Station 데 필요한 작업은 없습니다. 이러한 에페메리

스는 위성의 [NORAD ID](#)와 연결된 [2줄 요소 세트\(TLEs\)](#)입니다. 모든 기본 에페메리스의 우선 순위는 0입니다. 따라서 만료되지 않은 사용자 지정 에페메리스가 에페메리스 API를 통해 업로드되므로 항상 우선 순위가 1 이상이어야 합니다.

NORAD ID가 없는 위성은 사용자 지정 에페메리스 데이터에 업로드해야 합니다 AWS Ground Station. 예를 들어 [스페이스 트랙](#) 카탈로그에서 방금 시작했거나 의도적으로 생략된 위성은 NORAD ID가 없으며 사용자 지정 에페메리스를 업로드해야 합니다. 사용자 지정 에페메리스 데이터 제공에 대한 자세한 내용은 사용자 [지정 에페메리스 데이터 제공](#)을 참조하세요.

사용자 지정 에페메리스 데이터 제공

Important

에페메리스 API는 현재 프리뷰 상태입니다

에페메리스 API에 대한 액세스는 필요에 따라 제공됩니다. 사용자 지정 에페메리스 데이터를 업로드하는 기능이 필요한 경우를 통해 AWS Support 티켓을 개설하세요 [AWS Support Center Console](#). 당사 팀은 고객과 협력하여 특정 요구 사항에 맞게 기능을 활성화합니다.

개요

Ephemeris API를 사용하면 위성과 함께 사용하기 위해 사용자 지정 에페메리스를 AWS Ground Station에 업로드할 수 있습니다. 이러한 에페메리스는 [스페이스 트랙](#)의 기본 에페메리스를 재정의합니다(참조 [기본 에페메리스 데이터](#)). 오비트 에페메리스 메시지(OEM), 2줄 요소(TLE) 및 방위각 고도 형식으로 에페메리스 데이터 수신을 지원합니다.

AWS Ground Station는 에페메리스를 [개별화된 사용 데이터](#)로 처리합니다. 이 선택적 기능을 사용하는 경우 AWS는 에페메리스 데이터를 사용하여 문제 해결 지원을 제공합니다.

사용자 지정 에페메리스를 업로드하면 추적 품질을 개선하고, [스페이스 트랙](#) 에페메리스를 사용할 수 없는 작업을 처리하고 AWS Ground Station, 조작을 고려할 수 있습니다.

잘못된 에페메리스 문제를 해결하려면 다음을 참조하세요. [잘못된 에페메리스 문제 해결](#)

예:에서 고객 제공 에페메리스 사용 AWS Ground Station

에서 고객 제공 에페메리스를 사용하는 방법에 대한 자세한 지침은 [에서 고객 제공 에페메리스 사용 AWS Ground Station](#) 및 관련 GitHub 리포지토리 [aws-samples/aws-groundstation-cpe](#)를 AWS Ground Station참조하세요.

TLE 에페메리스 데이터 제공

Important

에페메리스 API는 현재 프리뷰 상태입니다

에페메리스 API에 대한 액세스는 필요에 따라 제공됩니다. 사용자 지정 에페메리스 데이터를 업로드하는 기능이 필요한 경우를 통해 AWS Support 티켓을 개설하세요 [AWS Support Center Console](#). 당사 팀은 고객과 협력하여 특정 요구 사항에 맞게 이 기능을 활성화합니다.

개요

2줄 요소(TLE) 세트는 위성 궤도를 설명하기 위한 표준화된 형식입니다. Ephemeris API를 사용하면 위성과 함께 사용하기 위해 TLE 에페메리스를 AWS Ground Station에 업로드할 수 있습니다. 이러한 에페메리스는 [스페이스 트랙](#)의 기본 에페메리스를 재정의합니다(참조 [기본 에페메리스 데이터](#)).

AWS Ground Station는 에페메리스를 [개별화된 사용 데이터](#)로 처리합니다. 이 선택적 기능을 사용하는 경우 AWS는 에페메리스 데이터를 사용하여 문제 해결 지원을 제공합니다.

사용자 지정 TLE 에페메리스를 업로드하면 추적 품질을 개선하고, [스페이스 트랙](#) 에페메리스를 사용할 수 없는 초기 작업을 처리하고 AWS Ground Station, 조작을 고려할 수 있습니다. 그러나 TLE 정확도는 전파 모델에 의해 제한되며 시간이 지남에 따라 epoch에서 떨어집니다. 더 긴 시간 동안 더 높은 충실도의 에페메리스 데이터를 얻으려면 OEM 형식을 사용하는 것이 좋습니다(참조 [OEM 에페메리스 데이터 제공](#)).

Note

위성 카탈로그 번호가 할당되기 전에 사용자 지정 에페메리스를 제공하는 경우 TLE의 00000 위성 카탈로그 번호 필드와 TLE의 국제 지정자 필드의 000 시작 번호 부분에 24000A를 사용할 수 있습니다(예: 2024년에 출시된 차량).

TLEs. https://en.wikipedia.org/wiki/Two-line_element_set

TLEs의 Alpha-5 위성 카탈로그 번호

2줄 요소 세트(TLEs)는 위성 카탈로그 번호 필드(1행과 2행의 3~7행)가 5자리로 제한되는 고정 너비 형식을 사용하므로 최대값은 99,999입니다. 이 한도를 초과하는 위성 카탈로그 번호를 나타내기 위해

미국 스페이스 포스는 [Alpha-5](#) 인코딩 체계를 개발했습니다. Alpha-5는 위성 카탈로그 번호 필드의 첫 번째 숫자를 알파벳 문자(A~Z, 숫자 1 및 0과 혼동되지 않도록 I 및 O 제외)로 바꿉니다. 이렇게 하면 표시 가능한 위성 카탈로그 번호의 범위가 기존 고정 너비 TLE 형식 내에서 100,000에서 339,999로 확장됩니다.

Alpha-5는 순전히 TLE 텍스트 형식의 디스플레이 인코딩입니다. 기본 숫자 위성 카탈로그 번호는 변경되지 않습니다. API의 `noradSatelliteID` AWS Ground Station 필드는 정수로 유지됩니다. `ListSatellites`와 같은 API 응답은 Alpha-5 인코딩 형식이 아닌 숫자 위성 카탈로그 번호를 `GetSatellite` 계속 반환합니다. Alpha-5 인코딩은 TLE 텍스트 줄 자체 내에서만 관련이 있습니다.

AWS Ground Station 는 Alpha-5 인코딩 위성 번호가 포함된 TLE 에페메리스를 지원합니다.

[CreateEphemeris](#) 작업을 통해 TLE를 업로드하면는 Alpha-5 인코딩 위성 번호를 AWS Ground Station 올바르게 구문 분석하고 이를 내부적으로 해당 정수 위성 카탈로그 번호로 확인합니다. Alpha-5 범위 (100,000~339,999)의 위성 카탈로그 번호가 있는 위성이 완벽하게 지원됩니다.

다음은 위성 카탈로그 번호가 100,000인 위성에 대한 예제 TLE 세트입니다. Alpha-5 인코딩은 각 TLE 선의 열 3~7A0000에서와 같이 이를 나타냅니다.

```
[
  {
    "tleLine1": "1 A0000U 26001A    26120.50000000    .00000072    00000-0    24500-4 0
9991",
    "tleLine2": "2 A0000    97.4500    85.2300 0012345    45.6789 314.5678 15.20000000
10013",
    "validTimeRange": {
      "startTime": 1777680000,
      "endTime": 1778284800
    }
  }
]
```

Alpha-5 인코딩 체계에 대한 자세한 내용은 [Space-Track Alpha-5 설명서](#)를 참조하세요.

TLE 에페메리스 생성

TLE 에페메리스는 API의 [CreateEphemeris](#) 작업을 사용하여 생성할 수 있습니다 AWS Ground Station . 이 작업은 요청 본문 또는 지정된 S3 버킷의 데이터를 사용하여 에페메리스를 업로드합니다.

한 가지 주의할 점은 에피메리스를 업로드하면 에피메리스가 VALIDATING로 설정되고 비동기 워크플로가 시작되어 에피메리스를 검증하고 이로부터 잠재적 접촉을 생성하는 비동기 워크플로가 시작된다

는 점입니다. 임시 저장소가 이 워크플로를 통과한 후 ENABLED가 된 후에만 접촉에 사용됩니다. 에페메리스 상태에 대해 [DescribeEphemeris](#)를 폴링하거나 CloudWatch 이벤트를 사용하여 에페메리스의 상태 변경을 추적해야 합니다.

잘못된 에페메리스 문제를 해결하려면 다음을 참조하세요. [잘못된 에페메리스 문제 해결](#)

예: API를 통해 2줄 요소(TLE) 세트 에페메리스 생성

AWS SDKs 및 CLI를 사용하여 [CreateEphemeris](#) 호출을 AWS Ground Station 통해 에페메리스를 로 설정한 두 줄 요소(TLE)를 업로드할 수 있습니다. 이 에페메리스는 위성의 기본 에페메리스 데이터 대신 사용됩니다(참조 [기본 에페메리스 데이터](#)). 이 예제에서는 [AWS SDK for Python\(Boto3\)](#)을 사용하여 이 작업을 수행하는 방법을 보여줍니다.

TLE 세트는 하나 이상의 TLE를 함께 묶어 연속 궤적을 구성하는 JSON 형식의 객체입니다. TLE 세트의 TLE는 궤적을 구성하는 데 사용할 수 있는 연속 세트를 형성해야 합니다(즉, TLE 세트의 TLE 간 시간 간격이 없음). TLE 세트의 예는 다음과 같습니다.

```
[
  {
    "tleLine1": "1 25994U 99068A 20318.54719794 .000000075 00000-0 26688-4 0 9997",
    "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934 14.57114995111906",
    "validTimeRange": {
      "startTime": 12345,
      "endTime": 12346
    }
  },
  {
    "tleLine1": "1 25994U 99068A 20318.54719794 .000000075 00000-0 26688-4 0 9997",
    "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934 14.57114995111906",
    "validTimeRange": {
      "startTime": 12346,
      "endTime": 12347
    }
  }
]
```

Note

TLE 세트의 TLE 시간 범위가 정확히 일치해야 유효하고 연속적인 궤적을 유지할 수 있습니다.

TLE 세트는 다음과 같이 AWS Ground Station boto3 클라이언트를 통해 업로드할 수 있습니다.

```
import boto3
from datetime import datetime, timedelta, timezone

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# Create TLE ephemeris
tle_ephemeris = ground_station_client.create_ephemeris(
    name="Example Ephemeris",
    satelliteId="2e925701-9485-4644-b031-EXAMPLE01",
    enabled=True,
    expirationTime=datetime.now(timezone.utc) + timedelta(days=3),
    priority=2,
    ephemeris={
        "tle": {
            "tleData": [
                {
                    "tleLine1": "1 25994U 99068A 20318.54719794 .000000075 00000-0
26688-4 0 9997",
                    "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934
14.57114995111906",
                    "validTimeRange": {
                        "startTime": datetime.now(timezone.utc),
                        "endTime": datetime.now(timezone.utc) + timedelta(days=7),
                    },
                }
            ]
        }
    },
)

print(f"Created TLE ephemeris with ID: {tle_ephemeris['ephemerisId']}")
```

이 호출은 향후 ephemerisId를 반환합니다. 예를 들어 위의 호출에서 제공된 ephemerisId를 사용하여 ephemeris 상태를 폴링할 수 있습니다.

```

import boto3
from datetime import datetime, timedelta, timezone
import time

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# First, create a TLE ephemeris
print("Creating TLE ephemeris...")

tle_ephemeris = ground_station_client.create_ephemeris(
    name="Example TLE Ephemeris for Description",
    satelliteId="2e925701-9485-4644-b031-EXAMPLE01",
    enabled=True,
    expirationTime=datetime.now(timezone.utc) + timedelta(days=3),
    priority=2,
    ephemeris={
        "tle": {
            "tleData": [
                {
                    "tleLine1": "1 25994U 99068A   20318.54719794   .000000075   000000-0
26688-4 0   9997",
                    "tleLine2": "2 25994   98.2007   30.6589 0001234   89.2782   18.9934
14.57114995111906",
                    "validTimeRange": {
                        "startTime": datetime.now(timezone.utc),
                        "endTime": datetime.now(timezone.utc) + timedelta(days=7),
                    },
                }
            ]
        }
    },
)

ephemeris_id = tle_ephemeris["ephemerisId"]
print(f"Created TLE ephemeris with ID: {ephemeris_id}")

# Describe the ephemeris immediately to check initial status
print("Describing ephemeris...")

response = ground_station_client.describe_ephemeris(ephemerisId=ephemeris_id)

print(f"Ephemeris ID: {response['ephemerisId']}")

```

```
print(f"Name: {response['name']}")
print(f"Status: {response['status']}")
```

[DescribeEphemeris](#) 작업의 응답 예제는 다음과 같습니다.

```
{
  "creationTime": 1620254718.765,
  "enabled": true,
  "name": "Example Ephemeris",
  "ephemerisId": "fde41049-14f7-413e-bd7b-EXAMPLE01",
  "priority": 2,
  "status": "VALIDATING",
  "suppliedData": {
    "tle": {
      "ephemerisData": "[{\\"tleLine1\\": \\"1 25994U 99068A 20318.54719794 .00000075
00000-0 26688-4 0 9997\\",\\"tleLine2\\": \\"2 25994 98.2007 30.6589 0001234 89.2782
18.9934 14.57114995111906\\",\\"validTimeRange\\": {\\"startTime\\": 1620254712000,
\\"endTime\\": 1620859512000}}]"
    }
  }
}
```

[DescribeEphemeris](#) 라우팅을 폴링하거나 CloudWatch 이벤트를 사용하여 업로드된 에페메리스의 상태를 추적하는 것이 좋습니다. 업로드된 에페메리스는 로 설정되고 고객 응대 예약 ENABLED 및 실행에 사용할 수 있게 되기 전에 비동기 검증 워크플로를 거쳐야 하기 때문입니다.

위의 예제 25994에서 TLEs 세트의 모든 TLE에 있는 NORAD ID는 위성이 [Space-Track](#) 데이터베이스에 할당된 NORAD ID와 일치해야 합니다.

예: S3 버킷에서 TLE 에페메리스 데이터 업로드

버킷 및 객체 키를 가리켜 S3 버킷에서 직접 TLE 에페메리스 파일을 업로드할 수도 있습니다. AWS Ground Station 는 사용자를 대신하여 객체를 검색합니다. 의 저장 데이터 암호화에 대한 자세한 내용은 단원을 AWS Ground Station 참조하십시오 [에 대한 저장 데이터 암호화 AWS Ground Station](#).

다음은 S3 버킷에서 TLE 에페메리스 파일을 업로드하는 예입니다.

```
import boto3
from datetime import datetime, timedelta, timezone
import json

# Create AWS clients
```

```

s3_client = boto3.client("s3")
ground_station_client = boto3.client("groundstation")

# Define S3 bucket and key
bucket_name = "ephemeris-bucket"
object_key = "test_data.tle"

# Create sample TLE set data
# Note: For actual satellites, use real TLE data from sources like Space-Track
tle_set_data = [
    {
        "tleLine1": "1 25994U 99068A   20318.54719794   .000000075   00000-0   26688-4 0
9997",
        "tleLine2": "2 25994   98.2007   30.6589 0001234   89.2782   18.9934
14.57114995111906",
        "validTimeRange": {
            "startTime": datetime.now(timezone.utc),
            "endTime": datetime.now(timezone.utc) + timedelta(days=3),
        },
    },
    {
        "tleLine1": "1 25994U 99068A   20321.54719794   .000000075   00000-0   26688-4 0
9998",
        "tleLine2": "2 25994   98.2007   33.6589 0001234   89.2782   18.9934
14.57114995112342",
        "validTimeRange": {
            "startTime": datetime.now(timezone.utc) + timedelta(days=3),
            "endTime": datetime.now(timezone.utc) + timedelta(days=7),
        },
    },
]

# Convert to JSON string for upload
tle_json = json.dumps(tle_set_data, indent=2)

# Upload sample TLE data to S3
print(f"Uploading TLE set data to s3://{bucket_name}/{object_key}")

s3_client.put_object(
    Bucket=bucket_name, Key=object_key, Body=tle_json, ContentType="application/json"
)
print("TLE set data uploaded successfully to S3")
print(f"Uploaded {len(tle_set_data)} TLE entries covering 7 days")

```

```
# Create TLE ephemeris from S3
print("Creating TLE ephemeris from S3...")

s3_tle_ephemeris = ground_station_client.create_ephemeris(
    name="2022-11-05 S3 TLE Upload",
    satelliteId="fde41049-14f7-413e-bd7b-EXAMPLE01",
    enabled=True,
    expirationTime=datetime.now(timezone.utc) + timedelta(days=5),
    priority=2,
    ephemeris={"tle": {"s3object": {"bucket": bucket_name, "key": object_key}}},
)

print(f"Created TLE ephemeris with ID: {s3_tle_ephemeris['ephemerisId']}")
```

OEM 에페메리스 데이터 제공

Important

에페메리스 API는 현재 프리뷰 상태입니다

에페메리스 API에 대한 액세스는 필요에 따라 제공됩니다. 사용자 지정 에페메리스 데이터를 업로드하는 기능이 필요한 경우를 통해 AWS Support 티켓을 개설하세요 [AWS Support Center Console](#). 당사 팀은 고객과 협력하여 특정 요구 사항에 맞게 이 기능을 활성화합니다.

개요

Orbit Ephemeris Message(OEM)는 우주선 궤적 데이터를 나타내는 표준화된 형식입니다. Ephemeris API를 사용하면 OEM 에페메리스를 AWS Ground Station에 업로드하여 위성과 함께 사용할 수 있습니다. 이러한 에페메리스는 [스페이스 트랙](#)의 기본 에페메리스를 재정의합니다(참조 [기본 에페메리스 데이터](#)).

AWS Ground Station는 에페메리스를 [개별화된 사용 데이터](#)로 처리합니다. 이 선택적 기능을 사용하는 경우 AWS는 에페메리스 데이터를 사용하여 문제 해결 지원을 제공합니다.

사용자 지정 OEM 에페메리스를 업로드하면 추적 품질을 개선하고, [스페이스 트랙](#) 에페메리스를 사용할 수 없는 초기 작업을 처리하고 AWS Ground Station, 조작을 고려할 수 있습니다.

Note

위성 카탈로그 번호가 위성에 할당되기 전에 사용자 지정 에페메리스를 제공하는 경우 OEM의 OBJECT_ID 부분에 satelliteId를 사용할 수 있습니다.

OEMs의 형식에 대한 자세한 내용은 단원을 참조하십시오 [OEM 에페메리스 형식](#).

OEM 에페메리스 형식

AWS Ground Station 는 몇 가지 추가 제한과 함께 [CCSDS 표준](#)에 따라 OEM 고객 제공 에페메리스를 처리합니다. OEM 파일은 KVN 형식이어야 합니다. 다음 표에는 OEM의 다양한 필드와 CCSDS 표준과 어떻게 AWS Ground Station 다른지 요약되어 있습니다.

섹션	Field	CCSDS 필요	AWS Ground Station 필수	참고
헤더	CCSDS_OEM_VERS	예	예	필수 값: 2.0
	COMMENT	아니요	아니요	
	분류	아니요	아니요	
	CREATION_DATE	예	예	
	생성자	예	예	
	MESSAGE_ID	아니요	아니요	
Metadata	META_START	예	예	
	COMMENT	아니요	아니요	
	OBJECT_NAME	예	예	
	OBJECT_ID	예	예	
	CENTER_NAME	예	예	필수 값: 지구

섹션	Field	CCSDS 필요	AWS Ground Station 필수	참고
	REF_FRAME	예	예	허용되는 값: EME2000, ITRF2000
	REF_FRAME_EPOCH	아니요	지원되지 않음*	허용되는 REF_FRAMEs에 는 암시적 에포크 가 있으므로 필요 하지 않음
	TIME_SYSTEM	예	예	필수 값: UTC
	START_TIME	예	예	
	USEABLE_START_TIME	아니요	아니요	
	USEABLE_STOP_TIME	아니요	아니요	
	STOP_TIME	예	예	
	보간	아니요	예	AWS Ground Station 가 접촉에 대한 정확한 가리 킴 각도를 생성하 는 데 필요합니 다.

섹션	Field	CCSDS 필요	AWS Ground Station 필수	참고
	인터폴레이션_도	아니요	예	AWS Ground Station 가 접촉에 대한 정확한 가리킴 각도를 생성하는 데 필요합니다. 가능하면 지정된 정도가 사용되지만 세그먼트에 데이터가 충분하지 않은 경우 더 낮은 정도가 사용됩니다.
	META_STOP	예	예	
데이터	X	예	예	에서 표시됨 km
	Y	예	예	에서 표시됨 km
	Z	예	예	에서 표시됨 km
	X_DOT	예	예	에서 표시됨 km/s
	Y_DOT	예	예	에서 표시됨 km/s
	Z_DOT	예	예	에서 표시됨 km/s
	X_DDOT	아니요	아니요	에서 표시됨 km/s ²
	Y_DDOT	아니요	아니요	에서 표시됨 km/s ²

섹션	Field	CCSDS 필요	AWS Ground Station 필수	참고
	Z_DDOT	아니요	아니요	에서 표시됨 km/s^2
공분산 행렬	COVARIANCE_START	아니요	아니요	
	EPOCH	아니요	아니요	
	COV_REF_FRAME	아니요	아니요	
	COVARIANCE_STOP	아니요	아니요	

*에서 지원하지 않는 행 AWS Ground Station 이 제공된 OEM에 포함된 경우 OEM은 검증에 실패합니다.

에 대한 CCSDS 표준과의 중요한 편차는 다음과 AWS Ground Station 같습니다.

- CCSDS_OEM_VERS는 이어야 합니다2.0.
- REF_FRAME는 EME2000 또는 중 하나여야 합니다 ITRF2000.
- REF_FRAME_EPOCH는에서 지원되지 않습니다 AWS Ground Station.
- CENTER_NAME는 이어야 합니다Earth.
- TIME_SYSTEM는 이어야 합니다UTC.
- INTERPOLATION 및 INTERPOLATION_DEGREE는 모두 AWS Ground Station 고객이 제공하는 에페메리스에 필요합니다.
- AWS Ground Station 는 지정된에서 보간을 수행하기에 충분한 에페메리스 데이터 레코드가 포함되지 않은 OEM 데이터 블록을 허용하여 CCSDS 5.2.4.7을 벗어납니다INTERPOLATION_DEGREE. 이 경우는 지정된 보다 작거나 같은 가장 높은 보간 정도를 AWS Ground Station 사용합니다INTERPOLATION_DEGREE.

KVN 형식의 OEM 에페메리스 예제

다음은 JPSS-1 퍼블릭 브로드캐스터 위성에 대한 KVN 형식의 OEM 에페메리스의 잘린 예입니다.

```
CCSDS_OEM_VERS = 2.0
```

```
COMMENT Orbit data are consistent with planetary ephemeris DE-430
```

```
CREATION_DATE = 2024-07-22T05:20:59
```

```
ORIGINATOR = Raytheon-JPSS/CGS
```

```
META_START
```

```
OBJECT_NAME = J1
```

```
OBJECT_ID = 2017-073A
```

```
CENTER_NAME = Earth
```

```
REF_FRAME = EME2000
```

```
TIME_SYSTEM = UTC
```

```
START_TIME = 2024-07-22T00:00:00.000000
```

```
STOP_TIME = 2024-07-22T00:06:00.000000
```

```
INTERPOLATION = Lagrange
```

```
INTERPOLATION_DEGREE = 5
```

```
META_STOP
```

```
2024-07-22T00:00:00.000000 5.905147360000000e+02 -1.860082793999999e+03
-6.944807075000000e+03 -5.784245796000000e+00 4.347501391999999e+00
-1.657256863000000e+00
```

```
2024-07-22T00:01:00.000000 2.425572045154201e+02 -1.595860765983339e+03
-7.030938457373539e+03 -5.810660250794190e+00 4.457103652219009e+00
-1.212889340333023e+00
```

```
2024-07-22T00:02:00.000000 -1.063224256538050e+02 -1.325569732497146e+03
-7.090262617183503e+03 -5.814973972202444e+00 4.549739160042560e+00
-7.639633689161465e-01
```

```
2024-07-22T00:03:00.000000 -4.547973959231161e+02 -1.050238305712201e+03
-7.122556683227951e+03 -5.797176562437553e+00 4.625064829516728e+00
-3.121687831090774e-01
```

```
2024-07-22T00:04:00.000000 -8.015427368657785e+02 -7.709137891269565e+02
-7.127699477194810e+03 -5.757338007808417e+00 4.682800822515077e+00
1.407953645161997e-01
```

```
2024-07-22T00:05:00.000000 -1.145240083085062e+03 -4.886583601179489e+02
-7.105671911254255e+03 -5.695608435738609e+00 4.722731329786999e+00
5.932259682105052e-01
```

```
2024-07-22T00:06:00.000000 -1.484582479061495e+03 -2.045451985605701e+02
-7.056557069672793e+03 -5.612218005854990e+00 4.744705579872771e+00
1.043421397392599e+00
```

OEM 에페메리스 생성

OEM 에페메리스는 API의 [CreateEphemeris](#) 작업을 사용하여 생성할 수 있습니다 AWS Ground Station . 이 작업은 요청 본문 또는 지정된 S3 버킷의 데이터를 사용하여 에페메리스를 업로드합니다.

한 가지 주의할 점은 에피메리스를 업로드하면 에피메리스가 VALIDATING로 설정되고 비동기 워크플로가 시작되어 에피메리스를 검증하고 이로부터 잠재적 접촉을 생성하는 비동기 워크플로가 시작된다는 점입니다. 임시 저장소가 이 워크플로를 통과한 후 ENABLED가 된 후에만 접촉에 사용됩니다. 에페메리스 상태에 대해 [DescribeEphemeris](#)를 폴링하거나 CloudWatch 이벤트를 사용하여 에페메리스의 상태 변경을 추적해야 합니다.

잘못된 에페메리스 문제를 해결하려면 다음을 참조하세요. [잘못된 에페메리스 문제 해결](#)

예: S3 버킷에서 OEM 에페메리스 데이터 업로드

버킷 및 객체 키를 가리켜 S3 버킷에서 OEM 에페메리스 파일을 직접 업로드할 수도 있습니다. AWS Ground Station 는 사용자를 대신하여 객체를 검색합니다. 의 저장 데이터 암호화에 대한 자세한 내용은 단원을 AWS Ground Station 참조하십시오 [에 대한 저장 데이터 암호화 AWS Ground Station](#).

다음은 S3 버킷에서 OEM 에페메리스 파일을 업로드하는 예제입니다

```
import boto3
from datetime import datetime, timedelta, timezone

# Create AWS clients
s3_client = boto3.client("s3")
ground_station_client = boto3.client("groundstation")

# Define S3 bucket and key
bucket_name = "ephemeris-bucket"
object_key = "test_data.oem"

# Create sample OEM data in KVN format
oem_data = """CCSDS_OEM_VERS = 2.0

COMMENT Orbit data are consistent with planetary ephemeris DE-430

CREATION_DATE   = 2024-07-22T05:20:59
ORIGINATOR      = Raytheon-JPSS/CGS
```

```

META_START
OBJECT_NAME      = J1
OBJECT_ID        = 2017-073A
CENTER_NAME      = Earth
REF_FRAME        = EME2000
TIME_SYSTEM      = UTC
START_TIME       = 2024-07-22T00:00:00.000000
STOP_TIME        = 2024-07-22T00:06:00.000000
INTERPOLATION    = Lagrange
INTERPOLATION_DEGREE = 5
META_STOP

2024-07-22T00:00:00.000000    5.905147360000000e+02    -1.860082793999999e+03
    -6.944807075000000e+03    -5.784245796000000e+00    4.347501391999999e+00
    -1.657256863000000e+00
2024-07-22T00:01:00.000000    2.425572045154201e+02    -1.595860765983339e+03
    -7.030938457373539e+03    -5.810660250794190e+00    4.457103652219009e+00
    -1.212889340333023e+00
2024-07-22T00:02:00.000000    -1.063224256538050e+02    -1.325569732497146e+03
    -7.090262617183503e+03    -5.814973972202444e+00    4.549739160042560e+00
    -7.639633689161465e-01
2024-07-22T00:03:00.000000    -4.547973959231161e+02    -1.050238305712201e+03
    -7.122556683227951e+03    -5.797176562437553e+00    4.625064829516728e+00
    -3.121687831090774e-01
2024-07-22T00:04:00.000000    -8.015427368657785e+02    -7.709137891269565e+02
    -7.127699477194810e+03    -5.757338007808417e+00    4.682800822515077e+00
    1.407953645161997e-01
2024-07-22T00:05:00.000000    -1.145240083085062e+03    -4.886583601179489e+02
    -7.105671911254255e+03    -5.695608435738609e+00    4.722731329786999e+00
    5.932259682105052e-01
2024-07-22T00:06:00.000000    -1.484582479061495e+03    -2.045451985605701e+02
    -7.056557069672793e+03    -5.612218005854990e+00    4.744705579872771e+00
    1.043421397392599e+00
""""

# Upload sample OEM data to S3
print(f"Uploading OEM data to s3://{bucket_name}/{object_key}")

s3_client.put_object(
    Bucket=bucket_name, Key=object_key, Body=oem_data, ContentType="text/plain"
)

print("OEM data uploaded successfully to S3")

```

```
# Create OEM ephemeris from S3
print("Creating OEM ephemeris from S3...")

s3_oem_ephemeris = ground_station_client.create_ephemeris(
    name="2024-07-22 S3 OEM Upload",
    satelliteId="fde41049-14f7-413e-bd7b-EXAMPLE01",
    enabled=True,
    expirationTime=datetime.now(timezone.utc) + timedelta(days=5),
    priority=2,
    ephemeris={"oem": {"s3Object": {"bucket": bucket_name, "key": object_key}}},
)

print(f"Created OEM ephemeris with ID: {s3_oem_ephemeris['ephemerisId']}")
```

다음은 이전 예제 코드 블록에 업로드된 OEM 에페메리스에 대해 호출되는 [DescribeEphemeris](#) 작업에서 반환된 데이터의 예입니다.

```
{
  "creationTime": 1620254718.765,
  "enabled": true,
  "name": "Example Ephemeris",
  "ephemerisId": "fde41049-14f7-413e-bd7b-EXAMPLE02",
  "priority": 2,
  "status": "VALIDATING",
  "suppliedData": {
    "oem": {
      "sourceS3Object": {
        "bucket": "ephemeris-bucket-for-testing",
        "key": "test_data.oem"
      }
    }
  }
}
```

방위각 고도 에페메리스 데이터 제공

Important

방위각 고도 에페메리스 기능은 현재 미리 보기 상태이며 명시적 온보딩이 필요합니다. 방위각 고도 에페메리스 기능은 제한된 수의 미리 결정된 특수 사용 사례에 대해 엄격한 액세스 제어가 적용됩니다. 액세스는 고객이 제공하는 표준 에페메리스 기능보다 훨씬 더 제한적입니다.

니다. 승인된 사용 사례 및 액세스 요청 프로세스에 대한 자세한 내용은 통해 AWS Support 티켓을 개설하세요 [AWS Support Center Console](#). 전문 사용 사례에 대한 승인 프로세스를 안내해 드립니다.

개요

방위각 고도 에페메리스는 위성 궤도 정보를 제공하지 않고 안테나 방향 지정을 직접 지정하는 방법을 제공합니다. 위성의 궤도를 설명하는 에페메리스 데이터를 업로드하는 대신 시간 태그가 지정된 방위각과 고도 각도를 제공하여 안테나에 접촉 전체에서 정확히 어디를 가리킬지 알려줍니다.

AWS Ground Station 는 에페메리스를 [개별화된 사용 데이터](#)로 처리합니다. 이 선택적 기능을 사용하는 경우 AWS는 에페메리스 데이터를 사용하여 문제 해결 지원을 제공합니다.

이 접근 방식은 다음 시나리오에 특히 유용합니다.

- 초기 작업 지원: 정확한 궤도 데이터를 사용할 수 없거나 궤도 파라미터가 빠르게 변화하는 시작 및 초기 궤도 단계(LEOP) 중.
- 사용자 지정 포인팅 패턴: 안테나 테스트 또는 비표준 작업을 위한 특정 포인팅 시퀀스 구현.

Note

방위각 고도 에페메리스를 사용하는 경우 연락처 예약 요청에서 위성 ARN을 생략할 수 있습니다. 위성 ARN이 생략되지 않은 경우에도 여전히 연락처 데이터의 일부로 포함되지만 방위각 고도 에페메리스는 에페메리스 우선 순위 확인을 수행하는 대신 안테나를 가리키는 데 사용됩니다. 방위각 고도 에페메리스는 특정 지상국과 연결되어 있으며 해당 위치에 대한 안테나 방향 지정을 정의합니다.

방위각 고도 에페메리스 데이터 형식

방위각 고도 에페메리스 데이터는 세그먼트로 구성된 시간 태그 방위각 및 고도 값으로 구성됩니다. 각 세그먼트에는 특정 시간 범위를 포함하는 일련의 방위각과 고도 각도가 포함되어 있습니다.

방위각 고도 에페메리스 데이터의 주요 구성 요소는 다음과 같습니다.

- Ground Station: 이 방위 고도 에페메리스가 사용될 특정 지상국입니다.
- 각도 단위: 각도(또는)의 측정 단위입니다 DEGREE_ANGLERADIAN.

- 세그먼트: 방위각 및 고도 각도로 구성된 하나 이상의 시간 경계 모음입니다.
- 시간 태그가 지정된 각도: 연결된 타임스탬프가 있는 개별 방위각 및 고도 값입니다.

각 세그먼트에는 다음이 필요합니다.

- 참조 에포크(세그먼트의 기본 시간)
- 유효한 시간 범위(세그먼트의 시작 및 종료 시간)
- 시간 태그가 지정된 방위각/상승 페어 5개 이상

방위각 고도 제약 조건:

- 각도: -180° ~ 360° 의 방위각
- 라디안 단위의 방위각: $-\pi \sim 2\pi$
- 각도 상승: -90° ~ 90°
- 라디안 단위의 상승: $-\pi/2 \sim \pi/2$
- 시간 값은 각 세그먼트 내에서 오름차순이어야 합니다.
- 세그먼트는 시간이 겹치지 않아야 합니다.

자세한 내용은 [CreateEphemeris](#) API 설명서 및 [TimeAzEl](#) 데이터 유형을 참조하세요.

방위각 고도 에페메리스 생성

방위각 고도 에페메리스는 동일한 [CreateEphemeris](#) API 작업을 사용하지만 azEl 에페메리스 유형을 사용하여 생성됩니다. TLE와 OEM 에페메리스의 주요 차이점은 다음과 같습니다.

- groundStation 파라미터를 지정해야 합니다.
- 요청에서 satelliteId 파라미터를 생략해야 합니다.
- 우선 순위 설정은 적용되지 않습니다(각 방위각 고도 에페메리스는 지상국에만 해당됨).
- 4차 Lagrange 보간을 지원하려면 각 세그먼트에 방위각/상승 지점이 5개 이상 포함되어야 합니다.
- 추가 제한 및 요구 사항은 [CreateEphemeris](#) API 설명서에 자세히 설명되어 있습니다.

한 가지 주의할 점은 에피메리스를 업로드하면 에피메리스가 VALIDATING로 설정되고 비동기 워크플로가 시작되어 에피메리스를 검증하고 이로부터 잠재적 접촉을 생성하는 비동기 워크플로가 시작된다는 점입니다. 에페메리스는 이 워크플로를 통과하고 상태가 된 후에만 고객 응대에 사용됩니

다ENABLED. 에페메리스 상태에 대해 [DescribeEphemeris](#)를 폴링하거나 CloudWatch 이벤트를 사용하여 에페메리스의 상태 변경을 추적해야 합니다.

잘못된 에페메리스 문제를 해결하려면 다음을 참조하세요. [잘못된 에페메리스 문제 해결](#)

예: API를 통해 방위각 고도 에페메리스 생성

다음 예제에서는 AWS SDK for Python(Boto3)을 사용하여 방위각 고도 에페메리스를 생성하는 방법을 보여줍니다.

```
import boto3

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# Create azimuth elevation ephemeris
azimuth_elevation_ephemeris = ground_station_client.create_ephemeris(
    name="Azimuth Elevation for Ohio Ground Station",
    ephemeris={
        "azEl": {
            "groundStation": "Ohio 1",
            "data": {
                "azElData": {
                    "angleUnit": "DEGREE_ANGLE",
                    "azElSegmentList": [
                        {
                            "referenceEpoch": "2024-03-15T10:00:00Z",
                            "validTimeRange": {
                                "startTime": "2024-03-15T10:00:00Z",
                                "endTime": "2024-03-15T10:15:00Z",
                            },
                        },
                    ],
                    "azElList": [
                        {"dt": 0.0, "az": 45.0, "el": 10.0},
                        {"dt": 180.0, "az": 50.0, "el": 15.0},
                        {"dt": 360.0, "az": 55.0, "el": 20.0},
                        {"dt": 540.0, "az": 60.0, "el": 25.0},
                        {"dt": 720.0, "az": 65.0, "el": 30.0},
                        {"dt": 900.0, "az": 70.0, "el": 35.0},
                    ],
                },
            },
        },
    },
)
```

```

    }
  },
)

print(f"Created ephemeris with ID: {azimuth_elevation_ephemeris['ephemerisId']}")

```

이 예시는 다음과 같이 설정되어 있습니다.

- 방위각 고도 데이터는 "Ohio 1" 지상국과 연결됩니다.
- 각도는 도 단위로 지정됩니다.
- 세그먼트는 15분 동안 진행됩니다.
- dt 값은 참조 에포크의 원자 초 오프셋입니다.
- 방위각/상승 쌍 6개가 제공됩니다(최소 5개).

예: S3에서 방위각 고도 데이터 업로드

더 큰 데이터 세트의 경우 S3 버킷에서 방위각 고도 데이터를 업로드할 수 있습니다.

```

import boto3
import json

# Create AWS clients
s3_client = boto3.client("s3")
ground_station_client = boto3.client("groundstation")

# Define S3 bucket and key
bucket_name = "azimuth-elevation-bucket"
object_key = "singapore-azimuth-elevation.json"

# Create sample azimuth elevation data
azimuth_elevation_data = {
    "angleUnit": "DEGREE_ANGLE",
    "azElSegmentList": [
        {
            "referenceEpoch": "2024-03-15T10:00:00Z",
            "validTimeRange": {
                "startTime": "2024-03-15T10:00:00Z",
                "endTime": "2024-03-15T10:15:00Z",
            },
            "azElList": [
                {"dt": 0.0, "az": 45.0, "el": 10.0},
            ],
        },
    ],
}

```

```

        {"dt": 180.0, "az": 50.0, "el": 15.0},
        {"dt": 360.0, "az": 55.0, "el": 20.0},
        {"dt": 540.0, "az": 60.0, "el": 25.0},
        {"dt": 720.0, "az": 65.0, "el": 30.0},
        {"dt": 900.0, "az": 70.0, "el": 35.0},
    ],
},
{
    "referenceEpoch": "2024-03-15T10:15:00Z",
    "validTimeRange": {
        "startTime": "2024-03-15T10:15:00Z",
        "endTime": "2024-03-15T10:30:00Z",
    },
    "azElList": [
        {"dt": 0.0, "az": 70.0, "el": 35.0},
        {"dt": 180.0, "az": 75.0, "el": 40.0},
        {"dt": 360.0, "az": 80.0, "el": 45.0},
        {"dt": 540.0, "az": 85.0, "el": 50.0},
        {"dt": 720.0, "az": 90.0, "el": 55.0},
        {"dt": 900.0, "az": 95.0, "el": 50.0},
    ],
},
],
}

# Upload sample data to S3
print(f"Uploading azimuth elevation data to s3://{bucket_name}/{object_key}")

s3_client.put_object(
    Bucket=bucket_name,
    Key=object_key,
    Body=json.dumps(azimuth_elevation_data, indent=2),
    ContentType="application/json",
)
print("Sample data uploaded successfully to S3")

# Create azimuth elevation ephemeris from S3
print("Creating azimuth elevation ephemeris from S3...")

s3_azimuth_elevation_ephemeris = ground_station_client.create_ephemeris(
    name="Large Azimuth Elevation Dataset",
    ephemeris={
        "azEl": {
            "groundStation": "Singapore 1",

```

```

        "data": {"s3Object": {"bucket": bucket_name, "key": object_key}},
    },
    },
)

print(f"Created ephemeris with ID: {s3_azimuth_elevation_ephemeris['ephemerisId']}")

```

S3 객체에는 직접 업로드 예제와 동일한 형식의 방위각 고도 데이터가 있는 JSON 구조가 포함되어야 합니다.

방위각 고도 에페메리스를 사용하여 연락처 예약

방위각 고도 에페메리스를 사용하여 고객 응대를 예약하는 경우 프로세스는 TLE 및 OEM 에페메리스와 다릅니다.

1. [CreateEphemeris](#)를 사용하여 방위각 고도 에페메리스 생성
2. 에페메리스가 ENABLED 상태에 도달할 때까지 기다립니다.
3. 추적 재정의와 함께 [ReserveContact](#)를 사용하여 고객 응대 예약

방위각 고도 에페메리스가 있는 고객 응대를 예약하는 예:

```

import boto3
from datetime import datetime
import time

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# First, create an azimuth elevation ephemeris
print("Creating azimuth elevation ephemeris...")

create_ephemeris_response = ground_station_client.create_ephemeris(
    name="Azimuth Elevation for Contact Reservation",
    ephemeris={
        "azEl": {
            "groundStation": "Ohio 1",
            "data": {
                "azElData": {
                    "angleUnit": "DEGREE_ANGLE",
                    "azElSegmentList": [
                        {

```

```
        "referenceEpoch": "2024-03-15T10:00:00Z",
        "validTimeRange": {
            "startTime": "2024-03-15T10:00:00Z",
            "endTime": "2024-03-15T10:15:00Z",
        },
        "azElList": [
            {"dt": 0.0, "az": 45.0, "el": 10.0},
            {"dt": 180.0, "az": 50.0, "el": 15.0},
            {"dt": 360.0, "az": 55.0, "el": 20.0},
            {"dt": 540.0, "az": 60.0, "el": 25.0},
            {"dt": 720.0, "az": 65.0, "el": 30.0},
            {"dt": 900.0, "az": 70.0, "el": 35.0},
        ],
    },
],
},
),
)

ephemeris_id = create_ephemeris_response["ephemerisId"]
print(f"Created ephemeris with ID: {ephemeris_id}")

# Wait for ephemeris to become ENABLED
print("Waiting for ephemeris to become ENABLED...")

while True:
    status = ground_station_client.describe_ephemeris(ephemerisId=ephemeris_id)[
        "status"
    ]
    if status == "ENABLED":
        print("Ephemeris is ENABLED")
        break
    elif status in ["INVALID", "ERROR"]:
        raise RuntimeError(f"Ephemeris failed: {status}")
    time.sleep(5)

# Reserve contact with azimuth elevation ephemeris
print("Reserving contact...")

contact = ground_station_client.reserve_contact(
    # Note: satelliteArn is omitted when using azimuth elevation ephemeris
```

```
missionProfileArn="arn:aws:groundstation:us-east-2:111122223333:mission-profile/
example-mission-profile",
groundStation="Ohio 1",
startTime=datetime(2024, 3, 15, 10, 0, 0),
endTime=datetime(2024, 3, 15, 10, 15, 0),
trackingOverrides={"programTrackSettings": {"azEl": {"ephemerisId":
ephemeris_id}}},
)

print(f"Reserved contact with ID: {contact['contactId']}")
```

Note

방위각 고도 에페메리스와 접촉을 예약할 때 `satelliteArn` 파라미터를 생략할 수 있습니다. 안테나는 접촉 중에 지정된 방위각 및 고도 각도를 따릅니다.

사용 가능한 연락처 나열

방위각 고도 에페메리스를 사용하는 경우 [ListContacts](#) API에는 특정 파라미터가 필요합니다.

- 요청에서 `satelliteArn` 파라미터를 생략할 수 있습니다.
- 사용할 에페메리스를 지정하려면 방위각 고도 에페메리스 ID와 함께 `ephemeris` 파라미터를 제공해야 합니다.
- 제공된 방위각 및 고도 각도가 요청된 지상국의 [사이트 마스크](#)보다 높을 때 사용 가능한 접촉 창이 표시됩니다.
- 여전히 `groundStation` 및를 제공해야 합니다. `missionProfileArn`

방위각 고도 에페메리스를 생성하고 여기에 사용 가능한 연락처를 나열하는 예:

```
import boto3
from datetime import datetime, timezone
import time

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# Step 1: Create azimuth elevation ephemeris
print("Creating azimuth elevation ephemeris...")
ephemeris_response = ground_station_client.create_ephemeris(
```

```

name="Stockholm AzEl Ephemeris",
ephemeris={
    "azEl": {
        "groundStation": "Stockholm 1",
        "data": {
            "azElData": {
                "angleUnit": "DEGREE_ANGLE",
                "azElSegmentList": [
                    {
                        "referenceEpoch": "2024-04-01T12:00:00Z",
                        "validTimeRange": {
                            "startTime": "2024-04-01T12:00:00Z",
                            "endTime": "2024-04-01T12:30:00Z",
                        },
                        "azElList": [
                            {"dt": 0.0, "az": 30.0, "el": 15.0},
                            {"dt": 360.0, "az": 45.0, "el": 30.0},
                            {"dt": 720.0, "az": 60.0, "el": 45.0},
                            {"dt": 1080.0, "az": 75.0, "el": 35.0},
                            {"dt": 1440.0, "az": 90.0, "el": 20.0},
                            {"dt": 1800.0, "az": 105.0, "el": 10.0},
                        ],
                    },
                ],
            },
        },
    },
},
),

ephemeris_id = ephemeris_response["ephemerisId"]
print(f"Created ephemeris: {ephemeris_id}")

# Step 2: Wait for ephemeris to become ENABLED
print("Waiting for ephemeris to become ENABLED...")
while True:
    describe_response = ground_station_client.describe_ephemeris(
        ephemerisId=ephemeris_id
    )
    status = describe_response["status"]

    if status == "ENABLED":
        print("Ephemeris is ENABLED")
        break

```

```

elif status in ["INVALID", "ERROR"]:
    # Check for validation errors
    if "invalidReason" in describe_response:
        print(f"Ephemeris validation failed: {describe_response['invalidReason']}")
        raise RuntimeError(f"Ephemeris failed with status: {status}")

print(f"Current status: {status}, waiting...")
time.sleep(5)

# Step 3: List available contacts using the azimuth elevation ephemeris
print("Listing available contacts with azimuth elevation ephemeris...")

# Convert epoch timestamps to datetime objects
start_time = datetime.fromtimestamp(1760710513, tz=timezone.utc)
end_time = datetime.fromtimestamp(1760883313, tz=timezone.utc)

contacts_response = ground_station_client.list_contacts(
    startTime=start_time,
    endTime=end_time,
    groundStation="Stockholm 1",
    statusList=["AVAILABLE"],
    ephemeris={"azEl": {"id": ephemeris_id}},
    # satelliteArn is optional
    satelliteArn="arn:aws:groundstation::111122223333:satellite/a88611b0-f755-404e-
b60d-57d8aEXAMPLE",
    missionProfileArn="arn:aws:groundstation:eu-north-1:111122223333:mission-
profile/966b72f6-6d82-4e7e-b072-f8240EXAMPLE",
)

# Process the results
if contacts_response["contactList"]:
    print(f"Found {len(contacts_response['contactList'])} available contacts:")
    for contact in contacts_response["contactList"]:
        print(f" - Contact from {contact['startTime']} to {contact['endTime']}")
        print(
            f"      Max elevation: {contact.get('maximumElevation', {}).get('value', 'N/
A')}°"
        )
else:
    print("No available contacts found for the specified azimuth elevation ephemeris")

```

Note

접촉 기간을 결정하는 데 사용할 방위각 고도 에페메리스를 지정하기 위해 접촉을 나열할 때 방위각 고도 ID가 있는 `ephemeris` 파라미터를 제공해야 합니다. `satelliteArn`이 포함된 경우 고객 응대 데이터와 연결되지만 방위각 고도 에페메리스는 에페메리스 우선 순위 확인을 수행하는 대신 안테나를 가리키는 데 사용됩니다.

사용자 지정 에페메리스를 사용하여 연락처 예약

개요

사용자 지정 에페메리스(TLE, OEM 또는 방위각 고도)를 사용하는 경우 [ReserveContact](#) API를 사용하여 연락처를 예약할 수 있습니다. 이 섹션에서는 고객 응대를 예약하기 위한 두 가지 일반적인 워크플로와 성공적인 고객 응대 예약을 보장하기 위한 중요한 고려 사항에 대해 설명합니다.

AWS Ground Station 안테나는 여러 고객 간에 공유되는 리소스입니다. 즉, 연락처를 나열할 때 연락처 창을 사용할 수 있는 경우에도 다른 고객이 예약하기 전에 예약할 수 있습니다. 따라서 예약 후 고객 응대가 SCHEDULED 상태에 도달하는지 확인하고 고객 응대 상태 변경에 대한 적절한 모니터링을 구현하는 것이 중요합니다.

Important

방위각 고도 에페메리스의 경우 `ReserveContact` 요청에서 `satelliteArn` 파라미터를 생략할 수 있으며 에페메리스 `IDtrackingOverrides`를 제공해야 합니다. TLE 및 OEM 에페메리스의 경우 여전히를 제공해야 합니다 `satelliteArn`.

고객 응대 예약 워크플로

사용자 지정 에페메리스로 연락처를 예약하기 위한 두 가지 기본 워크플로가 있습니다.

1. List-then-reserve 워크플로: 먼저 [ListContacts](#)를 사용하여 사용 가능한 연락처 창을 나열한 다음 특정 창을 선택하고 예약합니다. 이 접근 방식은 선택하기 전에 사용 가능한 모든 기회를 보고 싶을 때 유용합니다.
2. 직접 예약 워크플로: 사용 가능한 연락처를 먼저 나열하지 않고 특정 기간 동안 연락처를 직접 예약합니다. 이 접근 방식은 원하는 연락 시간을 이미 알고 있거나 미리 정해진 일정으로 작업하는 경우에 유용합니다.

두 워크플로 모두 유효하며 선택 사항은 운영 요구 사항에 따라 달라집니다. 다음 섹션에서는 각 접근 방식의 예를 제공합니다.

워크플로 1: 사용 가능한 연락처를 나열한 다음 예약

이 워크플로는 먼저 사용 가능한 연락 기간을 쿼리한 다음 특정 기간을 예약합니다. 이는 선택하기 전에 사용 가능한 모든 기회를 보고 싶을 때 유용합니다.

예: 방위각 고도 에페메리스를 사용하여 나열 및 예약

```
import boto3
from datetime import datetime, timezone
import time

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# Create azimuth elevation ephemeris
print("Creating azimuth elevation ephemeris...")
ephemeris_response = ground_station_client.create_ephemeris(
    name="AzEl Ephemeris for Contact",
    ephemeris={
        "azEl": {
            "groundStation": "Ohio 1",
            "data": {
                "azElData": {
                    "angleUnit": "DEGREE_ANGLE",
                    "azElSegmentList": [
                        {
                            "referenceEpoch": "2024-03-15T10:00:00Z",
                            "validTimeRange": {
                                "startTime": "2024-03-15T10:00:00Z",
                                "endTime": "2024-03-15T10:15:00Z",
                            },
                        },
                    ],
                    "azElList": [
                        {"dt": 0.0, "az": 45.0, "el": 10.0},
                        {"dt": 180.0, "az": 50.0, "el": 15.0},
                        {"dt": 360.0, "az": 55.0, "el": 20.0},
                        {"dt": 540.0, "az": 60.0, "el": 25.0},
                        {"dt": 720.0, "az": 65.0, "el": 30.0},
                        {"dt": 900.0, "az": 70.0, "el": 35.0},
                    ],
                },
            },
        },
    },
)
```

```

        ],
    },
},
),

ephemeris_id = ephemeris_response["ephemerisId"]
print(f"Created ephemeris: {ephemeris_id}")

# Wait for ephemeris to become ENABLED
while True:
    status = ground_station_client.describe_ephemeris(ephemerisId=ephemeris_id)[
        "status"
    ]
    if status == "ENABLED":
        print("Ephemeris is ENABLED")
        break
    elif status in ["INVALID", "ERROR"]:
        raise RuntimeError(f"Ephemeris failed: {status}")
    time.sleep(5)

# List available contacts
print("Listing available contacts...")
contacts = ground_station_client.list_contacts(
    # Note: satelliteArn is omitted for azimuth elevation ephemeris
    groundStation="Ohio 1",
    missionProfileArn="arn:aws:groundstation:us-east-2:111122223333:mission-profile/
example-profile",
    startTime=datetime(2024, 3, 15, 10, 0, 0, tzinfo=timezone.utc),
    endTime=datetime(2024, 3, 15, 10, 15, 0, tzinfo=timezone.utc),
    statusList=["AVAILABLE"],
    ephemeris={"azEl": {"id": ephemeris_id}},
)

if contacts["contactList"]:
    # Reserve the first available contact
    contact = contacts["contactList"][0]
    print(f"Reserving contact from {contact['startTime']} to {contact['endTime']}...")

    reservation = ground_station_client.reserve_contact(
        # Note: satelliteArn is omitted when using azimuth elevation ephemeris
        missionProfileArn="arn:aws:groundstation:us-east-2:111122223333:mission-
profile/example-profile",

```

```

        groundStation="Ohio 1",
        startTime=contact["startTime"],
        endTime=contact["endTime"],
        trackingOverrides={
            "programTrackSettings": {"azEl": {"ephemerisId": ephemeris_id}}
        },
    )

    print(f"Reserved contact: {reservation['contactId']}")
else:
    print("No available contacts found")

```

예: TLE 에페메리스를 사용하여 나열 및 예약

```

import boto3
from datetime import datetime, timedelta, timezone
import time

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

satellite_id = "12345678-1234-1234-1234-123456789012"
satellite_arn = f"arn:aws:groundstation::111122223333:satellite/{satellite_id}"

# Create TLE ephemeris
print("Creating TLE ephemeris...")
ephemeris_response = ground_station_client.create_ephemeris(
    name="TLE Ephemeris for Contact",
    satelliteId=satellite_id,
    enabled=True,
    expirationTime=datetime.now(timezone.utc) + timedelta(days=7),
    priority=1, # Higher priority than default ephemeris
    ephemeris={
        "tle": {
            "tleData": [
                {
                    "tleLine1": "1 25994U 99068A   24075.54719794   .000000075   00000-0
26688-4 0   9997",
                    "tleLine2": "2 25994   98.2007   30.6589 0001234   89.2782   18.9934
14.57114995111906",
                    "validTimeRange": {
                        "startTime": datetime.now(timezone.utc),
                        "endTime": datetime.now(timezone.utc) + timedelta(days=7),

```

```

        },
    }
]
},
)

ephemeris_id = ephemeris_response["ephemerisId"]
print(f"Created ephemeris: {ephemeris_id}")

# Wait for ephemeris to become ENABLED
while True:
    status = ground_station_client.describe_ephemeris(ephemerisId=ephemeris_id)[
        "status"
    ]
    if status == "ENABLED":
        print("Ephemeris is ENABLED")
        break
    elif status in ["INVALID", "ERROR"]:
        raise RuntimeError(f"Ephemeris failed: {status}")
    time.sleep(5)

# List available contacts
print("Listing available contacts...")
start_time = datetime.now(timezone.utc) + timedelta(hours=1)
end_time = start_time + timedelta(days=1)

contacts = ground_station_client.list_contacts(
    satelliteArn=satellite_arn, # Required for TLE/OEM ephemeris
    groundStation="Hawaii 1",
    missionProfileArn="arn:aws:groundstation:us-west-2:111122223333:mission-profile/
example-profile",
    startTime=start_time,
    endTime=end_time,
    statusList=["AVAILABLE"],
)

if contacts["contactList"]:
    # Reserve the first available contact
    contact = contacts["contactList"][0]
    print(f"Reserving contact from {contact['startTime']} to {contact['endTime']}...")

    reservation = ground_station_client.reserve_contact(
        satelliteArn=satellite_arn, # Required for TLE/OEM ephemeris

```

```

        missionProfileArn="arn:aws:groundstation:us-west-2:111122223333:mission-
profile/example-profile",
        groundStation="Hawaii 1",
        startTime=contact["startTime"],
        endTime=contact["endTime"],
        # Note: trackingOverrides is optional for TLE/OEM
        # The system will use the highest priority ephemeris automatically
    )

    print(f"Reserved contact: {reservation['contactId']}")
else:
    print("No available contacts found")

```

워크플로 2: 직접 연락처 예약

이 워크플로는 사용 가능한 기간을 먼저 나열하지 않고 연락처를 직접 예약합니다. 이 접근 방식은 원하는 연락 시간을 이미 알고 있거나 자동 일정을 구현할 때 유용합니다.

예: 방위각 고도 에페메리스를 사용한 직접 예약

```

import boto3
from datetime import datetime, timezone
import time

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

# Define contact window
contact_start = datetime(2024, 3, 20, 14, 0, 0, tzinfo=timezone.utc)
contact_end = datetime(2024, 3, 20, 14, 15, 0, tzinfo=timezone.utc)

# Create azimuth elevation ephemeris for the specific contact time
print("Creating azimuth elevation ephemeris...")
ephemeris_response = ground_station_client.create_ephemeris(
    name="Direct Contact AzEl Ephemeris",
    ephemeris={
        "azEl": {
            "groundStation": "Ohio 1",
            "data": {
                "azElData": {
                    "angleUnit": "DEGREE_ANGLE",
                    "azElSegmentList": [

```

```

        "referenceEpoch": contact_start.isoformat(),
        "validTimeRange": {
            "startTime": contact_start.isoformat(),
            "endTime": contact_end.isoformat(),
        },
        "azElList": [
            {"dt": 0.0, "az": 45.0, "el": 10.0},
            {"dt": 180.0, "az": 50.0, "el": 15.0},
            {"dt": 360.0, "az": 55.0, "el": 20.0},
            {"dt": 540.0, "az": 60.0, "el": 25.0},
            {"dt": 720.0, "az": 65.0, "el": 30.0},
            {"dt": 900.0, "az": 70.0, "el": 35.0},
        ],
    },
],
},
},
},
},
),

ephemeris_id = ephemeris_response["ephemerisId"]
print(f"Created ephemeris: {ephemeris_id}")

# Wait for ephemeris to become ENABLED
while True:
    status = ground_station_client.describe_ephemeris(ephemerisId=ephemeris_id)[
        "status"
    ]
    if status == "ENABLED":
        print("Ephemeris is ENABLED")
        break
    elif status in ["INVALID", "ERROR"]:
        raise RuntimeError(f"Ephemeris failed: {status}")
    time.sleep(5)

# Directly reserve the contact
print(f"Reserving contact from {contact_start} to {contact_end}...")

reservation = ground_station_client.reserve_contact(
    # Note: satelliteArn is omitted for azimuth elevation
    missionProfileArn="arn:aws:groundstation:us-east-2:111122223333:mission-profile/
example-profile",
    groundStation="Ohio 1",

```

```

        startTime=contact_start,
        endTime=contact_end,
        trackingOverrides={"programTrackSettings": {"azEl": {"ephemerisId":
ephemeris_id}}},
    )

print(f"Reserved contact: {reservation['contactId']}")

```

예: TLE 에페메리스를 사용한 직접 예약

```

import boto3
from datetime import datetime, timedelta, timezone
import time

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

satellite_id = "12345678-1234-1234-1234-123456789012"
satellite_arn = f"arn:aws:groundstation::111122223333:satellite/{satellite_id}"

# Define contact window (based on predicted pass)
contact_start = datetime(2024, 3, 21, 10, 30, 0, tzinfo=timezone.utc)
contact_end = datetime(2024, 3, 21, 10, 42, 0, tzinfo=timezone.utc)

# Create TLE ephemeris
print("Creating TLE ephemeris...")
ephemeris_response = ground_station_client.create_ephemeris(
    name="Direct Contact TLE Ephemeris",
    satelliteId=satellite_id,
    enabled=True,
    expirationTime=contact_end + timedelta(days=1),
    priority=1,
    ephemeris={
        "tle": {
            "tleData": [
                {
                    "tleLine1": "1 25994U 99068A   24080.50000000   .00000075   00000-0
26688-4 0   9999",
                    "tleLine2": "2 25994   98.2007   35.6589 0001234   89.2782   18.9934
14.57114995112000",
                    "validTimeRange": {
                        "startTime": (contact_start - timedelta(hours=1)).isoformat(),
                        "endTime": (contact_end + timedelta(hours=1)).isoformat(),

```

```

        },
    },
]
},
),

ephemeris_id = ephemeris_response["ephemerisId"]
print(f"Created ephemeris: {ephemeris_id}")

# Wait for ephemeris to become ENABLED
while True:
    status = ground_station_client.describe_ephemeris(ephemerisId=ephemeris_id)[
        "status"
    ]
    if status == "ENABLED":
        print("Ephemeris is ENABLED")
        break
    elif status in ["INVALID", "ERROR"]:
        raise RuntimeError(f"Ephemeris failed: {status}")
    time.sleep(5)

# Directly reserve the contact
print(f"Reserving contact from {contact_start} to {contact_end}...")

reservation = ground_station_client.reserve_contact(
    satelliteArn=satellite_arn, # Required for TLE ephemeris
    missionProfileArn="arn:aws:groundstation:us-west-2:111122223333:mission-profile/
example-profile",
    groundStation="Hawaii 1",
    startTime=contact_start,
    endTime=contact_end,
    # Note: trackingOverrides is optional for TLE
    # The system will use the highest priority ephemeris automatically
)

print(f"Reserved contact: {reservation['contactId']}")

```

고객 응대 상태 변경 모니터링

고객 응대를 예약한 후에는 상태를 모니터링하여 로 성공적으로 전환 SCHEDULED 되고 모든 문제에 대한 알림을 받는 것이 중요합니다. 모든 고객 응대 상태 변경에 대해 Amazon EventBridge에 이벤트를 AWS Ground Station 내보냅니다.

고객 응대 상태는 이 수명 주기를 따릅니다.

- SCHEDULING - 고객 응대가 예약을 위해 처리되고 있습니다.
- SCHEDULED - 고객 응대가 성공적으로 예약되었으며 실행됩니다.
- FAILED_TO_SCHEDULE - 고객 응대를 예약할 수 없음(터미널 상태)

고객 응대 상태 및 수명 주기에 대한 자세한 내용은 섹션을 참조하세요 [고객 응대 수명 주기 이해](#).

EventBridge를 사용한 고객 응대 상태 모니터링 구현

고객 응대 상태 변경을 실시간으로 모니터링하려면 Ground Station 고객 응대 상태가 변경될 때마다 Lambda 함수를 트리거하는 Amazon EventBridge 규칙을 설정할 수 있습니다. 이 접근 방식은 고객 응대 상태를 폴링하는 것보다 더 효율적이고 확장 가능합니다.

구현 단계

1. 고객 응대 상태 변경 이벤트를 처리하는 Lambda 함수 생성
2. Ground Station 고객 응대 상태 변경 이벤트와 일치하는 EventBridge 규칙 생성
3. Lambda 함수를 규칙의 대상으로 추가

Lambda 함수 핸들러 예제

고객 응대 상태 변경 이벤트를 처리하는 Lambda 함수의 전체 예는 AquaSnppJpssTerraDigIF.yml CloudFormation 템플릿의 GroundStationCloudWatchEventHandlerLambda 리소스를 참조하세요. 이 템플릿은 AWS Ground Station 고객 온보딩 Amazon S3 버킷에서 사용할 수 있습니다. 이 템플릿에 액세스하는 방법에 대한 지침은 데이터 흐름 엔드포인트 예제의 [함께 넣기](#) 섹션을 참조하세요.

EventBridge 규칙 구성

EventBridge 규칙은 모든 Ground Station 고객 응대 상태 변경과 일치하도록 다음 이벤트 패턴을 사용해야 합니다.

```
{
  "source": ["aws.groundstation"],
  "detail-type": ["Ground Station Contact State Change"]
}
```

특정 상태(예: 실패)만 필터링하려면 세부 정보 필터를 추가할 수 있습니다.

```
{
  "source": ["aws.groundstation"],
  "detail-type": ["Ground Station Contact State Change"],
  "detail": {
    "contactStatus": [
      "FAILED_TO_SCHEDULE",
      "FAILED",
      "AWS_FAILED",
      "AWS_CANCELLED"
    ]
  }
}
```

Lambda 대상으로 EventBridge 규칙을 생성하는 방법에 대한 자세한 지침은 Amazon EventBridge 사용 설명서의 [이벤트에 대응하는 규칙 생성을 참조하세요](#).

자동화를 위한 EventBridge 규칙 설정

EventBridge 규칙을 생성하여 고객 응대 상태 변경에 자동으로 응답할 수 있습니다. 예제:

- 고객 응대 예약 실패 시 알림 전송
- 고객 응대가 들어오면 Lambda 함수를 트리거하여 리소스를 준비합니다. PREPASS
- 감사 목적으로 고객 응대 완료 기록

AWS Ground Station 이벤트에 대한 EventBridge 규칙 설정에 대한 자세한 내용은 섹션을 참조하세요 [이벤트 AWS Ground Station 로 자동화](#).

모범 사례 및 고려 사항

일정 충돌 처리

AWS Ground Station 안테나는 공유 리소스이므로에서 사용할 ListContacts 수 있는 고객 응대 기간은 다른 고객이 예약한 후 예약할 수 있습니다. 이를 처리하려면:

1. 예약 후 항상 고객 응대 상태 확인
2. 대체 기간을 사용하여 재시도 로직 구현
3. 가능하면 미리 고객 응대를 예약하는 것이 좋습니다.
4. EventBridge 이벤트를 사용하여 FAILED_TO_SCHEDULE 상태 모니터링

에페메리스 검증 타이밍

에페메리스를 사용하여 연락처를 예약하려면 에페메리스가 ENABLED 상태여야 합니다. 검증 프로세스는 일반적으로 에페메리스 유형 및 크기에 따라 몇 초에서 몇 분 정도 걸립니다. 고객 응대를 예약하기 전에 항상 에페메리스 상태를 확인합니다.

고객 응대 타이밍 고려 사항

사용자 지정 에페메리스를 사용하는 경우:

- 에페메리스가 전체 연락 기간을 포함하는지 확인
- 방위각 고도 에페메리스의 경우 접촉 전체에서 각도가 안테나를 [사이트 마스크](#) 위에 유지하는지 확인합니다.
- 향후 연락을 예약할 때 에페메리스 만료 시간 고려

에페메리스 유형별 API 차이

ReserveContact API는 에페메리스 유형에 따라 다르게 동작합니다.

에페메리스 유형	satelliteArn 필수	trackingOverrides 필요
TLE	예	아니요(선택 사항)
OEM	예	아니요(선택 사항)
방위각 상승	아니요(선택 사항)	예

사용되는 에페메리스 이해

에페메리스에는 우선 순위, 만료 시간 및 활성화 플래그가 있습니다. 이를 통해 고객 응대 중에 추적에 사용되는 에페메리스가 결정됩니다.

TLE 및 OEM 에페메리스

OEM 및 TLE 에페메리스의 경우 각 위성에 대해 하나의 에페메리스만 활성화할 수 있습니다. 사용할 에페메리스는 우선 순위가 가장 높은 활성화된 에페메리스로, 만료 시간은 미래입니다. 우선 순위 값이 클수록 우선 순위가 높음을 나타냅니다. [ListContacts](#)에서 반환하는 사용 가능한 연락 시간은 이 에

에페메리스를 기반으로 합니다. 여러 개의 ENABLED 에페메리스의 우선 순위가 동일한 경우 가장 최근에 생성되거나 업데이트된 에페메리스가 사용됩니다.

Note

AWS Ground Station에는 위성당 ENABLED 고객 제공 에페메리스 수에 대한 서비스 할당량이 있습니다(: [Service Quotas](#) 참조). 이 할당량에 도달한 후 에페메리스 데이터를 업로드하려면 가장 낮은 우선순위/가장 빠른 생성 에페메리스를 삭제([DeleteEphemeris](#) 사용)하거나 비활성화([UpdateEphemeris](#) 사용)합니다.

에페메리스가 생성되지 않았거나 ENABLED 에페메리스 상태가 없는 경우 AWS Ground Station는 가능한 경우 위성([스페이스 트랙](#))에 기본 에페메리스를 사용합니다. 이 기본 에페메리스의 우선순위는 0입니다.

방위각 고도 에페메리스

방위각 고도 에페메리스는 OEM 및 TLE 에페메리스와 다르게 작동합니다. 각 방위각 고도 에페메리스는 특정 지상국과 연결되며 우선 순위가 없습니다. 방위각 고도 에페메리스로 연락처를 예약할 때 `trackingOverrides` 파라미터를 통해 사용할 방위각 고도 에페메리스를 명시적으로 지정합니다.

방위각 고도 에페메리스의 주요 차이점:

- 우선 순위 시스템 없음 - 각 연락처에 대해 에페메리스를 명시적으로 선택합니다.
- Ground Station별 - 각 에페메리스는 특정 Ground Station과 연결됩니다.
- 자동 폴백 없음 - 지정된 에페메리스를 사용할 수 없는 경우 고객 응대가 실패합니다.

Note

방위각 고도 에페메리스는 OEM 및 TLE 에페메리스와 경쟁하지 않습니다. 고객 응대를 예약할 때 명시적으로 선택되며 추적 재정의가 지정된 경우에만 사용됩니다.

새 에페메리스가 이전에 예약된 연락처에 미치는 영향

[DescribeContact API](#)를 사용하여 활성 가시성 시간을 반환하여 새 에페메리스가 이전에 예약된 연락처에 미치는 영향을 확인합니다.

OEM 및 TLE 에페메리스의 경우 새 에페메리스를 업로드하기 전에 예약된 연락처는 원래 예약된 연락 시간을 유지하는 반면 안테나 추적은 활성 에페메리스를 사용합니다. 활성 에페메리스를 기반으로 하는 우주선의 위치가 이전 에페메리스와 크게 다른 경우 전송/수신 사이트 마스크 외부에서 작동하는 우주선으로 인해 안테나와의 위성 접촉 시간이 단축될 수 있습니다. 따라서 이전 에페메리스와 크게 다른 새 에페메리스를 업로드한 후 향후 연락처를 취소하고 다시 예약하는 것이 좋습니다.

[DescribeContact API](#)를 사용하면 예약된 고객 응대 startTime 및를 반환된

visibilityStartTime 및 endTime와 비교하여 전송/수신 사이트 마스크 외부에서 작동하는 우주선으로 인해 사용할 수 없는 향후 고객 응대 부분을 확인할 수 있습니다visibilityEndTime. 향후 고객 응대(들)를 취소하고 다시 예약하기로 선택한 경우 고객 응대 시간 범위가 가시성 시간 범위를 30 초 이상 벗어나서는 안 됩니다. 취소된 연락처는 연락 시간에 너무 가깝게 취소되면 비용이 발생할 수 있습니다. 취소된 접촉에 대한 자세한 내용은 [Ground Station FAQ](#)를 참조하세요.

방위각 고도 에페메리스의 경우 예약된 고객 응대는 고객 응대가 예약될 때 선택한 특정 에페메리스를 사용합니다. 예약된 연락처의 방위각 고도 데이터를 업데이트해야 하는 경우 새 에페메리스로 연락처를 취소하고 다시 예약할 수 있습니다.

위성의 현재 에페메리스 가져오기

[GetSatellite](#) 또는 [ListSatellites](#) 작업을 호출하여 특정 위성 AWS Ground Station 에 대해에서 사용 중인 현재 에페메리스를 검색할 수 있습니다. 이 두 메서드는 모두 현재 사용 중인 에페메리스에 대한 메타데이터를 반환합니다. 이 에페메리스 메타데이터에는 업로드된 사용자 지정 에페메리스 AWS Ground Station 와 기본 에페메리스에 대해 다릅니다.

Note

방위각 고도 에페메리스는 위성과 연결되지 않으므로 [GetSatellite](#) 또는 [ListSatellites](#)에서 반환되지 않습니다. 방위각 고도 에페메리스에 대한 정보를 검색하려면 특정 에페메리스 ID와 함께 [DescribeEphemeris](#) API를 사용하거나 [ListEphemerides](#)를 사용하여 계정에 사용 가능한 모든 에페메리스를 확인합니다.

기본 에페메리스는 source 및 epoch 필드만 포함합니다. epoch는 [스페이스 트랙](#)에서 가져온 [두 줄 요소 세트](#)의 [에포크](#)이며, 현재 위성의 궤적을 계산하는 데 사용되고 있습니다.

사용자 지정 에페메리스는 CUSTOMER_PROVIDED의 source 값을 가지며 ephemerisId 필드에 고유한 식별자가 포함됩니다. 이 고유 식별자를 사용하여 [DescribeEphemeris](#) 작업을 통해 에페메리스를 쿼리할 수 있습니다. [CreateEphemeris](#) 작업을 AWS Ground Station 통해 업로드하는 동안 에페메리스에 이름이 할당된 경우 선택적 name 필드가 반환됩니다.

에페메리스는에 의해 동적으로 업데이트 AWS Ground Station 되므로 반환된 데이터는 API 호출 시 사용되는 에페메리스의 스냅샷일 뿐입니다.

기본 에페메리스를 사용하는 위성에 대한 [GetSatellite](#) 반환 예제

```
{
  "satelliteId": "e1cfe0c7-67f9-4d98-bad2-EXAMPLE",
  "satelliteArn": "arn:aws:groundstation::111122223333:satellite/e1cfe0c7-67f9-4d98-bad2-EXAMPLE",
  "noradSatelliteID": 25994,
  "groundStations": [
    "Ohio 1",
    "Oregon 1"
  ],
  "currentEphemeris": {
    "source": "SPACE_TRACK",
    "epoch": 1528245583.619
  }
}
```

사용자 지정 에페메리스를 사용하는 위성에 대한 [GetSatellite](#) 예제

```
{
  "satelliteId": "e1cfe0c7-67f9-4d98-bad2-EXAMPLE",
  "satelliteArn": "arn:aws:groundstation::111122223333:satellite/e1cfe0c7-67f9-4d98-bad2-EXAMPLE",
  "noradSatelliteID": 25994,
  "groundStations": [
    "Ohio 1",
    "Oregon 1"
  ],
  "currentEphemeris": {
    "source": "CUSTOMER_PROVIDED",
    "ephemerisId": "e1cfe0c7-67f9-4d98-bad2-EXAMPLE",
    "name": "My Ephemeris"
  }
}
```

방위각 고도 에페메리스 나열

방위각 고도 에페메리스는 위성과 연결되지 않으므로 다른 APIs를 사용하여 위성에 대한 정보를 검색하고 검색해야 합니다.

1. [ListEphemerides](#)를 사용하여 방위각 고도 에페메리스를 포함하여 계정의 모든 에페메리스를 나열합니다. 상태 및 에페메리스 유형별로 필터링할 수 있습니다.
2. 특정 에페메리스 ID와 함께 [DescribeEphemeris](#)를 사용하여 방위각 고도 에페메리스에 대한 자세한 정보를 가져옵니다.
3. 특정 연락처 ID와 함께 [DescribeContact](#)를 사용하여 연락처에 사용되는 에페메리스에 대한 자세한 정보를 가져옵니다.

방위각 고도 에페메리스를 포함한 [ListEphemerides](#) 응답의 예:

```
{
  "ephemerides": [
    {
      "ephemerisId": "abc12345-6789-def0-1234-5678EXAMPLE",
      "ephemerisType": "AZ_EL",
      "name": "Azimuth Elevation for Ohio Ground Station",
      "status": "ENABLED",
      "creationTime": 1620254718.765
    },
    {
      "ephemerisId": "def45678-9012-abc3-4567-8901EXAMPLE",
      "ephemerisType": "TLE",
      "name": "TLE for Satellite 12345",
      "status": "ENABLED",
      "creationTime": 1620254700.123
    }
  ]
}
```

Note

[ListEphemerides](#) 응답에서 방위각 고도 에페메리스에는 groundStation 필드 대신 satelliteId 필드가 있으므로 쉽게 식별할 수 있습니다.

기본 에페메리스 데이터로 되돌리기

사용자 지정 에페메리스 데이터를 업로드하면 해당 특정 위성에 대해 AWS Ground Station 사용하는 기본 에페메리스를 재정의합니다. AWS Ground Station 는 현재 활성화되고 만료되지 않은 고객 제공 에페메리스를 사용할 수 없을 때까지 기본 에페메리스를 다시 사용하지 않습니다. AWS Ground Station 또한 해당 만료 시간 이후에 사용할 수 있는 기본 에페메리스가 있더라도 현재 고객이 제공한 에페메리스의 만료 시간 이후의 고객 응대를 나열하지 않습니다.

Note

방위각 고도 에페메리스에는 기본값이 없으며 위성 에페메리스를 재정의하지 않습니다. `trackingOverrides` 파라미터를 사용하여 연락처를 예약할 때 명시적으로 선택됩니다. 방위각 고도 에페메리스를 더 이상 사용하지 않으려면 추적 재정의 지정하지 않고 연락처를 예약하면 시스템에서 활성 위성 에페메리스를 대신 사용합니다.

TLE 및 OEM 에페메리스 되돌리기

위성의 기본 [스페이스 트랙](#) 에페메리스로 되돌리려면 다음 중 하나를 수행해야 합니다.

- 활성화된 모든 고객 제공 에페메리스를 삭제([DeleteEphemeris](#) 사용)하거나 비활성화([UpdateEphemeris](#) 사용)합니다. [ListEphemerides](#).
- 고객이 제공한 기존 에페메리스가 모두 만료될 때까지 기다리세요.

[GetSatellite](#)를 호출하고 위성에 대한 현재 에페메리스source의가 인지 확인하여 기본 에페메리스가 사용되고 있는지 확인할 수 있습니다SPACE_TRACK. 기본 에페메리스에 [기본 에페메리스 데이터](#) 대한 자세한 내용은 섹션을 참조하세요.

방위각 고도 에페메리스 관리

방위각 고도 에페메리스는 각 접촉에 대해 명시적으로 선택되고 위성과 연결되지 않으므로 "되돌아가기"라는 개념은 없습니다. 대신 다음과 같이 방위각 고도 에페메리스를 관리할 수 있습니다.

- 방위각 고도 에페메리스 사용을 중지하려면:를 지정trackingOverrides하고 지정하지 않고 새 연락처를 예약하기만 하면 됩니다satelliteArn. 연락처는 대신 지정된 위성에 활성 에페메리스를 사용합니다.

- 미사용 방위각 고도 에페메리스 제거: [DeleteEphemeris](#)를 사용하여 더 이상 필요하지 않은 방위각 고도 에페메리스를 삭제합니다. 예약된 고객 응대에서 현재 사용 중인 에페메리스는 삭제할 수 없습니다.

계정의 모든 방위각 고도 에페메리스를 나열하려면 [ListEphemerides](#)를 사용합니다. 방위각 고도 에페메리스는 `ephemerisType` 필드 또는 응답에 `groundStation` 필드 대신 `satelliteId` 필드의 존재로 식별할 수 있습니다.

데이터 흐름 작업

AWS Ground Station 는 노드 및 엣지 관계를 사용하여 데이터 흐름을 구성하여 데이터의 스트림 처리를 활성화합니다. 각 노드는 예상 처리를 설명하는 구성으로 표시됩니다. 이 개념을 설명하려면의 데이터 흐름을 로 고려antenna-downlink하세요s3-recording. antenna-downlink 노드는 구성에 정의된 파라미터에 따른 무선 주파수 스펙트럼의 아날로그-디지털 변환을 나타냅니다. 는 수신 데이터를 수신하여 S3 버킷에 저장하는 컴퓨팅 노드를 s3-recording 나타냅니다. 결과 데이터 흐름은 사양에 따라 디지털화된 RF 데이터를 S3 버킷으로 비동기식으로 전송하는 것입니다.

미션 프로파일 내에서 필요에 맞게 많은 데이터 흐름을 생성할 수 있습니다. 다음 섹션에서는 AWS Ground Station 와 함께 사용할 다른 AWS 리소스를 설정하는 방법을 설명하고 데이터 흐름을 구성하기 위한 권장 사항을 제공합니다. 소스 또는 대상 노드로 간주되는지 여부를 포함하여 각 노드의 작동 방식에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station Configs 사용](#).

주제

- [AWS Ground Station 데이터 영역 인터페이스](#)
- [리전 간 데이터 전송 사용](#)
- [Amazon S3 설정 및 구성](#)
- [Amazon VPC 설정 및 구성](#)
- [Amazon EC2 설정 및 구성](#)

AWS Ground Station 데이터 영역 인터페이스

선택한 데이터 흐름의 결과 데이터 구조는 데이터 흐름의 소스에 따라 달라집니다. 이러한 형식에 대한 세부 정보는 위성 온보딩 중에 제공됩니다. 다음은 각 데이터 흐름 유형에 사용되는 형식을 요약한 것입니다.

- 안테나 다운링크
 - (대역폭less-than-or-equal-to) 데이터는 [VITA-49 신호 데이터/IP](#) 형식 패킷으로 전송됩니다. 40MHz
 - (대역폭 40MHz 초과) 데이터는 AWS Ground Station 클래스 2 패킷으로 전송됩니다.
- antenna-downlink-demod-decode
 - 데이터는 복조/디코딩된 데이터/IP 형식 패킷으로 전송됩니다.
- 안테나 링크

- 데이터는 [VITA-49 신호 데이터/IP](#) 형식 패킷으로 전송되어야 합니다.
- antenna-uplink-echo
 - 데이터는 [VITA-49 신호 데이터/IP](#) 형식 패킷으로 전송됩니다.

리전 간 데이터 전송 사용

리전 AWS Ground Station 간 데이터 전송 기능을 사용하면 안테나에서 AWS Ground Station 지원되는 모든 AWS 리전으로 데이터를 유연하게 전송할 수 있습니다. 즉, 단일 AWS 리전에서 인프라를 유지하고 온보딩된 모든에 대한 연락처를 예약할 수 [AWS Ground Station 위치](#) 있습니다.

Amazon S3 버킷에서 연락처 데이터를 수신하면 AWS Ground Station 가 모든 전송 측면을 관리합니다.

Amazon EC2 인스턴스로 리전 간 데이터 전송을 사용하려면(에 AWS Ground Station 이진트 또는 데이터 흐름 엔드포인트 사용) 현재 AWS 리전에서 데이터 흐름 엔드포인트를 생성해야 하며 dataflow-endpoint-config은 동일한 리전을 지정해야 합니다. AWS Ground Station 는 데이터 리전 간 전송을 관리합니다.

Amazon S3 설정 및 구성

Amazon S3 버킷을 활용하여를 사용하여 다운로드 신호를 수신할 수 있습니다 AWS Ground Station. 대상 s3-recording-config를 생성하려면가 버킷에 파일을 AWS Ground Station 쓸 수 있는 권한을 부여하는 Amazon S3 버킷과 IAM 역할을 지정할 수 있어야 합니다.

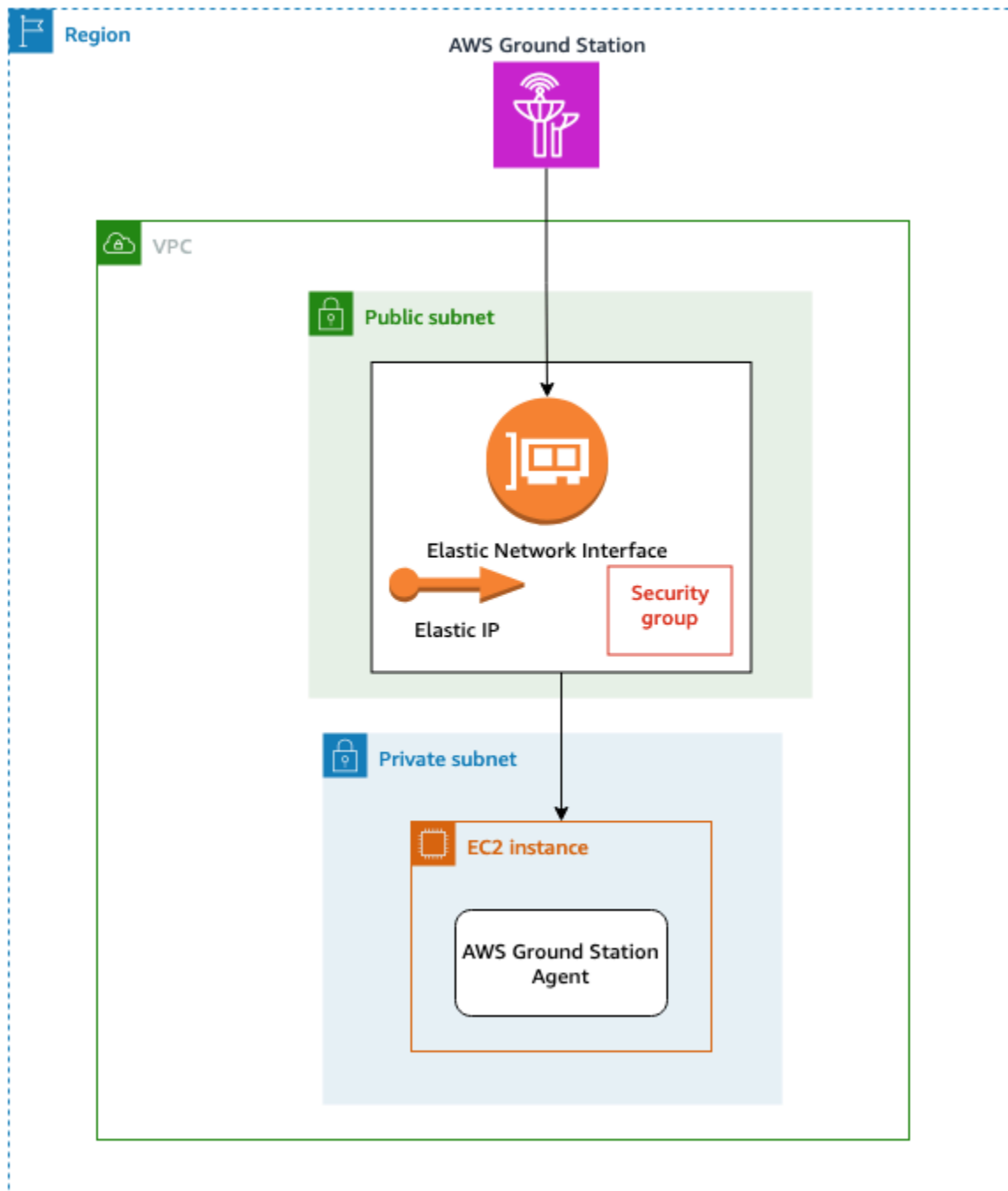
Amazon S3 버킷, IAM 역할 또는 구성 생성에 대한 제한 사항은 AWS Ground Station 섹션을 참조 [Amazon S3 레코딩 구성](#)하세요.

Amazon VPC 설정 및 구성

VPC를 설정하는 전체 가이드는이 가이드의 범위를 벗어납니다. 자세한 내용은 [Amazon VPC 사용 설명서](#)를 참조하세요.

이 섹션에서는 Amazon EC2 및 데이터 흐름 엔드포인트가 VPC 내에 존재할 수 있는 방법에 대해 설명합니다.는 지정된 데이터 흐름에 대해 여러 전송 지점을 지원하지 AWS Ground Station 않습니다. 각 데이터 흐름은 단일 EC2 수신기로 종료될 것으로 예상됩니다. 단일 EC2 수신기가 예상대로 구성은 다중 AZ 중복이 아닙니다. VPC를 사용할 전체 예제는 섹션을 참조하세요 [미션 프로파일 구성 예](#).

AWS Ground Station 에이전트를 사용한 VPC 구성



위성 데이터는 안테나에 근접한 AWS Ground Station 에이전트 인스턴스에 제공됩니다. AWS Ground Station 에이전트는 사용자가 제공한 AWS KMS 키를 사용하여 데이터를 스트라이프한 다음 암호화합니다. 각 스트라이프는 AWS Network 백본을 통해 소스 안테나에서 [Amazon EC2 탄력적 IP\(EIP\)](#)로 전송됩니다. 데이터는 연결된 Amazon EC2 Elastic Network Interface(ENI)를 통해 EC2 인스턴스에 도착합니다. [Amazon EC2](#) EC2 인스턴스에서 설치된 AWS Ground Station 에이전트는 데이터를 해독하고

전달 오류 수정(FEC)을 수행하여 삭제된 데이터를 복구한 다음 설정에서 지정한 IP 및 포트로 전달합니다.

아래 목록은 에이전트 전송을 위해 AWS Ground Station VPC를 설정할 때 고유한 설정 고려 사항을 호출합니다.

보안 그룹 - AWS Ground Station 트래픽 전용 보안 그룹을 설정하는 것이 좋습니다. 이 보안 그룹은 Dataflow 엔드포인트 그룹에서 지정한 것과 동일한 포트 범위에서 UDP 수신 트래픽을 허용해야 합니다. 이는 권한을 AWS Ground Station IP 주소로만 제한하도록 AWS 관리형 접두사 목록을 AWS Ground Station 유지합니다. 배포 리전의 PrefixListId를 [교체하는 방법에 대한 자세한 내용은 AWS 관리형 접두사 목록을](#) 참조하세요.

탄력적 네트워크 인터페이스(ENI) - 위의 보안 그룹을 이 ENI와 연결하고 퍼블릭 서브넷에 배치해야 합니다.

Note

ENI당 연결된 보안 그룹 수의 기본 할당량은 5입니다. 이는 최대 16개의 조정 가능한 제한입니다. [Amazon VPC 할당량을 참조하세요.](#)

다음 CloudFormation 템플릿은 이 섹션에 설명된 인프라를 생성하는 방법을 보여줍니다.

ReceiveInstanceEIP:

```
Type: AWS::EC2::EIP
Properties:
  Domain: 'vpc'
```

InstanceSecurityGroup:

```
Type: AWS::EC2::SecurityGroup
Properties:
  GroupDescription: AWS Ground Station receiver instance security group.
  VpcId: YourVpcId
  SecurityGroupIngress:
    # Add additional items here.
    - IpProtocol: udp
      FromPort: your-port-start-range
      ToPort: your-port-end-range
      PrefixListIds:
        - PrefixListId: com.amazonaws.global.groundstation
  Description: "Allow AWS Ground Station Downlink ingress."
```

InstanceNetworkInterface:

Type: AWS::EC2::NetworkInterface

Properties:

Description: *ENI for AWS Ground Station to connect to.*

GroupSet:

- !Ref *InstanceSecurityGroup*SubnetId: *A Public Subnet***ReceiveInstanceEIPAllocation:**

Type: AWS::EC2::EIPAssociation

Properties:

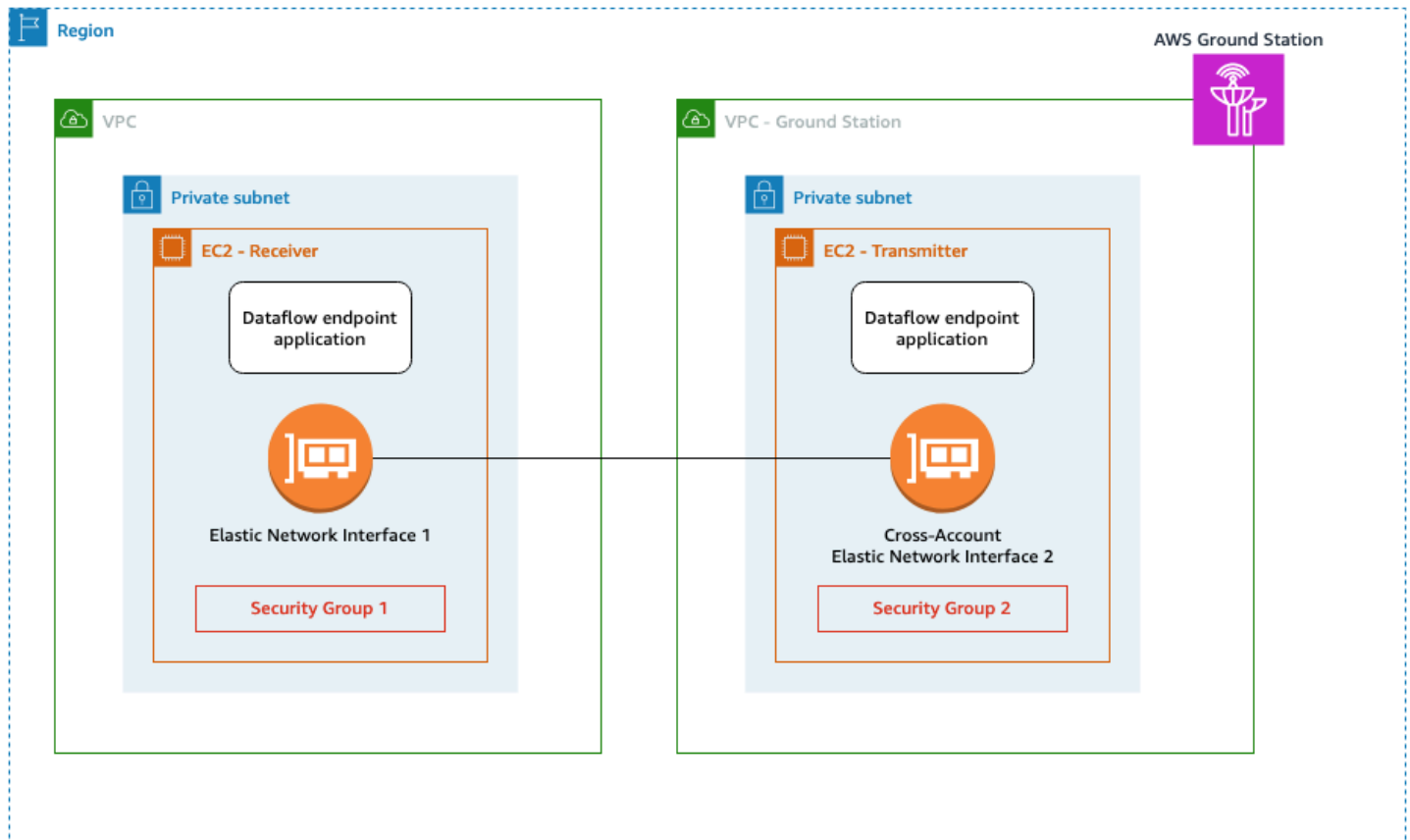
AllocationId:

Fn::GetAtt: [*ReceiveInstanceEIP*, AllocationId]

NetworkInterfaceId:

Ref: *InstanceNetworkInterface*

데이터 흐름 엔드포인트를 사용한 VPC 구성



위성 데이터는 안테나에 근접한 데이터 흐름 엔드포인트 애플리케이션 인스턴스에 제공됩니다. 그런 다음 데이터는 소유 VPC에서 교차 계정 [Amazon EC2 탄력적 네트워크 인터페이스\(ENI\)](#)를 통해 전송됩니다. AWS Ground Station. 그런 다음 데이터는 Amazon EC2 인스턴스에 연결된 ENI를 통해 EC2 인스턴스에 도착합니다. Amazon EC2 그러면 설치된 데이터 흐름 엔드포인트 애플리케이션이 설정에서 지정한 IP 및 포트로 이를 전달합니다. 이 흐름의 역방향은 업링크 연결에 대해 발생합니다.

아래 목록은 데이터 흐름 엔드포인트 전송을 위해 VPC를 설정할 때 고유한 설정 고려 사항을 호출합니다.

Note

ENI당 연결된 보안 그룹 수의 기본 할당량은 5입니다. 이는 최대 16개의 조정 가능한 제한입니다. [Amazon VPC 할당량을 참조하세요.](#)

IAM 역할 - IAM 역할은 데이터 흐름 엔드포인트의 일부이며 다이어그램에 표시되지 않습니다. 교차 계정 ENI를 생성하고 AWS Ground Station Amazon EC2 인스턴스에 연결하는 데 사용되는 IAM 역할입니다.

보안 그룹 1 -이 보안 그룹은 계정의 Amazon EC2 인스턴스에 연결될 ENI에 연결됩니다. dataflow-endpoint-group에 지정된 포트에서 보안 그룹 2의 UDP 트래픽을 허용해야 합니다.

탄력적 네트워크 인터페이스(ENI) 1 - 보안 그룹 1을 ENI와 연결하고 서브넷에 배치해야 합니다.

서브넷 - 계정의 Amazon EC2 인스턴스에 대해 데이터 흐름당 사용 가능한 IP 주소가 하나 이상 있는지 확인해야 합니다. 서브넷 크기 조정에 대한 자세한 내용은 [서브넷 CIDR 블록](#)을 참조하세요.

보안 그룹 2 -이 보안 그룹은 데이터 흐름 엔드포인트에서 참조됩니다. 이 보안 그룹은 계정에 데이터를 배치하는 데 AWS Ground Station 사용할 ENI에 연결됩니다.

리전 - 교차 리전 연결에 지원되는 리전에 대한 자세한 내용은 [리전 간 데이터 전송 사용](#)을 참조하세요.

다음 CloudFormation 템플릿은 이 섹션에 설명된 인프라를 생성하는 방법을 보여줍니다.

DataFlowEndpointSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Security Group for AWS Ground Station registration of Dataflow Endpoint Groups

VpcId: *YourVpcId*

AWSGroundStationSecurityGroupEgress:

Type: AWS::EC2::SecurityGroupEgress

Properties:

GroupId: !Ref: *DataflowEndpointSecurityGroup*

IpProtocol: udp

FromPort: *55555*

ToPort: *55555*

CidrIp: *10.0.0.0/8*

Description: *"Allow AWS Ground Station to send UDP traffic on port 55555 to the 10/8 range."*

InstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: *AWS Ground Station receiver instance security group.*

VpcId: *YourVpcId*

SecurityGroupIngress:

- IpProtocol: udp

FromPort: *55555*

ToPort: *55555*

SourceSecurityGroupId: *!Ref DataflowEndpointSecurityGroup*

Description: *"Allow AWS Ground Station Ingress from DataflowEndpointSecurityGroup"*

ReceiverSubnet:

Type: AWS::EC2::Subnet

Properties:

Ensure your CidrBlock will always have at least one available IP address per dataflow endpoint.

See <https://docs.aws.amazon.com/vpc/latest/userguide/subnet-sizing.html> for subnet sizing guidelines.

CidrBlock: *"10.0.0.0/24"*

Tags:

- Key: "Name"

Value: *"AWS Ground Station - Dataflow endpoint Example Subnet"*

- Key: "Description"

Value: *"Subnet for EC2 instance receiving AWS Ground Station data"*

VpcId: !Ref ReceiverVPC

Amazon EC2 설정 및 구성

AWS Ground Station 에이전트 또는 데이터 흐름 엔드포인트를 통해 VITA-49 Signal/IP 데이터 또는 VITA-49 Extension 데이터/IP를 동기식으로 전송하려면 Amazon EC2 인스턴스를 올바르게 구성해야 합니다. 특정 요구 사항에 따라 동일한 인스턴스에서 직접 프론트 엔드(FE) 프로세서 또는 소프트웨어 정의 라디오(SDR)를 수행하거나 추가 EC2 인스턴스를 활용해야 할 수 있습니다. FE 또는 SDR의 선택 및 설치에 이 사용 설명서의 범위를 벗어납니다. 특정 데이터 형식에 대한 자세한 내용은 섹션을 참조하십시오 [AWS Ground Station 데이터 영역 인터페이스](#).

서비스 약관에 대한 자세한 내용은 [AWS 서비스 약관](#)을 참조하십시오.

제공 공통 소프트웨어

AWS Ground Station 는 Amazon EC2 인스턴스를 쉽게 설정할 수 있는 공통 소프트웨어를 제공합니다.

AWS Ground Station 에이전트

AWS Ground Station 에이전트는 디지털 중간 주파수(DigiF) 다운링크 데이터를 수신하고 복호화된 데이터를 내보내 다음을 활성화합니다.

- 40MHz ~ 400MHz 대역폭의 DigiF 다운링크 기능을 제공합니다.
- AWS 네트워크의 모든 퍼블릭 IP(AWS 탄력적 IP)로 높은 속도, 낮은 지터 DigiF 데이터 전송.
- 순방향 오류 수정(FEC)을 사용한 안정적인 데이터 전송.
- 암호화를 위해 고객 관리형 AWS KMS 키를 사용하여 데이터를 안전하게 전송합니다.

자세한 내용은 [AWS Ground Station 에이전트 사용 설명서](#)를 참조하십시오.

데이터 흐름 엔드포인트 애플리케이션

에서 AWS Ground Station 안테나 위치와 Amazon EC2 인스턴스 간에 데이터를 보내고 받는 AWS Ground Station 데 사용하는 네트워킹 애플리케이션입니다. 데이터의 업링크 및 다운링크에 사용할 수 있습니다.

소프트웨어 정의 라디오(SDR)

위성과 통신하는 데 사용되는 신호를 조절/디모듈링하는 데 사용할 수 있는 소프트웨어 정의 라디오(SDR)입니다.

AWS Ground Station Amazon Machine Image(AMIs)

이러한 설치의 빌드 및 구성 시간을 줄이기 위해 사전 구성된 AMIs AWS Ground Station 도 제공합니다. 데이터 흐름 엔드포인트 네트워킹 애플리케이션과 소프트웨어 정의 라디오(SDR)가 있는 AMIs 는 온보딩이 완료된 후 계정에 제공됩니다. 프라이빗 Amazon Machine Image(AMI)에서 Groundstation 을 검색하여 Amazon EC2 [AMIs](#). AWS Ground Station 에이전트AMIs는 퍼블릭이며 퍼블릭 Amazon Machine Image(AMI)에서 Groundstation을 검색하여 Amazon EC2 [AMIs](#).

원격 측정 작업

AWS Ground Station 원격 측정은 위성 접촉 중에 AWS Ground Station 안테나에서 거의 실시간에 가까운 지표를 제공합니다. 원격 측정 데이터를 사용하여 고객 응대 성능을 모니터링하고, 이상을 감지하고, 위성 통신에 대해 정보에 입각한 결정을 내릴 수 있습니다.

주제

- [원격 측정 작동 방식](#)
- [사용 가능한 원격 측정 유형](#)
- [리전별 가용성](#)
- [원격 측정 설정](#)
- [원격 측정 데이터 이해](#)

원격 측정 작동 방식

원격 측정을 사용하려면 원격 측정 데이터를 전송하는 위치를 지정하는 TelemetrySinkConfig AWS Ground Station 를 구성합니다. 그런 다음 telemetrySinkConfigArn 필드를 사용하여이 구성을 미션 프로파일에 추가합니다. 원격 측정 지원 미션 프로파일을 사용하는 고객 응대 중에는 원격 측정 데이터를 계정으로 AWS Ground Station 스트리밍합니다.

원격 측정 전송 프로세스는 다음과 같이 작동합니다.

1. AWS 계정에 Kinesis Data Streams 스트림을 생성하여 원격 측정 데이터를 수신합니다. 스트림은 연락처를 예약하는 것과 동일한 계정 및 리전에서 생성해야 합니다.
2. 스트림에 데이터를 쓸 수 있는 AWS Ground Station 권한을 부여하는 IAM 역할을 생성합니다.
3. 스트림 및 IAM 역할을 참조하는 TelemetrySinkConfig를 생성합니다.
4. TelemetrySinkConfig를 미션 프로파일에 추가합니다.
5. 새로운 원격 측정 지원 미션 프로파일을 사용하여 연락처를 나열하고 예약합니다.
6. 이 미션 프로파일을 사용하는 고객 응대 중에는 원격 측정 데이터를 거의 실시간으로 Kinesis Data Streams 스트림으로 AWS Ground Station 스트리밍합니다.
7. AWS 서비스 또는 자체 애플리케이션을 사용하여 스트림의 원격 측정 데이터를 사용하고 처리합니다.

사용 가능한 원격 측정 유형

AWS Ground Station 는 고객 응대 중에 다음과 같은 원격 측정 유형을 제공합니다.

Note

AWS Ground Station 는 지원되는 원격 측정 유형의 수를 확장하기 위해 노력하고 있습니다.

텔레메트리 가리키기

위성 접촉 중 안테나 방향에 대한 정보를 제공합니다. 이 원격 측정 유형은 항상 접촉 중에 전송되며 실제 및 명령 방위각과 고도 각도를 포함합니다. 자세한 내용은 [텔레메트리 가리키기](#) 섹션을 참조하세요.

원격 측정 추적

안테나 추적 상태 및 추적 오류에 대한 정보를 제공합니다. 이 원격 측정 유형은 추적 구성에서 자동 추적이 활성화된 경우 전송됩니다. 자세한 내용은 [원격 측정 추적](#) 섹션을 참조하세요.

리전별 가용성

텔레메트리는가 AWS Ground Station 실행되는 모든 AWS 리전에서 사용할 수 있습니다. 고객 응대 실행 중에 원격 측정이 AWS Ground Station 안테나에서 고객 응대를 예약한 리전으로 전송되어 교차 리전 지원을 제공합니다.

AWS Ground Station 리전 및 지상국 위치의 전체 목록은 섹션을 참조하세요 [AWS Ground Station 위치](#).

원격 측정 설정

다음 단계에 따라 AWS Ground Station 연락처에 대한 원격 측정을 구성합니다. 이 설정을 완료하면 원격 측정이 활성화된 미션 프로파일을 사용하는 고객 응대 중에 원격 측정 데이터가 Kinesis Data Streams 스트림으로 전송됩니다. Kinesis Data Streams에 대한 자세한 내용은 [Kinesis Data Streams 사용 설명서](#)를 참조하세요.

1단계: 사전 조건 AWS 리소스 생성

다음 CloudFormation 코드 조각은 원격 측정 전송을 위한 사전 필수 AWS 리소스를 생성하는 방법을 보여줍니다. 이 코드 조각은 Kinesis Data Streams 스트림과 스트림에 원격 측정 데이터를 쓸 수 있는 AWS Ground Station 권한을 부여하는 IAM 역할을 생성합니다.

TelemetryStream:

```
Type: AWS::Kinesis::Stream
Properties:
  Name: GroundStationTelemetryStream
  StreamModeDetails:
    StreamMode: ON_DEMAND
  RetentionPeriodHours: 24
```

TelemetryRole:

```
Type: AWS::IAM::Role
Properties:
  RoleName: GroundStationTelemetryRole
  AssumeRolePolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal:
          Service: groundstation.amazonaws.com
        Action: sts:AssumeRole
  Policies:
    - PolicyName: KinesisWritePolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - kinesis:DescribeStream
              - kinesis:PutRecord
              - kinesis:PutRecords
            Resource: !GetAtt TelemetryStream.Arn
```

아래 목록은 원격 측정 전송을 구성할 때 고유한 설정 고려 사항을 호출합니다 AWS Ground Station.

Kinesis Data Streams 스트림 - 스트림은 처리량에 따라 자동으로 조정되는 온디맨드 용량 모드를 사용합니다. 이는 대부분의 사용 사례에 권장됩니다. 스트림은 24시간 동안 데이터를 보존하도록 구성됩니다. 기본적으로 스트림은 AWS 관리형 암호화를 사용합니다. 에서 고객 관리형 암호화를 사용하려면

StreamEncryption 속성을 AWS Key Management Service 추가하고 kms:GenerateDataKey 권한을 포함하도록 IAM 역할 정책을 업데이트합니다. 자세한 내용은 [Amazon Kinesis Data Streams의 데이터 보호](#)를 참조하세요.

IAM 역할 - IAM 역할을 통해 groundstation.amazonaws.com 서비스 보안 주체는 역할을 수임하고 Kinesis Data Streams 스트림에 원격 측정 데이터를 쓸 수 있습니다. 역할 정책은 스트림의 kinesis:DescribeStream, kinesis:PutRecord 및 kinesis:PutRecords 작업에 대한 권한을 부여합니다. 신뢰 정책 및 역할 정책 설정에 대한 지침은 [원격 측정 싱크 구성](#) 섹션을 참조하세요.

추가 구성 - AWS Ground Station API 호출에 사용하는 IAM 사용자 또는 역할에 iam:PassRole 권한을 추가합니다. 이렇게 하면 TelemetrySinkConfig를 생성할 AWS Ground Station 때 원격 측정 역할에 전달할 수 있습니다. TelemetrySinkConfig

PassRole 정책 예제

역할 정책을 업데이트하거나 연결하는 방법에 대한 자세한 내용은 [IAM 사용 설명서의 IAM 정책 관리](#)를 참조하세요. iam:PassRole 권한에 대한 자세한 내용은 [AWS 서비스에 역할을 전달할 수 있는 사용자 권한 부여를 참조하세요](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::999999999999:role/your-telemetry-delivery-role-name"
    }
  ]
}
```

2단계: TelemetrySinkConfig 생성

가 원격 측정 데이터를 Kinesis Data Streams 스트림에 전송하는 방법을 정의하는 TelemetrySinkConfig를 생성합니다. AWS Ground Station 1단계의 CloudFormation 스택 출력에서 스트림 ARN 및 역할 ARN을 사용합니다.

Note

TelemetrySinkConfig를 생성하면 AWS Ground Station 는 파티션 키가 인 빈 테스트 레코드를 전송하여 Kinesis Data Streams 스트림에 대한 액세스를 확인합니다test.

TelemetrySinkConfig 생성에 대한 자세한 내용은 섹션을 참조하세요[원격 측정 싱크 구성](#).

3단계: 미션 프로파일에 원격 측정 추가

미션 프로파일을 생성합니다. 미션 프로파일 생성에 대한 자세한 내용은 섹션을 참조하세요[AWS Ground Station 미션 프로파일 사용](#). 미션 프로파일telemetrySinkConfigArn에를 추가하여 연락 중에 원격 측정 전송을 활성화합니다. 2단계에서 생성한 TelemetrySinkConfig의 ARN을 사용합니다.

4단계: 고객 응대 예약

원격 측정이 활성화된 미션 프로파일을 사용하여 고객 응대를 예약합니다. 고객 응대 중에 AWS Ground Station 는 원격 측정 데이터를 Kinesis Data Streams 스트림으로 스트리밍합니다.

고객 응대 중에 예상되는 사항

- 원격 측정 시작 - 고객 응대가 시작되면 데이터가 스트리밍을 시작합니다.
- 실시간에 가까운 전송 - 원격 측정은 거의 실시간으로 Kinesis Data Streams 스트림에 도착합니다.
- 고객 응대 기간 - 데이터는 전체 고객 응대에서 계속됩니다.
- 자동 중지 - 고객 응대가 종료되면 원격 측정이 스트리밍을 중지합니다.

전송 모니터링

다음을 사용하여 원격 측정 전송을 모니터링할 수 있습니다.

- Kinesis Data Streams 스트림 지표 - CloudWatch에서 수신 레코드를 확인합니다. 자세한 내용은 [Amazon Kinesis Data Streams 모니터링](#)을 참조하세요.
- 애플리케이션 로그 - 스트림에서 사용하는 애플리케이션의 데이터 처리를 확인합니다.
- Kinesis Data Viewer - Kinesis Data Streams 스트림 콘솔을 사용하여 스트림의 샘플 레코드를 봅니다.

다음 단계

설정을 완료한 후 다음을 수행할 수 있습니다.

- 원격 측정 데이터 형식과 사용 가능한 원격 측정 유형에 대해 알아봅니다. [원격 측정 데이터 이해](#) 섹션을 참조하세요.
- 애플리케이션을 구축하여 Kinesis Data Streams 스트림의 원격 측정 데이터를 처리합니다. 자세한 내용은 [Amazon Kinesis Data Streams용 소비자 구축을 참조하세요](#).
- CloudWatch 및 기타 AWS 서비스를 사용하여 대시보드 및 알림을 생성합니다.
- 문제가 발생할 경우 문제 해결 지침을 검토하세요. [원격 측정 문제 해결](#) 섹션을 참조하세요.

원격 측정 데이터 이해

원격 측정 데이터는 Kinesis Data Streams 스트림에 Base64-encoded JSON 레코드로 전송됩니다. 각 레코드에는 고객 응대 및 샘플링된 원격 측정에 대한 메타데이터를 포함하여 위성 고객 응대 중에 수집된 정보가 포함됩니다.

데이터 형식 개요

각 원격 측정 레코드에는 다음 구성 요소가 포함됩니다.

원격 측정 유형 및 버전

특정 유형의 원격 측정 데이터와 스키마 버전을 식별합니다. 이를 통해 다양한 원격 측정 유형을 적절하게 구문 분석할 수 있습니다. 스키마 버전 관리에 대한 자세한 내용은 섹션을 참조하세요 [스키마 버전 관리 및 진화](#).

범위 ID

원격 측정 범위의 고유 식별자입니다. 이를 통해 원격 측정 데이터를 특정 연락처와 연결할 수 있습니다.

Metadata

원격 측정에 대한 컨텍스트 정보입니다.

데이터

원격 측정 유형과 관련된 샘플링된 원격 측정입니다.

파티션 키

원격 측정 레코드는 다음과 같은 형식의 파티션 키를 사용하여 Kinesis Data Streams 스트림으로 전송됩니다.

```
SCOPE#scopeId#TELEMETRY_ID#telemetryId#TELEMETRY_VERSION#telemetryVersion
```

이 파티션 키는 단일 고객 응대에 대해 지정된 유형의 모든 원격 측정이 Kinesis Data Streams 스트림 내의 동일한 샤드로 전송되도록 하여 해당 고객 응대의 원격 측정 스트림에 대한 최선의 작업 순서를 제공합니다.

텔레메트리 가리키기

조준 원격 측정은 위성 접촉 중 안테나 조준 방향에 대한 정보를 제공합니다. 이 원격 측정 유형은 고객 응대 중에 항상 전송됩니다.

데이터 필드

sampleTimestamp

원격 측정 데이터가 샘플링된 시간으로, ISO-8601 형식은 밀리초 정밀도의 UTC입니다.

방위각

도 단위의 안테나의 실제 방위각입니다.

고도

안테나의 실제 고도 각도입니다.

commandedAzimuth

도 단위의 명령 방위각입니다. 안테나가 달성하려고 시도하는 목표 방위각입니다.

commandedElevation

도 단위의 명령된 고도 각도입니다. 이는 안테나가 달성하려고 시도하는 목표 고도 각도입니다.

Note

실제 안테나 위치는 접촉 중 물리적 제한 또는 기계적 지연으로 인해 명령된 위치와 다를 수 있습니다.

메타데이터 필드

groundStation

지상국의 이름(예: "Ohio 1").

satelliteId

의 위성 리소스 식별자입니다 AWS Ground Station.

contactId

연락처의 식별자입니다.

예제 JSON

```
{
  "telemetryTypeAndVersion": "POINTING#1.0.0",
  "telemetryType": "POINTING",
  "telemetryVersion": "1.0.0",
  "scopeId": "12345678-1234-1234-1234-123456789012",
  "metadata": {
    "groundStation": "Ohio 1",
    "satelliteId": "87654321-4321-4321-4321-210987654321",
    "contactId": "12345678-1234-1234-1234-123456789012"
  },
  "data": {
    "sampleTimestamp": "2025-12-08T12:00:00.123Z",
    "azimuth": 180.5,
    "elevation": 45.2,
    "commandedAzimuth": 180.0,
    "commandedElevation": 45.0
  }
}
```

원격 측정 추적

추적 원격 측정은 안테나 추적 상태 및 추적 오류에 대한 정보를 제공합니다. 이 원격 측정 유형은 추적 구성에서 자동 추적이 활성화되어 있고 안테나가 자동 추적을 적극적으로 사용하고 있을 때 전송됩니다.

Note

TrackingConfig의 autotrack 파라미터가 로 설정된 경우 추적 원격 측정이 전송REMOVED되지 않습니다. 구성 추적에 대한 자세한 내용은 섹션을 참조하세요 [추적 구성](#).

데이터 필드**sampleTimestamp**

원격 측정 데이터가 샘플링된 시간으로, ISO-8601 형식은 밀리초 정밀도의 UTC입니다.

trackingStatus

안테나의 현재 추적 상태입니다. 가능한 값은 다음과 같습니다.

- TRACKING - 안테나가 미션 프로파일과 일치하는 신호에 성공적으로 잠겼으며 하늘에서 이를 적극적으로 따르고 있습니다. 고객 응대 중 공칭 작동 상태입니다.
- ACQUIRING - 안테나가 신호를 찾아 잠그는 중입니다. 시스템은 현재 에페메리스 데이터를 기반으로 하는 프로그래밍 방식 추적을 사용하고 있습니다.
- MASKED - 위성의 예측 위치는 자동 추적 마스크 뒤에 있습니다. 즉, 안테나는 특정 방향에서 자동 추적을 안정적으로 활용할 수 없습니다. 이는 일반적으로 낮은 고도와 같이 RF 간섭이 높은 영역에서 발생합니다.

trackingErrorAzimuth

각도로 측정되는 방위각 측의 추적 오류입니다.

trackingErrorElevation

고도 측의 추적 오류로, 각도로 측정됩니다.

Note

추적 오류 값은 신호 강도를 극대화하기 위해 자동 추적 중에 AWS Ground Station 적용되는 에페메리스 기반 프로그램 트랙의 조정을 나타냅니다.

메타데이터 필드

원격 측정 추적에는 원격 측정을 가리키는 것과 동일한 메타데이터 필드인 groundStation, 및 satelliteId가 포함됩니다contactId.

예제 JSON

```
{
  "telemetryTypeAndVersion": "TRACKING#1.0.0",
  "telemetryType": "TRACKING",
  "telemetryVersion": "1.0.0",
  "scopeId": "12345678-1234-1234-1234-123456789012",
  "metadata": {
    "groundStation": "Ohio 1",
    "satelliteId": "87654321-4321-4321-4321-210987654321",
    "contactId": "12345678-1234-1234-1234-123456789012"
  },
  "data": {
    "sampleTimestamp": "2025-12-08T12:00:00.123Z",
    "trackingStatus": "TRACKING",
    "trackingErrorAzimuth": 0.2,
    "trackingErrorElevation": 0.1
  }
}
```

Kinesis Data Streams 스트림에서 데이터 읽기

원격 측정 데이터는 Kinesis Data Streams 스트림으로 전송되며 표준 스트림 소비 패턴을 사용하여 사용할 수 있습니다. 스트림에서 데이터를 읽을 때는 다음 사항을 고려해야 합니다.

Base64 디코딩

Kinesis Data Streams 스트림의 데이터는 Base64-encoded됩니다. JSON으로 구문 분석하기 전에 데이터를 디코딩해야 합니다. 자세한 내용은 [Amazon Kinesis Data Streams 작업을](#) 참조하세요.

Kinesis Data Viewer 사용

원격 측정 데이터에 빠르게 액세스할 수 있도록 Kinesis Data Streams 스트림 콘솔은 데이터 뷰어 기능을 제공합니다. 이 기능을 사용하는 경우:

- 텔레메트리 전송은 스트림 내의 모든 샤드에 발생할 수 있습니다.
- 기본 시작 위치는 샤드의 최신 레코드에서 읽습니다.
- 선택한 샤드를 조정하고 “타임스탬프 시작” 시작 위치를 사용하여 수신된 레코드를 확인해야 할 수 있습니다.

Kinesis Client Library 사용

Kinesis Client Library(KCL)는 샤드 관리, 체크포인트 및 로드 밸런싱을 포함하여 Kinesis Data Streams 스트림의 데이터 소비와 관련된 많은 복잡성을 관리합니다. 프로덕션 원격 측정 소비 애플리케이션에 KCL을 사용하는 것이 좋습니다.

자세한 내용은 [Kinesis Client Library를 사용하여 소비자 개발을 참조하세요](#).

사용 모범 사례

- 지연 시간 최소화 - 향상된 팬아웃을 사용하여 폴링에 비해 전용 처리량과 짧은 지연 시간으로 Kinesis Data Streams 스트림에서 읽을 수 있습니다. 자세한 내용은 [향상된 팬아웃 소비자 개발을 참조하세요](#).
- 전용 스트림 - AWS Ground Station 원격 측정 통합에 전용 Kinesis Data Streams 스트림을 사용합니다. 스트림을 다른 애플리케이션과 공유하면 쓰기 처리량 포화 및 원격 측정 전송 실패가 발생할 수 있습니다.
- 온디맨드 용량 - 처리량에 따라 샤드를 자동으로 조정할 수 있도록 Kinesis Data Streams 스트림을 온디맨드 프로비저닝 모드로 배포합니다.
- 처리량 모니터링 - CloudWatch 지표를 사용하여 스트림의 제한을 모니터링합니다. 자세한 내용은 [Amazon Kinesis Data Streams 모니터링을 참조하세요](#).

스키마 버전 관리 및 진화

원격 측정 스키마는 시간 경과에 따른 변화를 지원하도록 버전이 지정됩니다. 각 레코드의 `telemetryVersion` 필드는 스키마 버전을 나타냅니다.

스키마 변경 처리

- 향후 새로운 원격 측정 유형이 도입될 수 있습니다.
- 기존 원격 측정 유형은 주요 변경 사항이 있는 새 버전을 수신할 수 있습니다.
- 애플리케이션은 알 수 없는 원격 측정 유형 및 버전을 허용해야 합니다.
- `telemetryTypeAndVersion`, `telemetryType` 및 `telemetryVersion` 필드를 구문 분석하여 각 레코드를 처리하는 방법을 결정합니다.

여러 스키마 버전을 정상적으로 처리할 수 있는 버전 인식 페이로드 직렬화를 구현하여 새 버전이 도입될 때 애플리케이션이 계속 작동할 수 있도록 하는 것이 좋습니다.

연락처 작업

AWS Ground Station 콘솔 또는 원하는 언어로 된 AWS SDK를 사용하여 위성 데이터를 입력하고, 안테나 위치를 식별하고, 통신하고 AWS CLI, 선택한 위성의 안테나 시간을 예약할 수 있습니다. 고객 응대 시작* 15분 전까지 고객 응대 예약을 검토, 취소 및 다시 예약할 수 있습니다. 연락처를 업데이트하여 방위각/상승, OEM 또는 TLE 추적 데이터를 포함한 에페메리스 재정의의 지정하거나 대상 위성을 변경할 수도 있습니다. 자세한 내용은 [연락처 및 연락처 버전 관리 업데이트](#) 섹션을 참조하세요. 또한 AWS Ground Station 예약 시간(분) 요금 모델을 사용하는 경우 예약 시간(분) 요금제의 세부 정보도 볼 수 있습니다.

AWS Ground Station 는 리전 간 데이터 전송을 지원합니다. 선택한 미션 프로파일의 일부인 데이터 흐름 엔드포인트 구성에 따라 데이터가 전송되는 리전이 결정됩니다. 리전 간 데이터 전송 사용에 대한 자세한 내용은 섹션을 참조하세요 [리전 간 데이터 전송 사용](#).

접촉을 예약하려면 리소스를 구성해야 합니다. 리소스를 구성하지 않은 경우 섹션을 참조하세요 [시작하기](#). [ReserveContact](#)가 호출되면는 고객 응대의 수명 주기 동안 사용할 미션 프로파일 및 구성 리소스의 스냅샷을 AWS Ground Station 생성합니다. [UpdateMissionProfile](#) 및 [UpdateConfig](#) APIs를 사용하여 이러한 리소스를 변경해도 업데이트 전에 예약된 연락처에는 반영되지 않습니다. 이미 예약된 연락처에 리소스 변경 사항을 적용해야 하는 경우 먼저 [CancelContact](#)를 사용하여 연락처를 취소한 다음 [ReserveContact](#)를 사용하여 다시 예약해야 합니다.

* 취소된 연락처는 연락 시간에 너무 가까워 취소되면 비용이 발생할 수 있습니다. 취소된 접촉에 대한 자세한 내용은 [Ground Station FAQ](#)를 참조하세요.

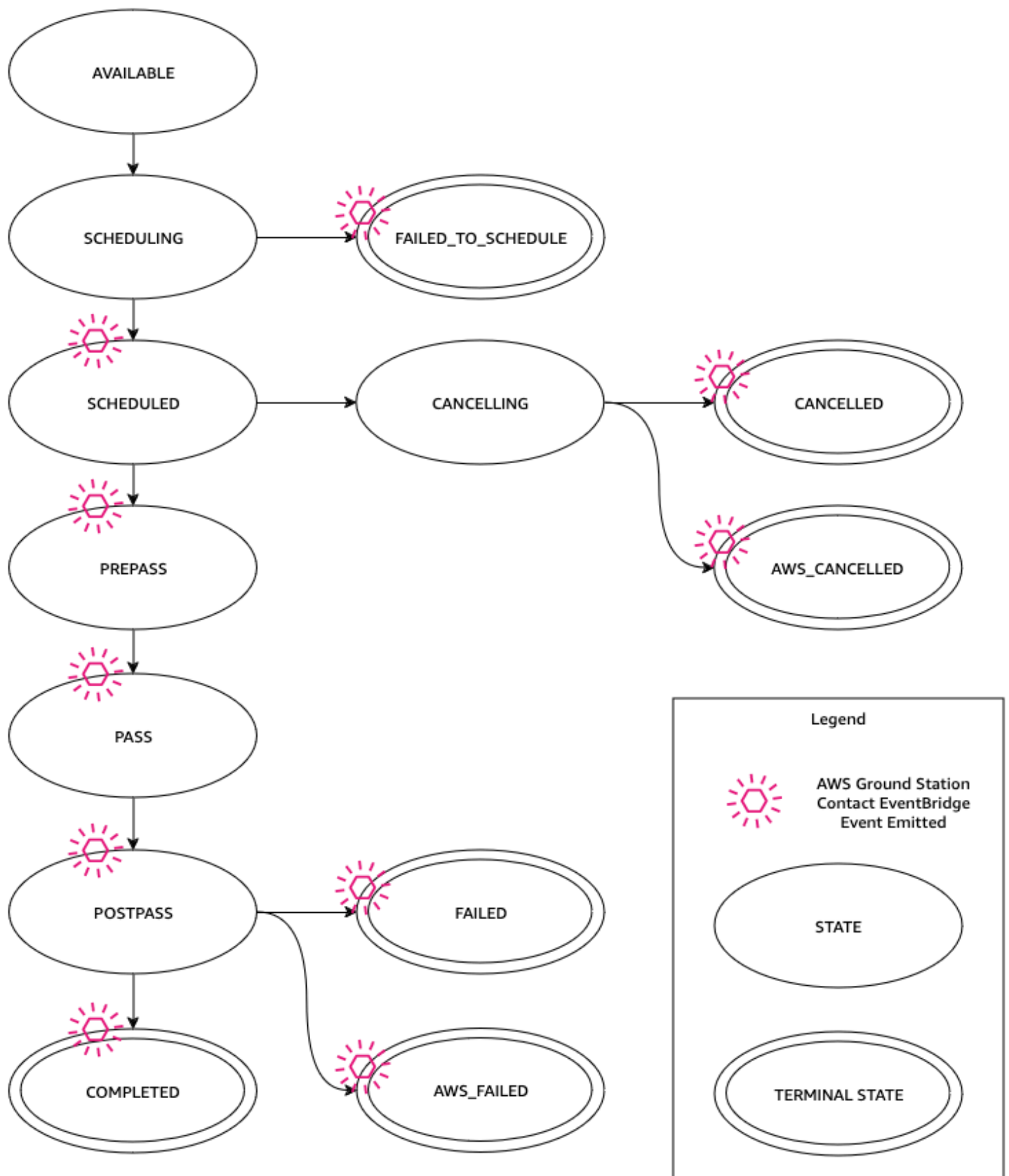
주제

- [고객 응대 수명 주기 이해](#)
- [고객 응대 결제 이해](#)
- [연락처 및 연락처 버전 관리 업데이트](#)

고객 응대 수명 주기 이해

고객 응대 수명 주기를 이해하면를 사용하는 동안 다양한 문제를 자동화하고 해결하는 데 도움이 될 수 있습니다 AWS Ground Station. 다음 다이어그램은 고객 AWS Ground Station 응대 수명 주기와 수명 주기 중에 발생하는 이벤트 브리지 이벤트를 보여줍니다. COMPLETED, FAILED, FAILED_TO_SCHEDULE, CANCELLED, AWS_CANCELLED 및 AWS_FAILED는 터미널 상태입니다. 고객 응대는 터미널 상태에서 전환되지 않습니다. 각 상태가 나타내는 내용과 [CancelContact](#)를 사용

하여 중지 가능 또는 취소 가능 여부에 [AWS Ground Station 고객 응대 상태](#) 대한 자세한 내용은 섹션을 참조하세요.



AWS Ground Station 고객 응대 상태

AWS Ground Station 고객 응대 상태는 지정된 시간에 해당 고객 응대에 어떤 일이 일어나고 있는지에 대한 통찰력을 제공합니다.

고객 응대 상태

다음 표에서는 연락처가 가질 수 있는 상태를 설명합니다.

Status	설명	터미널	취소 가능	중지 가능
AVAILABLE	연락처를 예약할 수 있습니다.	아니요	해당 사항 없음	해당 사항 없음
예약	고객 응대가 예약 중입니다.	아니요	예	아니요
SCHEDULED	고객 응대가 성공적으로 예약되었습니다.	아니요	예	아니요
FAILED_TO_SCHEDULE	고객 응대를 예약하지 못했습니다.	예	아니요	아니요
사전 통과	고객 응대가 곧 시작되고 리소스가 준비 중입니다.	아니요	예	아니요
PASS	고객 응대가 현재 실행 중이며 위성과 통신 중입니다.	아니요	아니요	예
사후 통과	통신이 완료되었으며 사용된 리소스가 정리되고 있습니다.	아니요	아니요	아니요
COMPLETED	고객 응대가 오류 없이 완료되었습니다.	예	아니요	아니요
FAILED	리소스 구성 문제로 인해 연락이 실패했습니다.	예	아니요	아니요
AWS_FAILED	AWS Ground Station 서비스 문제로 인해 연락이 실패했습니다.	예	아니요	아니요

Status	설명	터미널	취소 가능	중지 가능
취소 중	고객 응대가 취소되는 중입니다.	아니요	아니요	아니요
AWS_CANCELED	AWS Ground Station 서비스에서 고객 응대를 취소했습니다. 안테나 또는 사이트 유지 관리 및 에페메리스 드리프트는 이러한 상황이 발생할 수 있는 경우의 예입니다.	예	아니요	아니요
취소됨	고객 응대가 취소되었습니다.	예	아니요	아니요

Note

취소되거나 중지된 연락처의 결제 영향에 대한 자세한 내용은 섹션을 참조하세요 [고객 응대 결제 이해](#).

고객 응대 데이터 보존

AWS Ground Station 는 고객 응대 예약 [ReserveContact](#) 요청이 이루어진 후 1년 동안 고객 응대 데이터를 보관합니다. 1년이 지나면 연락처 데이터가 삭제됩니다.

1년 이상 연락처 데이터를 보존해야 하는 경우 보존 기간이 만료되기 전에 데이터를 내보내는 것이 좋습니다. 연락처 데이터에 액세스하고 내보내는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [AWS Ground Station API Reference](#)
- [AWS Ground Station CLI 명령 참조](#)

고객 응대 결제 이해

를 사용하면 사용하는 안테나 시간에 대해서만 AWS Ground Station 비용을 지불합니다. AWS Ground Station 미터는 분당 고객 응대 사용량을 기준으로 합니다. 각 고객 응대에 대해 서비스는 시작 시간부터 종료 시간까지 고객 응대 지속 시간을 계산하고 가장 가까운 분으로 올림합니다. 이 측정 기간은 해당 고객 응대에 대한 요금을 결정합니다.

속도는 다음 두 가지 주요 요인에 따라 달라집니다.

- 대역폭 - 고객 응대에 예약된 대역폭의 양(협대역 또는 광대역)
- 지상국 위치 - 요금은 지상국 위치에 따라 다릅니다.

대역폭 정의

AWS Ground Station 는 순간 대역폭을 기반으로 고객 응대를 두 개의 대역폭 계층으로 분류합니다.

- 협대역 - 순간 대역폭이 40MHz 이하인 모든 고객 응대
- 광대역 - 순간 대역폭이 40MHz보다 큰 모든 고객 응대

예약 모드

AWS Ground Station 는 두 가지 예약 모드를 제공합니다.

- 온디맨드 - 장기 약정 없이 안테나 액세스 비용 지불
- 예약됨 - 월별 약정과 함께 온디맨드에 비해 할인된 요금과 향상된 일정을 제공합니다. 일정 기간 동안 월별 사용량을 약정하는 고객은 예약 분 요금을 사용할 수 있습니다.

계정의 특정 요금 정보 또는 예약 예약 모드에 대한 자세한 내용은 AWS 담당자에게 문의하세요.

CancelContact

[CancelContact](#) API의 사용은 호출 시 연락 상태에 따라 달라집니다.

- 고객 응대 시작 전 - 고객 응대를 완전히 취소합니다.
- 고객 응대 시작 후 및 고객 응대 종료 전 - 진행 중인 고객 응대를 중지합니다.

고객 응대를 취소하는 경우 결제는 예약 모드와 취소 시기에 따라 달라집니다. 자세한 내용은 AWS 담당자에게 문의하세요.

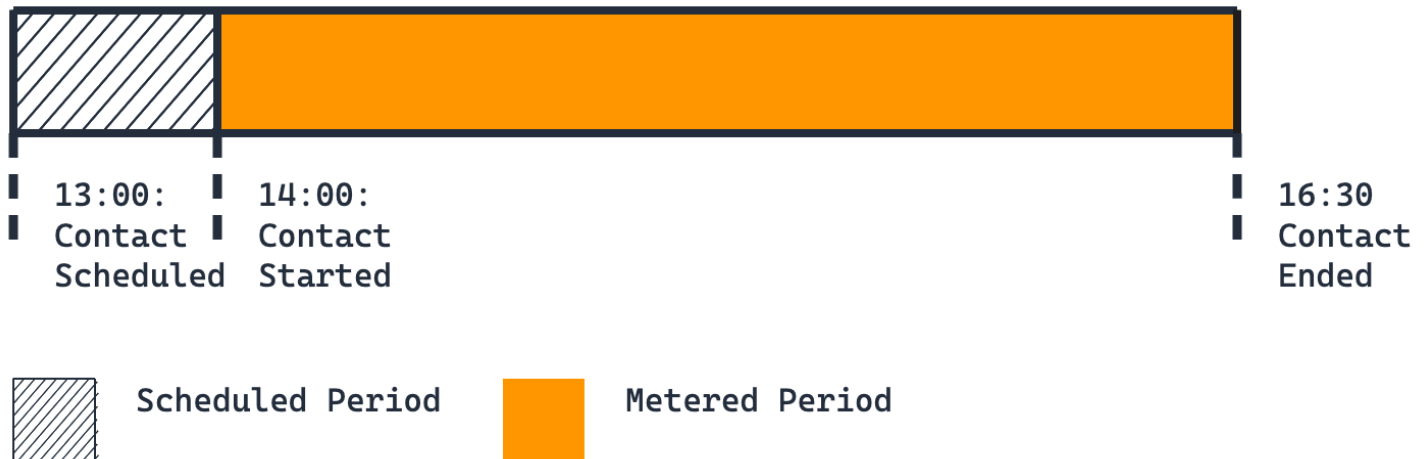
고객 응대를 중지하면 실행된 고객 응대 부분과 중복 고객 응대에서 다루지 않는 남은 시간에 대한 요금이 청구됩니다. 이 컨텍스트에서 중복 연락처는 다음과 같습니다.

- 원래 중지된 고객 응대와 동일한 지상국에서 예약됨
- 원래 중지된 연락처와 동일한 AWS 계정 ID로 예약됨
- 원래 고객 응대를 중지하기 위해 명령이 실행된 후 예약됨

다음 시나리오에서는이 측정이 실제로 어떻게 작동하는지 보여줍니다.

시나리오 1: 단일 고객 응대

Ground Station Anytown 1에서 150분 연락을 14:00에 시작하여 16:30에 종료하도록 예약합니다.



결제 내역:

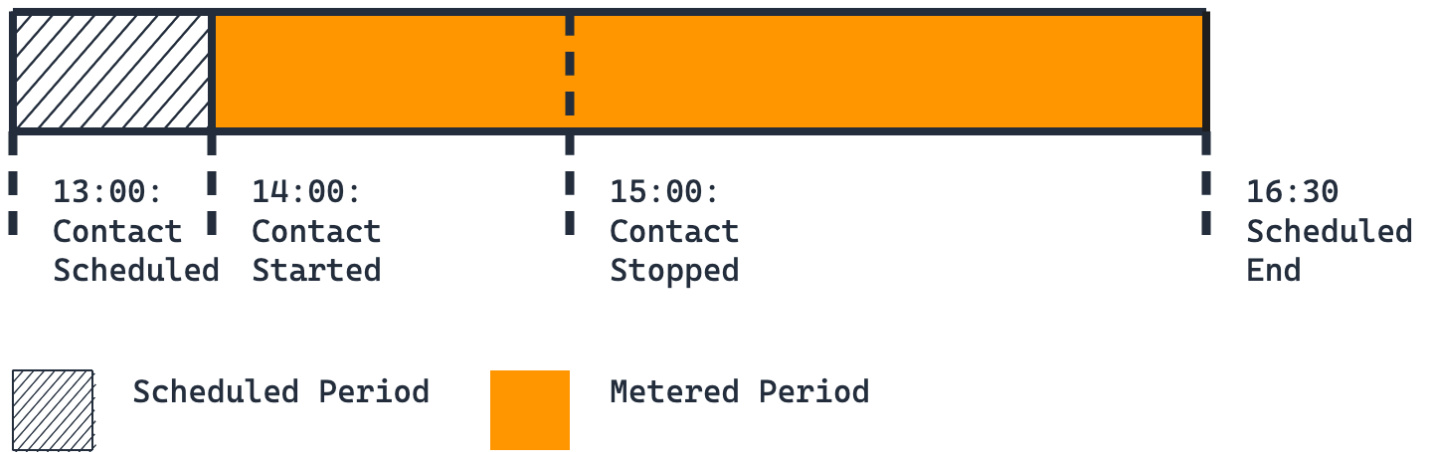
- 첫 번째 연락: 150분(전체 기간)

150분 요금이 청구됩니다. 이는 연락이 중지 또는 취소 없이 예약된 완료까지 실행되는 기준 시나리오입니다.

시나리오 2: 중지된 단일 고객 응대

Ground Station Anytown 1에서 150분 연락을 14:00에 시작하여 16:30에 종료하도록 예약합니다.

15:00에 CancelContact API를 호출하여 고객 응대를 중지합니다.



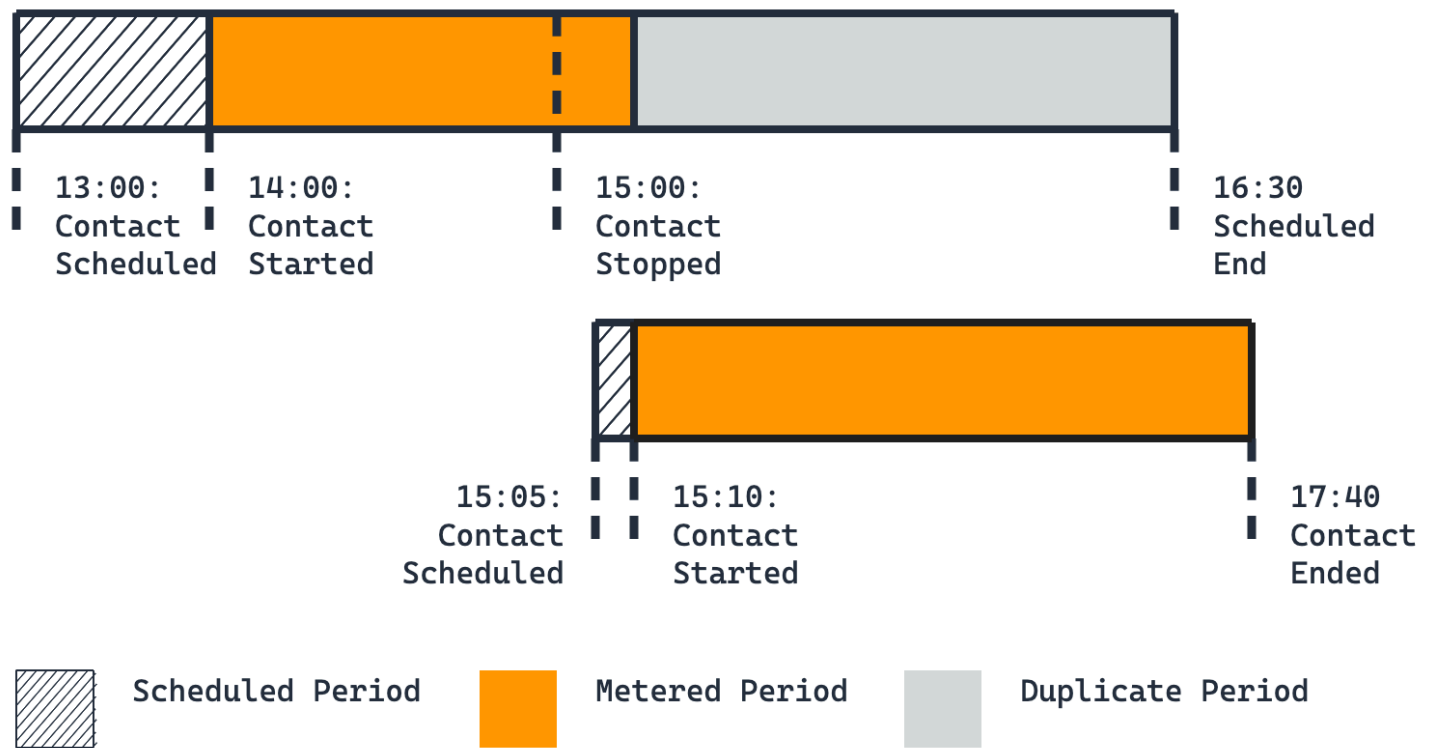
결제 내역:

- 첫 번째 연락: 150분(전체 원래 기간)

고객 응대를 중지했지만 남은 시간(15:00~16:30)을 처리하기 위해 중복 고객 응대를 예약하지 않았기 때문에 전체 150분에 대한 요금이 청구됩니다. 중복을 예약하지 않고 고객 응대를 중지하면 원래 예약된 전체 기간에 대한 책임은 사용자에게 있습니다.

시나리오 3: 단일 중복

Ground Station Anytown 1에서 150분 연락을 14:00에 시작하여 16:30에 종료하도록 예약합니다. 15:00에 CancelContact API를 호출하여 첫 번째 연락처를 중지합니다. CancelContact를 호출한 후 동일한 Ground Station에서 15:10부터 150분 동안 다른 연락처를 예약합니다.



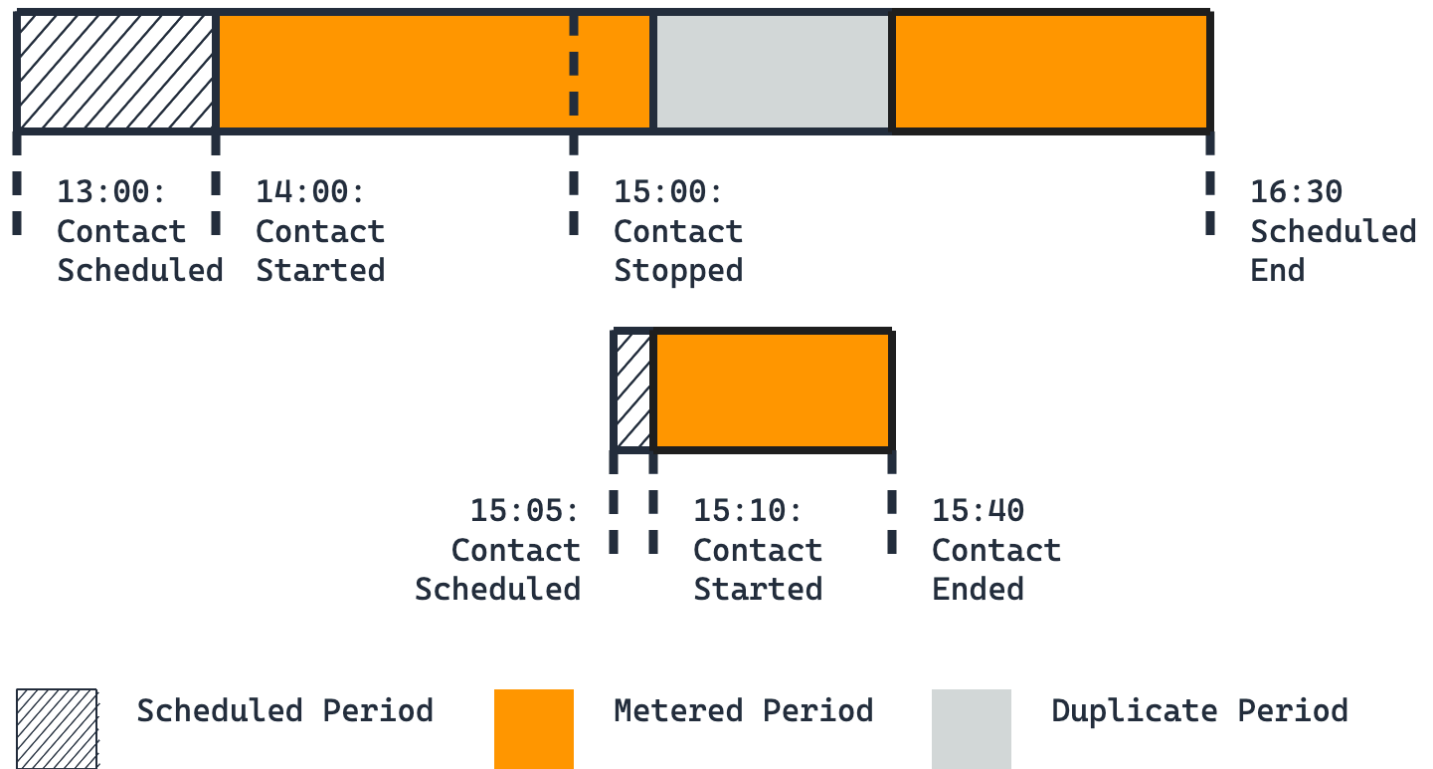
결제 내역:

- 첫 번째 연락: 70분(실행된 60분 + 두 번째 연락이 시작되기 전 가동 중지 시간 10분)
- 두 번째 연락: 150분(전체 기간)

두 번째 고객 응대는 첫 번째 고객 응대를 중지한 후 예약했기 때문에 중복됩니다. 중복은 15:10부터 16:30까지의 남은 시간을 다루므로 첫 번째 고객 응대가 실제로 실행된 시간과 중지와 재시작 사이의 10분 간격에 대해서만 요금이 청구됩니다.

시나리오 4: 짧은 중복

Ground Station Anytown 1에서 150분 연락을 14:00에 시작하여 16:30에 종료하도록 예약합니다. 15:00에 CancelContact API를 호출하여 첫 번째 연락처를 중지합니다. CancelContact를 호출한 후 15:10부터 동일한 Ground Station에서 30분 연락처를 예약합니다.



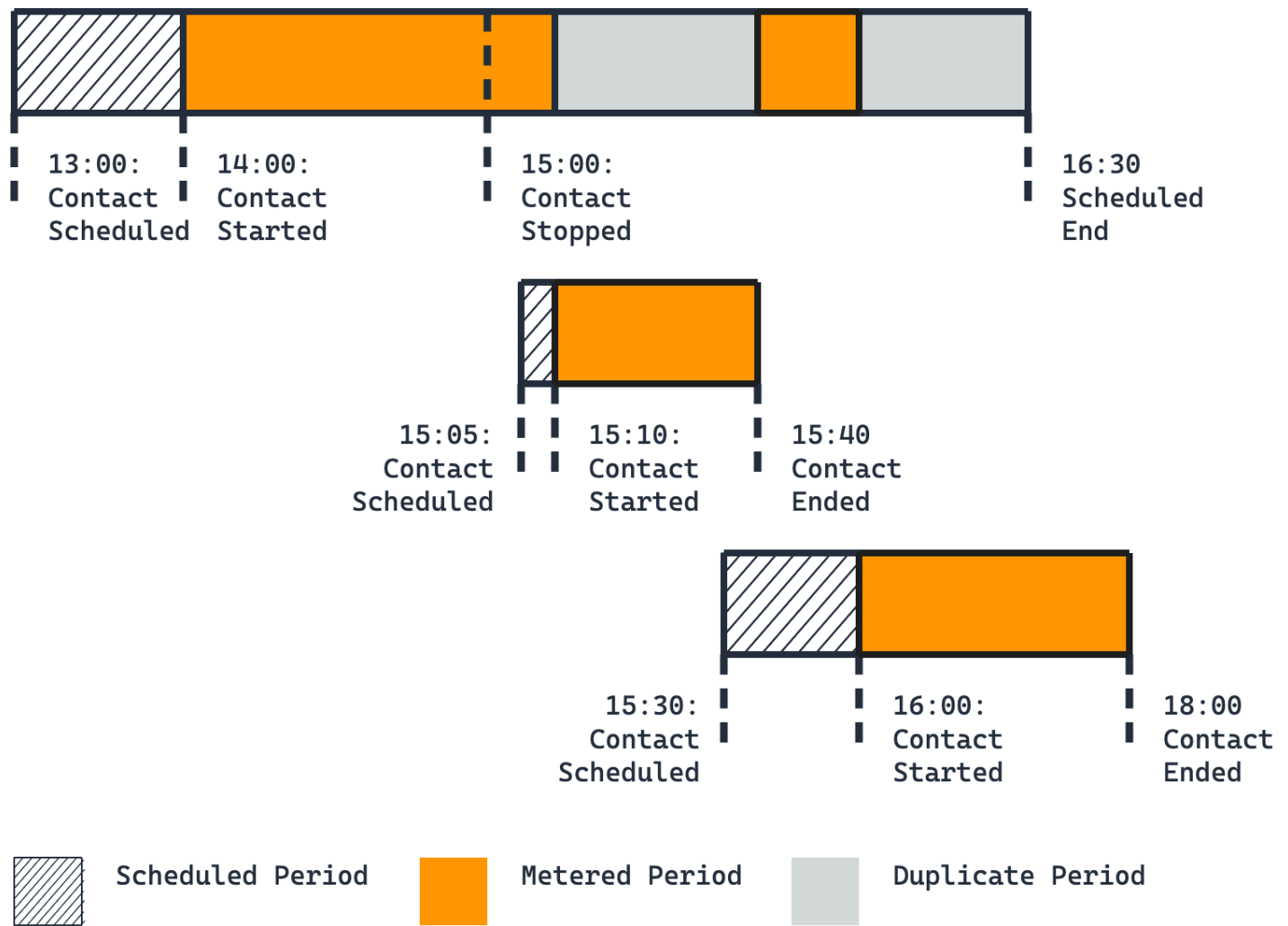
결제 내역:

- 첫 번째 연락: 120분(실행된 60분 + 두 번째 연락이 시작되기 전 가동 중지 시간 10분 + 중복이 처리되지 않은 남은 시간 50분)
- 두 번째 연락: 30분(전체 기간)

중복 고객 응대는 첫 번째 고객 응대를 중지한 후 남은 90분 중 30분(15:10~15:40)만 포함합니다. 중복이 시작되기 전 10분 간격과 중복이 종료된 후 50분의 미확인 시간(15:40~16:30) 모두에 대해 요금이 청구됩니다.

시나리오 5: 여러 중복

Ground Station Anytown 1에서 150분 연락을 14:00에 시작하여 16:30에 종료하도록 예약합니다. 15:00에 CancelContact API를 호출하여 첫 번째 연락처를 중지합니다. CancelContact를 호출한 후 15:10부터 동일한 Ground Station에서 30분 연락처를 예약합니다. 나중에 15:30에 16:00부터 120분 동안 다른 고객 응대를 예약합니다.



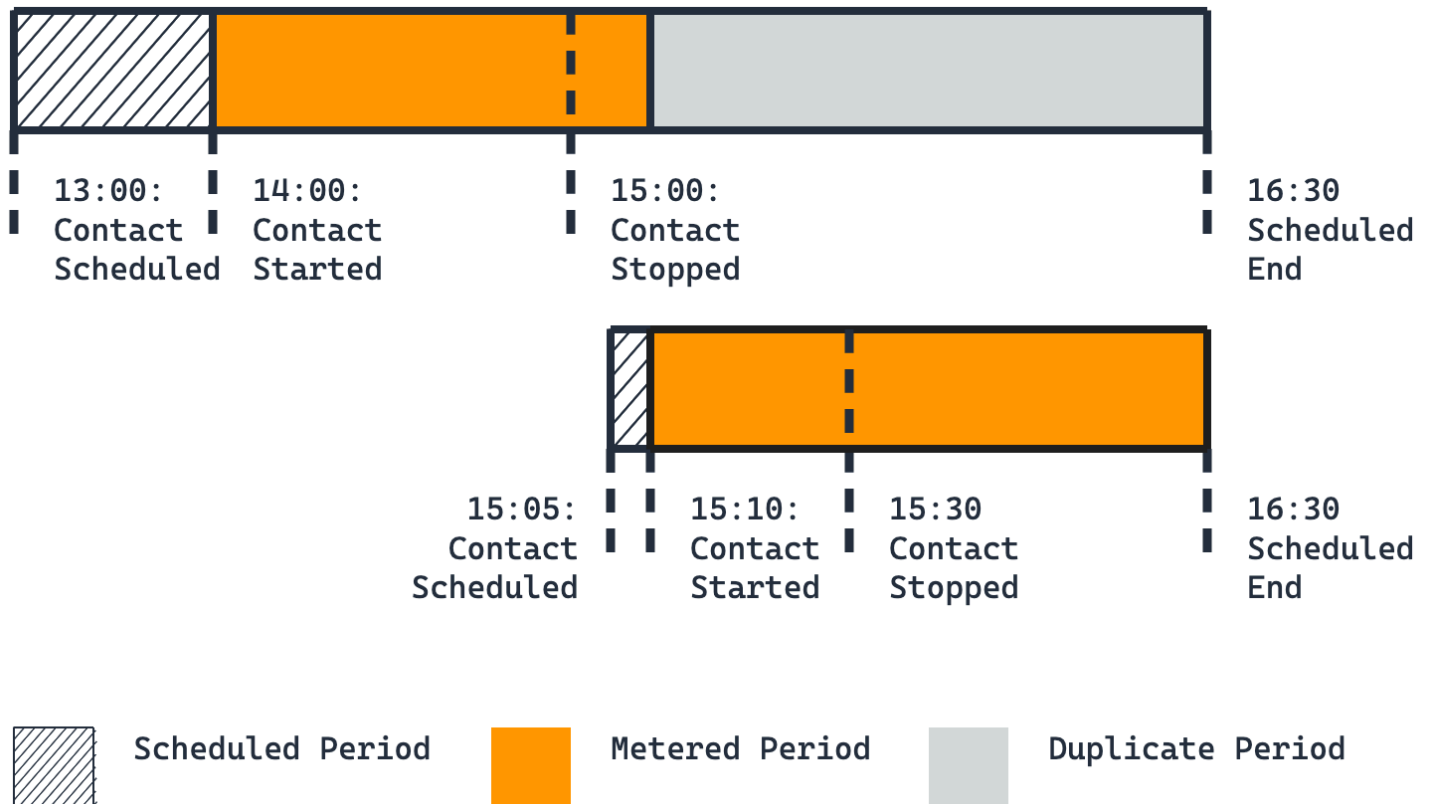
결제 내역:

- 첫 번째 연락: 90분(실행 60분 + 두 번째 연락이 시작되기 전 가동 중지 시간 10분 + 두 번째 연락과 세 번째 연락 사이의 가동 중지 시간 20분)
- 두 번째 연락: 30분(전체 기간)
- 세 번째 연락처: 120분(전체 기간)

두 번째 및 세 번째 고객 응대는 모두 첫 번째 고객 응대를 중지한 후 예약했기 때문에 중복으로 계산됩니다. 그러나 첫 번째 중지(15:00)와 두 번째 시작(15:10) 사이에 10분, 두 번째 종료(15:40)와 세 번째 시작(16:00) 사이에 20분이라는 고객 응대 간의 간격에 대해서는 여전히 요금이 청구됩니다.

시나리오 6: 다중 중지

Ground Station Anytown 1에서 150분 연락을 14:00에 시작하여 16:30에 종료하도록 예약합니다. 15:00에 CancelContact API를 호출하여 첫 번째 연락처를 중지합니다. CancelContact를 호출한 후 Ground Station Anytown 1에서 15:10에 시작하여 16:30에 종료되는 80분 연락처를 예약합니다. 15:30에 CancelContact API를 다시 호출하여 중복 연락처를 중지합니다.



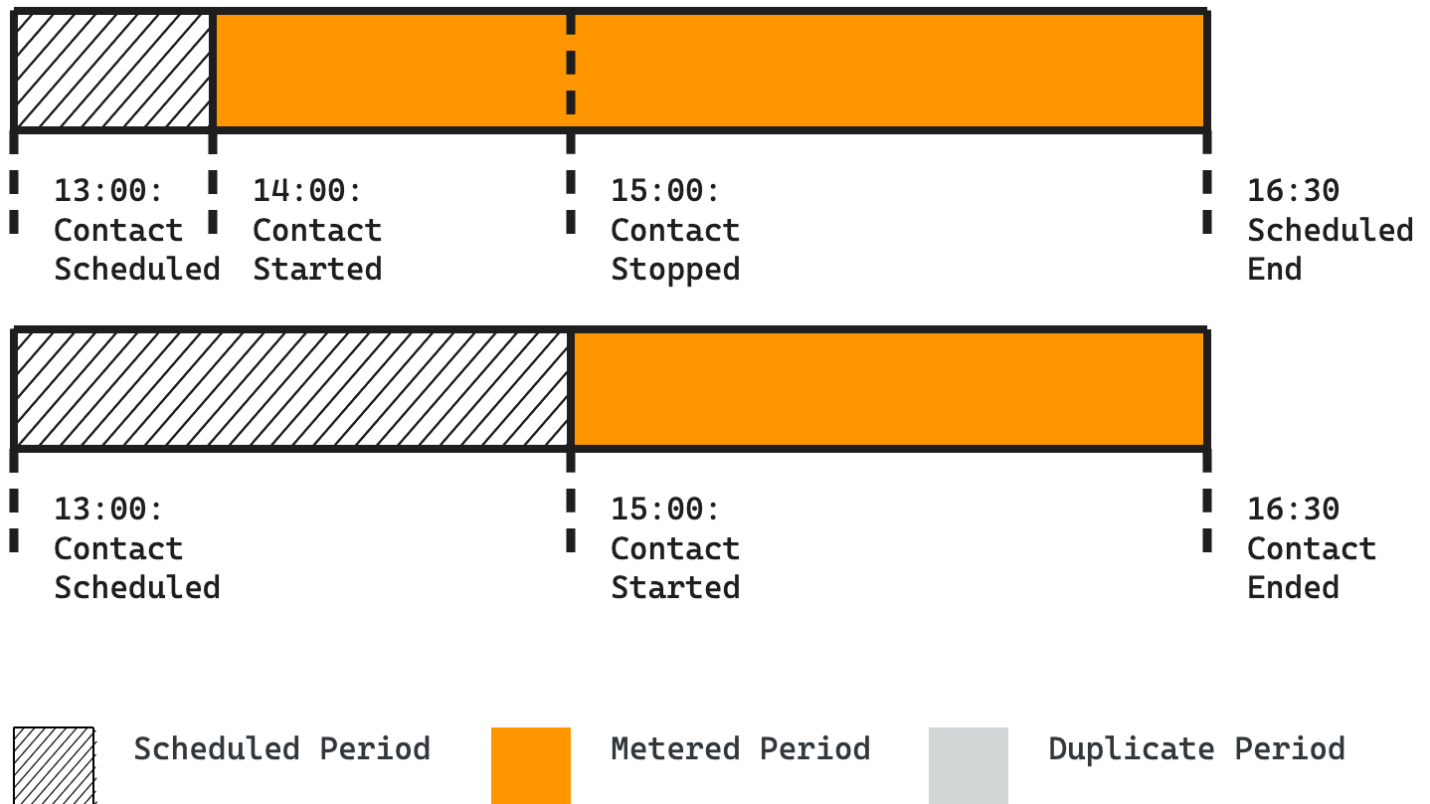
결제 내역:

- 첫 번째 연락: 70분(실행된 60분 + 두 번째 연락이 시작되기 전 가동 중지 시간 10분)
- 두 번째 연락: 80분(전체 원래 기간)

두 번째 고객 응대는 15:30에 중지하여 원래 예약된 시간(15:30~16:30)의 60분을 채우지 않았기 때문에 전체 80분 동안 청구됩니다. 나머지 시간을 포함하도록 다른 중복 고객 응대를 예약하지 않는 한, 중지된 고객 응대의 전체 지속 시간에 대한 책임은 사용자에게 있습니다.

시나리오 7: 중복이 없는 다중 안테나 지상국

13:00에 Ground Station Anytown 1에서 두 연락처를 예약합니다. 첫 번째는 14:00에 시작하여 16:30에 끝나는 150분 고객 응대입니다. 두 번째는 15:00에 시작하여 16:30에 끝나는 90분 고객 응대입니다. 15:00에 CancelContact API를 호출하여 첫 번째 연락처를 중지합니다. Ground Station Anytown 1은 다중 안테나 지상국으로, 두 고객 응대를 동시에 실행할 수 있습니다.



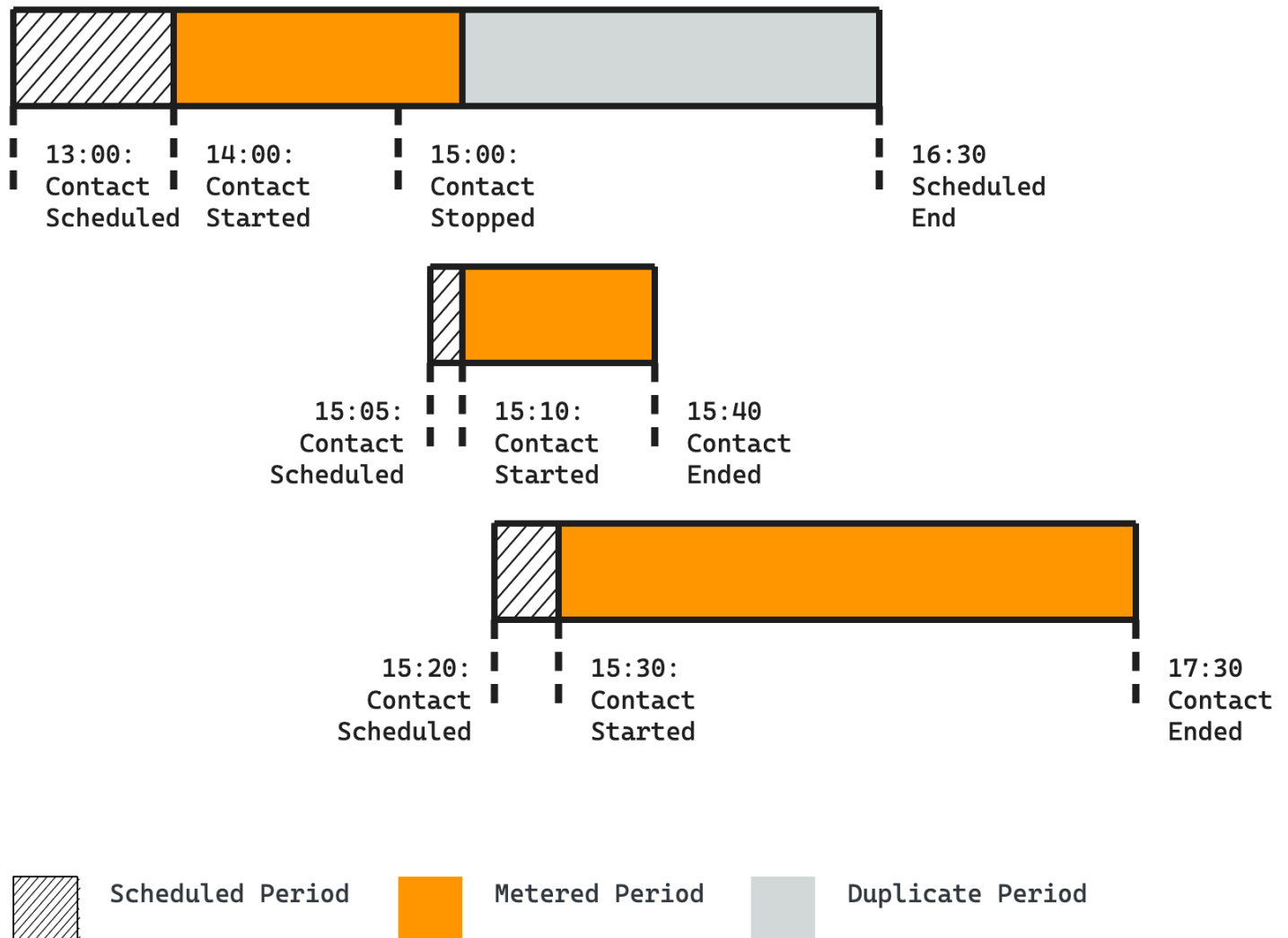
결제 내역:

- 첫 번째 연락: 150분(전체 원래 기간)
- 두 번째 연락: 90분(전체 기간)

두 번째 고객 응대가 첫 번째 고객 응대의 중지된 부분과 겹치더라도 중복으로 계산되지 않습니다. 두 번째 고객 응대는 중복에 대한 첫 번째 기준을 충족하지 못합니다. 첫 번째 고객 응대를 15:00에 중지하기 전에 13:00에 예약되었습니다. 중복이 아니기 때문에 언제 중지했는지에 관계없이 첫 번째 연락의 전체 원래 기간에 대한 요금이 청구됩니다.

시나리오 8: 중복 접촉이 있는 다중 안테나 지상국

Ground Station Anytown 1에서 150분 연락을 14:00에 시작하여 16:30에 종료하도록 예약합니다. 15:00에 CancelContact API를 호출하여 첫 번째 연락처를 중지합니다. CancelContact를 호출한 후 Ground Station Anytown 1에서 15:10부터 15:40까지 30분 연락처를 예약합니다. 나중에 15:30에 시작하여 17:00에 끝나는 Ground Station Anytown 1에서 또 다른 90분 연락을 예약합니다. Ground Station Anytown 1은 다중 안테나 지상국으로, 중복 접촉을 중복 시간과 동시에 실행할 수 있습니다.



결제 내역:

- 첫 번째 연락: 70분(실행된 60분 + 두 번째 연락이 시작되기 전 가동 중지 시간 10분)
- 두 번째 연락: 30분(전체 기간)
- 세 번째 연락처: 90분(전체 기간)

두 번째 및 세 번째 고객 응대는 모두 첫 번째 고객 응대를 중지한 후 예약했기 때문에 중복으로 계산됩니다. 첫 번째 고객 응대 중지(15:00)와 두 번째 고객 응대 시작(15:10) 사이의 10분 간격은 원래 고객 응대에 대해 청구되는 가동 중지 시간을 나타냅니다.

연락처 및 연락처 버전 관리 업데이트

AWS Ground Station 는 SCHEDULED, PREPASS 또는 고객 응대 상태의 PASS 고객 응대 업데이트를 지원합니다. [UpdateContact](#) API를 사용하여 방위각/상승, OEM 또는 TLE 추적 데이터를 포함한 연락처에 대한 에페메리스 재정의의 취소하고 다시 예약하지 않고도 지정할 수 있습니다. 이는 접촉 중에 안테나를 다른 위성으로 재작업해야 하는 지리동기식(GEO) 위성 작업 또는 포인팅 조정이 필요한 Launch and Early Operations(LEOPs)에 유용합니다.

연락처를 예약하거나 업데이트할 때마다 새 연락처 버전을 AWS Ground Station 생성합니다. 고객 응대 버전은 고객 응대에 대한 변경 기록을 제공하며 각 업데이트의 상태를 추적할 수 있습니다.

고객 응대 버전 관리 작동 방식

[ReserveContact](#)를 호출하면가 연락처의 첫 번째 버전(버전 1)을 AWS Ground Station 생성하고 응답versionId에를 반환합니다. 이후에 [UpdateContact](#)를 호출할 때마다 버전 번호가 증가하는 새 버전이 생성됩니다.

[DescribeContact](#) API는 응답 version 필드의 버전 정보를 포함하여 현재 ACTIVE 연락처 버전을 반환합니다. [ListContacts](#) API에는 각 연락처에 대한 버전 정보도 포함되어 있습니다.

연락처의 특정 버전을 보려면 [DescribeContactVersion](#) API를 사용합니다. 연락처의 모든 버전을 나열하려면 [ListContactVersions](#) API를 사용합니다.

연락처 업데이트

연락처가 예약됨, PREPASS 또는 PASS 상태일 때 [UpdateContact](#)를 호출할 수 있습니다. API는 다음 파라미터를 허용합니다.

- **contactId** - 업데이트할 연락처의 식별자입니다.
- **clientToken** - 요청이 한 번만 처리되도록 하는 멍등성 토큰입니다. 동일한 클라이언트 토큰으로 요청을 다시 시도하면는 업데이트를 다시 수행하지 않고 원래 응답을 AWS Ground Station 반환합니다. AWS SDKs 제공되지 않은 경우 클라이언트 토큰이 자동으로 생성됩니다.
- **trackingOverrides** - 연락처에 대한 새 추적 구성입니다. 여기에는 프로그램 트랙 설정(azimuth/elevation, TLE 또는 OEM 에페메리스)이 포함됩니다.

- **satelliteArn** - 연락처에 대한 위성의 ARN입니다. 프로그램 트랙 설정과 함께 대상 위성을 변경할 때 새 위성의 ARN을 제공합니다. 방위각/상승 각도에 대해 승인된 고객만이 값을 null로 설정할 수 있습니다. 다른 모든 고객은 연락처의 위성 ARN을 포함해야 합니다.

Important

UpdateContact API는 요청의 모든 파라미터를 적용합니다. 생략되거나 null로 명시적으로 설정된 모든 파라미터는 변경되지 않고 해당 값을 지우는 요청으로 처리됩니다. 예를 들어를 제공trackingOverrides했지만 satelliteArn를 생략하면 위성 ARN이 지워집니다. 각 업데이트 요청에 원하는 값을 모두 포함해야 합니다.

해당와 satelliteArn 함께 새를 제공하여 고객 응대 중에 대상 위성을 변경할 수 있습니다다trackingOverrides. 고객 응대 시작 및 종료 시간은이 API로 변경되지 않으므로 고객 응대 기간 동안 지상국에서 새 위성을 볼 수 있어야 합니다. 또한 새 위성을 지상국에 온보딩하고 미션 프로파일에 필요한 라이선스를 보유해야 합니다. 접촉의 미션 프로파일은 변경할 수 없으므로 위성 전환은 두 위성이 동일한 미션 프로파일을 사용하는 경우에만 적용됩니다.

Important

UpdateContact API는 연락의 시작 시간, 종료 시간 또는 미션 프로파일 변경을 지원하지 않습니다. 이러한 값을 변경하려면 연락처를 취소하고 새 연락처를 예약합니다. UpdateContact API는 위성 간 전환 또는 휘발성 데이터 업데이트와 같은 안테나 포인팅 구성을 재작업하도록 설계되었습니다.

Important

UpdateContact API는 [안테나 다운링크 복조 디코드 구성](#) 구성을 사용하는 미션 프로파일이 있는 연락처를 지원하지 않습니다. 이러한 연락처의 구성을 변경하려면 연락처를 취소하고 새 연락처를 예약합니다.

UpdateContact API는 contactId 및 새를 반환합니다versionId. 업데이트는 비동기적으로 처리됩니다. [DescribeContactVersion](#)을 사용하여 업데이트 상태를 확인합니다. AWS SDKs 및는 버전이 ACTIVE 또는 FAILED_TO_UPDATE 상태에 도달할 때까지 폴링하는 ContactUpdated웨이터를 AWS Command Line Interface 제공합니다.

Note

한 번에 하나의 업데이트만 진행할 수 있습니다. 최신 연락처 버전이 업데이트 중 상태인 경우 API는 `ConflictException`을 반환합니다. 다른 업데이트를 제출하기 전에 현재 업데이트가 `ACTIVE` 또는 `FAILED_TO_UPDATE` 상태에 도달할 때까지 기다립니다.

고객 응대 버전 상태

각 연락처 버전에는 다음 상태 중 하나가 있습니다.

Status	설명
업데이트 중	버전이 연락처에 적용되고 있습니다. 업데이트가 제출되어에서 처리 중입니다 AWS Ground Station.
ACTIVE	버전은 연락처에 대한 현재 활성 구성입니다. 지상국이이 버전의 설정을 사용하고 있습니다.
SUPERSEDED	버전은 이전에 활성화되었지만 최신 버전으로 대체되었습니다.
FAILED_TO_UPDATE	업데이트를 적용할 수 없습니다. 연락처가 이전 활성 버전으로 돌아갑니다. 자세한 내용은 <code>failureCodes</code> 및 <code>failureMessage</code> 필드를 확인하세요.

코드 예제

다음 예제에서는 AWS SDK for Python(Boto3)과 함께 고객 응대 버전 관리 APIs를 사용하는 방법을 보여줍니다.

예: 연락처 업데이트

다음 예시에서는 새 추적 재정의로 연락처를 업데이트하고 Boto3 `ContactUpdated` 웨이터를 사용하여 업데이트가 완료될 때까지 기다립니다.

```
import boto3
import uuid

# Create AWS Ground Station client
```

```

ground_station_client = boto3.client("groundstation")

# The contact ID of an existing scheduled contact to update
contact_id = "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"

# Generate a unique client token for idempotency.
# If you retry the same request with the same client token,
# the API returns the same response without creating a duplicate version.
client_token = str(uuid.uuid4())

# Update the contact to use a different TLE ephemeris for tracking.
# The UpdateContact API applies all parameters in the request.
# Any parameter set to null is treated as a request to clear that value,
# not to leave it unchanged. Include all desired values in each request.
print(f"Updating contact {contact_id}...")

update_response = ground_station_client.update_contact(
    contactId=contact_id,
    clientToken=client_token,
    satelliteArn="arn:aws:groundstation::111122223333:satellite/a88611b0-f755-404e-
b60d-57d8aEXAMPLE",
    trackingOverrides={
        "programTrackSettings": {
            "tle": {"ephemerisId": "b2c3d4e5-6789-01ab-cdef-EXAMPLE22222"}
        }
    },
)

contact_id = update_response["contactId"]
version_id = update_response["versionId"]
print(f"Update submitted. Contact: {contact_id}, Version: {version_id}")

# Wait for the update to complete using the ContactUpdated waiter.
# The waiter polls DescribeContactVersion until the version reaches
# ACTIVE (success) or FAILED_TO_UPDATE (failure) status.
# The waiter raises WaiterError if the version reaches FAILED_TO_UPDATE
# or if MaxAttempts is exceeded, so we use try/except to handle both cases.
print("Waiting for update to complete...")

from botocore.exceptions import WaiterError

waiter = ground_station_client.get_waiter("contact_updated")

try:

```

```

waiter.wait(
    contactId=contact_id,
    versionId=version_id,
    WaiterConfig={
        "Delay": 5,
        "MaxAttempts": 180,
    },
)
print(f"Contact updated successfully. Version {version_id} is now active.")
except WaiterError as e:
    # WaiterError is raised when the version reaches FAILED_TO_UPDATE
    # or when MaxAttempts is exceeded. Retrieve the current version to inspect.
    version_response = ground_station_client.describe_contact_version(
        contactId=contact_id,
        versionId=version_id,
    )
    version_status = version_response["version"]["status"]
    if version_status == "FAILED_TO_UPDATE":
        failure_codes = version_response["version"].get("failureCodes", [])
        failure_message = version_response["version"].get("failureMessage", "")
        print(f"Update failed. Codes: {failure_codes}, Message: {failure_message}")
    else:
        print(f"Waiter timed out. Current version status: {version_status}. Error:
{e}")

```

예: 연락처 버전 설명

다음 예시에서는 상태, 구성 및 실패 정보를 포함하여 특정 연락처 버전의 세부 정보를 검색합니다.

```

import boto3

# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

contact_id = "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
version_id = 2

# Describe a specific version of a contact.
# Use this API to check the status of an update or to view the
# configuration that was active at a specific point in time.
print(f"Describing version {version_id} of contact {contact_id}...")

response = ground_station_client.describe_contact_version(

```

```

        contactId=contact_id,
        versionId=version_id,
    )

# Display version details
version = response["version"]
print(f"Version ID: {version['versionId']}")
print(f"Status: {version['status']}")
print(f"Created: {version.get('created', 'N/A')}")

if version.get("activated"):
    print(f"Activated: {version['activated']}")

if version.get("superseded"):
    print(f"Superseded: {version['superseded']}")

# Display contact details for this version
print(f"\nContact ID: {response['contactId']}")
print(f"Contact Status: {response['contactStatus']}")
print(f"Ground Station: {response['groundStation']}")
print(f"Start Time: {response['startTime']}")
print(f"End Time: {response['endTime']}")

if response.get("satelliteArn"):
    print(f"Satellite ARN: {response['satelliteArn']}")

if response.get("trackingOverrides"):
    print(f"Tracking Overrides: {response['trackingOverrides']}")

# Check for failure details if the version failed to update
if version["status"] == "FAILED_TO_UPDATE":
    failure_codes = version.get("failureCodes", [])
    failure_message = version.get("failureMessage", "")
    print(f"\nFailure Codes: {failure_codes}")
    print(f"Failure Message: {failure_message}")

```

예: 연락처 버전 나열

다음 예제에서는 페이지 매김을 사용하여 대규모 결과 집합을 처리하여 전체 변경 기록을 볼 수 있는 연락처의 모든 버전을 나열합니다.

```
import boto3
```

```
# Create AWS Ground Station client
ground_station_client = boto3.client("groundstation")

contact_id = "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"

# List all versions of a contact to view the full history of changes.
# Results are paginated. Use the nextToken to retrieve additional pages.
print(f"Listing versions for contact {contact_id}...")

paginator = ground_station_client.get_paginator("list_contact_versions")
page_iterator = paginator.paginate(
    contactId=contact_id,
    PaginationConfig={
        "MaxItems": 100,
        "PageSize": 20,
    },
)

for page in page_iterator:
    for version in page["contactVersionsList"]:
        version_id = version["versionId"]
        status = version["status"]
        created = version.get("created", "N/A")

        print(f"  Version {version_id}: status={status}, created={created}")

        if version.get("activated"):
            print(f"    Activated: {version['activated']}")

        if version.get("superseded"):
            print(f"    Superseded: {version['superseded']}")

        if status == "FAILED_TO_UPDATE":
            failure_codes = version.get("failureCodes", [])
            failure_message = version.get("failureMessage", "")
            print(f"      Failure Codes: {failure_codes}")
            print(f"      Failure Message: {failure_message}")

        if status == "UPDATING":
            print(f"      Update is currently in progress.")
```

고려 사항

- 고객 응대 버전 관리 기능이 도입되기 전에 생성된 고객 응대에는 버전 정보가 없습니다. 이러한 연락처에 대해 [DescribeContactVersion](#) 또는 [ListContactVersions](#)를 호출하면 `ResourceNotFoundException`이 반환됩니다.
- 진행 중인 고객 응대의 추적 재정의 업데이트하면 안테나가 새 포인팅 구성으로 조정되는 동안 전환 기간이 짧아집니다. 이 시간 동안 신호 수신 또는 전송이 중단될 수 있습니다.
- 연락처 버전은 삭제할 수 없습니다. [ListContactVersions](#)를 사용하여 연락처에 대한 전체 변경 기록을 봅니다.
- `UpdateContact` API는 연락처의 예약 리전에서만 호출할 수 있습니다.

AWS Ground Station 디지털 트윈 기능 사용

의 디지털 트윈 기능은 위성 미션 관리와 명령 및 제어 소프트웨어를 테스트하고 통합할 수 있는 환경을 AWS Ground Station에 제공합니다. 디지털 트윈 기능을 사용하면 프로덕션 안테나 용량을 사용하지 않고도 예약, 구성 확인 및 적절한 오류 처리를 테스트할 수 있습니다. 디지털 트윈 기능과의 AWS Ground Station 통합을 테스트하면 위성 작업을 원활하게 관리하는 시스템의 기능에 대한 신뢰도를 높일 수 있습니다. 또한 프로덕션 용량을 사용하거나 스펙트럼 라이선스를 요구하지 않고 AWS Ground Station APIs를 테스트할 수 있습니다.

시작하려면 아래 디지털 트윈 기능에 대한 온보딩을 [위성 온보딩](#) 요청합니다. 위성이 디지털 트윈 기능에 온보딩되면 디지털 트윈 지상국에 대한 연락을 예약할 수 있습니다. 액세스할 수 있는 지상국 목록은 AWS SDK [ListGroundStations](#) 응답을 통해 검색할 수 있습니다. 디지털 트윈 지상국에는 나열된 지상국의 정확한 사본 [AWS Ground Station 위치](#)으로, 지상국 이름 “디지털 트윈”에 대한 수정 접두사가 붙습니다. 여기에는 사이트 마스크 및 실제 GPS 좌표를 포함하되 이에 국한되지 않는 안테나 기능과 메타데이터가 포함됩니다. 현재 디지털 트윈 기능에는 설명된 대로 데이터 전송을 지원하지 않습니다. [데이터 흐름 작업](#). 또한 설명된 원격 측정 전송 [원격 측정 작업](#)은 디지털 트윈 지상국에서 지원되지 않습니다.

온보딩되면 디지털 트윈 기능에는 설명된 대로 프로덕션 서비스와 동일한 Amazon EventBridge 이벤트 및 API 응답을 내보냅니다. [이벤트 AWS Ground Station 로 자동화](#). 이러한 이벤트를 통해 구성 및 데이터 흐름 엔드포인트 그룹을 미세 조정할 수 있습니다.

AWS Ground Station 전용 안테나

AWS Ground Station 전용 안테나는 AWS 사용자를 대신하여 관리하는 사용자 지정 안테나 시스템입니다. 안테나 시간을 다른 고객과 공유하는 퍼블릭 AWS Ground Station 안테나와 달리 전용 안테나는 사양에 맞게 구축된 안테나에 대한 전용 액세스를 제공합니다. 하나 이상의 전용 안테나가 있을 수 있습니다. 전용 안테나는 글로벌 AWS 네트워크에 연결되며, 퍼블릭 안테나와 함께 사용하는 것과 동일한 AWS Ground Station APIs 및 워크플로를 사용하여 상호 작용하고 안테나 사용률에 대한 향상된 가시성을 제공합니다.

전용 안테나란?

전용 안테나는 조직에 맞게 사용자 지정 빌드되고에서 완벽하게 관리하는 물리적 안테나 시스템입니다 AWS. 하나 이상의 전용 안테나가 있을 수 있습니다. 전용 안테나는 다음과 같은 점에서 퍼블릭 AWS Ground Station 안테나와 다릅니다.

- 사용자 지정 빌드 - 전용 안테나는 사양에 맞게 빌드됩니다. 전용 안테나의 기능은에 설명된 대로 퍼블릭 안테나의 기능에 국한되지 않습니다 [AWS Ground Station 사이트 기능](#).
- 유연한 위치 - 전용 안테나는 기존 AWS Ground Station 안테나 위치로 제한되지 않습니다. 전용 안테나를 현재 사용할 수 없는 AWS 리전을 포함하여 기존 리전 AWS Ground Station 에 연결할 수 있습니다. AWS 와 협력하여 요구 사항에 맞는 위치와 리전을 결정할 수 있습니다.
- 전용 액세스 - 고객 응대별로 안테나 시간을 다른 AWS Ground Station 고객과 공유하는 대신 안테나에 대한 전용 액세스 권한을 가집니다.

전용 AWS Ground Station 안테나를 포함한 모든 안테나에서 완벽하게 관리합니다 AWS. 여기에는 글로벌 AWS 네트워크에 대한 유지 관리 및 연결이 포함됩니다. 퍼블릭 안테나와 함께 사용하는 것과 동일한 AWS Ground Station APIs 및 워크플로를 사용하여 전용 안테나와 상호 작용합니다. 연락을 예약하고, 미션 프로파일을 구성하고, 동일한 방식으로 데이터를 전송합니다.

전용 안테나 계약을 보유한 고객이 온보딩할 계정을 선택하는 여러 AWS 계정에서 전용 안테나를 공유할 수 있습니다. 온보딩된 각 계정은 안테나에서 고객 응대를 독립적으로 예약할 수 있으며 안테나를 공유하는 모든 계정의 예약을 볼 수 있습니다.

전용 안테나에 대해 자세히 알아보거나 시작하려면 AWS Support 통해 문의하세요 [AWS Support Center Console](#).

예약 가시성 향상

전용 안테나에 대해 [ListGroundStationReservations](#) API를 사용하면 퍼블릭 안테나에서 사용할 수 없는 추가 정보가 표시됩니다. 다음 표에는 퍼블릭 AWS Ground Station 안테나와 비교한 전용 안테나용 [ListGroundStationReservations](#) API의 동작 차이가 요약되어 있습니다.

동작	전용 안테나	퍼블릭 안테나
자체 연락처	다음에 포함된 전체 세부 정보가 표시됩니다. <code>contactId</code>	다음에 포함된 전체 세부 정보가 표시됩니다. <code>contactId</code>
다른 계정의 연락처	시간 슬롯으로만 표시, 없이 표시 <code>contactId</code>	표시되지 않음
유지 관리 기간	PLANNED 또는 로 표시 UNPLANNED <code>maintenanceType</code>	표시되지 않음

유지 관리 기간은 안테나를 위성 통신에 사용할 수 없는 기간을 나타냅니다. `maintenanceType` 필드는 유지 관리가 PLANNED 또는 였는지 여부를 나타냅니다. UNPLANNED. 계획되지 않은 유지 관리가 예약되면 유지 관리 기간과 겹치는 고객 응대가에 의해 취소될 수 있습니다 AWS Ground Station.

전용 안테나를 공유하는 다른 AWS 계정의 연락처를 보는 경우 예약에는 시간 슬롯과 안테나 정보가 포함되지만 `contactId`는 포함되지 않습니다.

Important

향상된 예약 가시성은 전용 안테나에만 적용됩니다. 퍼블릭 AWS Ground Station 안테나를 사용하면 다른 고객과 동일한 가시성을 확보할 수 있습니다. 퍼블릭 안테나에서는 유지 관리 기간이나 다른 계정의 예약이 표시되지 않습니다.

예약 목록에 대한 자세한 내용은 섹션을 참조하세요 [지상국 예약 보기](#).

관련 리소스

- [지상국 예약 보기](#)

- [AWS Ground Station 위치](#)
- [AWS Ground Station 사이트 기능](#)
- AWS Ground Station API 참조의 [ListGroundStationReservations](#)

를 사용한 모니터링 이해 AWS Ground Station

모니터링은 AWS Ground Station의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. AWS는 다음과 같은 모니터링 도구를 제공하여 모니터링 AWS Ground Station, 문제 발생 시 보고, 적절한 경우 자동 조치를 취합니다.

- Amazon EventBridge Events는 AWS 리소스의 변경 사항을 설명하는 시스템 이벤트 스트림을 거의 실시간으로 제공합니다. EventBridge 이벤트는 특정 이벤트를 감시하고 이러한 이벤트가 발생할 때 다른 AWS 서비스에서 자동화된 작업을 트리거하는 규칙을 작성할 수 있으므로 자동화된 이벤트 기반 컴퓨팅을 지원합니다. EventBridge Events에 대한 자세한 내용은 [Amazon EventBridge 사용 설명서](#)를 참조하세요.
- AWS CloudTrail는 AWS 계정에서 또는 계정을 대신하여 수행된 API 호출 및 관련 이벤트를 캡처하고 사용자가 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 호출한 사용자 및 계정 AWS, 호출이 수행된 소스 IP 주소, 호출이 발생한 시기를 식별할 수 있습니다. 에 대한 자세한 내용은 [AWS CloudTrail 사용 설명서](#)를 AWS CloudTrail참조하세요.
- Amazon CloudWatch 지표는 사용 시 예약된 연락처에 대한 지표를 캡처합니다 AWS Ground Station. CloudWatch Metrics를 사용하면 채널, 편광 및 위성 ID를 기반으로 데이터를 분석하여 접축의 신호 강도 및 오류를 식별할 수 있습니다. 자세한 내용은 [Amazon CloudWatch 지표 사용](#)을 참조하세요.
- [AWS 사용자 알림](#)를 사용하여 AWS Ground Station 이벤트에 대한 알림을 받을 전송 채널을 설정할 수 있습니다. 이벤트가 지정한 규칙과 일치하면 알림을 받습니다. 이메일, [채팅 애플리케이션의 Amazon Q Developer](#) 채팅 알림 또는 [AWS Console Mobile Application](#) 푸시 알림을 비롯한 여러 채널을 통해 이벤트에 대한 알림을 받을 수 있습니다. AWS 콘솔 [알림 센터에서](#) 알림을 볼 수도 있습니다. 사용자 알림 지원 집계를 통해 특정 이벤트 중에 수신하는 알림 수를 줄일 수 있습니다.

다음 주제를 사용하여 AWS Ground Station을 모니터링합니다.

주제

- [이벤트 AWS Ground Station 로 자동화](#)
- [를 사용하여 AWS Ground Station API 호출 로깅 AWS CloudTrail](#)
- [Amazon CloudWatch를 사용하여 지표 보기](#)

이벤트 AWS Ground Station 로 자동화

Note

이 문서에서는 전체적으로 “이벤트”라는 용어를 사용합니다. CloudWatch Events와 EventBridge는 기본 서비스 및 API가 동일합니다. 두 서비스를 사용하면 수신 이벤트를 확인한 후 처리 대상으로 라우팅하는 규칙을 생성할 수 있습니다.

이벤트를 사용하면 AWS 서비스를 자동화하고 애플리케이션 가용성 문제 또는 리소스 변경과 같은 시스템 이벤트에 자동으로 대응할 수 있습니다. AWS 서비스의 이벤트는 거의 실시간으로 전달됩니다. 원하는 이벤트만 표시하도록 간단한 규칙을 작성한 후 규칙과 일치하는 이벤트 발생 시 실행할 자동화 작업을 지정할 수 있습니다. 자동으로 트리거될 수 있는 작업 중 일부는 다음과 같습니다.

- AWS Lambda 함수 호출
- Amazon EC2 Run Command 호출
- Amazon Kinesis Data Streams로 이벤트 릴레이
- AWS Step Functions 상태 시스템 활성화
- SNS 주제 또는 Amazon SQS 대기열 알림

에서 이벤트를 사용하는 몇 가지 예는 다음과 AWS Ground Station 같습니다.

- Lambda 함수를 간접적으로 호출하여 이벤트 상태를 기반으로 Amazon EC2 인스턴스의 시작 및 종지를 자동화합니다.
- 접촉의 상태가 변경될 때마다 Amazon SNS 주제에 게시합니다. 이러한 주제는 접촉의 시작 또는 끝 부분에 이메일 공지를 보내도록 설정할 수 있습니다.

자세한 내용은 [Amazon EventBridge Events 사용 설명서](#)를 참조하세요.

AWS Ground Station 이벤트 유형

Note

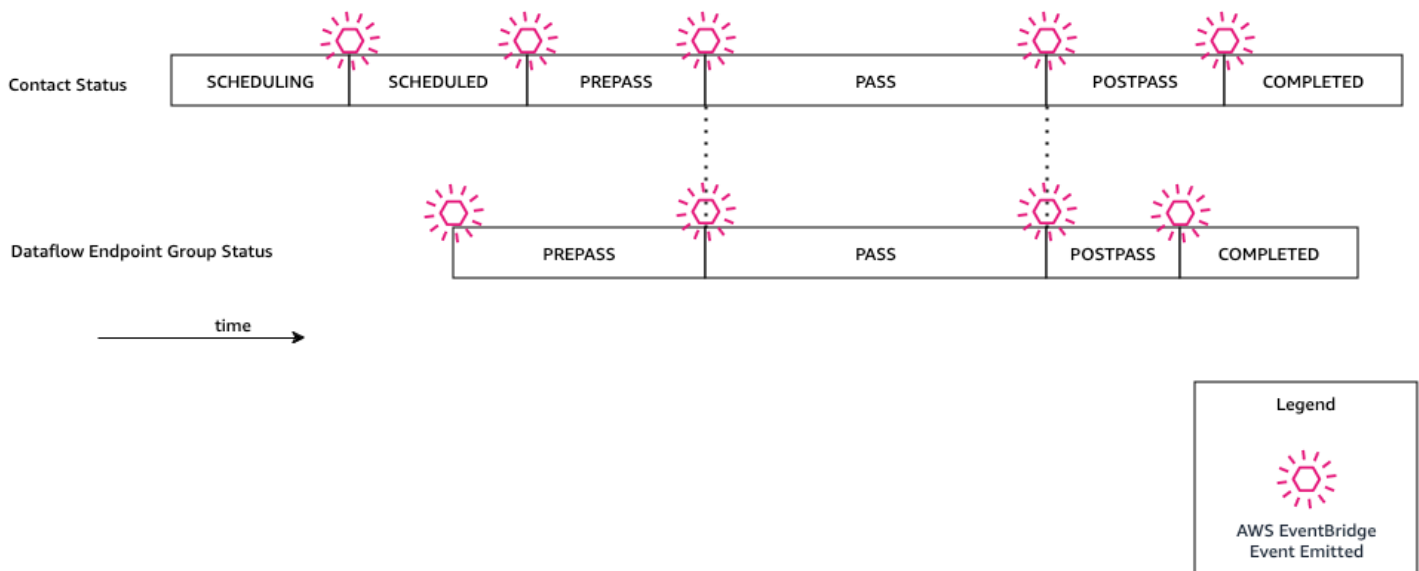
AWS Ground Station에서 생성되는 모든 이벤트는 “소스”의 값으로 “aws.groundstation”을 사용합니다.

AWS Ground Station 는 상태 변경과 관련된 이벤트를 내보내 자동화를 사용자 지정하는 기능을 지원합니다. 현재는 고객 응대 상태 변경 이벤트, 데이터 흐름 엔드포인트 그룹 변경 이벤트 및 에페메리스 상태 변경 이벤트를 AWS Ground Station 지원합니다. 다음 섹션에서는 각 유형에 대한 자세한 정보를 제공합니다.

연락 이벤트 타임라인

AWS Ground Station 는 고객 응대 상태가 변경될 때 이벤트를 내보냅니다. 이러한 상태 변경의 정의와 상태 자체의 의미에 대한 자세한 내용은 [고객 응대 수명 주기 이해](#) 섹션을 참조하세요. 고객 응대에서 사용되는 모든 데이터 흐름 엔드포인트 그룹에는 독립적인 이벤트 세트가 있으며, 이 이벤트 세트를 내보내집니다. 동일한 기간 동안 데이터 흐름 엔드포인트 그룹에 대한 이벤트도 내보냅니다. 사전 통과 및 사후 통과 이벤트의 정확한 시간은 미션 프로파일 및 데이터 흐름 엔드포인트 그룹을 설정할 때 사용자가 구성할 수 있습니다.

다음 다이어그램은 공칭 고객 응대 및 관련 데이터 흐름 엔드포인트 그룹에 대해 생성된 상태 및 이벤트를 보여줍니다.



Ground Station 접촉 상태 변경

예정된 고객 응대의 상태가 변경될 때 특정 작업을 수행하려면 작업을 자동화하는 규칙을 설정할 수 있습니다. 이 기능은 접촉의 상태 변경에 대한 알림을 수신하려는 경우에 유용합니다. 이러한 이벤트를 수신할 때 변경하려는 경우 미션 프로파일의 [contactPrePassDurationSeconds](#) 및 [contactPostPassDurationSeconds](#)를 수정할 수 있습니다. 이벤트는 접촉이 예약된 리전으로 전송됩니다.

다음은 예제 이벤트입니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123",
  "account": "123456789012",
  "time": "2019-05-30T17:40:30Z",
  "region": "us-west-2",
  "source": "aws.groundstation",
  "resources": [
    "arn:aws:groundstation:us-west-2:123456789012:contact/11111111-1111-1111-1111-111111111111"
  ],
  "detailType": "Ground Station Contact State Change",
  "detail": {
    "contactId": "11111111-1111-1111-1111-111111111111",
    "groundstationId": "Ground Station 1",
    "missionProfileArn": "arn:aws:groundstation:us-west-2:123456789012:mission-profile/11111111-1111-1111-1111-111111111111",
    "satelliteArn":
      "arn:aws:groundstation::123456789012:satellite/11111111-1111-1111-1111-111111111111",
    "contactStatus": "PASS"
  }
}
```

contactStatus에 대해 가능한 값은 [the section called “AWS Ground Station 고객 응대 상태”](#)에 정의됩니다.

Ground Station 데이터 흐름 엔드포인트 그룹 상태 변경

데이터 흐름 엔드포인트 그룹이 데이터 수신에 사용 중일 때 작업을 수행하고 싶으면 이 작업을 자동화하도록 규칙을 설정할 수 있습니다. 이렇게 하면 데이터 흐름 엔드포인트 그룹의 상태 변경에 따라 다른 작업을 수행할 수 있습니다. 이러한 이벤트를 수신할 때 변경하려면 [contactPrePassDurationSeconds](#)와 [contactPostPassDurationSeconds](#)가 다른 데이터 흐름 엔드포인트 그룹을 사용합니다. 이 이벤트는 데이터 흐름 엔드포인트 그룹의 리전으로 전송됩니다.

아래에 예제가 나와 있습니다.

```
{
  "version": "0",
  "id": "01234567-0123-0123",
  "account": "123456789012",
```

```

    "time": "2019-05-30T17:40:30Z",
    "region": "us-west-2",
    "source": "aws.groundstation",
    "resources": [
      "arn:aws:groundstation:us-west-2:123456789012:dataflow-endpoint-group/bad957a8-1d60-4c45-a92a-39febd98921d",
      "arn:aws:groundstation:us-west-2:123456789012:contact/98ddd10f-f2bc-479c-bf7d-55644737fb09",
      "arn:aws:groundstation:us-west-2:123456789012:mission-profile/c513c84c-eb40-4473-88a2-d482648c9234"
    ],
    "detailType": "Ground Station Dataflow Endpoint Group State Change",
    "detail": {
      "dataflowEndpointGroupId": "bad957a8-1d60-4c45-a92a-39febd98921d",
      "groundstationId": "Ground Station 1",
      "contactId": "98ddd10f-f2bc-479c-bf7d-55644737fb09",
      "dataflowEndpointGroupArn": "arn:aws:groundstation:us-west-2:680367718957:dataflow-endpoint-group/bad957a8-1d60-4c45-a92a-39febd98921d",
      "missionProfileArn": "arn:aws:groundstation:us-west-2:123456789012:mission-profile/c513c84c-eb40-4473-88a2-d482648c9234",
      "dataflowEndpointGroupState": "PREPASS"
    }
  }
}

```

dataflowEndpointGroupState에서 가능한 상태로는 PREPASS, PASS, POSTPASS 및 COMPLETED가 있습니다.

에페메리스 이벤트

Ground Station 에페메리스 상태 변경

에페메리스가 상태를 변경할 때 특정 작업을 수행하려는 경우 이 작업을 자동화하도록 규칙을 설정할 수 있습니다. 이렇게 하면 상태를 변경하는 에페메리스에 따라 다양한 작업을 수행할 수 있습니다. 예를 들어, 에페메리스의 검증이 완료되었는데 지금은 ENABLED일 때 작업을 수행할 수 있습니다. 이 이벤트에 대한 알림은 에페메리스가 업로드된 리전으로 전송됩니다.

아래에 예제가 나와 있습니다.

```

{
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "Ground Station Ephemeris State Change",

```

```

    "source": "aws.groundstation",
    "account": "123456789012",
    "time": "2019-12-03T21:29:54Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:groundstation::123456789012:satellite/10313191-c9d9-4ecb-a5f2-bc55cab050ec",
        "arn:aws:groundstation::123456789012:ephemeris/111111-cccc-bbbb-a555-bcccca005000"
    ],
    "detail": {
        "ephemerisStatus": "ENABLED",
        "ephemerisId": "111111-cccc-bbbb-a555-bcccca005000",
        "satelliteId": "10313191-c9d9-4ecb-a5f2-bc55cab050ec"
    }
}

```

ephemerisStatus에서 가능한 상태로는 ENABLED, VALIDATING, INVALID, ERROR, DISABLED, EXPIRED가 있습니다.

를 사용하여 AWS Ground Station API 호출 로깅 AWS CloudTrail

AWS Ground Station 는 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다 AWS Ground Station. CloudTrail은 AWS Ground Station 에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 AWS Ground Station 콘솔의 호출과 AWS Ground Station API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하면 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다 AWS Ground Station. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail 에서 수집한 정보를 사용하여 수행된 요청, 요청이 수행된 AWS Ground Station IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 설명은 [AWS CloudTrail 사용자 가이드](#)를 참조하십시오.

AWS Ground Station CloudTrail의 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. 에서 활동이 발생하면 AWS Ground Station 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 설명은 [CloudTrail 이벤트 기록으로 이벤트 보기](#)를 참조하세요.

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

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에서 Amazon SNS 알림 구성](#)
- [여러 리전으로부터 CloudTrail 로그 파일 받기](#) 및 [여러 계정으로부터 CloudTrail 로그 파일 받기](#)

모든 AWS Ground Station 작업은 CloudTrail에서 로깅되며 [AWS Ground Station API 참조](#)에 문서화됩니다. 예를 들어 ReserveContact, CancelContact, ListConfigs 작업을 직접 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

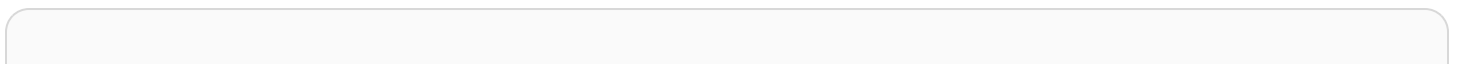
자세한 설명은 [CloudTrail userIdentity 요소](#)를 참조하세요.

AWS Ground Station 로그 파일 항목 이해

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 ReserveContact 작업을 보여주는 CloudTrail 로그 항목이 나타냅니다.

예제: ReserveContact



```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:sts::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-05-15T21:11:59Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Alice",
        "accountId": "123456789012",
        "userName": "Alice"
      }
    }
  },
  "eventTime": "2019-05-15T21:14:37Z",
  "eventSource": "groundstation.amazonaws.com",
  "eventName": "ReserveContact",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Mozilla/5.0 Gecko/20100101 Firefox/123.0",
  "requestParameters": {
    "satelliteArn":
"arn:aws:groundstation::123456789012:satellite/11111111-2222-3333-4444-555555555555",
    "groundStation": "Ohio 1",
    "startTime": 1558356107,
    "missionProfileArn": "arn:aws:groundstation:us-east-2:123456789012:mission-
profile/11111111-2222-3333-4444-555555555555",
    "endTime": 1558356886
  },
  "responseElements": {
    "contactId": "11111111-2222-3333-4444-555555555555"
  },
  "requestID": "11111111-2222-3333-4444-555555555555",
  "eventID": "11111111-2222-3333-4444-555555555555",
  "readOnly": false,
  "eventType": "AwsApiCall",

```

```
"recipientAccountId": "11111111-2222-3333-4444-555555555555"
}
```

Amazon CloudWatch를 사용하여 지표 보기

고객 응대 중에는 데이터를 AWS Ground Station 자동으로 캡처하여 분석을 위해 CloudWatch로 전송합니다. Amazon CloudWatch 콘솔에서 데이터를 볼 수 있습니다. 액세스 및 CloudWatch 지표에 대한 자세한 내용은 [Amazon CloudWatch 지표 사용](#)을 참조하세요.

AWS Ground Station 원격 측정 기능을 사용하여 연락 중에 거의 실시간에 가까운 지표를 수신할 수도 있습니다. CloudWatch 지표는 거의 실시간으로 사용할 수 없으며 전송이 지연될 수 있습니다. 또한 CloudWatch는 1초 동안 지표를 집계하여 잠재적으로 데이터 세분화를 줄일 수 있습니다. 원격 측정 기능은 개별 지표를 제공하고 거의 실시간으로 AWS 계정에 직접 제공합니다. 자세한 내용은 [원격 측정 작업](#) 섹션을 참조하세요.

Important

AWS Ground Station 는 고객 응대가 예약된 AWS 리전이 아닌 고객 응대의 지상국 위치와 연결된 AWS 리전으로 CloudWatch 지표를 내보냅니다. 연락처에 대한 지표를 보려면 지상국 리전의 CloudWatch에 액세스해야 합니다. 각 지상국 위치와 연결된 리전에 대한 AWS 자세한 내용은 섹션을 참조하세요 [지상국 위치의 AWS 리전 찾기](#). 연락처를 예약하는 리전에서 원격 측정 데이터를 수신하려면 AWS Ground Station 원격 측정 기능을 사용할 수 있습니다. 자세한 내용은 [원격 측정 작업](#) 섹션을 참조하세요.

AWS Ground Station 지표 및 차원

사용할 수 있는 지표는 무엇입니까?

다음 지표는에서 사용할 수 있습니다 AWS Ground Station.

Note

내보내는 특정 지표는 사용 중인 AWS Ground Station 기능에 따라 달라집니다. 구성에 따라 아래 지표의 하위 집합만 내보내질 수 있습니다.

지표	측정치 차원	설명
AzimuthAngle	Satelliteld	안테나의 방위각. 진북쪽은 0도이고 동쪽은 90도입니다. 단위: 도
BitErrorRate	채널, 편광, Satelliteld	지정된 비트 전송 횟수에서 비트에 대한 오류율입니다. 비트 오류는 노이즈, 왜곡 또는 간섭으로 인해 발생합니다. 단위: 단위 시간당 비트 오류
BlockErrorRate	채널, 편광, Satelliteld	지정된 수신 블록의 수에서 블록의 오류율입니다. 블록 오류는 간섭으로 인해 발생합니다. 단위: 잘못된 블록/총 블록 수
CarrierFrequencyRecovery_Cn0	범주, 구성, Satelliteld	단위 대역폭당 캐리어 대 잡음 밀도 비율. 단위: 데시벨-헤르츠(dB-Hz)
CarrierFrequencyRecovery_Locked	범주, 구성, Satelliteld	복조기 캐리어 주파수 복구 루프가

지표	측정치 차원	설명
		잠겨 있으면 1로 설정하고 잠금 해제되면 0으로 설정합니다. 단위: 단위 없음
CarrierFrequencyRecovery_OffsetFrequency_Hz	범주, 구성, Satelliteld	추정된 신호 중심과 이상적인 중심 주파수 사이의 오프셋. 이는 우주선과 안테나 시스템 사이의 도플러 시프트 및 로컬 오실레이터 오프셋으로 인해 발생합니다. 단위: 헤르츠(Hz)
ElevationAngle	Satelliteld	안테나의 고도 각도. 수평선은 0도이고 천정은 90도입니다. 단위: 도
Es/N0	채널, 편광, Satelliteld	심볼당 에너지 대 노이즈 파워 스펙트럼 밀도의 비율. 단위: 데시벨(dB)

지표	측정치 차원	설명
ReceivedPower	편광, Satelliteld	복조기/디코더에서 측정된 신호 강도입니다. 단위: 밀리와트에 상응하는 데시벨 (dBm)
SymbolTimingRecovery_ErrorVectorMagnitude	범주, 구성, Satelliteld	수신된 심볼과 이상적인 정상점 사이의 오차 벡터 크기. 단위: 백분율
SymbolTimingRecovery_Locked	범주, 구성, Satelliteld	복조기 심볼 타이밍 복구 루프가 잠겨 있으면 1로 설정하고, 잠금이 해제되면 0으로 설정합니다. 단위: 단위 없음
SymbolTimingRecovery_OffsetSymbolRate	범주, 구성, Satelliteld	추정된 심볼 레이트와 이상적인 신호 심볼 레이트 사이의 오프셋. 이는 우주선과 안테나 시스템 사이의 도플러 시프트 및 로컬 오실레이터 오프셋으로 인해 발생합니다. 단위: 기호/초

어떤 차원에 사용되나요 AWS Ground Station?

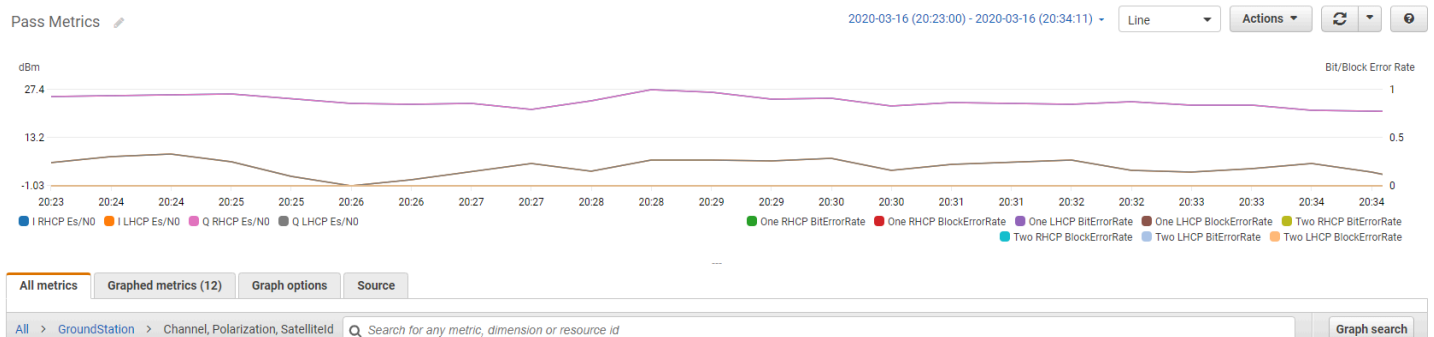
다음 차원을 사용하여 AWS Ground Station 데이터를 필터링할 수 있습니다.

차원	설명
Category	복조 또는 디코딩.
Channel	각 접촉에 대한 채널에는 1, 2, I(동상) 및 Q(직교)가 포함됩니다.
Config	안테나 다운링크 데모 디코딩 구성 ARN.
Polarization	각 접촉에 대한 편광에는 LHCP(Left Hand Circular Polarized) 또는 RHCP(Right Hand Circular Polarized)가 포함됩니다.
SatelliteId	위성 ID에는 접촉에 대한 위성의 ARN이 포함됩니다.

지표 보기

그래프로 표시된 지표를 볼 때 집계 창에서 지표가 표시되는 방법이 결정된다는 점에 유의해야 합니다. 접촉의 각 지표는 데이터를 받은 후 3시간 동안 초당 데이터로 표시될 수 있습니다. 데이터는 3시간이 경과한 후 CloudWatch 지표에 의해 분당 데이터로 집계됩니다. 초당 데이터에 대한 지표를 확인해야 하는 경우 데이터를 수신한 후 3시간 이내에 데이터를 보거나 CloudWatch 지표 외부에서 유지하는 것이 좋습니다. CloudWatch 보존에 대한 자세한 내용은 [Amazon CloudWatch 개념 - 지표 보존](#)을 참조하세요.

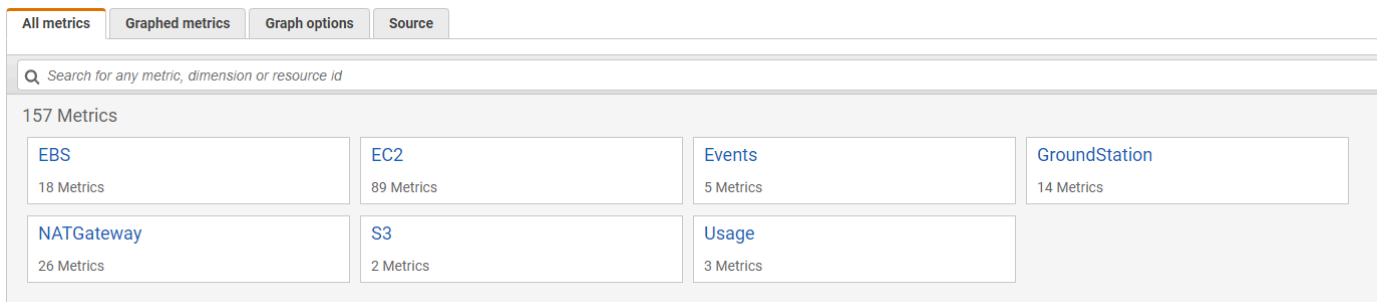
또한 처음 60초 내에 캡처된 데이터에는 의미 있는 지표를 생성하기에 충분한 정보가 포함되지 않으며 표시되지 않을 수 있습니다. 의미 있는 지표를 보려면 60초 후에 데이터를 보는 것이 좋습니다.



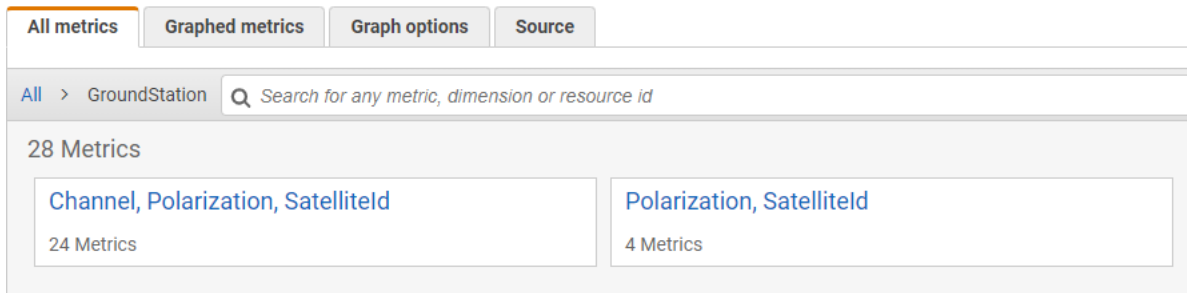
CloudWatch의 지표 그래프 작성 AWS Ground Station 에 대한 자세한 내용은 [지표 그래프 작성을 참조하세요](#).

콘솔을 사용한 메트릭 확인

1. 지상국 위치와 연결된 AWS 리전을 결정합니다. AWS Ground Station 는 연락처의 지상국 위치와 연결된 리전에서 CloudWatch 지표를 내보냅니다. 지상국 위치 및 관련 AWS 리전 목록은 섹션을 참조하세요 [지상국 위치의 AWS 리전 찾기](#).
2. [CloudWatch 콘솔](#)을 엽니다.
3. 탐색 창에서 지표를 선택합니다.
4. GroundStation 네임스페이스를 선택합니다.



5. 원하는 지표 차원(예: 채널, 편광, SatelliteId)을 선택합니다.



6. 모든 지표 탭에 네임스페이스의 해당 측정기준에 대한 모든 지표가 표시됩니다. 다음을 수행할 수 있습니다.
 - a. 테이블을 정렬하려면 열 머리글을 사용합니다.
 - b. 지표를 그래프로 표시하려면 지표와 연결된 확인란을 선택합니다. 모든 지표를 선택하려면 테이블의 제목 행에서 확인란을 선택합니다.
 - c. 리소스로 필터링하려면 리소스 ID를 선택한 후 검색에 추가를 선택합니다.
 - d. 지표로 필터링하려면 지표 이름을 선택한 후 검색에 추가를 선택합니다.

를 사용하여 지표를 보려면 AWS CLI

AWS Ground Station 는 연락처의 지상국 위치와 연결된 리전에서 CloudWatch 지표를 내보냅니다. 연결된 AWS 리전의 지상국 위치 목록은 섹션을 참조하세요 [지상국 위치의 AWS 리전 찾기](#). *ground-station-region-code*를 해당 지상국 위치의 AWS 리전 코드로 바꿉니다(예: 오레곤 1, 하와이 1 또는 알래스카 1의 us-west-2 경우). 이 절차의 모든 후속 AWS CLI 명령은 동일한 리전을 사용해야 합니다.

1. AWS CLI 가 설치되어 있는지 확인합니다. 설치에 대한 자세한 내용은 AWS CLI 버전 2 설치를 AWS CLI참조하세요. <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>
2. 지상국 위치와 연결된 AWS 리전을 식별합니다.
3. CloudWatch CLI의 [get-metric-data](#) 메서드를 사용하여 관심 있는 지표를 지정하도록 수정할 수 있는 파일을 생성한 다음 해당 지표를 쿼리하는 데 사용됩니다.

이렇게 하려면 실행합니다 `aws cloudwatch get-metric-data --region ground-station-region-code --generate-cli-skeleton`. 이렇게 하면 다음과 비슷한 출력이 생성됩니다.

```
{
  "MetricDataQueries": [
    {
      "Id": "",
      "MetricStat": {
        "Metric": {
          "Namespace": "",
          "MetricName": "",
          "Dimensions": [
            {
              "Name": "",
              "Value": ""
            }
          ]
        },
        "Period": 0,
        "Stat": "",
        "Unit": "Seconds"
      },
      "Expression": "",
      "Label": "",
      "ReturnData": true,
    }
  ]
}
```

```

        "Period": 0,
        "AccountId": ""
    } ],
    "StartTime": "1970-01-01T00:00:00",
    "EndTime": "1970-01-01T00:00:00",
    "NextToken": "",
    "ScanBy": "TimestampDescending",
    "MaxDatapoints": 0,
    "LabelOptions": {
        "Timezone": ""
    }
}

```

4. 를 실행하여 사용 가능한 CloudWatch 지표를 나열
`aws cloudwatch list-metrics --region ground-station-region-code`합니다.

최근에 AWS Ground Station를 사용한 경우 메서드는 다음과 같은 항목이 포함된 출력을 반환해야 합니다.

```

...
{
    "Namespace": "AWS/GroundStation",
    "MetricName": "ReceivedPower",
    "Dimensions": [
        {
            "Name": "Polarization",
            "Value": "LHCP"
        },
        {
            "Name": "SatelliteId",
            "Value": "arn:aws:groundstation::111111111111:satellite/aaaaaaaa-
bbbb-cccc-dddd-eeeeeeeeeeeeee"
        }
    ]
},
...

```

Note

마지막으로 사용한 후 2주가 지난 경우 사용 [가능한 지표 테이블](#)을 수동으로 검사하여 지표 AWS/GroundStation 네임스페이스에서 지표 이름과 차원을 찾아 AWS Ground Station야 합니다. CloudWatch 제한 사항에 대한 자세한 내용은 다음을 참조하세요. [사용 가능한 지표 보기](#)

5. 2단계에서 생성한 JSON 파일을와 같은 3단계SatelliteId의 필수 값과 지표Polarization의 필수 값과 일치하도록 수정합니다. 또한 연락처와 일치하도록 StartTime, 및 EndTime 값을 업데이트해야 합니다. 예제:

```
{
  "MetricDataQueries": [
    {
      "Id": "receivedPowerExample",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/GroundStation",
          "MetricName": "ReceivedPower",
          "Dimensions": [
            {
              "Name": "SatelliteId",
              "Value":
                "arn:aws:groundstation::111111111111:satellite/aaaaaaaa-bbbb-cccc-dddd-
                eeeeeeeeeeee"
            },
            {
              "Name": "Polarization",
              "Value": "RHCP"
            }
          ]
        },
        "Period": 300,
        "Stat": "Maximum",
        "Unit": "None"
      },
      "Label": "ReceivedPowerExample",
      "ReturnData": true
    }
  ],
}
```

```

    "StartTime": "2024-02-08T00:00:00",
    "EndTime": "2024-04-09T00:00:00"
  }

```

Note

AWS Ground Station 는 지표에 따라 1~60초마다 지표를 게시합니다. Period 필드에 지표의 게시 기간보다 작은 값이 있는 경우 지표가 반환되지 않습니다.

- 이전 단계에서 생성한 `aws cloudwatch get-metric-data` 구성 파일을 실행합니다. 아래에 예제가 나와 있습니다.

```

aws cloudwatch get-metric-data --region ground-station-region-code --cli-input-json
file://<nameOfConfigurationFileCreatedInStep2>.json

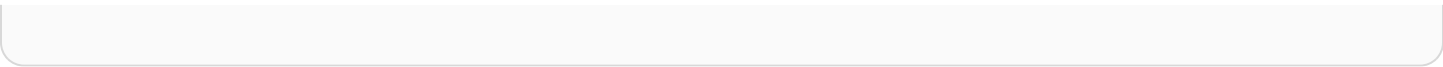
```

지표는 접축의 타임스탬프와 함께 제공됩니다. AWS Ground Station 지표 출력의 예는 아래에 나와 있습니다.

```

{
  "MetricDataResults": [
    {
      "Id": "receivedPowerExample",
      "Label": "ReceivedPowerExample",
      "Timestamps": [
        "2024-04-08T18:35:00+00:00",
        "2024-04-08T18:30:00+00:00",
        "2024-04-08T18:25:00+00:00"
      ],
      "Values": [
        -33.30191555023193,
        -31.46100273132324,
        -32.13915576934814
      ],
      "StatusCode": "Complete"
    }
  ],
  "Messages": []
}

```



의 보안 AWS Ground Station

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다. AWS는 보안 목표를 달성하는 데 도움이 되는 보안 관련 도구 및 기능을 제공합니다. 이러한 도구와 기능에는 네트워크 보안, 구성 관리, 액세스 제어 및 데이터 보안이 포함됩니다.

를 사용할 때는 업계 모범 사례를 따르고 end-to-end 암호화를 구현하는 AWS Ground Station 것이 좋습니다. AWS는 암호화와 데이터 보호를 통합할 수 있는 API를 제공합니다. AWS 보안에 대한 자세한 내용은 [AWS 보안 소개](#) 백서를 참조하세요.

다음 주제에서 리소스 보안 방법에 대해 알아보십시오.

주제

- [에 대한 자격 증명 및 액세스 관리 AWS Ground Station](#)
- [AWS 에 대한 관리형 정책 AWS Ground Station](#)
- [Ground Station에 대한 서비스 연결 역할 사용](#)
- [에 대한 저장 데이터 암호화 AWS Ground Station](#)
- [에 대한 전송 중 데이터 암호화 AWS Ground Station](#)

에 대한 자격 증명 및 액세스 관리 AWS Ground Station

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 되는 서비스입니다. IAM 관리자는 누가 AWS Ground Station 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

- [대상](#)
- [ID를 통한 인증](#)
- [정책을 사용하여 액세스 관리](#)
- [AWS Ground Station 에서 IAM을 사용하는 방법](#)
- [에 대한 자격 증명 기반 정책 예제 AWS Ground Station](#)
- [AWS Ground Station 자격 증명 및 액세스 문제 해결](#)

대상

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

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청([참조 AWS Ground Station 자격 증명 및 액세스 문제 해결](#))
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출([AWS Ground Station 에서 IAM을 사용하는 방법 참조](#))
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성([에 대한 자격 증명 기반 정책 예제 AWS Ground Station 참조](#))

ID를 통한 인증

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

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

프로그래밍 방식 액세스를 위해서는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 [API 요청용 AWS Signature Version 4](#) 섹션을 참조하세요.

AWS 계정 루트 사용자

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

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명에 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 [What is IAM Identity Center?](#)를 참조하세요.

IAM 사용자 및 그룹

[IAM 사용자](#)는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여 액세스하도록 인간 사용자에게 요구](#)하기를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자 사용 사례](#) 섹션을 참조하세요.

IAM 역할

[IAM 역할](#)은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. [사용자에서 IAM 역할\(콘솔\)로 전환하거나 또는 API 작업을 호출하여 역할을](#) 수입할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 [역할 수입 방법](#)을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 [교차 계정 리소스 액세스](#)를 참조하세요.

정책을 사용하여 액세스 관리

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

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

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수입할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책

을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 [고객 관리형 정책으로 사용자 지정 IAM 권한 정의](#)를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책 및 인라인 정책 중에서 선택](#) 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

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

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 [IAM 엔터티의 권한 범위](#)를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [서비스 제어 정책](#)을 참조하세요.
- 리소스 제어 정책(RCP) - 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [리소스 제어 정책\(RCP\)](#)을 참조하세요.
- 세션 정책 - 역할 또는 페더레이션 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 [세션 정책](#)을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하세요.

AWS Ground Station 에서 IAM을 사용하는 방법

IAM을 사용하여에 대한 액세스를 관리하기 전에 사용할 수 있는 IAM 기능을 AWS Ground Station알아 보니다 AWS Ground Station.

에서 사용할 수 있는 IAM 기능 AWS Ground Station

IAM 특성	AWS Ground Station 지원
자격 증명 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
엔터티 권한	예
서비스 역할	아니요
서비스 연결 역할	예

AWS Ground Station 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 [AWS IAM으로 작업하는 서비스를](#) 참조하세요.

에 대한 자격 증명 기반 정책 AWS Ground Station

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지

를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 [고객 관리형 정책으로 사용자 지정 IAM 권한 정의](#)를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하세요.

에 대한 자격 증명 기반 정책 예제 AWS Ground Station

자격 AWS Ground Station 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Ground Station](#).

내의 리소스 기반 정책 AWS Ground Station

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 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, 페더레이션 사용자 또는 이 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM에서 교차 계정 리소스 액세스](#)를 참조하세요.

에 대한 정책 작업 AWS Ground Station

정책 작업 지원: 예

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 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

AWS Ground Station 작업 목록을 보려면 서비스 승인 참조의에서 [정의한 작업을 AWS Ground Station](#) 참조하세요.

의 정책 작업은 작업 앞에 다음 접두사를 AWS Ground Station 사용합니다.

```
groundstation
```

단일 문에서 여러 작업을 지정하려면 심표로 구분합니다.

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

자격 AWS Ground Station 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Ground Station](#).

에 대한 정책 리소스 AWS Ground Station

정책 리소스 지원: 예

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.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 [Amazon 리소스 이름\(ARN\)](#)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"

```

AWS Ground Station 리소스 유형 및 해당 ARNs 목록을 보려면 서비스 승인 참조의에서 [정의한 리소스를 AWS Ground Station](#) 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 [AWS Ground Station에서 정의한 작업](#)을 참조하세요.

자격 AWS Ground Station 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Ground Station](#).

에 대한 정책 조건 키 AWS Ground Station

서비스별 정책 조건 키 지원: 예

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.

Condition 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 [AWS 전역 조건 컨텍스트 키](#)를 참조하세요.

AWS Ground Station 조건 키 목록을 보려면 서비스 승인 참조의 [대한 조건 키를 AWS Ground Station](#) 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 [에서 정의한 작업을 AWS Ground Station](#) 참조하세요.

자격 AWS Ground Station 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [에 대한 자격 증명 기반 정책 예제 AWS Ground Station](#).

ACLs AWS Ground Station

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

를 사용한 ABAC AWS Ground Station

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 [ABAC 권한 부여를 통한 권한 정의](#)를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 [속성 기반 액세스 제어\(ABAC\) 사용](#)을 참조하세요.

에서 임시 자격 증명 사용 AWS Ground Station

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 [IAM의 임시 보안 자격 증명](#) 및 [IAM으로 작업하는 AWS 서비스](#) 섹션을 참조하세요.

에 대한 교차 서비스 보안 주체 권한 AWS Ground Station

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스 함께 사용합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

에 대한 서비스 역할 AWS Ground Station

서비스 역할 지원: 아니요

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 내용은 IAM 사용 설명서의 [AWS 서비스 AWS에 권한을 위임할 역할 생성](#)을 참조하세요.

Warning

서비스 역할에 대한 권한을 변경하면 AWS Ground Station 기능이 중단될 수 있습니다. 에서 관련 지침을 AWS Ground Station 제공하는 경우에만 서비스 역할을 편집합니다.

에 대한 서비스 연결 역할 AWS Ground Station

서비스 연결 역할 지원: 예

서비스 연결 역할은 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 [IAM으로 작업하는 AWS 서비스](#)를 참조하세요. 서비스 연결 역할 열에서 Yes가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

에 대한 자격 증명 기반 정책 예제 AWS Ground Station

기본적으로 사용자 및 역할에는 AWS Ground Station 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성\(콘솔\)](#)을 참조하세요.

각 리소스 유형에 대한 ARNs 형식을 포함하여 AWS Ground Station에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 [에 대한 작업, 리소스 및 조건 키를 AWS Ground Station](#) 참조하세요.

주제

- [정책 모범 사례](#)
- [AWS Ground Station 콘솔 사용](#)
- [사용자가 자신의 고유한 권한을 볼 수 있도록 허용](#)

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS Ground Station 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 또는 [AWS 직무에 대한 관리형 정책](#)을 참조하세요.
- 최소 권한 적용 - IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 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. AWS 서비스와 같은 특징을 통해 사용되는 경우 조건을 사용하여 서

비즈니스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.

- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 [IAM Access Analyzer에서 정책 검증](#)을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정입니다. API 작업을 직접적으로 호출할 때 MFA가 필요하다면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 [MFA를 통한 보안 API 액세스](#)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하세요.

AWS Ground Station 콘솔 사용

AWS Ground Station 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은의 AWS Ground Station 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 AWS Ground Station 콘솔을 계속 사용할 수 있도록 하려면 AWS Ground Station **ConsoleAccess** 또는 **ReadOnly** AWS 관리형 정책도 엔티티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 [사용자에게 권한 추가](#)를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 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": "*"
}
]
}

```

AWS Ground Station 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단 AWS Ground Station 하고 수정할 수 있습니다.

주제

- [에서 작업을 수행할 권한이 없음 AWS Ground Station](#)
- [iam:PassRole을 수행하도록 인증되지 않음](#)
- [내 외부의 사람이 내 AWS Ground Station 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.](#)

에서 작업을 수행할 권한이 없음 AWS Ground Station

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 `groundstation:GetWidget` 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
groundstation:GetWidget on resource: my-example-widget
```

이 경우, `groundstation:GetWidget` 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

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

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS Ground Station에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

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

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

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우, Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

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

내 외부의 사람이 내 AWS Ground Station 리소스에 액세스 AWS 계정 하도록 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 제

어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- 에서 이러한 기능을 AWS Ground Station 지원하는지 여부를 알아보려면 섹션을 참조하세요 [AWS Ground Station에서 IAM을 사용하는 방법](#).
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 [IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요](#).
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [타사가 AWS 계정 소유한에 대한 액세스 권한 제공을](#) AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [외부에서 인증된 사용자에게 액세스 권한 제공\(ID 페더레이션\)](#)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 [IAM의 크로스 계정 리소스 액세스](#)를 참조하세요.

AWS에 대한 관리형 정책 AWS Ground Station

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 [고객 관리형 정책](#)을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 미칩니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 가이드의 [AWS 관리형 정책](#)을 참조하세요.

AWS 관리형 정책: AWSGroundStationAgentInstancePolicy

AWSGroundStationAgentInstancePolicy 정책을 IAM ID에 연결할 수 있습니다.

이 정책은 AWS Ground Station 에이전트에게 Amazon EC2 인스턴스에 Ground Station 고객 응대 중에 인스턴스가 데이터를 보내고 받을 수 있는 권한을 부여합니다. 이 정책의 모든 권한은 Ground Station 서비스에서 부여됩니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `groundstation` - 데이터 흐름 엔드포인트 인스턴스가 Ground Station Agent API를 직접적으로 호출할 수 있도록 합니다.

최신 버전의 JSON 정책 문서를 보려면 AWS 관리형 정책 참조 안내서의 [AWSGroundStationAgentInstancePolicy](#)를 참조하세요.

AWS 관리형 정책:

AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy

AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy를 IAM 엔터티에 연결할 수 없습니다. 이 정책은가 사용자를 대신하여 작업을 AWS Ground Station 수행하도록 허용하는 서비스 연결 역할에 연결됩니다. 자세한 내용은 [서비스 연결 역할 사용](#)을 참조하십시오.

이 정책은가 퍼블릭 IPv4 주소를 찾을 AWS Ground Station 수 있도록 허용하는 EC2 권한을 부여합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `ec2:DescribeAddresses` -가 사용자를 대신하여 EIPs와 연결된 모든 IPs를 나열 AWS Ground Station 하도록 허용합니다.

- `ec2:DescribeNetworkInterfaces` -가 사용자를 대신하여 EC2 인스턴스와 연결된 네트워크 인터페이스에 대한 정보를 가져오 AWS Ground Station 도록 허용합니다.

최신 버전의 JSON 정책 문서를 보려면 AWS 관리형 정책 참조 안내서의 [AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy](#)를 참조하세요.

AWS Ground Station AWS 관리형 정책에 대한 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 AWS Ground Station 이후부터의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 AWS Ground Station 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경	설명	Date
AWSGroundStationAgentInstancePolicy – 기존 정책에 대한 업데이트	AWS Ground Station 는 에이전트가 향상된 Ground Station 작업을 위해 작업 응답 URLs을 검색할 수 있도록 새로운 권한을 추가했습니다.	2025년 11월 13일
AWSGroundStationAgentInstancePolicy — 새 정책	AWS Ground Station 는 AWS Ground Station 에이전트를 사용할 수 있는 데이터 흐름 엔드포인트 인스턴스 권한을 제공하는 새 정책을 추가했습니다.	2023년 8월 12일
AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy — 새 정책	AWS Ground Station 는가 EIP와 연결된 퍼블릭 IPv4 주소 및 EC2 인스턴스와 연결된 네트워크 인터페이스를 AWS Ground Station 찾을 수 있는 권한을 EC2에 부여하는 새 정책을 추가했습니다. EIPs	2022년 11월 2일

변경	설명	Date
AWS Ground Station 에서 변경 내용 추적 시작	AWS Ground Station 가 AWS 관리형 정책에 대한 변경 내용 추적을 시작했습니다.	2021년 3월 1일

Ground Station에 대한 서비스 연결 역할 사용

AWS Ground Station 는 AWS Identity and Access Management (IAM) [서비스 연결 역할](#)을 사용합니다. 서비스 연결 역할은 에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 Ground Station에서 사전 정의하며 서비스가 사용자를 대신하여 다른 AWS 서비스를 호출하는 데 필요한 모든 권한을 포함합니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할은 Ground Station를 더 쉽게 설정할 수 있습니다. Ground Station에서 서비스 연결 역할의 권한을 정의하므로 다르게 정의되지 않은 한, Ground Station만 해당 역할을 수입할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 [AWS IAM으로 작업하는 서비스를](#) 참조하고 서비스 연결 역할 열에서 예인 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

Ground Station에 대한 서비스 연결 역할 권한

Ground Station은 AWSServiceRoleForGroundStationDataflowEndpointGroup라는 서비스 연결 역할을 사용합니다 — AWS GroundStation은 이 서비스 연결 역할을 사용하여 EC2를 간접적으로 호출하여 퍼블릭 IPv4 주소를 찾습니다.

AWSServiceRoleForGroundStationDataflowEndpointGroup 서비스 연결 역할은 역할을 수입하기 위해 다음 서비스를 신뢰합니다.

- `groundstation.amazonaws.com`

이름이 AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy인 연결 권한 정책은 Ground Station이 지정된 리소스에 대해 다음 작업을 수행하도록 허용합니다.

- 작업: `all AWS resources (*)`에 대한 `ec2:DescribeAddresses`

조치를 통해 Ground Station은 EIP와 관련된 모든 IP를 나열할 수 있습니다.

- 작업: `all AWS resources (*)`에 대한 `ec2:DescribeNetworkInterfaces`

작업을 통해 Ground Station은 EC2 인스턴스와 연결된 네트워크 인터페이스에 대한 정보를 가져올 수 있습니다

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 연결 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 권한](#)을 참조하세요.

Ground Station에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS CLI 또는 AWS API에서 `DataflowEndpointGroup`을 생성하면 Ground Station이 서비스 연결 역할을 생성합니다.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. `DataflowEndpointGroup`을 생성하면 Ground Station이 서비스 연결 역할을 다시 자동으로 생성합니다.

또한 IAM 콘솔을 사용해 Amazon EC2로 데이터 전달 사용 사례로 서비스 연결 역할을 생성할 수도 있습니다. AWS CLI 또는 AWS API에서 서비스 이름을 사용하여 `groundstation.amazonaws.com` 서비스 연결 역할을 생성합니다. 자세한 내용은 IAM 사용자 설명서의 [서비스 연결 역할 생성](#) 섹션을 참조하세요. 이 서비스 연결 역할을 삭제하면 동일한 프로세스를 사용하여 역할을 다시 생성할 수 있습니다.

Ground Station에 대한 서비스 연결 역할 편집

Ground Station에서는 `AWSServiceRoleForGroundStationDataflowEndpointGroup` 서비스 연결 역할을 편집할 수 없습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 편집](#)을 참조하세요.

Ground Station에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔터티가 없도록 합니다.

먼저 서비스 연결 역할을 사용하여 `DataflowEndpointGroups`를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 따라서 `DataflowEndpointGroups`에 대한 권한을 실수로 취소하는 것을 방지할 수 있습니다.

습니다. 한 서비스 연결 역할을 여러 DataflowEndpointGroups 그룹에 사용하는 경우 서비스 연결 역할을 삭제하기 전에 해당 역할을 사용하는 모든 DataflowEndpointGroups 그룹을 삭제해야 합니다.

Note

리소스를 삭제하려 할 때 Ground Station 서비스가 역할을 사용 중이면 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

AWSServiceRoleForGroundStationDataflowEndpointGroup에서 사용하는 Ground Station 리소스를 삭제하려면

- AWS CLI 또는 AWS API를 통해 DataflowEndpointGroups를 삭제합니다.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여

AWSServiceRoleForGroundStationDataflowEndpointGroup 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 [서비스 연결 역할 삭제](#)를 참조하십시오.

Ground Station 서비스 연결 역할이 지원되는 리전

Ground Station에서는 서비스를 사용할 수 있는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 [리전 표](#)를 참조하세요.

문제 해결

NOT_AUTHORIZED_TO_CREATE_SLR - 이는 CreateDataflowEndpointGroup API를 간접적으로 호출하는 데 사용되는 계정 내 역할에 iam:CreateServiceLinkedRole 권한이 없음을 나타냅니다. iam:CreateServiceLinkedRole 권한이 있는 관리자는 계정의 서비스 연결 역할을 수동으로 생성해야 합니다.

에 대한 저장 데이터 암호화 AWS Ground Station

AWS Ground Station 는 기본적으로 암호화를 제공하여 AWS 소유 암호화 키를 사용하여 저장된 민감한 데이터를 보호합니다.

- AWS 소유 키 - 기본적으로 이러한 키를 AWS Ground Station 사용하여 직접 식별 가능한 개인 데이터 및 에페메리스를 자동으로 암호화합니다. AWS소유 키를 보거나 관리하거나 사용할 수 없으며

사용을 감사할 수 없습니다. 그러나 데이터를 암호화하는 키를 보호하기 위해 조치를 취하거나 프로그램을 변경할 필요는 없습니다. 자세한 내용은 [AWS Key Management Service 개발자 안내서](#)의 [AWS소유 키](#)를 참조하세요.

기본적으로 저장 데이터를 암호화하면 민감한 데이터 보호와 관련된 운영 오버헤드와 복잡성을 줄이는 데 도움이 됩니다. 동시에 엄격한 암호화 규정 준수 및 규제 요구 사항을 충족하는 안전한 애플리케이션을 구축할 수 있습니다.

AWS Ground Station 는 모든 민감한 유희 데이터에 암호화를 적용하지만, 에페메리스와 같은 일부 AWS Ground Station 리소스의 경우 기본 관리형 키 대신 고객 AWS 관리형 키를 사용하도록 선택할 수 있습니다.

- 고객 관리형 키 -는 기존 소유 암호화 대신 생성, 소유 및 관리하는 대칭 고객 관리 AWS 형 키 사용을 AWS Ground Station 지원합니다. 이 암호화 계층을 완전히 제어할 수 있으므로 다음과 같은 작업을 수행할 수 있습니다.
 - 키 정책 수립 및 유지
 - IAM 정책 및 권한 부여 수립 및 유지
 - 키 정책 활성화 및 비활성화
 - 키 암호화 자료 교체
 - 태그 추가
 - 키 별칭 만들기
 - 키 삭제 일정 수립

자세한 내용은 [AWS Key Management Service 개발자 안내서](#)의 [고객 관리 키](#)를 참조하세요.

다음 표에는에서 고객 관리형 키 사용을 AWS Ground Station 지원하는 리소스가 요약되어 있습니다.

데이터 유형	AWS 소유 키 암호화	고객 관리형 키 암호화 (선택 사항)
위성의 궤적을 계산하는 데 사용되는 Ephemeris 데이터	활성화됨	활성화됨
안테나를 명령하는 데 사용되는 방위각 고도 에페메리스	활성화됨	활성화됨

Note

AWS Ground Station 는를 사용하여 저장 데이터 암호화를 자동으로 활성화 AWS 소유 키 하여 개인 식별 데이터를 무료로 보호합니다. 그러나 고객 관리형 키 사용에는 AWS KMS 요금이 적용됩니다. 요금에 대한 자세한 내용은 [AWS Key Management Service 요금](#)을 참조하십시오. 에 대한 자세한 내용은 [AWS Key Management Service 개발자 안내서](#)를 AWS KMS참조하세요.

각 리소스 유형에 대한 자세한 내용은 다음을 참조하세요.

- [TLE 및 OEM 에페메리스 데이터에 대한 저장 데이터 암호화](#)
- [방위각 고도 에페메리스에 대한 유희 시 암호화](#)

고객 관리형 키 생성

AWS Management Console또는 AWS KMS APIs.

대칭형 고객 관리형 키를 생성하려면

[AWS Key Management Service 개발자 안내서](#)의 대칭 고객 관리형 키를 생성하는 단계를 따릅니다.

주요 정책 개요

키 정책에서는 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 만들 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 [고객 관리형 키에 대한 액세스 관리](#)를 참조하세요.

AWS Ground Station 리소스와 함께 고객 관리형 키를 사용하려면 AWS Ground Station 서비스에 적절한 권한을 부여하도록 키 정책을 구성해야 합니다. 특정 권한 및 정책 구성은 암호화하려는 리소스 유형에 따라 다릅니다.

- TLE 및 OEM 에페메리스 데이터의 경우 - 특정 키 정책 요구 사항 및 예제[TLE 및 OEM 에페메리스 데이터에 대한 저장 데이터 암호화](#)는 섹션을 참조하세요.
- 방위각 고도 에페메리스 데이터의 경우 - 특정 키 정책 요구 사항 및 예제[방위각 고도 에페메리스에 대한 유희 시 암호화](#)는 섹션을 참조하세요.

Note

키 정책 구성은 에페메리스 유형마다 다릅니다. TLE 및 OEM 에페메리스 데이터는 키 액세스에 권한 부여를 사용하는 반면 방위각 고도 에페메리스는 직접 키 정책 권한을 사용합니다. 암호화하려는 특정 리소스 유형에 따라 키 정책을 구성해야 합니다.

[정책에서 권한을 지정](#)하고 [키 액세스 문제를 해결하는](#) 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하세요.

에 대한 고객 관리형 키 지정 AWS Ground Station

다음 리소스를 암호화하기 위해 고객 관리형 키를 지정할 수 있습니다.

- 에페메리스(TLE, OEM 및 방위각 고도)

리소스를 생성할 때 KmsKeyarn을 제공하여 데이터 키를 지정할 수 있습니다

- kmsKeyArn - AWS KMS 고객 관리형 [키의 키 식별자](#)

AWS Ground Station 암호화 컨텍스트

[암호화 컨텍스트](#)는 데이터에 대한 추가 컨텍스트 정보를 포함하는 선택적 키-값 페어 세트입니다.는 암호화 컨텍스트를 추가 인증 데이터로 AWS KMS 사용하여 인증된 암호화를 지원합니다. 데이터 암호화 요청에 암호화 컨텍스트를 포함하면는 암호화 컨텍스트를 암호화된 데이터에 AWS KMS 바인딩합니다. 요청에 동일한 암호화 컨텍스트를 포함해야 이 데이터를 해독할 수 있습니다.

AWS Ground Station 는 암호화되는 리소스에 따라 서로 다른 암호화 컨텍스트를 사용하고 생성된 각 키 권한에 대해 특정 암호화 컨텍스트를 지정합니다.

리소스별 암호화 컨텍스트 세부 정보는 다음을 참조하세요.

- [TLE 및 OEM 에페메리스 데이터에 대한 저장 데이터 암호화](#)
- [방위각 고도 에페메리스에 대한 유희 시 암호화](#)

TLE 및 OEM 에페메리스 데이터에 대한 저장 데이터 암호화

TLE 및 OEM 에페메리스에 대한 주요 정책 요구 사항

고객 관리형 키를 에페메리스 데이터와 함께 사용하려면 키 정책이 AWS Ground Station 서비스에 다음 권한을 부여해야 합니다.

- [kms:CreateGrant](#) - 고객 관리형 키에 대한 액세스 권한을 생성합니다. 암호화된 데이터를 읽고 저장하기 위해 고객 관리형 키에 대한 [권한 부여 작업을](#) 수행할 수 있는 AWS Ground Station 액세스 권한을 부여합니다.
- [kms:DescribeKey](#) - 제공된 키를 사용하기 전에 키를 AWS Ground Station 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.

[권한 부여 사용에](#) 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하세요.

고객 관리형 키를 사용하여 에페메리스를 생성하기 위한 IAM 사용자 권한

가 암호화 작업에 고객 관리형 키를 AWS Ground Station 사용하는 경우 에페메리스 리소스를 생성하는 사용자를 대신합니다.

고객 관리형 키를 사용하여 에페메리스 리소스를 생성하려면 사용자에게 고객 관리형 키에서 다음 작업을 호출할 수 있는 권한이 있어야 합니다.

- [kms:CreateGrant](#) - 사용자가 대신 고객 관리형 키에 대한 권한 부여를 생성할 수 있습니다 AWS Ground Station.
- [kms:DescribeKey](#) - 사용자가 고객 관리형 키 세부 정보를 보고 키를 검증할 수 있습니다.

키 정책에서 허용하는 경우 키 정책 또는 IAM 정책에서 이러한 필수 권한을 지정할 수 있습니다. 이러한 권한을 통해 사용자는 대신 암호화 작업에 고객 관리형 키를 사용하도록 권한을 부여 AWS Ground Station 할 수 있습니다.

가 에페메리스에 AWS KMS 대해에서 권한 부여를 AWS Ground Station 사용하는 방법

AWS Ground Station 고객 관리형 [키를 사용하려면에 키 권한이](#) 필요합니다.

고객 관리형 키로 암호화된 에페메리스를 업로드하면 [CreateGrant](#) 요청에 전송하여 사용자를 대신하여 키 부여를 AWS Ground Station 생성합니다 AWS KMS. 의 권한 부여 AWS KMS 는 계정의 AWS KMS 키에 대한 액세스 권한을 부여하는 AWS Ground Station 데 사용됩니다.

이렇게 하면 AWS Ground Station 가 다음을 수행할 수 있습니다.

- 데이터 키는 암호화에 즉시 사용되지 않으므로 [GenerateDataKey](#)를 호출하여 암호화된 데이터 키를 생성하고 저장합니다.
- [Decrypt](#)를 호출하여 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스합니다.
- 데이터 키를 사용하여 데이터를 암호화하려면 [Encrypt](#)를 호출합니다.
- 서비스에서 [RetireGrant](#)를 허용하도록 사용 중지 보안 주체를 설정합니다.

언제든지 권한 부여에 대한 액세스를 취소할 수 있습니다. 이렇게 하면 고객 관리형 키로 암호화된 데이터에 액세스할 수 AWS Ground Station 없으며, 이는 해당 데이터에 의존하는 작업에 영향을 미칩니다. 예를 들어 현재 고객 응대에 사용 중인 에페메리스에서 키 부여를 제거하면 AWS Ground Station 는 제공된 에페메리스 데이터를 사용하여 고객 응대 중에 안테나를 가리키지 못합니다. 이렇게 하면 연락이 실패 상태로 종료됩니다.

에페메리스 암호화 컨텍스트

에페메리스 리소스를 암호화하기 위한 키 부여는 특정 위성 ARN에 바인딩됩니다.

```
"encryptionContext": {
  "aws:groundstation:arn":
    "arn:aws:groundstation::111122223333:satellite/00a770b0-082d-45a4-80ed-SAMPLE",
  "aws:s3:arn":
    "arn:aws:s3:::customerephemerisbucket/0034abcd-12ab-34cd-56ef-123456SAMPLE"
}
```

Note

키 부여는 동일한 키-위성 쌍에 재사용됩니다.

모니터링을 위한 암호화 컨텍스트 사용

대칭형 고객 관리형 키를 사용하여 Ephemeris를 암호화하는 경우 감사 레코드 및 로그의 암호화 컨텍스트를 사용하여 고객 관리형 키가 사용되는 방식을 식별할 수도 있습니다. 암호화 컨텍스트는 [AWS CloudTrail 또는 Amazon CloudWatch Logs에서 생성된 로그](#)에도 나타납니다.

암호화 컨텍스트를 사용하여 고객 관리형 키에 대한 액세스 제어

그러나 암호화 컨텍스트를 사용하여 키 정책 및 IAM 정책에서 대칭 conditions에 대한 액세스를 제어할 수도 있습니다. 또한 권한 부여에서 암호화 컨텍스트 제약 조건을 사용할 수 있습니다.

AWS Ground Station 는 권한 부여에 암호화 컨텍스트 제약 조건을 사용하여 계정 또는 리전의 고객 관리형 키에 대한 액세스를 제어합니다. 권한 부여 제약 조건에 따라 권한 부여가 허용하는 작업은 지정된 암호화 컨텍스트를 사용해야 합니다.

다음은 특정 암호화 컨텍스트에서 고객 관리형 키에 대한 액세스 권한을 부여하는 키 정책 설명의 예입니다. 이 정책 설명의 조건에 따라 권한 부여에는 암호화 컨텍스트를 지정하는 암호화 컨텍스트 제약 조건이 있어야 합니다.

다음 예제에서는 위성에 바인딩된 에페메리스 데이터에 대한 키 정책을 보여줍니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow AWS Ground Station to Describe key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*"
    },
    {
      "Sid": "Allow AWS Ground Station to Create Grant on key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": "kms:CreateGrant",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:aws:groundstation:arn":
            "arn:aws:groundstation::123456789012:satellite/satellite-id"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

암호화 키의 에페메리스 모니터링

AWS Key Management Service 고객 관리형 키를 에페메리스 리소스와 함께 사용하는 경우 [AWS CloudTrail](#) 또는 [Amazon CloudWatch logs](#)를 사용하여 AWS Ground Station 보내는 요청을 추적할 수 있습니다 AWS KMS. 다음 예제는 [CreateGrant](#), [GenerateDataKey](#), [Decrypt](#) 및 [DescribeKey](#)에 대한 CloudTrail 이벤트로, 고객 관리형 키로 암호화된 데이터에 액세스하기 AWS Ground Station 위해에서 호출한 작업을 모니터링합니다 AWS KMS .

CreateGrant

AWS KMS 고객 관리형 키를 사용하여 에페메리스 리소스를 암호화하는 경우는 AWS 계정의 AWS KMS 키에 액세스하기 위해 사용자를 대신하여 [CreateGrant](#) 요청을 AWS Ground Station 보냅니다. 가 AWS Ground Station 생성하는 권한 부여는 AWS KMS 고객 관리형 키와 연결된 리소스에 따라 다릅니다. 또한 AWS Ground Station 는 리소스를 삭제할 때 [RetireGrant](#) 작업을 사용하여 권한 부여를 제거합니다.

다음 예제 이벤트는 에페메리스에 대한 [CreateGrant](#) 작업을 기록합니다.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ASIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/SampleUser01",
    "accountId": "111122223333",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ASIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {

```

```

        "creationDate": "2022-02-22T22:22:22Z",
        "mfaAuthenticated": "false"
    },
    "invokedBy": "AWS Internal"
},
"eventTime": "2022-02-22T22:22:22Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "AWS Internal",
"userAgent": "ExampleDesktop/1.0 (V1; OS)",
"requestParameters": {
    "operations": [
        "GenerateDataKeyWithoutPlaintext",
        "Decrypt",
        "Encrypt"
    ],
    "constraints": {
        "encryptionContextSubset": {
            "aws:groundstation:arn":
"arn:aws:groundstation::111122223333:satellite/00a770b0-082d-45a4-80ed-SAMPLE"
        }
    },
    "granteePrincipal": "groundstation.us-west-2.amazonaws.com",
    "retiringPrincipal": "groundstation.us-west-2.amazonaws.com",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
},
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": false,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],

```

```

    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }

```

DescribeKey

AWS KMS 고객 관리형 키를 사용하여 에페메리스 리소스를 암호화하는 경우는 요청된 키가 계정에 존재하는지 확인하기 위해 사용자를 대신하여 [DescribeKey](#) 요청을 AWS Ground Station 보냅니다.

다음 예제 이벤트는 [DescribeKey](#) 작업을 기록합니다.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ASIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:sts::111122223333:assumed-role/User/Role",
    "accountId": "111122223333",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ASIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Role",
        "accountId": "111122223333",
        "userName": "User"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-02-22T22:22:22Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2022-02-22T22:22:22Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",

```

```

    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}

```

GenerateDataKey

AWS KMS 고객 관리형 키를 사용하여 에페메리스 리소스를 암호화하는 경우는 데이터를 암호화할 데이터 키를 생성하기 위해 [GenerateDataKey](#) 요청을 AWS Ground Station 보냅니다.

다음 예제 이벤트는 에페메리스에 대한 [GenerateDataKey](#) 작업을 기록합니다.

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2022-02-22T22:22:22Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keySpec": "AES_256",

```

```

    "encryptionContext": {
      "aws:groundstation:arn":
"arn:aws:groundstation::111122223333:satellite/00a770b0-082d-45a4-80ed-SAMPLE",
      "aws:s3:arn":
"arn:aws:s3:::customerephemerisbucket/0034abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventCategory": "Management"
}

```

Decrypt

AWS KMS 고객 관리형 키를 사용하여 에페메리스 리소스를 암호화하는 경우는 [Decrypt](#) 작업을 AWS Ground Station 사용하여 이미 동일한 고객 관리형 키로 암호화된 경우 제공된 에페메리스를 복호화합니다. 예를 들어 S3 버킷에서 epemeris를 업로드하고 해당 버킷에서 지정된 키를 사용하여 암호화하는 경우를 예로 들 수 있습니다.

다음 예제 이벤트는 에페메리스에 대한 [복호화](#) 작업을 기록합니다.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "AWS Internal"
  },

```

```

    "eventTime": "2022-02-22T22:22:22Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": {
      "encryptionContext": {
        "aws:groundstation:arn":
"arn:aws:groundstation::111122223333:satellite/00a770b0-082d-45a4-80ed-SAMPLE",
        "aws:s3:arn":
"arn:aws:s3:::customerephemerisbucket/0034abcd-12ab-34cd-56ef-123456SAMPLE"
      },
      "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventCategory": "Management"
  }

```

방위각 고도 에페메리스에 대한 유희 시 암호화

방위각 고도 에페메리스에 대한 주요 정책 요구 사항

방위각 고도 에페메리스 데이터와 함께 고객 관리형 키를 사용하려면 키 정책이 AWS Ground Station 서비스에 다음 권한을 부여해야 합니다. 권한 부여를 사용하는 TLE 및 OEM 에페메리스 데이터와 달리 방위각 고도 에페메리스는 암호화 작업에 직접 키 정책 권한을 사용합니다. 이는의 권한을 관리하고 키를 사용하는 더 간단한 방법입니다.

- [kms:GenerateDataKey](#) - 방위각 고도 에페메리스 데이터를 암호화하기 위한 데이터 키를 생성합니다.
- [kms:Decrypt](#) - 방위각 고도 에페메리스 데이터에 액세스할 때 암호화된 데이터 키를 해독합니다.

고객 관리형 키에 대한 AWS Ground Station 액세스 권한을 부여하는 키 정책 예제

Note

방위각 고도 에페메리스를 사용하면 키 정책에서 이러한 권한을 직접 구성해야 합니다. 키 정책 설명에서 리전 AWS Ground Station 서비스 보안 주체(예: `groundstation.region.amazonaws.com`)에게 이러한 권한을 부여해야 합니다. 이러한 문이 키 정책에 추가 AWS Ground Station 되지 않으면 사용자 지정 방위각 고도 에페메리스를 저장하거나 액세스할 수 없습니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow AWS Ground Station to Describe key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*"
    },
    {
      "Sid": "Allow AWS Ground Station to Encrypt and Decrypt with key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
]
}
```

고객 관리형 키를 사용하여 방위각 고도 에페메리스를 생성하기 위한 IAM 사용자 권한이 암호화 작업에 고객 관리형 키를 AWS Ground Station 사용하는 경우 방위각 고도 에페메리스 리소스를 생성하는 사용자를 대신합니다.

고객 관리형 키를 사용하여 방위각 고도 에페메리스 리소스를 생성하려면 사용자에게 고객 관리형 키에서 다음 작업을 호출할 수 있는 권한이 있어야 합니다.

- [kms:GenerateDataKey](#) - 사용자가 방위각 고도 에페메리스 데이터를 암호화하기 위한 데이터 키를 생성할 수 있습니다.
- [kms:Decrypt](#) - 방위각 고도 에페메리스 데이터에 액세스할 때 사용자가 데이터 키를 복호화할 수 있도록 허용합니다.
- [kms:DescribeKey](#) - 사용자가 고객 관리형 키 세부 정보를 보고 키를 검증할 수 있습니다.

키 정책에서 허용하는 경우 키 정책 또는 IAM 정책에서 이러한 필수 권한을 지정할 수 있습니다. 이러한 권한을 통해 사용자는 대신 암호화 작업에 고객 관리형 키를 사용하도록 권한을 부여 AWS Ground Station 할 수 있습니다.

가 방위각 고도 에페메리스에 대한 키 정책을 AWS Ground Station 사용하는 방법

고객 관리형 키로 방위각 고도 에페메리스 데이터를 제공하면 키 정책을 AWS Ground Station 사용하여 암호화 키에 액세스합니다. 권한은 TLE 또는 OEM 에페메리스 데이터와 마찬가지로 권한 부여를 AWS Ground Station 통하지 않고 키 정책 설명을 통해 직접 부여됩니다.

고객 관리형 키에 대한 AWS Ground Station의 액세스를 제거하면 해당 키로 암호화된 데이터에 액세스할 수 AWS Ground Station 없으며, 이는 해당 데이터에 의존하는 작업에 영향을 미칩니다. 예를 들어 접촉에 현재 사용 중인 방위각 고도 에페메리스에 대한 키 정책 권한을 제거하면 AWS Ground Station 는 접촉 중에 안테나를 명령하기 위해 제공된 방위각 고도 데이터를 사용할 수 없습니다. 이렇게 하면 연락이 실패 상태로 종료됩니다.

방위각 고도 에페메리스 암호화 컨텍스트

가 AWS KMS 키를 AWS Ground Station 사용하여 방위각 고도 에페메리스 데이터를 암호화하는 경우 서비스는 [암호화 컨텍스트](#)를 지정합니다. 암호화 컨텍스트는 데이터 무결성을 보장하기 위해 AWS KMS 사용하는 추가 인증 데이터(AAD)입니다. 암호화 작업에 대해 암호화 컨텍스트가 지정되면 서비

스가 암호화 해제 작업에 대해 동일한 암호화 컨텍스트를 지정해야 합니다. 그렇지 않으면 암호화 해제가 실패합니다. 암호화 컨텍스트는 지정된 AWS KMS 키가 사용된 이유를 이해하는 데 도움이 되도록 CloudTrail 로그에도 기록됩니다. CloudTrail 로그에는 AWS KMS 키 사용을 설명하는 많은 항목이 포함될 수 있지만 각 로그 항목의 암호화 컨텍스트는 해당 특정 사용의 이유를 결정하는 데 도움이 될 수 있습니다.

AWS Ground Station 는 방위각 고도 에페메리스에서 고객 관리형 키로 암호화 작업을 수행할 때 다음 암호화 컨텍스트를 지정합니다.

```
{
  "encryptionContext": {
    "aws:groundstation:ground-station-id": "Ohio 1",
    "aws:groundstation:arn": "arn:aws:groundstation:us-east-2:111122223333:ephemeris/00a770b0-082d-45a4-80ed-SAMPLE",
    "aws:s3:arn": "arn:aws:s3:::customerephemerisbucket/00a770b0-082d-45a4-80ed-SAMPLE/raw"
  }
}
```

암호화 컨텍스트에는 다음이 포함됩니다.

`aws:groundstation:ground-station-id`

방위각 고도 에페메리스와 연결된 지상국의 이름입니다.

`aws:groundstation:arn`

에페메리스 리소스의 ARN입니다.

`aws:s3:arn`

Amazon S3에 저장된 에페메리스의 ARN입니다.

암호화 컨텍스트를 사용하여 고객 관리형 키에 대한 액세스 제어

IAM 조건문을 사용하여 고객 관리형 키에 대한 AWS Ground Station 액세스를 제어할 수 있습니다. `kms:GenerateDataKey` 및 `kms:Decrypt` 작업에 조건문을 추가하면 사용할 AWS KMS 수 있는 지상 스테이션이 제한됩니다.

다음은 특정 지상국의 특정 리전에서 고객 관리형 키에 대한 AWS Ground Station 액세스 권한을 부여하는 키 정책 설명의 예입니다. 이 정책 설명의 조건을 사용하려면 키 정책의 조건과 일치하는 암호화 컨텍스트를 지정하는 키에 대한 모든 암호화 및 복호화 액세스가 필요합니다.

특정 지상국의 고객 관리형 키에 대한 AWS Ground Station 액세스 권한을 부여하는 키 정책 예제

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow AWS Ground Station to Describe key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*"
    },
    {
      "Sid": "Allow AWS Ground Station to Encrypt and Decrypt with key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:aws:groundstation:ground-station-id":
            "specific-ground-station-name"
        }
      }
    }
  ]
}
```

여러 지상국의 고객 관리형 키에 대한 AWS Ground Station 액세스 권한을 부여하는 키 정책 예제

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow AWS Ground Station to Describe key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*"
    },
    {
      "Sid": "Allow AWS Ground Station to Encrypt and Decrypt with key",
      "Effect": "Allow",
      "Principal": {
        "Service": "groundstation.us-east-1.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:groundstation:ground-station-id":
            [
              "specific-ground-station-name-1",
              "specific-ground-station-name-2"
            ]
        }
      }
    }
  ]
}
```

방위각 고도 에페메리스에 대한 암호화 키 모니터링

방위각 고도 에페메리스 리소스와 함께 AWS KMS 고객 관리형 키를 사용하는 경우 [CloudTrail](#) 또는 [CloudWatch 로그](#)를 사용하여가 AWS Ground Station 로 보내는 요청을 추적할 수 있습니다 AWS KMS. 다음 예제는 고객 관리형 키로 암호화된 데이터에 액세스하기 위해에서 호출한 작업을 모니터링 AWS KMS AWS Ground Station 하기 위한 [GenerateDataKey](#) 및 [Decrypt](#)에 대한 CloudTrail 이벤트입니다.

GenerateDataKey

AWS KMS 고객 관리형 키를 사용하여 방위각 고도 에페메리스 리소스를 암호화하는 경우는 데이터를 암호화할 데이터 키를 생성하기 AWS KMS 위해 [GenerateDataKey](#) 요청을에 AWS Ground Station 보냅니다.

다음 예제 이벤트는 방위각 고도 에페메리스에 대한 [GenerateDataKey](#) 작업을 기록합니다.

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ASIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/SampleUser01",
    "accountId": "111122223333",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ASIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2025-08-25T14:45:48Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2025-08-25T14:52:02Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
```

```

"sourceIPAddress": "AWS Internal",
"userAgent": "AWS Internal",
"requestParameters": {
  "keySpec": "AES_256",
  "encryptionContext": {
    "aws:groundstation:arn": "arn:aws:groundstation:us-
west-2:111122223333:ephemeris/bb650670-7a4b-4152-bd60-SAMPLE",
    "aws:groundstation:ground-station-id": "Ohio 1",
    "aws:s3:arn": "arn:aws:s3:::customerephemerisbucket/bb650670-7a4b-4152-
bd60-SAMPLE/raw"
  },
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"responseElements": null,
"requestID": "ef6f9a8f-8ef6-46a1-bdcb-123456SAMPLE",
"eventID": "952842d4-1389-3232-b885-123456SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"sharedEventID": "8424f6b6-2280-4d1d-b9fd-0348b1546cba",
"eventCategory": "Management"
}

```

Decrypt

AWS KMS 고객 관리형 키를 사용하여 방위각 고도 에페메리스 리소스를 암호화하는 경우는 [Decrypt](#) 작업을 AWS Ground Station 사용하여 이미 동일한 고객 관리형 키로 암호화된 경우 제공된 방위각 고도 에페메리스 데이터를 복호화합니다.

다음 예제 이벤트는 방위각 고도 에페메리스에 대한 [Decrypt](#) 작업을 기록합니다.

```

{
  "eventVersion": "1.11",

```

```

"userIdentity": {
  "type": "AssumedRole",
  "principalId": "ASIAIOSFODNN7EXAMPLE",
  "arn": "arn:aws:sts::111122223333:assumed-role/Admin/SampleUser01",
  "accountId": "111122223333",
  "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "ASIAIOSFODNN7EXAMPLE",
      "arn": "arn:aws:iam::111122223333:role/Admin",
      "accountId": "111122223333",
      "userName": "Admin"
    }
  },
  "attributes": {
    "creationDate": "2025-08-25T14:45:48Z",
    "mfaAuthenticated": "false"
  }
},
"invokedBy": "AWS Internal",
"eventTime": "2025-08-25T14:54:01Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-west-2",
"sourceIPAddress": "AWS Internal",
"userAgent": "AWS Internal",
"requestParameters": {
  "encryptionContext": {
    "aws:groundstation:arn": "arn:aws:groundstation:us-west-2:111122223333:ephemeris/bb650670-7a4b-4152-bd60-SAMPLE",
    "aws:groundstation:ground-station-id": "Ohio 1",
    "aws:s3:arn": "arn:aws:s3:::customerephemerisbucket/bb650670-7a4b-4152-bd60-SAMPLE/raw"
  },
  "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
},
"responseElements": null,
"requestID": "a2f46066-49fb-461a-93cb-123456SAMPLE",
"eventID": "e997b426-e3ad-31c7-a308-123456SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",

```

```

        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "477b568e-7f56-4f04-905c-623ff146f30d",
    "eventCategory": "Management"
}

```

에 대한 전송 중 데이터 암호화 AWS Ground Station

AWS Ground Station 는 전송 중에 민감한 데이터를 보호하기 위해 기본적으로 암호화를 제공합니다. 데이터는 미션 프로파일 구성에 따라 두 가지 방법으로 AWS Ground Station 안테나 위치와 Amazon EC2 인스턴스 간에 스트리밍할 수 있습니다.

- AWS Ground Station 에이전트
- 데이터 흐름 엔드포인트

데이터를 스트리밍하는 각 방법은 전송 중 데이터 암호화를 다르게 처리합니다. 다음 섹션에서 각 방법에 대해 설명합니다.

AWS Ground Station 에이전트 스트림

AWS Ground Station 에이전트는 고객 관리형 AWS KMS 키를 사용하여 스트림을 암호화합니다. Amazon EC2 인스턴스에서 실행되는 AWS Ground Station 에이전트는 스트림을 자동으로 복호화하여 복호화된 데이터를 제공합니다.

스트림 암호화에 사용되는 AWS KMS 키는 [streamsKmsKey](#) 파라미터 `MissionProfile`에서 생성할 때 지정됩니다. 키에 대한 AWS Ground Station 액세스 권한을 부여하는 모든 권한은 연결된 AWS KMS 키 정책을 통해 처리됩니다 `streamsKmsKey`.

데이터 흐름 엔드포인트 스트림

데이터 흐름 엔드포인트 스트림은 [Datagram Transport Layer Security\(DTLS\)](#) 를 사용하여 암호화됩니다. 이 작업은 자체 서명된 인증서를 사용하여 수행되며 추가 구성이 필요하지 않습니다.

미션 프로파일 구성 예

제공된 예제는 퍼블릭 브로드캐스트 위성을 가져와 이를 지원하는 미션 프로파일을 생성하는 방법을 보여줍니다. 결과 템플릿은 공공 방송 위성 연락을 취하는 데 도움이 되고 위성에 대한 결정을 내리는 데 도움이 됩니다.

주제

- [JPSS-1 - 퍼블릭 브로드캐스트 위성\(PBS\) - 평가](#)
- [Amazon S3 데이터 전송을 활용하는 퍼블릭 브로드캐스트 위성](#)
- [데이터 흐름 엔드포인트\(협대역\)를 사용하는 퍼블릭 브로드캐스트 위성](#)
- [데이터 흐름 엔드포인트를 사용하는 퍼블릭 브로드캐스트 위성\(변조 및 디코딩됨\)](#)
- [AWS Ground Station 에이전트\(와이드밴드\)를 활용하는 퍼블릭 브로드캐스트 위성](#)

JPSS-1 - 퍼블릭 브로드캐스트 위성(PBS) - 평가

이 예제 섹션은와 일치합니다 [고객 온보딩 프로세스 개요](#). AWS Ground Station 와의 간략한 호환성 분석을 제공하고 이어지는 특정 예제의 단계를 설정합니다.

[퍼블릭 브로드캐스트 위성](#) 단원에서 언급했듯이 공개적으로 사용할 수 있는 일부 위성 또는 위성의 통신 경로를 활용할 수 있습니다. 이 단원에서는 AWS Ground Station 용어로 [JPSS-1](#)을 설명합니다. 참고로 [관절 폴라 위성 시스템 1\(JPSS-1\) 우주선 고속 데이터\(HRD\)를 다이렉트 브로드캐스트 스테이션\(DBS\) 무선 주파수\(RF\) 인터페이스 제어 문서\(ICD\)](#)에 활용하여 예제를 완료합니다. 또한 JPSS-1이 NORAD ID 43013에 연결되어 있다는 점에 유의해야 합니다.

JPSS-1 위성은 ICD의 그림 1-1과 같이 업링크 1개와 직접 다운링크 통신 경로 3개를 제공합니다. 이 4개의 통신 경로 중 단일 HRD(High Rate Data) 다운링크 통신 경로만 퍼블릭 소비에 사용할 수 있습니다. 이를 기반으로 이 경로에는 훨씬 더 구체적인 데이터도 연결되어 있습니다. 네 가지 경로는 다음과 같습니다.

- 데이터 속도가 2~128kbps인 2067.27MHz 중심 주파수의 명령 경로(업링크). 이 경로는 공개적으로 액세스할 수 없습니다.
- 데이터 속도가 1~524kbps인 2247.5MHz 중심 주파수의 원격 측정 경로(다운링크). 이 경로는 공개적으로 액세스할 수 없습니다.
- 데이터 속도가 150~300Mbps인 26.7034GHz 중심 주파수의 SMD 경로(다운링크). 이 경로는 공개적으로 액세스할 수 없습니다.

- 데이터 속도가 15Mbps인 7812MHz 중심 주파수의 HRD 경로(다운링크)에 대한 RF입니다. 대역폭은 30MHz이며 right-hand-circular-polarized. JPSS-1에 온보딩할 때 액세스할 수 있는 통신 경로 AWS Ground Station입니다. 이 통신 경로에는 계측 과학 데이터, 계측 엔지니어링 데이터, 계측 원격 측정 데이터 및 실시간 우주선 정리 데이터가 포함됩니다.

잠재적 데이터 경로를 비교할 때 명령(업링크), 원격 측정(다운링크) 및 HRD(다운링크) 경로가의 빈도, 대역폭 및 다중 채널 동시 사용 기능을 충족하는 것을 확인할 수 있습니다 AWS Ground Station. SMD 경로는 중앙 주파수가 기존 수신기 범위를 벗어나기 때문에 호환되지 않습니다. 지원되는 기능에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station 사이트 기능](#).

Note

SMD 경로는 호환되지 AWS Ground Station 않으므로 예제 구성에 표시되지 않습니다.

Note

명령(업링크) 및 원격 측정(다운링크) 경로는 ICD에 정의되어 있지 않으며 공개적으로 사용할 수 없기 때문에 사용 시 제공되는 값은 개념적입니다.

Amazon S3 데이터 전송을 활용하는 퍼블릭 브로드캐스트 위성

이 예제는 사용 설명서의 [JPSS-1 - 퍼블릭 브로드캐스트 위성\(PBS\) - 평가](#) 섹션에서 수행된 분석을 기반으로 합니다.

이 예제에서는 HRD 통신 경로를 디지털 중간 빈도로 캡처하고 향후 배치 처리를 위해 저장하려는 시나리오를 가정해야 합니다. 이렇게 하면 디지털화된 후 원시 무선 주파수(RF) I/Q(in-phase quadrature) 샘플이 절약됩니다. Amazon S3 버킷에 데이터가 저장되면 원하는 소프트웨어를 사용하여 데이터를 복조 및 디코딩할 수 있습니다. 자세한 처리 예제는 [MathWorks 자습서를](#) 참조하세요. 이 예제를 사용한 후에는 Amazon EC2 스팟 요금 구성 요소를 추가하여 데이터를 처리하고 전체 처리 비용을 절감하는 것이 좋습니다.

통신 경로

이 섹션은 시작하기 [데이터 흐름 통신 경로 계획](#)를 나타냅니다.

다음 템플릿 코드 조각은 모두 템플릿의 리소스 섹션에 속합니다 CloudFormation .

Resources:

Resources that you would like to create should be placed within the Resources section.

Note

CloudFormation 템플릿의 내용에 대한 자세한 내용은 [템플릿 섹션을](#) 참조하세요.

Amazon S3에 단일 통신 경로를 전달하는 시나리오를 고려할 때 단일 비동기 전송 경로가 있음을 알 수 있습니다. [비동기 데이터 전송](#) 섹션에 따라 Amazon S3 버킷을 정의해야 합니다.

```
# The S3 bucket where AWS Ground Station will deliver the downlinked data.
GroundStationS3DataDeliveryBucket:
  Type: AWS::S3::Bucket
  DeletionPolicy: Retain
  UpdateReplacePolicy: Retain
  Properties:
    # Results in a bucket name formatted like: aws-groundstation-data-{account id}-{region}-{random 8 character string}
    BucketName: !Join ["-", ["aws-groundstation-data", !Ref AWS::AccountId, !Ref
AWS::Region, !Select [0, !Split ["-", !Select [2, !Split ["/", !Ref AWS::StackId]]]]]]
```

또한가 버킷을 AWS Ground Station 사용하도록 허용하려면 적절한 역할 및 정책을 생성해야 합니다.

```
# The IAM role that AWS Ground Station will assume to have permission find and write
# data to your S3 bucket.
GroundStationS3DataDeliveryRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Action:
            - 'sts:AssumeRole'
          Effect: Allow
          Principal:
```

```

    Service:
      - groundstation.amazonaws.com
    Condition:
      StringEquals:
        "aws:SourceAccount": !Ref AWS::AccountId
      ArnLike:
        "aws:SourceArn": !Sub "arn:aws:groundstation:${AWS::Region}:
${AWS::AccountId}:config/s3-recording/*"

# The S3 bucket policy that defines what actions AWS Ground Station can perform on
your S3 bucket.
GroundStationS3DataDeliveryBucketPolicy:
  Type: AWS::IAM::Policy
  Properties:
    PolicyDocument:
      Statement:
        - Action:
            - 's3:GetBucketLocation'
          Effect: Allow
          Resource:
            - !GetAtt GroundStationS3DataDeliveryBucket.Arn
        - Action:
            - 's3:PutObject'
          Effect: Allow
          Resource:
            - !Join [ "/", [ !GetAtt GroundStationS3DataDeliveryBucket.Arn, "*" ] ]
    PolicyName: GroundStationS3DataDeliveryPolicy
  Roles:
    - !Ref GroundStationS3DataDeliveryRole

```

AWS Ground Station 구성

이 섹션은 시작하기 [구성 생성](#)를 나타냅니다.

자동 추적을 사용하여 기본 설정을 지정하려면 추적 구성이 필요합니다. PREFERRED를 자동 추적으
로 선택하면 신호 품질이 향상될 수 있지만 충분한 JPSS-1 에페메리스 품질로 인해 신호 품질을 충족
할 필요는 없습니다.

```

TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:

```

```

Name: "JPSS Tracking Config"
ConfigData:
  TrackingConfig:
    Autotrack: "PREFERRED"

```

통신 경로를 기반으로 위성 부분을 나타내는 안테나 다운링크 구성을 정의하고 방금 생성한 Amazon S3 버킷을 참조하는 s3 레코딩을 정의해야 합니다.

```

# The AWS Ground Station Antenna Downlink Config that defines the frequency spectrum
used to
# downlink data from your satellite.
JpssDownlinkDigIfAntennaConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Downlink DigIF Antenna Config"
    ConfigData:
      AntennaDownlinkConfig:
        SpectrumConfig:
          Bandwidth:
            Units: "MHz"
            Value: 30
          CenterFrequency:
            Units: "MHz"
            Value: 7812
          Polarization: "RIGHT_HAND"

# The AWS Ground Station S3 Recording Config that defines the S3 bucket and IAM role
to use
# when AWS Ground Station delivers the downlink data.
S3RecordingConfig:
  Type: AWS::GroundStation::Config
  DependsOn: GroundStationS3DataDeliveryBucketPolicy
  Properties:
    Name: "JPSS S3 Recording Config"
    ConfigData:
      S3RecordingConfig:
        BucketArn: !GetAtt GroundStationS3DataDeliveryBucket.Arn
        RoleArn: !GetAtt GroundStationS3DataDeliveryRole.Arn

```

AWS Ground Station 미션 프로파일

이 섹션은 시작하기 [미션 프로파일 생성](#)를 나타냅니다.

이제 연결된 구성이 있으므로 이를 사용하여 데이터 흐름을 구성할 수 있습니다. 나머지 파라미터에 기본값을 사용합니다.

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to downlink data.
JpssAsynchMissionProfile:
  Type: AWS::GroundStation::MissionProfile
  Properties:
    Name: "43013 JPSS Asynchronous Data"
    MinimumViableContactDurationSeconds: 180
    TrackingConfigArn: !Ref TrackingConfig
    DataflowEdges:
      - Source: !Ref JpssDownlinkDigIfAntennaConfig
        Destination: !Ref S3RecordingConfig
```

함께 넣기

위의 리소스를 사용하면 온보딩된 모든에서 비동기 데이터 전송을 위해 JPSS-1 연락처를 예약할 수 있습니다 AWS Ground Station [AWS Ground Station 위치](#).

다음은이 섹션에 설명된 모든 리소스를 직접 사용할 수 있는 단일 CloudFormation 템플릿으로 결합한 전체 템플릿입니다 CloudFormation.

라는 CloudFormation 템플릿에는 Amazon S3 버킷과 고객 응대를 예약하고 VITA-49 Signal/IP 다이렉트 브로드캐스트 데이터를 수신하는 데 필요한 AWS Ground Station 리소스가 AquaSnppJpss-1TerraDigIfS3DataDelivery.yml 포함되어 있습니다.

Aqua, SNPP, JPSS-1/NOAA-20 및 Terra가 계정에 온보딩되지 않은 경우 섹션을 참조하세요 [위성 온보딩](#).

Note

유효한 자격 AWS 증명을 사용하여 Amazon S3 버킷을 온보딩하는 고객에 액세스하여 템플릿에 액세스할 수 있습니다. 아래 링크는 리전별 Amazon S3 버킷을 사용합니다. us-west-2 스택을 생성할 CloudFormation 해당 리전을 나타내도록 리전 코드를 변경합니다.

또한 다음 지침은 YAML을 사용합니다. 그러나 템플릿은 YAML 형식과 JSON 형식으로 모두 제공됩니다. JSON을 사용하려면 템플릿을 다운로드할 때 `.json` 대신 `.yaml` 파일 확장명을 로 바꿉니다.

를 사용하여 템플릿을 다운로드하려면 다음 명령을 AWS CLI 사용하여 실행합니다.

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpss-1TerraDigIfS3DataDelivery.yaml .
```

브라우저에서 다음 URL로 이동하면 콘솔에서 템플릿을 보고 다운로드할 수 있습니다.

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss-1TerraDigIfS3DataDelivery.yaml
```

다음 링크를 CloudFormation 사용하여에서 직접 템플릿을 지정할 수 있습니다.

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss-1TerraDigIfS3DataDelivery.yaml
```

데이터 흐름 엔드포인트(협대역)를 사용하는 퍼블릭 브로드캐스트 위성

이 예제는 사용 설명서의 [JPSS-1 - 퍼블릭 브로드캐스트 위성\(PBS\) - 평가](#) 섹션에서 수행된 분석을 기반으로 합니다.

이 예제를 완료하려면 시나리오를 가정해야 합니다. 즉, HRD 통신 경로를 디지털 중간 주파수(DigIF)로 캡처하고 SDR을 사용하여 Amazon EC2 인스턴스의 데이터 흐름 엔드포인트 애플리케이션에서 수신하는 대로 처리해야 합니다.

통신 경로

이 섹션은 시작하기 [데이터 흐름 통신 경로 계획](#)를 나타냅니다. 이 예제에서는 CloudFormation 템플릿에 파라미터 및 리소스 섹션이라는 두 섹션을 생성합니다.

Note

CloudFormation 템플릿의 내용에 대한 자세한 내용은 [템플릿 섹션](#)을 참조하세요.

파라미터 섹션에서 다음 파라미터를 추가합니다. CloudFormation 콘솔을 통해 스택을 생성할 때 이러한 값에 대한 값을 지정합니다.

Parameters:

EC2Key:

Description: The SSH key used to access the EC2 receiver instance. Choose any SSH key if you are not creating an EC2 receiver instance. For instructions on how to create an SSH key see <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html>

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

ReceiverAMI:

Description: The Ground Station DDX AMI ID you want to use. Please note that AMIs are region specific. For instructions on how to retrieve an AMI see <https://docs.aws.amazon.com/ground-station/latest/ug/dataflows.ec2-configuration.html#dataflows.ec2-configuration.amis>

Type: AWS::EC2::Image::Id

 Note

키 페어를 생성하고 Amazon EC2 EC2Key 파라미터의 이름을 제공해야 합니다. [Amazon EC2 인스턴스에 대한 키 페어 생성을 참조하세요.](#)

또한 CloudFormation 스택을 생성할 때 올바른 리전별 AMI ID를 제공해야 합니다. [AWS Ground Station Amazon Machine Image\(AMIs\)](#) 섹션을 참조하세요.

나머지 템플릿 코드 조각은 템플릿의 리소스 섹션에 속합니다 CloudFormation .

Resources:

Resources that you would like to create should be placed within the resource section.

EC2 인스턴스에 단일 통신 경로를 전달하는 시나리오를 고려할 때 단일 동기식 전송 경로가 있습니다. [동기식 데이터 전송](#) 섹션에 따라 데이터 흐름 엔드포인트 애플리케이션을 사용하여 Amazon EC2 인스턴스를 설정 및 구성하고 하나 이상의 데이터 흐름 엔드포인트 그룹을 생성해야 합니다.

```

# The EC2 instance that will send/receive data to/from your satellite using AWS
Ground Station.
ReceiverInstance:
  Type: AWS::EC2::Instance
  Properties:
    DisableApiTermination: false
    IamInstanceProfile: !Ref GeneralInstanceProfile
    ImageId: !Ref ReceiverAMI
    InstanceType: m5.4xlarge
    KeyName: !Ref EC2Key
    Monitoring: true
    PlacementGroupName: !Ref ClusterPlacementGroup
    SecurityGroupIds:
      - Ref: InstanceSecurityGroup
    SubnetId: !Ref ReceiverSubnet
    BlockDeviceMappings:
      - DeviceName: /dev/xvda
        Ebs:
          VolumeType: gp2
          VolumeSize: 40
    Tags:
      - Key: Name
        Value: !Join [ "-", [ "Receiver" , !Ref "AWS::StackName" ] ]
  UserData:
    Fn::Base64:
      |
      #!/bin/bash
      exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console)
2>&1
      echo `date +%F %R:%S` "INFO: Logging Setup" >&2

      GROUND_STATION_DIR="/opt/aws/groundstation"
      GROUND_STATION_BIN_DIR="${GROUND_STATION_DIR}/bin"
      STREAM_CONFIG_PATH="${GROUND_STATION_DIR}/customer_stream_config.json"

      echo "Creating ${STREAM_CONFIG_PATH}"
      cat << STREAM_CONFIG > "${STREAM_CONFIG_PATH}"
      {
        "ddx_streams": [
          {
            "streamName": "Downlink",
            "maximumWanRate": 4000000000,
            "lanConfigDevice": "lo",
            "lanConfigPort": 50000,

```

```

        "wanConfigDevice": "eth1",
        "wanConfigPort": 55888,
        "isUplink": false
    }
]
}
STREAM_CONFIG

echo "Waiting for dataflow endpoint application to start"
while netstat -lnt | awk '$4 ~ /:80$/ {exit 1}'; do sleep 10; done

OS_VERSION_ID=$( [ -f /etc/os-release ] && . /etc/os-release && echo
"${VERSION_ID}")
if [ "${OS_VERSION_ID}" = "2023" ]; then PYTHON_BIN="python3"; else
PYTHON_BIN="python"; fi

echo "Configuring dataflow endpoint application streams"
streams_configured=false
for attempt in $(seq 1 6); do
    if ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/configure_streams.py" --
configFileName "${STREAM_CONFIG_PATH}"; then
        streams_configured=true
        break
    fi
    echo "configure_streams attempt ${attempt} failed; retrying in 10s"
    sleep 10
done
if [ "${streams_configured}" = "true" ]; then
    sleep 2
    ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/save_default_config.py"
else
    echo "ERROR: configure_streams failed after ${attempt} attempts; not saving
a default config"
fi

exit 0

# The AWS Ground Station Dataflow Endpoint Group that defines the endpoints that AWS
Ground
# Station will use to send/receive data to/from your satellite.
DataflowEndpointGroup:
  Type: AWS::GroundStation::DataflowEndpointGroup
  Properties:
    ContactPostPassDurationSeconds: 180

```

```

    ContactPrePassDurationSeconds: 120
    EndpointDetails:
      - Endpoint:
          Name: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ] # needs to
match DataflowEndpointConfig name
          Address:
            Name: !GetAtt ReceiverInstanceNetworkInterface.PrimaryPrivateIpAddress
            Port: 55888
          SecurityDetails:
            SecurityGroupIds:
              - Ref: "DataflowEndpointSecurityGroup"
            SubnetIds:
              - !Ref ReceiverSubnet
            RoleArn: !GetAtt DataDeliveryServiceRole.Arn

# The security group for your EC2 instance.
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: AWS Ground Station receiver instance security group.
    VpcId: !Ref ReceiverVPC
    SecurityGroupIngress:
      # To allow SSH access to the instance, add another rule allowing tcp port 22
from your CidrIp
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        SourceSecurityGroupId: !Ref DataflowEndpointSecurityGroup
        Description: "AWS Ground Station Downlink Stream"

# The security group that the ENI created by AWS Ground Station belongs to.
DataflowEndpointSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security Group for AWS Ground Station registration of Dataflow
Endpoint Groups
    VpcId: !Ref ReceiverVPC
    SecurityGroupEgress:
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        CidrIp: 10.0.0.0/8
        Description: "AWS Ground Station Downlink Stream To 10/8"
      - IpProtocol: udp

```

```
    FromPort: 55888
    ToPort: 55888
    CidrIp: 172.16.0.0/12
    Description: "AWS Ground Station Downlink Stream To 172.16/12"
  - IpProtocol: udp
    FromPort: 55888
    ToPort: 55888
    CidrIp: 192.168.0.0/16
    Description: "AWS Ground Station Downlink Stream To 192.168/16"
```

The placement group in which your EC2 instance is placed.

ClusterPlacementGroup:

Type: AWS::EC2::PlacementGroup

Properties:

Strategy: cluster

ReceiverVPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: **"10.0.0.0/16"**

Tags:

- Key: "Name"

Value: "AWS Ground Station - PBS to dataflow endpoint Example VPC"

- Key: "Description"

Value: "VPC for EC2 instance receiving AWS Ground Station data"

ReceiverSubnet:

Type: AWS::EC2::Subnet

Properties:

Ensure your CidrBlock will always have at least one available IP address per dataflow endpoint.

See <https://docs.aws.amazon.com/vpc/latest/userguide/subnet-sizing.html> for subnet sizing guidelines.

CidrBlock: **"10.0.0.0/24"**

Tags:

- Key: "Name"

Value: "AWS Ground Station - PBS to dataflow endpoint Example Subnet"

- Key: "Description"

Value: "Subnet for EC2 instance receiving AWS Ground Station data"

VpcId: !Ref ReceiverVPC

An ENI providing a fixed IP address for AWS Ground Station to connect to.

ReceiverInstanceNetworkInterface:

Type: AWS::EC2::NetworkInterface

Properties:

Description: Floating network interface providing a fixed IP address for AWS Ground Station to connect to.

GroupSet:

- !Ref InstanceSecurityGroup

SubnetId: !Ref ReceiverSubnet

Attach the ENI to the EC2 instance.

ReceiverInstanceInterfaceAttachment:

Type: AWS::EC2::NetworkInterfaceAttachment

Properties:

DeleteOnTermination: false

DeviceIndex: "1"

InstanceId: !Ref ReceiverInstance

NetworkInterfaceId: !Ref ReceiverInstanceNetworkInterface

또한가 계정에 탄력적 네트워크 인터페이스(ENI)를 생성할 수 있도록 적절한 정책 및 역할을 AWS Ground Station 생성해야 합니다.

AWS Ground Station assumes this role to create/delete ENIs in your account in order to stream data.

DataDeliveryServiceRole:

Type: AWS::IAM::Role

Properties:**Policies:**

- PolicyDocument:

Statement:

- Action:

- ec2:CreateNetworkInterface

- ec2>DeleteNetworkInterface

- ec2:CreateNetworkInterfacePermission

- ec2>DeleteNetworkInterfacePermission

- ec2:DescribeSubnets

- ec2:DescribeVpcs

- ec2:DescribeSecurityGroups

Effect: Allow

Resource: '*'

Version: '2012-10-17'

PolicyName: DataDeliveryServicePolicy

AssumeRolePolicyDocument:

Version: 2012-10-17

```

Statement:
  - Effect: Allow
    Principal:
      Service:
        - groundstation.amazonaws.com
    Action:
      - sts:AssumeRole

# The EC2 instance assumes this role.
InstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Principal:
            Service:
              - "ec2.amazonaws.com"
          Action:
            - "sts:AssumeRole"
    Path: "/"
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
      - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
      - arn:aws:iam::aws:policy/service-role/AmazonEC2RoleforSSM

# The instance profile for your EC2 instance.
GeneralInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref InstanceRole

```

AWS Ground Station 구성

이 섹션은 시작하기 [구성 생성](#)를 나타냅니다.

자동 추적을 사용하여에 대한 기본 설정을 지정하려면 추적 구성이 필요합니다. PREFERRED를 자동 추적으로 선택하면 신호 품질이 향상될 수 있지만 충분한 JPSS-1 에페메리스 품질로 인해 신호 품질을 충족할 필요는 없습니다.

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"
```

통신 경로를 기반으로 위성 부분을 나타내는 안테나 다운링크 구성과 엔드포인트 세부 정보를 정의하는 데이터 흐름 엔드포인트 그룹을 참조하는 데이터 흐름 엔드포인트 구성을 정의해야 합니다.

```
# The AWS Ground Station Antenna Downlink Config that defines the frequency spectrum
used to
# downlink data from your satellite.
SnppJpssDownlinkDigIfAntennaConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "SNPP JPSS Downlink DigIF Antenna Config"
    ConfigData:
      AntennaDownlinkConfig:
        SpectrumConfig:
          Bandwidth:
            Units: "MHz"
            Value: 30
          CenterFrequency:
            Units: "MHz"
            Value: 7812
          Polarization: "RIGHT_HAND"

# The AWS Ground Station Dataflow Endpoint Config that defines the endpoint used to
downlink data
# from your satellite.
DownlinkDigIfEndpointConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "Aqua SNPP JPSS Downlink DigIF Endpoint Config"
    ConfigData:
      DataflowEndpointConfig:
        DataflowEndpointName: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ]
        DataflowEndpointRegion: !Ref AWS::Region
```

AWS Ground Station 미션 프로파일

이 섹션은 시작하기 [미션 프로파일 생성](#)를 나타냅니다.

이제 연결된 구성이 있으므로 이를 사용하여 데이터 흐름을 구성할 수 있습니다. 나머지 파라미터에 기본값을 사용합니다.

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnppJpssMissionProfile:
  Type: AWS::GroundStation::MissionProfile
  Properties:
    Name: "37849 SNPP And 43013 JPSS"
    ContactPrePassDurationSeconds: 120
    ContactPostPassDurationSeconds: 60
    MinimumViableContactDurationSeconds: 180
    TrackingConfigArn: !Ref TrackingConfig
    DataflowEdges:
      - Source: !Ref SnppJpssDownlinkDigIfAntennaConfig
        Destination: !Ref DownlinkDigIfEndpointConfig
```

함께 넣기

위의 리소스를 사용하면 온보딩된 모든에서 동기식 데이터 전송을 위해 JPSS-1 연락처를 예약할 수 있습니다 AWS Ground Station [AWS Ground Station 위치](#).

다음은이 섹션에 설명된 모든 리소스를 직접 사용할 수 있는 단일 CloudFormation 템플릿으로 결합한 전체 템플릿입니다 CloudFormation.

라는 CloudFormation 템플릿AquaSnppJpssTerraDigIF.yml은 Aqua, SNPP, JPSS-1/NOAA-20 및 Terra 위성에 대한 디지털화된 중간 주파수(DigIF) 데이터 수신을 빠르게 시작할 수 있도록 설계되었습니다. 여기에는 Amazon EC2 인스턴스와 원시 DigIF 다이렉트 브로드캐스트 데이터를 수신하는 데 필요한 CloudFormation 리소스가 포함되어 있습니다.

Aqua, SNPP, JPSS-1/NOAA-20 및 Terra가 계정에 온보딩되지 않은 경우 섹션을 참조하세요 [위성 온보딩](#).

Note

유효한 자격 AWS 증명을 사용하여 Amazon S3 버킷을 온보딩하는 고객에 액세스하여 템플릿에 액세스할 수 있습니다. 아래 링크는 리전별 Amazon S3 버킷을 사용합니다. us-west-2 스택을 생성할 CloudFormation 해당 리전을 나타내도록 리전 코드를 변경합니다.

또한 다음 지침에서는 YAML을 사용합니다. 그러나 템플릿은 YAML 형식과 JSON 형식으로 모두 제공됩니다. JSON을 사용하려면 템플릿을 다운로드할 .json 때 .yaml 파일 확장명을로 바꿉니다.

를 사용하여 템플릿을 다운로드하려면 다음 명령을 AWS CLI 사용합니다.

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/
AquaSnppJpssTerraDigIF.yaml .
```

브라우저에서 다음 URL로 이동하면 콘솔에서 템플릿을 보고 다운로드할 수 있습니다.

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpssTerraDigIF.yaml
```

다음 링크를 CloudFormation 사용하여에서 직접 템플릿을 지정할 수 있습니다.

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/
AquaSnppJpssTerraDigIF.yaml
```

템플릿이 정의하는 추가 리소스는 무엇입니까?

AquaSnppJpssTerraDigIF 템플릿에는 다음과 같은 추가 리소스가 포함되어 있습니다.

- (선택 사항) CloudWatch 이벤트 트리거 - AWS Lambda 고객 응대 AWS Ground Station 전후에에서 전송한 CloudWatch 이벤트를 사용하여 트리거되는 함수입니다. AWS Lambda 함수는 수신기 인스턴스를 시작하고 선택적으로 중지합니다.
- (선택 사항) 접촉에 대한 EC2 확인 - Lambda를 사용하여 Amazon EC2 인스턴스의 SNS 알림을 통한 접촉에 대한 확인 시스템을 설정하는 옵션입니다. 이때 현재 사용량에 따라 요금이 부과될 수 있다는 점에 유의하세요.
- Ground Station Amazon 머신 이미지 검색 Lambda - 인스턴스와 원하는 AMI에 설치할 소프트웨어를 선택할 수 있는 옵션입니다. 소프트웨어 옵션에는 소프트웨어 정의 라디오가 있거나 없는 데이터

흐름 엔드포인트 애플리케이션이 포함됩니다. 이러한 옵션은 추가 소프트웨어 업데이트 및 기능이 릴리스됨에 따라 변경될 수 있습니다.

- 추가 미션 프로파일 - 추가 퍼블릭 브로드캐스트 위성(Aqua, SNPP 및 Terra)을 위한 미션 프로파일입니다.
- 추가 안테나 다운링크 구성 - 추가 퍼블릭 브로드캐스트 위성(Aqua, SNPP 및 Terra)에 대한 안테나 다운링크 구성입니다.

이 템플릿의 위성에 대한 값과 매개 변수가 이미 채워져 있습니다. 이러한 파라미터를 사용하면 이러한 위성에서 AWS Ground Station 즉시 쉽게 사용할 수 있습니다. 이 템플릿을 사용할 AWS Ground Station 때를 사용하기 위해 자체 값을 구성할 필요가 없습니다. 그러나 사용 사례에 맞게 템플릿을 작동하도록 값을 사용자 지정할 수 있습니다.

내 데이터는 어디에서 수신합니까?

데이터 흐름 엔드포인트 그룹은 템플릿의 일부로 생성되는 수신기 인스턴스 네트워크 인터페이스를 사용하도록 설정됩니다. 수신기 인스턴스는 데이터 흐름 엔드포인트 애플리케이션을 사용하여 데이터 흐름 엔드포인트 AWS Ground Station에 의해 정의된 포트의에서 데이터 스트림을 수신합니다. 수신된 데이터는 수신기 인스턴스의 루프백 어댑터에서 UDP 포트 50000을 통해 사용할 수 있습니다. 데이터 흐름 엔드포인트 그룹 설정에 대한 자세한 내용은 [AWS::GroundStation::DataflowEndpointGroup](#)을 참조하세요.

데이터 흐름 엔드포인트를 사용하는 퍼블릭 브로드캐스트 위성(변조 및 디코딩됨)

이 예제는 사용 설명서의 [JPSS-1 - 퍼블릭 브로드캐스트 위성\(PBS\) - 평가](#) 섹션에서 수행된 분석을 기반으로 합니다.

이 예제를 완료하려면 dataflow 엔드포인트를 사용하여 HRD 통신 경로를 복조 및 디코딩된 다이렉트 브로드캐스트 데이터로 캡처하려는 시나리오를 가정해야 합니다. 이 예제는 NASA Direct Readout Labs 소프트웨어(RT-STPS 및 IPOPP)를 사용하여 데이터를 처리하려는 경우 좋은 출발점입니다.

통신 경로

이 섹션은 시작하기 [데이터 흐름 통신 경로 계획](#)를 나타냅니다. 이 예제에서는 CloudFormation 템플릿에 파라미터 및 리소스 섹션이라는 두 섹션을 생성합니다.

Note

CloudFormation 템플릿의 내용에 대한 자세한 내용은 [템플릿 섹션](#)을 참조하세요.

파라미터 섹션에서 다음 파라미터를 추가합니다. CloudFormation 콘솔을 통해 스택을 생성할 때 이러한 값에 대한 값을 지정합니다.

Parameters:**EC2Key:**

Description: The SSH key used to access the EC2 receiver instance. Choose any SSH key if you are not creating an EC2 receiver instance. For instructions on how to create an SSH key see <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html>

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

ReceiverAMI:

Description: The Ground Station DDX AMI ID you want to use. Please note that AMIs are region specific. For instructions on how to retrieve an AMI see <https://docs.aws.amazon.com/ground-station/latest/ug/dataflows.ec2-configuration.html#dataflows.ec2-configuration.amis>

Type: AWS::EC2::Image::Id

Note

키 페어를 생성하고 Amazon EC2 EC2Key 파라미터의 이름을 제공해야 합니다. [Amazon EC2 인스턴스에 대한 키 페어 생성을 참조하세요](#).

또한 CloudFormation 스택을 생성할 때 올바른 리전별 AMI ID를 제공해야 합니다. [AWS Ground Station Amazon Machine Image\(AMIs\)](#) 섹션을 참조하세요.

나머지 템플릿 코드 조각은 템플릿의 리소스 섹션에 속합니다 CloudFormation .

Resources:

Resources that you would like to create should be placed within the resource section.

EC2 인스턴스에 단일 통신 경로를 전달하는 시나리오를 고려할 때 단일 동기식 전송 경로가 있습니다. [동기식 데이터 전송](#) 섹션에 따라 데이터 흐름 엔드포인트 애플리케이션을 사용하여 Amazon EC2 인스턴스를 설정 및 구성하고 하나 이상의 데이터 흐름 엔드포인트 그룹을 생성해야 합니다.

```
# The EC2 instance that will send/receive data to/from your satellite using AWS
Ground Station.
ReceiverInstance:
  Type: AWS::EC2::Instance
  Properties:
    DisableApiTermination: false
    IamInstanceProfile: !Ref GeneralInstanceProfile
    ImageId: !Ref ReceiverAMI
    InstanceType: m5.4xlarge
    KeyName: !Ref EC2Key
    Monitoring: true
    PlacementGroupName: !Ref ClusterPlacementGroup
    SecurityGroupIds:
      - Ref: InstanceSecurityGroup
    SubnetId: !Ref ReceiverSubnet
    BlockDeviceMappings:
      - DeviceName: /dev/xvda
        Ebs:
          VolumeType: gp2
          VolumeSize: 40
    Tags:
      - Key: Name
        Value: !Join [ "-", [ "Receiver" , !Ref "AWS::StackName" ] ]
  UserData:
    Fn::Base64:
      |
      #!/bin/bash
      exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console)
2>&1

      echo `date +%F %R:%S` "INFO: Logging Setup" >&2

      GROUND_STATION_DIR="/opt/aws/groundstation"
      GROUND_STATION_BIN_DIR="${GROUND_STATION_DIR}/bin"
      STREAM_CONFIG_PATH="${GROUND_STATION_DIR}/customer_stream_config.json"

      echo "Creating ${STREAM_CONFIG_PATH}"
      cat << STREAM_CONFIG > "${STREAM_CONFIG_PATH}"
      {
```

```

    "ddx_streams": [
        {
            "streamName": "Downlink",
            "maximumWanRate": 4000000000,
            "lanConfigDevice": "lo",
            "lanConfigPort": 50000,
            "wanConfigDevice": "eth1",
            "wanConfigPort": 55888,
            "isUplink": false
        }
    ]
}
STREAM_CONFIG

echo "Waiting for dataflow endpoint application to start"
while netstat -lnt | awk '$4 ~ /:80$/ {exit 1}'; do sleep 10; done

OS_VERSION_ID=$( [ -f /etc/os-release ] && . /etc/os-release && echo
"${VERSION_ID}")
if [ "${OS_VERSION_ID}" = "2023" ]; then PYTHON_BIN="python3"; else
PYTHON_BIN="python"; fi

echo "Configuring dataflow endpoint application streams"
streams_configured=false
for attempt in $(seq 1 6); do
    if ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/configure_streams.py" --
configFileName "${STREAM_CONFIG_PATH}"; then
        streams_configured=true
        break
    fi
    echo "configure_streams attempt ${attempt} failed; retrying in 10s"
    sleep 10
done
if [ "${streams_configured}" = "true" ]; then
    sleep 2
    ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/save_default_config.py"
else
    echo "ERROR: configure_streams failed after ${attempt} attempts; not saving
a default config"
fi

exit 0

```

```

# The AWS Ground Station Dataflow Endpoint Group that defines the endpoints that AWS
Ground
# Station will use to send/receive data to/from your satellite.
DataflowEndpointGroup:
  Type: AWS::GroundStation::DataflowEndpointGroup
  Properties:
    ContactPostPassDurationSeconds: 180
    ContactPrePassDurationSeconds: 120
    EndpointDetails:
      - Endpoint:
          Name: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ] # needs to
match DataflowEndpointConfig name
          Address:
            Name: !GetAtt ReceiverInstanceNetworkInterface.PrimaryPrivateIpAddress
            Port: 55888
    SecurityDetails:
      SecurityGroupIds:
        - Ref: "DataflowEndpointSecurityGroup"
      SubnetIds:
        - !Ref ReceiverSubnet
      RoleArn: !GetAtt DataDeliveryServiceRole.Arn

# The security group that the ENI created by AWS Ground Station belongs to.
DataflowEndpointSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security Group for AWS Ground Station registration of Dataflow
Endpoint Groups
    VpcId: !Ref ReceiverVPC
    SecurityGroupEgress:
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        CidrIp: 10.0.0.0/8
        Description: "AWS Ground Station Downlink Stream To 10/8"
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        CidrIp: 172.16.0.0/12
        Description: "AWS Ground Station Downlink Stream To 172.16/12"
      - IpProtocol: udp
        FromPort: 55888

```

```
ToPort: 55888
CidrIp: 192.168.0.0/16
Description: "AWS Ground Station Downlink Stream To 192.168/16"

# The placement group in which your EC2 instance is placed.
ClusterPlacementGroup:
  Type: AWS::EC2::PlacementGroup
  Properties:
    Strategy: cluster

# The security group for your EC2 instance.
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: AWS Ground Station receiver instance security group.
    VpcId: !Ref ReceiverVPC
    SecurityGroupIngress:
      # To allow SSH access to the instance, add another rule allowing tcp port 22
      from your CidrIp
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        SourceSecurityGroupId: !Ref DataflowEndpointSecurityGroup
        Description: "AWS Ground Station Downlink Stream"

ReceiverVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: "10.0.0.0/16"
    Tags:
      - Key: "Name"
        Value: "AWS Ground Station - PBS to dataflow endpoint Demod Decode Example
VPC"
      - Key: "Description"
        Value: "VPC for EC2 instance receiving AWS Ground Station data"

ReceiverSubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock: "10.0.0.0/24"
    Tags:
      - Key: "Name"
        Value: "AWS Ground Station - PBS to dataflow endpoint Demod Decode Example
Subnet"
```

```

    - Key: "Description"
      Value: "Subnet for EC2 instance receiving AWS Ground Station data"
    VpcId: !Ref ReceiverVPC

# An ENI providing a fixed IP address for AWS Ground Station to connect to.
ReceiverInstanceNetworkInterface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    Description: Floating network interface providing a fixed IP address for AWS
Ground Station to connect to.
    GroupSet:
      - !Ref InstanceSecurityGroup
    SubnetId: !Ref ReceiverSubnet

# Attach the ENI to the EC2 instance.
ReceiverInstanceInterfaceAttachment:
  Type: AWS::EC2::NetworkInterfaceAttachment
  Properties:
    DeleteOnTermination: false
    DeviceIndex: "1"
    InstanceId: !Ref ReceiverInstance
    NetworkInterfaceId: !Ref ReceiverInstanceNetworkInterface

# The instance profile for your EC2 instance.
GeneralInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref InstanceRole

```

또한가 계정에서 탄력적 네트워크 인터페이스(ENI)를 AWS Ground Station 생성할 수 있도록 허용하는 적절한 정책, 역할 및 프로필이 필요합니다.

```

# AWS Ground Station assumes this role to create/delete ENIs in your account in order
to stream data.
DataDeliveryServiceRole:
  Type: AWS::IAM::Role
  Properties:
    Policies:
      - PolicyDocument:
          Statement:

```

```
- Action:
  - ec2:CreateNetworkInterface
  - ec2>DeleteNetworkInterface
  - ec2:CreateNetworkInterfacePermission
  - ec2>DeleteNetworkInterfacePermission
  - ec2:DescribeSubnets
  - ec2:DescribeVpcs
  - ec2:DescribeSecurityGroups
Effect: Allow
Resource: '*'
Version: '2012-10-17'
PolicyName: DataDeliveryServicePolicy
AssumeRolePolicyDocument:
  Version: 2012-10-17
  Statement:
    - Effect: Allow
      Principal:
        Service:
          - groundstation.amazonaws.com
      Action:
        - sts:AssumeRole
```

The EC2 instance assumes this role.

```
InstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Principal:
            Service:
              - "ec2.amazonaws.com"
          Action:
            - "sts:AssumeRole"
    Path: "/"
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
      - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
      - arn:aws:iam::aws:policy/service-role/AmazonEC2RoleforSSM
```

AWS Ground Station 구성

이 섹션은 [구성 생성](#) 사용 설명서를 나타냅니다.

자동 추적을 사용하여에 대한 기본 설정을 지정하려면 추적 구성이 필요합니다. PREFERRED를 자동 추적으로 선택하면 신호 품질이 향상될 수 있지만 충분한 JPSS-1 에페메리스 품질로 인해 신호 품질을 충족할 필요는 없습니다.

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"
```

통신 경로를 기반으로 위성 부분을 나타내는 antenna-downlink-demod-decode 구성과 엔드포인트 세부 정보를 정의하는 데이터 흐름 엔드포인트 그룹을 참조하는 데이터 흐름 엔드포인트 구성을 정의해야 합니다.

Note

DemodulationConfig, 및의 값을 설정하는 방법에 대한 자세한 내용은 섹션을 참조DecodeConfig하세요 [안테나 다운링크 복조 디코드 구성](#).

```
# The AWS Ground Station Antenna Downlink Config that defines the frequency spectrum
used to
# downlink data from your satellite.
JpssDownlinkDemodDecodeAntennaConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Downlink Demod Decode Antenna Config"
    ConfigData:
      AntennaDownlinkDemodDecodeConfig:
        SpectrumConfig:
          CenterFrequency:
            Value: 7812
```

```
    Units: "MHz"
    Polarization: "RIGHT_HAND"
    Bandwidth:
      Value: 30
      Units: "MHz"
  DemodulationConfig:
    UnvalidatedJSON: '{
      "type": "QPSK",
      "qpsk": {
        "carrierFrequencyRecovery": {
          "centerFrequency": {
            "value": 7812,
            "units": "MHz"
          },
          "range": {
            "value": 250,
            "units": "kHz"
          }
        },
        "symbolTimingRecovery": {
          "symbolRate": {
            "value": 15,
            "units": "Msps"
          },
          "range": {
            "value": 0.75,
            "units": "ksps"
          },
          "matchedFilter": {
            "type": "ROOT_RAISED_COSINE",
            "rolloffFactor": 0.5
          }
        }
      }
    }'
  DecodeConfig:
    UnvalidatedJSON: '{
      "edges": [
        {
          "from": "I-Ingress",
          "to": "IQ-Recombiner"
        },
        {
          "from": "Q-Ingress",
```

```

        "to": "IQ-Recombiner"
    },
    {
        "from": "IQ-Recombiner",
        "to": "CcsdsViterbiDecoder"
    },
    {
        "from": "CcsdsViterbiDecoder",
        "to": "NrzmDecoder"
    },
    {
        "from": "NrzmDecoder",
        "to": "UncodedFramesEgress"
    }
],
"nodeConfigs": {
    "I-Ingress": {
        "type": "CODED_SYMBOLS_INGRESS",
        "codedSymbolsIngress": {
            "source": "I"
        }
    },
    "Q-Ingress": {
        "type": "CODED_SYMBOLS_INGRESS",
        "codedSymbolsIngress": {
            "source": "Q"
        }
    },
    "IQ-Recombiner": {
        "type": "IQ_RECOMBINER"
    },
    "CcsdsViterbiDecoder": {
        "type": "CCSDS_171_133_VITERBI_DECODER",
        "ccsds171133ViterbiDecoder": {
            "codeRate": "ONE_HALF"
        }
    },
    "NrzmDecoder": {
        "type": "NRZ_M_DECODER"
    },
    "UncodedFramesEgress": {
        "type": "UNCODED_FRAMES_EGRESS"
    }
}
}

```

```
}'
```

```
# The AWS Ground Station Dataflow Endpoint Config that defines the endpoint used to
downlink data
# from your satellite.
DownlinkDemodDecodeEndpointConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "Aqua SNPP JPSS Downlink Demod Decode Endpoint Config"
    ConfigData:
      DataflowEndpointConfig:
        DataflowEndpointName: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ]
        DataflowEndpointRegion: !Ref AWS::Region
```

AWS Ground Station 미션 프로파일

이 섹션은 [미션 프로파일 생성](#) 사용 설명서를 나타냅니다.

이제 연결된 구성이 있으므로 이를 사용하여 데이터 흐름을 구성할 수 있습니다. 나머지 파라미터에 기본값을 사용합니다.

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnppJpssMissionProfile:
  Type: AWS::GroundStation::MissionProfile
  Properties:
    Name: "37849 SNPP And 43013 JPSS"
    ContactPrePassDurationSeconds: 120
    ContactPostPassDurationSeconds: 60
    MinimumViableContactDurationSeconds: 180
    TrackingConfigArn: !Ref TrackingConfig
    DataflowEdges:
      - Source: !Join [ "/", [ !Ref JpssDownlinkDemodDecodeAntennaConfig,
"UncodedFramesEgress" ] ]
        Destination: !Ref DownlinkDemodDecodeEndpointConfig
```

함께 넣기

위의 리소스를 사용하면 온보딩된 모든에서 동기식 데이터 전송을 위해 JPSS-1 연락처를 예약할 수 있습니다 AWS Ground Station [AWS Ground Station 위치](#).

다음은이 섹션에 설명된 모든 리소스를 직접 사용할 수 있는 단일 CloudFormation 템플릿으로 결합한 전체 템플릿입니다 CloudFormation.

라는 CloudFormation 템플릿AquaSnppJpss.yml은 Aqua, SNPP 및 JPSS-1/NOAA-20 위성에 대한 데이터 수신을 시작할 수 있는 빠른 액세스를 제공하도록 설계되었습니다. 여기에는 Amazon EC2 인스턴스와 고객 응대를 예약하고 복조 및 디코딩된 다이렉트 브로드캐스트 데이터를 수신하는 데 필요한 AWS Ground Station 리소스가 포함되어 있습니다.

Aqua, SNPP, JPSS-1/NOAA-20 및 Terra가 계정에 온보딩되지 않은 경우 섹션을 참조하세요 [위성 온보딩](#).

Note

유효한 자격 AWS 증명을 사용하여 Amazon S3 버킷을 온보딩하는 고객에 액세스하여 템플릿에 액세스할 수 있습니다. 아래 링크는 리전별 Amazon S3 버킷을 사용합니다. us-west-2 스택을 생성할 CloudFormation 해당 리전을 나타내도록 리전 코드를 변경합니다. 또한 다음 지침에서는 YAML을 사용합니다. 그러나 템플릿은 YAML 형식과 JSON 형식으로 모두 제공됩니다. JSON을 사용하려면 템플릿을 다운로드할 .json 때 .yaml 파일 확장명을 로 바꿉니다.

를 사용하여 템플릿을 다운로드하려면 다음 명령을 AWS CLI사용합니다.

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yaml .
```

브라우저에서 다음 URL로 이동하면 콘솔에서 템플릿을 보고 다운로드할 수 있습니다.

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yaml
```

다음 링크를 CloudFormation 사용하여에서 직접 템플릿을 지정할 수 있습니다.

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss.yaml
```

템플릿이 정의하는 추가 리소스는 무엇입니까?

AquaSnppJpss 템플릿에는 다음과 같은 추가 리소스가 포함되어 있습니다.

- (선택 사항) CloudWatch 이벤트 트리거 - AWS Lambda 고객 응대 AWS Ground Station 전후에에서 전송한 CloudWatch 이벤트를 사용하여 트리거되는 함수입니다. AWS Lambda 함수는 수신기 인스턴스를 시작하고 선택적으로 중지합니다.
- (선택 사항) 접촉에 대한 EC2 확인 - Lambda를 사용하여 Amazon EC2 인스턴스의 SNS 알림을 통한 접촉에 대한 확인 시스템을 설정하는 옵션입니다. 이때 현재 사용량에 따라 요금이 부과될 수 있다는 점에 유의하세요.
- Ground Station Amazon 머신 이미지 검색 Lambda - 인스턴스와 원하는 AMI에 설치할 소프트웨어를 선택할 수 있는 옵션입니다. 소프트웨어 옵션에는 소프트웨어 정의 라디오가 있거나 없는 데이터 흐름 엔드포인트 애플리케이션이 포함됩니다. 광대역 DigIF 데이터 전송 및 AWS Ground Station 에이전트를 사용하려면 섹션을 참조하세요 [AWS Ground Station 에이전트\(와이드밴드\)를 활용하는 퍼블릭 브로드캐스트 위성](#). 이러한 옵션은 추가 소프트웨어 업데이트 및 기능이 릴리스됨에 따라 변경될 수 있습니다.
- 추가 미션 프로파일 - 추가 퍼블릭 브로드캐스트 위성(Aqua, SNPP 및 Terra)을 위한 미션 프로파일입니다.
- 추가 안테나 다운링크 구성 - 추가 퍼블릭 브로드캐스트 위성(Aqua, SNPP 및 Terra)에 대한 안테나 다운링크 구성입니다.

이 템플릿의 위성에 대한 값과 매개 변수가 이미 채워져 있습니다. 이러한 파라미터를 사용하면 이러한 위성에서 AWS Ground Station 즉시 쉽게 사용할 수 있습니다. 이 템플릿을 사용할 AWS Ground Station 때를 사용하기 위해 자체 값을 구성할 필요가 없습니다. 그러나 사용 사례에 맞게 템플릿을 작동하도록 값을 사용자 지정할 수 있습니다.

내 데이터는 어디에서 수신합니까?

데이터 흐름 엔드포인트 그룹은 템플릿의 일부로 생성되는 수신기 인스턴스 네트워크 인터페이스를 사용하도록 설정됩니다. 수신기 인스턴스는 데이터 흐름 엔드포인트 애플리케이션을 사용하여 데이터 흐름 엔드포인트 AWS Ground Station에 의해 정의된 포트의에서 데이터 스트림을 수신합니다. 수신된 데이터는 수신기 인스턴스의 루프백 어댑터에서 UDP 포트 50000을 통해 사용할 수 있습니다. 데이터 흐름 엔드포인트 그룹 설정에 대한 자세한 내용은 [AWS::GroundStation::DataflowEndpointGroup](#)을 참조하세요.

AWS Ground Station 에이전트(와이드밴드)를 활용하는 퍼블릭 브로드캐스트 위성

이 예제는 사용 설명서의 [JPSS-1 - 퍼블릭 브로드캐스트 위성\(PBS\) - 평가](#) 섹션에서 수행된 분석을 기반으로 합니다.

이 예제를 완료하려면 시나리오를 가정해야 합니다. 즉, HRD 통신 경로를 광대역 디지털 중간 주파수(DigIF)로 캡처하고 SDR을 사용하여 AWS Ground Station 에이전트가 Amazon EC2 인스턴스에서 수신한 대로 처리해야 합니다.

Note

실제 JPSS HRD 통신 경로 신호의 대역폭은 30MHz이지만 이 예제에서는 AWS Ground Station 에이전트가 수신할 올바른 경로를 통해 흐를 수 있도록 안테나 다운링크 구성을 100MHz 대역폭의 신호로 처리하도록 구성합니다.

통신 경로

이 섹션은 시작하기 [데이터 흐름 통신 경로 계획](#)를 나타냅니다. 이 예제에서는 CloudFormation 템플릿에 다른 예제인 매핑 섹션에서 사용되지 않은 추가 섹션이 필요합니다.

Note

CloudFormation 템플릿의 내용에 대한 자세한 내용은 [템플릿 섹션을](#) 참조하세요.

먼저 리전별 AWS Ground Station 접두사 목록에 대해 CloudFormation 템플릿에서 매핑 섹션을 설정합니다. 이렇게 하면 Amazon EC2 인스턴스 보안 그룹에서 접두사 목록을 쉽게 참조할 수 있습니다. 접두사 목록 사용에 대한 자세한 내용은 섹션을 참조하세요 [AWS Ground Station 에이전트를 사용한 VPC 구성](#).

Mappings:

PrefixListId:

us-east-2:

groundstation: pl-087f83ba4f34e3bea

us-west-2:

groundstation: pl-0cc36273da754ebdc

```

us-east-1:
  groundstation: pl-0e5696d987d033653
eu-central-1:
  groundstation: pl-03743f81267c0a85e
sa-east-1:
  groundstation: pl-098248765e9effc20
ap-northeast-2:
  groundstation: pl-059b3e0b02af70e4d
ap-southeast-1:
  groundstation: pl-0d9b804fe014a6a99
ap-southeast-2:
  groundstation: pl-08d24302b8c4d2b73
me-south-1:
  groundstation: pl-02781422c4c792145
eu-west-1:
  groundstation: pl-03fa6b266557b0d4f
eu-north-1:
  groundstation: pl-033e44023025215c0
af-south-1:
  groundstation: pl-0382d923a9d555425

```

파라미터 섹션에서 다음 파라미터를 추가합니다. CloudFormation 콘솔을 통해 스택을 생성할 때 이러한 값에 대한 값을 지정합니다.

Parameters:

EC2Key:

Description: The SSH key used to access the EC2 receiver instance. Choose any SSH key if you are not creating an EC2 receiver instance. For instructions on how to create an SSH key see <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html>

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

AZ:

Description: "The AvailabilityZone that the resources of this stack will be created in. (e.g. us-east-2a)"

Type: AWS::EC2::AvailabilityZone::Name

ReceiverAMI:

Description: The Ground Station Agent AMI ID you want to use. Please note that AMIs are region specific. For instructions on how to retrieve an AMI

see <https://docs.aws.amazon.com/ground-station/latest/ug/dataflows.ec2-configuration.html#dataflows.ec2-configuration.amis>

Type: AWS::EC2::Image::Id

Note

키 페어를 생성하고 Amazon EC2 EC2Key 파라미터의 이름을 제공해야 합니다. [Amazon EC2 인스턴스에 대한 키 페어 생성을 참조하세요.](#)

또한 CloudFormation 스택을 생성할 때 올바른 리전별 AMI ID를 제공해야 합니다. [AWS Ground Station Amazon Machine Image\(AMIs\)](#)을 참조하세요.

나머지 템플릿 코드 조각은 템플릿의 리소스 섹션에 속합니다 CloudFormation .

Resources:

Resources that you would like to create should be placed within the Resources section.

Amazon EC2 인스턴스에 단일 통신 경로를 전달하는 시나리오를 고려할 때 단일 동기식 전송 경로가 있음을 알 수 있습니다. [동기식 데이터 전송](#) 섹션에 따라 AWS Ground Station 에이전트를 사용하여 Amazon EC2 인스턴스를 설정 및 구성하고 하나 이상의 데이터 흐름 엔드포인트 그룹을 생성해야 합니다. 먼저 AWS Ground Station 에이전트에 대한 Amazon VPC를 설정합니다.

ReceiverVPC:

Type: AWS::EC2::VPC

Properties:

EnableDnsSupport: 'true'

EnableDnsHostnames: 'true'

CidrBlock: 10.0.0.0/16

Tags:

- Key: "Name"

Value: "AWS Ground Station Example - PBS to AWS Ground Station Agent VPC"

- Key: "Description"

Value: "VPC for EC2 instance receiving AWS Ground Station data"

PublicSubnet:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref ReceiverVPC

```
MapPublicIpOnLaunch: 'true'
AvailabilityZone: !Ref AZ
CidrBlock: 10.0.0.0/20
Tags:
  - Key: "Name"
    Value: "AWS Ground Station Example - PBS to AWS Ground Station Agent Public Subnet"
  - Key: "Description"
    Value: "Subnet for EC2 instance receiving AWS Ground Station data"

RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref ReceiverVPC
    Tags:
      - Key: Name
        Value: AWS Ground Station Example - RouteTable

RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref PublicSubnet

Route:
  Type: AWS::EC2::Route
  DependsOn: InternetGateway
  Properties:
    RouteTableId: !Ref RouteTable
    DestinationCidrBlock: '0.0.0.0/0'
    GatewayId: !Ref InternetGateway

InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: AWS Ground Station Example - Internet Gateway

GatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId: !Ref ReceiverVPC
```

```
InternetGatewayId: !Ref InternetGateway
```

Note

AWS Ground Station 에이전트가 지원하는 VPC 구성에 대한 자세한 내용은 [AWS Ground Station 에이전트 요구 사항 - VPC 다이어그램을 참조하세요](#).

다음으로 수신기 Amazon EC2 인스턴스를 설정합니다.

```
# The placement group in which your EC2 instance is placed.
ClusterPlacementGroup:
  Type: AWS::EC2::PlacementGroup
  Properties:
    Strategy: cluster

# This is required for the EIP if the receiver EC2 instance is in a private subnet.
# This ENI must exist in a public subnet, be attached to the receiver and be
associated with the EIP.
ReceiverInstanceNetworkInterface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    Description: Floating network interface
    GroupSet:
      - !Ref InstanceSecurityGroup
    SubnetId: !Ref PublicSubnet

# An EIP providing a fixed IP address for AWS Ground Station to connect to. Attach it
to the receiver instance created in the stack.
ReceiverInstanceElasticIp:
  Type: AWS::EC2::EIP
  Properties:
    Tags:
      - Key: Name
        Value: !Join [ "-", [ "EIP" , !Ref "AWS::StackName" ] ]

# Attach the ENI to the EC2 instance if using a separate public subnet.
# Requires the receiver instance to be in a public subnet (SubnetId should be the id
of a public subnet)
ReceiverNetworkInterfaceAttachment:
  Type: AWS::EC2::NetworkInterfaceAttachment
```

```

Properties:
  DeleteOnTermination: false
  DeviceIndex: 1
  InstanceId: !Ref ReceiverInstance
  NetworkInterfaceId: !Ref ReceiverInstanceNetworkInterface

# Associate EIP with the ENI if using a separate public subnet for the ENI.
ReceiverNetworkInterfaceElasticIpAssociation:
  Type: AWS::EC2::EIPAssociation
  Properties:
    AllocationId: !GetAtt [ReceiverInstanceElasticIp, AllocationId]
    NetworkInterfaceId: !Ref ReceiverInstanceNetworkInterface

# The EC2 instance that will send/receive data to/from your satellite using AWS
Ground Station.
ReceiverInstance:
  Type: AWS::EC2::Instance
  DependsOn: PublicSubnet
  Properties:
    DisableApiTermination: false
    IamInstanceProfile: !Ref GeneralInstanceProfile
    ImageId: !Ref ReceiverAMI
    AvailabilityZone: !Ref AZ
    InstanceType: c5.24xlarge
    KeyName: !Ref EC2Key
    Monitoring: true
    PlacementGroupName: !Ref ClusterPlacementGroup
    SecurityGroupIds:
      - Ref: InstanceSecurityGroup
    SubnetId: !Ref PublicSubnet
    Tags:
      - Key: Name
        Value: !Join [ "-", [ "Receiver" , !Ref "AWS::StackName" ] ]

# agentCpuCores list in the AGENT_CONFIG below defines the cores that the AWS
Ground Station Agent is allowed to run on. This list can be changed to suit your use-
case, however if the agent isn't supplied with enough cores data loss may occur.
UserData:
  Fn::Base64:
    Fn::Sub:
      - |
        #!/bin/bash
        yum -y update

        AGENT_CONFIG_PATH="/opt/aws/groundstation/etc/aws-gs-agent-config.json"

```

```

cat << AGENT_CONFIG > "$AGENT_CONFIG_PATH"
{
  "capabilities": [
    "arn:aws:groundstation:${AWS::Region}:${AWS::AccountId}:dataflow-
endpoint-group/${DataflowEndpointGroupId}"
  ],
  "device": {
    "privateIps": [
      "127.0.0.1"
    ],
    "publicIps": [
      "${EIP}"
    ],
    "agentCpuCores": [
24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,72,73,74,75,76,77,78,79,80,81,8
    ]
  }
}
AGENT_CONFIG

systemctl start aws-groundstation-agent
systemctl enable aws-groundstation-agent

# <Tuning Section Start>
# Visit the AWS Ground Station Agent Documentation in the User Guide for
more details and guidance updates

# Set IRQ affinity with list of CPU cores and Receive Side Scaling mask
# Core list should be the first two cores (and hyperthreads) on each
socket

# Mask set to everything currently
# https://github.com/torvalds/linux/blob/v4.11/Documentation/networking/
scaling.txt#L80-L96
echo "@reboot sudo /opt/aws/groundstation/bin/set_irq_affinity.sh '0 1 48
49' 'ffffffff,ffffffff,ffffffff' >>/var/log/user-data.log 2>&1" >>/var/spool/cron/root

# Reserving the port range defined in the GS agent ingress address in
the Dataflow Endpoint Group so the kernel doesn't steal any of them from the GS agent.
These ports are the ports that the GS agent will ingress data
# across, so if the kernel steals one it could cause problems ingressing
data onto the instance.
echo net.ipv4.ip_local_reserved_ports="42000-50000" >> /etc/sysctl.conf

```

```
# </Tuning Section End>
```

```
# We have to reboot for linux kernel settings to apply
shutdown -r now
```

```
- DataflowEndpointGroupId: !Ref DataflowEndpointGroup
  EIP: !Ref ReceiverInstanceElasticIp
```

```
# The AWS Ground Station Dataflow Endpoint Group that defines the endpoints that AWS
Ground
```

```
# Station will use to send/receive data to/from your satellite.
```

```
DataflowEndpointGroup:
```

```
  Type: AWS::GroundStation::DataflowEndpointGroup
```

```
  Properties:
```

```
    ContactPostPassDurationSeconds: 180
```

```
    ContactPrePassDurationSeconds: 120
```

```
    EndpointDetails:
```

```
      - AwsGroundStationAgentEndpoint:
```

```
        Name: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ] # needs to
match DataflowEndpointConfig name
```

```
        EgressAddress:
```

```
          SocketAddress:
```

```
            Name: 127.0.0.1
```

```
            Port: 55000
```

```
        IngressAddress:
```

```
          SocketAddress:
```

```
            Name: !Ref ReceiverInstanceElasticIp
```

```
            PortRange:
```

```
              Minimum: 42000
```

```
              Maximum: 55000
```

또한가 계정에서 탄력적 네트워크 인터페이스(ENI)를 AWS Ground Station 생성할 수 있도록 적절한 정책, 역할 및 프로필이 필요합니다.

```
# The security group for your EC2 instance.
```

```
InstanceSecurityGroup:
```

```
  Type: AWS::EC2::SecurityGroup
```

```
  Properties:
```

```
    GroupDescription: AWS Ground Station receiver instance security group.
```

```
    VpcId: !Ref ReceiverVPC
```

```

SecurityGroupEgress:
  - CidrIp: 0.0.0.0/0
    Description: Allow all outbound traffic by default
    IpProtocol: "-1"
SecurityGroupIngress:
  # To allow SSH access to the instance, add another rule allowing tcp port 22
  from your CidrIp
  - IpProtocol: udp
    Description: Allow AWS Ground Station Incoming Dataflows
    ToPort: 50000
    FromPort: 42000
    SourcePrefixListId:
      Fn::FindInMap:
        - PrefixListId
        - Ref: AWS::Region
        - groundstation

# The EC2 instance assumes this role.
InstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Principal:
            Service:
              - "ec2.amazonaws.com"
          Action:
            - "sts:AssumeRole"
    Path: "/"
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
      - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
      - arn:aws:iam::aws:policy/service-role/AmazonEC2RoleforSSM
      - arn:aws:iam::aws:policy/AWSGroundStationAgentInstancePolicy
    Policies:
      - PolicyDocument:
          Statement:
            - Action:
                - sts:AssumeRole
              Effect: Allow
              Resource: !GetAtt GroundStationKmsKeyRole.Arn

```

```
Version: "2012-10-17"
PolicyName: InstanceGroundStationApiAccessPolicy

# The instance profile for your EC2 instance.
GeneralInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref InstanceRole

# The IAM role that AWS Ground Station will assume to access and use the KMS Key for
data delivery
GroundStationKmsKeyRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Action: sts:AssumeRole
          Effect: Allow
          Principal:
            Service:
              - groundstation.amazonaws.com
          Condition:
            StringEquals:
              "aws:SourceAccount": !Ref AWS::AccountId
            ArnLike:
              "aws:SourceArn": !Sub "arn:${AWS::Partition}:groundstation:
${AWS::Region}:${AWS::AccountId}:mission-profile/*"
        - Action: sts:AssumeRole
          Effect: Allow
          Principal:
            AWS: !Sub "arn:${AWS::Partition}:iam:${AWS::AccountId}:root"

GroundStationKmsKeyAccessPolicy:
  Type: AWS::IAM::Policy
  Properties:
    PolicyDocument:
      Statement:
        - Action:
            - kms:Decrypt
          Effect: Allow
          Resource: !GetAtt GroundStationDataDeliveryKmsKey.Arn
    PolicyName: GroundStationKmsKeyAccessPolicy
    Roles:
```

- Ref: GroundStationKmsKeyRole

GroundStationDataDeliveryKmsKey:

Type: AWS::KMS::Key

Properties:

KeyPolicy:

Statement:

- Action:

- kms:CreateAlias
- kms:Describe*
- kms:Enable*
- kms:List*
- kms:Put*
- kms:Update*
- kms:Revoke*
- kms:Disable*
- kms:Get*
- kms>Delete*
- kms:ScheduleKeyDeletion
- kms:CancelKeyDeletion
- kms:GenerateDataKey
- kms:TagResource
- kms:UntagResource

Effect: Allow

Principal:

AWS: !Sub "arn:\${AWS::Partition}:iam:\${AWS::AccountId}:root"

Resource: "*"

- Action:

- kms:Decrypt
- kms:GenerateDataKeyWithoutPlaintext

Effect: Allow

Principal:

AWS: !GetAtt GroundStationKmsKeyRole.Arn

Resource: "*"

Condition:

StringEquals:

"kms:EncryptionContext:sourceAccount": !Ref AWS::AccountId

ArnLike:

"kms:EncryptionContext:sourceArn": !Sub "arn:

\${AWS::Partition}:groundstation:\${AWS::Region}:\${AWS::AccountId}:mission-profile/*"

- Action:

- kms>CreateGrant

Effect: Allow

Principal:

```

    AWS: !Sub "arn:${AWS::Partition}:iam:${AWS::AccountId}:root"
Resource: "*"
Condition:
  ForAllValues:StringEquals:
    "kms:GrantOperations":
      - Decrypt
      - GenerateDataKeyWithoutPlaintext
    "kms:EncryptionContextKeys":
      - sourceArn
      - sourceAccount
  ArnLike:
    "kms:EncryptionContext:sourceArn": !Sub "arn:
${AWS::Partition}:groundstation:${AWS::Region}:${AWS::AccountId}:mission-profile/*"
  StringEquals:
    "kms:EncryptionContext:sourceAccount": !Ref AWS::AccountId
  Version: "2012-10-17"
  EnableKeyRotation: true

```

AWS Ground Station 구성

이 섹션은 시작하기 [구성 생성](#)를 나타냅니다.

자동 추적을 사용하여 기본 설정을 지정하려면 추적 구성이 필요합니다. PREFERRED를 자동 추적으로 선택하면 신호 품질이 향상될 수 있지만 충분한 JPSS-1 에페메리스 품질로 인해 신호 품질을 충족할 필요는 없습니다.

```

TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"

```

통신 경로에 따라 위성 부분을 나타내는 안테나 다운링크 구성과 엔드포인트 세부 정보를 정의하는 데이터 흐름 엔드포인트 그룹을 참조하는 데이터 흐름 엔드포인트 구성을 정의해야 합니다.

```
# The AWS Ground Station Antenna Downlink Config that defines the frequency spectrum
used to
# downlink data from your satellite.
SnppJpssDownlinkDigIfAntennaConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "SNPP JPSS Downlink WBDigIF Antenna Config"
    ConfigData:
      AntennaDownlinkConfig:
        SpectrumConfig:
          Bandwidth:
            Units: "MHz"
            Value: 100
          CenterFrequency:
            Units: "MHz"
            Value: 7812
          Polarization: "RIGHT_HAND"

# The AWS Ground Station Dataflow Endpoint Config that defines the endpoint used to
downlink data
# from your satellite.
DownlinkDigIfEndpointConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "Aqua SNPP JPSS Terra Downlink DigIF Endpoint Config"
    ConfigData:
      DataflowEndpointConfig:
        DataflowEndpointName: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ]
        DataflowEndpointRegion: !Ref AWS::Region
```

AWS Ground Station 미션 프로파일

이 섹션은 시작하기 [미션 프로파일 생성](#)를 나타냅니다.

이제 연결된 구성이 있으므로 이를 사용하여 데이터 흐름을 구성할 수 있습니다. 나머지 파라미터에 기본값을 사용합니다.

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnppJpssMissionProfile:
```

```
Type: AWS::GroundStation::MissionProfile
Properties:
  Name: !Sub 'JPSS WBDigIF gs-agent EC2 Delivery'
  ContactPrePassDurationSeconds: 120
  ContactPostPassDurationSeconds: 120
  MinimumViableContactDurationSeconds: 180
  TrackingConfigArn: !Ref TrackingConfig
  DataflowEdges:
    - Source: !Ref SnpJPSSDownlinkDigIfAntennaConfig
      Destination: !Ref DownlinkDigIfEndpointConfig
  StreamsKmsKey:
    KmsKeyArn: !GetAtt GroundStationDataDeliveryKmsKey.Arn
  StreamsKmsRole: !GetAtt GroundStationKmsKeyRole.Arn
```

함께 넣기

위의 리소스를 사용하면 온보딩된 모든에서 동기식 데이터 전송을 위해 JPSS-1 연락처를 예약할 수 있습니다 AWS Ground Station [AWS Ground Station 위치](#).

다음은이 섹션에 설명된 모든 리소스를 직접 사용할 수 있는 단일 CloudFormation 템플릿으로 결합한 전체 템플릿입니다 CloudFormation.

라는 CloudFormation 템플릿DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml은 Aqua, SNPP, JPSS-1/NOAA-20 및 Terra 위성에 대한 디지털화된 중간 주파수(DigIF) 데이터 수신을 빠르게 시작할 수 있도록 설계되었습니다. 여기에는 Amazon EC2 인스턴스와 AWS Ground Station Agent를 사용하여 원시 DigIF 다이렉트 브로드캐스트 데이터를 수신하는 데 필요한 CloudFormation 리소스가 포함되어 있습니다.

Aqua, SNPP, JPSS-1/NOAA-20 및 Terra가 계정에 온보딩되지 않은 경우 섹션을 참조하세요 [위성 온보딩](#).

Note

유효한 자격 AWS 증명을 사용하여 Amazon S3 버킷을 온보딩하는 고객에 액세스하여 템플릿에 액세스할 수 있습니다. 아래 링크는 리전별 Amazon S3 버킷을 사용합니다. us-west-2 스택을 생성할 CloudFormation 해당 리전을 나타내도록 리전 코드를 변경합니다.

또한 다음 지침은 YAML을 사용합니다. 그러나 템플릿은 YAML 형식과 JSON 형식으로 모두 제공됩니다. JSON을 사용하려면 템플릿을 다운로드할 .json 때 .yaml 파일 확장명을 로 바꿉니다.

를 사용하여 템플릿을 다운로드하려면 다음 명령을 AWS CLI사용합니다.

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/agent/ec2_delivery/
DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml .
```

브라우저에서 다음 URL로 이동하면 콘솔에서 템플릿을 보고 다운로드할 수 있습니다.

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-
west-2/agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

다음 링크를 CloudFormation 사용하여에서 직접 템플릿을 지정할 수 있습니다.

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/
agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

템플릿이 정의하는 추가 리소스는 무엇입니까?

DirectBroadcastSatelliteWbDigIfEc2DataDelivery 템플릿에는 다음과 같은 추가 리소스가 포함되어 있습니다.

- 수신기 인스턴스 탄력적 네트워크 인터페이스 - (조건부) 탄력적 네트워크 인터페이스는 제공된 경우 PublicSubnetId에서 지정한 서브넷에 생성됩니다. 이는 수신기 인스턴스가 프라이빗 서브넷에 있는 경우에 필요합니다. 탄력적 네트워크 인터페이스는 EIP와 연결되고 수신기 인스턴스에 연결됩니다.
- 수신기 인스턴스 탄력적 IP - 연결할 탄력적 IP AWS Ground Station 입니다. 이는 수신기 인스턴스 또는 탄력적 네트워크 인터페이스에 연결됩니다.
- 다음 탄력적 IP 연결 중 하나:
 - Receiver Instance to Elastic IP Association - PublicSubnetId가 지정되지 않은 경우 Elastic IP를 수신기 인스턴스에 연결하는 것입니다. 이를 위해서는 SubnetId가 퍼블릭 서브넷을 참조해야 합니다.
 - Receiver Instance Elastic Network Interface to Elastic IP Association - PublicSubnetId가 지정된 경우 수신기 인스턴스 탄력적 네트워크 인터페이스에 대한 탄력적 IP의 연결입니다.
- (선택 사항) CloudWatch 이벤트 트리거 - AWS Lambda 고객 응대 AWS Ground Station 전후에서 전송한 CloudWatch 이벤트를 사용하여 트리거되는 함수입니다. AWS Lambda 함수는 수신기 인스턴스를 시작하고 선택적으로 중지합니다.

- (선택 사항) 연락처에 대한 Amazon EC2 확인 - Lambda를 사용하여 SNS 알림이 있는 연락처에 대한 Amazon EC2 인스턴스(들)의 확인 시스템을 설정하는 옵션입니다. 이때 현재 사용량에 따라 요금이 부과될 수 있다는 점에 유의하세요.
- 추가 미션 프로필 - 추가 퍼블릭 브로드캐스트 위성(Aqua, SNPP 및 Terra)을 위한 미션 프로필입니다.
- 추가 안테나 다운링크 구성 - 추가 퍼블릭 브로드캐스트 위성(Aqua, SNPP 및 Terra)에 대한 안테나 다운링크 구성입니다.

이 템플릿의 위성에 대한 값과 매개 변수가 이미 채워져 있습니다. 이러한 파라미터를 사용하면 이러한 위성에서 AWS Ground Station 즉시 쉽게 사용할 수 있습니다. 이 템플릿을 사용할 AWS Ground Station 때를 사용하기 위해 자체 값을 구성할 필요가 없습니다. 그러나 사용 사례에 맞게 템플릿을 작동하도록 값을 사용자 지정할 수 있습니다.

내 데이터는 어디에서 수신합니까?

데이터 흐름 엔드포인트 그룹은 템플릿의 일부로 생성되는 수신기 인스턴스 네트워크 인터페이스를 사용하도록 설정됩니다. 수신기 인스턴스는 AWS Ground Station 에이전트를 사용하여 데이터 흐름 엔드포인트에 의해 정의된 포트 AWS Ground Station 에서의 데이터 스트림을 수신합니다. 데이터 흐름 엔드포인트 그룹 설정에 대한 자세한 내용은 [AWS::GroundStation::DataflowEndpointGroup](#)을 참조하세요. 에이전트에 AWS Ground Station 대한 자세한 내용은 [AWS Ground Station 에이전트란 무엇입니까?](#)를 참조하십시오.

문제 해결

다음 설명서는 사용 중에 발생할 수 있는 문제를 해결하는 데 도움이 될 수 있습니다 AWS Ground Station.

주제

- [Amazon EC2로 데이터를 전송하는 고객 응대 문제 해결](#)
- [실패 고객 응대 문제 해결](#)
- [실패한 고객 응대 업데이트 문제 해결](#)
- [FAILED_TO_SCHEDULE 연락처 문제 해결](#)
- [정상 상태가 아닌 DataflowEndpointGroups 문제 해결](#)
- [잘못된 에페메리스 문제 해결](#)
- [데이터를 수신하지 못한 고객 응대 문제 해결](#)
- [원격 측정 문제 해결](#)

Amazon EC2로 데이터를 전송하는 고객 응대 문제 해결

AWS Ground Station연락을 성공적으로 완료할 수 없는 경우 Amazon EC2 인스턴스가 실행 중인지 확인하고, 데이터 흐름 엔드포인트 애플리케이션이 실행 중인지 확인하고, 데이터 흐름 엔드포인트 애플리케이션의 스트림이 올바르게 구성되었는지 확인해야 합니다.

Note

DataDefender(DDX)는 현재에서 지원되는 데이터 흐름 엔드포인트 애플리케이션의 예입니다. AWS Ground Station

사전 조건

다음 절차에서는 Amazon EC2 인스턴스가 이미 설정되어 있다고 가정합니다. 에서 Amazon EC2 인스턴스를 설정하려면 시작하기를AWS Ground Station참조하세요. <https://docs.aws.amazon.com/ground-station/latest/ug/getting-started.html>

1단계: EC2 인스턴스가 실행 중인지 확인

다음 절차에서는 콘솔에서 Amazon EC2 인스턴스를 찾고 실행 중이 아닌 경우 시작하는 방법을 보여줍니다.

1. 문제 해결 중인 접촉에 사용된 Amazon EC2 인스턴스를 찾습니다. 다음 단계를 사용합니다.
 - a. CloudFormation 대시보드에서 Amazon EC2 인스턴스가 포함된 스택을 선택합니다.
 - b. 리소스 탭을 선택하고 논리 ID 열에서 Amazon EC2 인스턴스를 찾습니다. 인스턴스가 상태 열에 생성되었는지 확인합니다.
 - c. 물리적 ID 열에서 Amazon EC2 인스턴스에 대한 링크를 선택합니다. 그러면 Amazon EC2 관리 콘솔로 이동합니다.
2. Amazon EC2 관리 콘솔에서 Amazon EC2 인스턴스 상태가 실행 중인지 확인합니다.
3. 인스턴스가 실행 중이면 다음 단계로 계속합니다. 인스턴스가 실행 중이 아니면 다음 단계를 사용하여 인스턴스를 시작합니다.
 - Amazon EC2 인스턴스를 선택한 상태에서 작업 > 인스턴스 상태 > 시작을 선택합니다.

2단계: 사용되는 데이터 흐름 애플리케이션 유형 결정

데이터 전송에 AWS Ground Station 에이전트를 사용하는 경우 [AWS Ground Station 에이전트 문제 해결](#) 섹션으로 리디렉션하세요. 그렇지 않으면 DataDefender(DDX) 애플리케이션을 사용하는 경우로 계속 진행합니다. [the section called “3단계: 데이터 흐름 애플리케이션이 실행 중인지 확인”](#).

3단계: 데이터 흐름 애플리케이션이 실행 중인지 확인

DataDefender의 상태를 확인하려면 Amazon EC2의 인스턴스에 연결해야 합니다. 인스턴스 연결에 대한 자세한 내용은 [Linux 인스턴스에 연결을 참조하세요](#).

다음 절차에서는 SSH 클라이언트에서 명령을 사용하여 문제를 해결하는 단계를 제공합니다.

1. 터미널 또는 명령 프롬프트를 열고 SSH를 사용하여 Amazon EC2 인스턴스에 연결합니다. DataDefender 웹 UI를 보려면 원격 호스트의 포트 80을 전달합니다. 다음 명령은 SSH를 사용하여 포트 전달이 활성화된 접속을 통해 Amazon EC2 인스턴스에 연결하는 방법을 보여줍니다.

Note

<SSH KEY>, <BASTION HOST> 및 <HOST>를 특정 ssh 키, 접속 호스트 이름 및 Amazon EC2 인스턴스 호스트 이름으로 바꿔야 합니다.

Windows의 경우

```
ssh -L 8080:localhost:80 -o ProxyCommand="C:\Windows\System32\OpenSSH\ssh.exe -o
\"ForwardAgent yes\" -W %h:%p -i \"<SSH KEY>\" ec2-user@<BASTION HOST>" -i "<SSH
KEY>" ec2-user@<HOST>
```

Mac의 경우

```
ssh -L 8080:localhost:80 -o ProxyCommand="ssh -A -o 'ForwardAgent yes' -W %h:%p -i
<SSH KEY> ec2-user@<BASTION HOST>" -i <SSH KEY> ec2-user@<HOST>
```

- 출력에서 ddx라는 실행 중인 프로세스를 grepping(확인)하여 DataDefender(DDX라고도 함)가 실행 중인지 확인합니다. 실행 중인 프로세스를 검사하는 명령과 성공적인 예제 출력은 다음과 같습니다.

```
[ec2-user@Receiver-Instance ~]$ ps -ef | grep ddx
      Rtlogic   4977      1 10 Oct16 ?          2-00:22:14 /opt/rtlogic/ddx/
bin/ddx -m/opt/rtlogic/ddx/modules -p/opt/rtlogic/ddx/plugins -c/opt/rtlogic/
ddx/bin/ddx.xml -umask=077 -daemon -f installed=true -f security=true -f enable
HttpsForwarding=true
      Ec2-user 18787 18657  0 16:51 pts/0      00:00:00 grep -color=auto ddx
```

DataDefender가 실행 중인 경우 [the section called “4단계: 데이터 흐름 애플리케이션 스트림이 구성되어 있는지 확인”](#) 그렇지 않은 경우 다음 단계로 계속 진행합니다.

- 아래 명령을 사용하여 DataDefender를 시작합니다.

```
sudo systemctl start rtlogic-ddx
```

명령을 사용한 후 DataDefender가 실행 중인 경우로 건너뛵니다. [the section called “4단계: 데이터 흐름 애플리케이션 스트림이 구성되어 있는지 확인”](#). 그렇지 않다면 계속해서 다음 단계로 이동하십시오.

4. 아래 명령을 사용하여 다음 파일을 검사하여 DataDefender를 설치하고 구성하는 동안 오류가 있는지 확인합니다.

```
cat /var/log/user-data.log
cat /opt/aws/groundstation/.startup.out
```

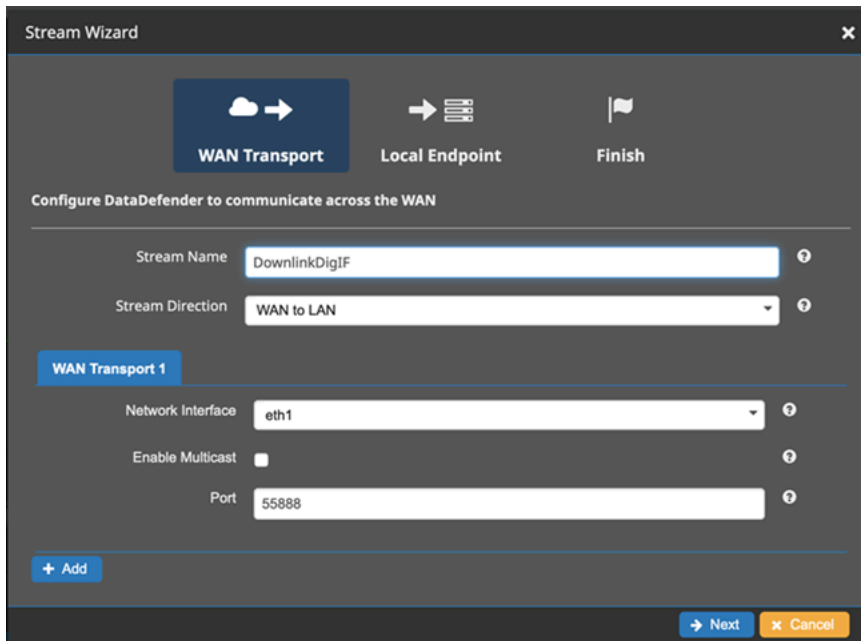
Note

이러한 파일을 검사할 때 발견되는 일반적인 문제는 Amazon EC2 인스턴스가 실행 중인 Amazon VPC에 설치 파일을 다운로드하기 위해 Amazon S3에 액세스할 수 있는 권한이 없다는 것입니다. 로그에서 이 문제가 발견되면 EC2 인스턴스의 Amazon VPC 및 보안 그룹 설정을 확인하여 Amazon S3에 대한 액세스를 차단하지 않는지 확인합니다.

Amazon VPC 설정을 확인한 후 DataDefender가 실행 중인 경우로 계속 진행합니다. [the section called “4단계: 데이터 흐름 애플리케이션 스트림이 구성되어 있는지 확인”](#). 문제가 지속되면 [AWS Support에 문의](#)하여 문제에 대한 설명과 함께 로그 파일을 전송합니다.

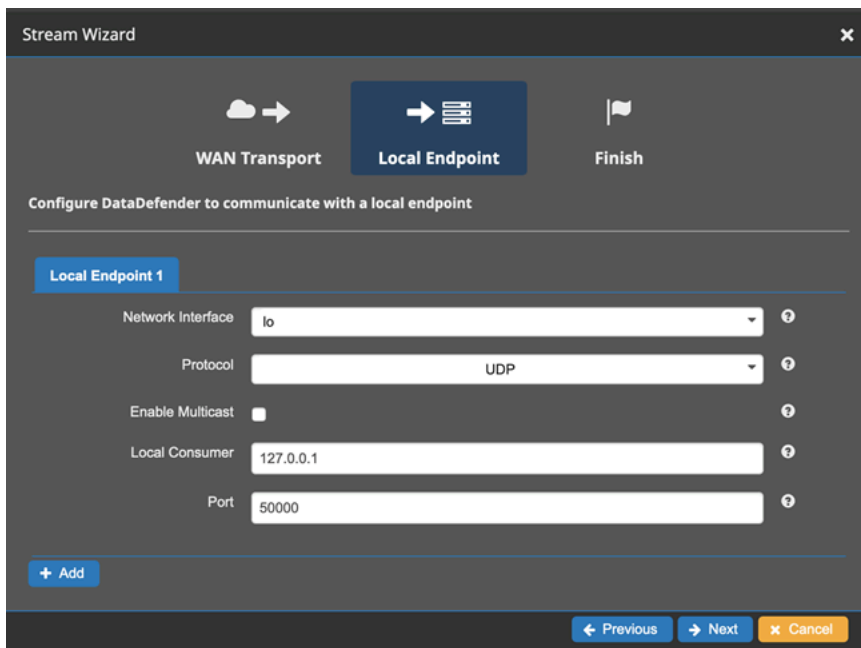
4단계: 데이터 흐름 애플리케이션 스트림이 구성되어 있는지 확인

1. 웹 브라우저에서 주소 표시줄에 localhost:8080 주소를 입력하여 DataDefender 웹 사용자 인터페이스에 액세스합니다. 그런 다음 Enter 키를 누릅니다.
2. DataDefender 대시보드에서 세부 정보로 이동을 선택합니다.
3. 스트림 목록에서 스트림을 선택하고 스트림 편집을 선택합니다.
4. 스트림 마법사 대화 상자에서 다음을 수행합니다.
 - a. WAN 전송 창에서 스트림 방향으로 WAN에서 LAN으로가 선택되어 있는지 확인합니다.
 - b. 포트 상자에 데이터 흐름 엔드포인트 그룹에 대해 선택한 WAN 포트가 존재하는지 확인합니다. 기본적으로 이 포트는 55888입니다. 그리고 다음을 선택합니다.



The screenshot shows the 'Stream Wizard' window with the 'WAN Transport' step selected. The title bar says 'Stream Wizard'. At the top, there are three tabs: 'WAN Transport' (active), 'Local Endpoint', and 'Finish'. Below the tabs, the text reads 'Configure DataDefender to communicate across the WAN'. The 'Stream Name' field is 'DownlinkDigIF'. The 'Stream Direction' dropdown is 'WAN to LAN'. Under 'WAN Transport 1', the 'Network Interface' is 'eth1', 'Enable Multicast' is unchecked, and the 'Port' is '55888'. At the bottom, there is a '+ Add' button, a 'Next' button, and a 'Cancel' button.

- c. 로컬 엔드포인트 창에서 포트 상자에 올바른 포트가 표시되는지 확인합니다. 기본적으로 이 포트는 50000입니다. DataDefender가AWS Ground Station서비스에서 데이터를 수신한 후 데이터를 수신할 포트입니다. 그리고 다음을 선택합니다.



The screenshot shows the 'Stream Wizard' window with the 'Local Endpoint' step selected. The title bar says 'Stream Wizard'. At the top, there are three tabs: 'WAN Transport', 'Local Endpoint' (active), and 'Finish'. Below the tabs, the text reads 'Configure DataDefender to communicate with a local endpoint'. Under 'Local Endpoint 1', the 'Network Interface' is 'lo', 'Protocol' is 'UDP', 'Enable Multicast' is unchecked, 'Local Consumer' is '127.0.0.1', and the 'Port' is '50000'. At the bottom, there is a '+ Add' button, a 'Previous' button, a 'Next' button, and a 'Cancel' button.

- d. 값을 변경한 경우 나머지 메뉴에서 완료를 선택합니다. 그렇지 않으면 스트림 마법사 메뉴를 취소할 수 있습니다.

이제 Amazon EC2 인스턴스와 DataDefender가 모두 실행 중이고 데이터를 수신하도록 올바르게 구성되었는지 확인했습니다 AWS Ground Station. 계속해서 [the section called “5단계: 수신기 인스턴스\(들\) 서브넷에 사용 가능한 IP 주소가 충분한지 확인”](#)로 이동하세요.

5단계: 수신기 인스턴스(들) 서브넷에 사용 가능한 IP 주소가 충분한지 확인

다음 절차에서는 콘솔의 Amazon EC2 수신기 인스턴스에서 사용 가능한 IP 주소 수를 찾는 방법을 보여줍니다.

1. 고객 응대에 사용된 각 Amazon EC2 수신기 인스턴스에 대해 문제 해결 중입니다. 다음 단계를 사용합니다.
 - a. CloudFormation 대시보드에서 Amazon EC2 인스턴스가 포함된 스택을 선택합니다.
 - b. 리소스 탭을 선택하고 논리 ID 열에서 Amazon EC2 인스턴스를 찾습니다. 인스턴스가 상태 열에 생성되었는지 확인합니다.
 - c. 물리적 ID 열에서 Amazon EC2 인스턴스에 대한 링크를 선택합니다. 그러면 Amazon EC2 관리 콘솔로 이동합니다.
2. Amazon EC2 관리 콘솔에서 Amazon EC2 수신기 인스턴스의 인스턴스 요약에서 서브넷 ID 링크를 찾아 클릭합니다. 그러면 해당 Amazon VPC 관리 콘솔로 이동합니다.
3. Amazon VPC 관리 콘솔에서 일치하는 서브넷을 선택하고 서브넷의 세부 정보에서 사용 가능한 IPv4 주소를 확인합니다. 이 수가 이 Amazon EC2 수신기 인스턴스를 사용하는 데이터 흐름 엔드 포인트보다 많지 않은 경우 다음을 수행합니다.
 - a. CloudFormation 템플릿의 해당 서브넷 CidrBlock을 올바른 크기로 업데이트합니다. 서브넷 크기 조정에 대한 자세한 내용은 [서브넷 CIDR 블록](#)을 참조하세요.
 - b. 업데이트된 CloudFormation 템플릿으로 스택을 재배포합니다.

문제가 계속 발생하면 [AWS Support에 문의하십시오](#).

실패 고객 응대 문제 해결

가 리소스 구성 문제를 AWS Ground Station 감지하면 고객 응대의 터미널 고객 응대 상태는 실패입니다. FAILED 연락의 원인이 될 수 있는 일반적인 사용 사례가 문제 해결에 도움이 되는 단계와 함께 아래에 나와 있습니다.

Note

이 가이드는 FAILED 연락 상태를 위한 것으로, AWS_FAILED, AWS_CANCELLED 또는 FAILED_TO_SCHEDULE 등과 같은 다른 장애 상태를 대상으로 하지 않습니다. 연락 상태에 대한 자세한 설명은 [the section called “AWS Ground Station 고객 응대 상태”](#) 섹션을 참조하십시오.

데이터 흐름 엔드포인트 FAILED 사용 사례

다음은 데이터 흐름 엔드포인트 기반 데이터 흐름의 연락 실패 상태가 될 수 있는 일반적인 사용 사례 목록입니다.

- 데이터 흐름 엔드포인트가 연결되지 않음 - 하나 이상의 데이터 흐름에 대한 AWS Ground Station 안테나와 데이터 흐름 엔드포인트 그룹 간의 연결이 설정되지 않았습니다.
- 데이터 흐름 엔드포인트 연결 지연 - 고객 응대 시작 시간 이후에 하나 이상의 데이터 흐름에 대한 AWS Ground Station 안테나와 데이터 흐름 엔드포인트 그룹 간의 연결이 설정되었습니다.
- 데이터 흐름 엔드포인트의 서브넷에 사용 가능한 IP 주소가 없습니다. AWS Ground Station의 데이터 전송 솔루션은 수신자 인스턴스의 서브넷에 사용 가능한 IP 주소가 없어 프라이빗 네트워크에서 ENI를 생성할 수 없습니다.
- 데이터 흐름 엔드포인트의 서브넷이 잘못되었습니다. 데이터 흐름 엔드포인트 그룹에 지정된 제공된 서브넷에 액세스할 수 없어 AWS Ground Station의 데이터 전송 솔루션이 프라이빗 네트워크에서 ENI를 생성할 수 없습니다.

데이터 흐름 엔드포인트 실패 사례의 경우 다음을 살펴보는 것이 좋습니다.

- 연락 시작 시간 전에 수신자 Amazon EC2 인스턴스가 성공적으로 시작되었는지 확인합니다.
- 고객 응대 중에 데이터 흐름 엔드포인트 소프트웨어가 실행 중이었는지 확인합니다.
- 수신자 인스턴스 서브넷당 데이터 흐름 엔드포인트당 사용 가능한 IP 주소가 하나 이상 있는지 확인합니다.
- 에 구성된 데이터 흐름을 통해 Dataflow 엔드포인트 그룹에 연결된 서브넷이 활성 상태로 [Amazon VPC 설정 및 구성](#) 유지되고 사용할 수 있도록 합니다 AWS Ground Station.

자세한 문제 해결 단계는 [Amazon EC2로 데이터를 전송하는 고객 응대 문제 해결](#) 관련 섹션을 참조하십시오.

AWS Ground Station 에이전트 FAILED 사용 사례

다음은 에이전트 기반 데이터 흐름의 FAILED 연락 상태가 될 수 있는 일반적인 사용 사례 목록입니다.

- AWS Ground Station 에이전트 보고되지 않음 상태 - 하나 이상의 데이터 흐름에 대해 데이터 흐름 엔드포인트 그룹에서 데이터 전송을 오케스트레이션할 책임이 있는 에이전트가 상태를 성공적으로 보고하지 않았습니다 AWS Ground Station. 이 상태 업데이트는 연락 종료 시간으로부터 몇 초 이내에 이루어져야 합니다.
- AWS Ground Station 에이전트가 늦게 시작됨 - 고객 응대 시작 시간 이후에 하나 이상의 데이터 흐름에 대해 데이터 흐름 엔드포인트 그룹에서 데이터 전송을 오케스트레이션하는 에이전트가 늦게 시작되었습니다.

AWS Ground Station 에이전트 데이터 흐름 실패 사례의 경우 다음을 살펴보는 것이 좋습니다.

- 연락 시작 시간 전에 수신자 Amazon EC2 인스턴스가 성공적으로 시작되었는지 확인합니다.
- 시작 시점과 연락 중에 에이전트 애플리케이션이 가동되어 실행 중인지 확인하십시오.
- 연락 종료 후 15초 이내에 에이전트 애플리케이션과 Amazon EC2 인스턴스가 종료되지 않았는지 확인합니다. 이렇게 하면 에이전트가 AWS Ground Station에 상태를 보고할 충분한 시간을 확보할 수 있습니다.

자세한 문제 해결 단계는 [Amazon EC2로 데이터를 전송하는 고객 응대 문제 해결](#) 관련 섹션을 참조하십시오.

실패한 고객 응대 업데이트 문제 해결

[UpdateContact](#) API를 호출하면 요청에 대해 동기 검증을 AWS Ground Station 수행합니다. 검증이 통과하면 업데이트가 비동기적으로 처리되어 변경 사항이 안테나 리전에 전파됩니다. 동기 검증 오류는 HTTP 응답에서 직접 반환됩니다. 비동기 실패는 연락처 버전의 `failureCodes` 및 `failureMessage` 필드를 통해 보고되며, 업데이트에 실패한 버전에 대해 [DescribeContactVersion](#)을 호출하여 볼 수 있습니다.

고객 응대 버전 관리에 대한 자세한 내용은 섹션을 참조하세요 [연락처 및 연락처 버전 관리 업데이트](#).

동기 검증 오류

[UpdateContact](#) 요청이 검증에 실패하면 HTTP 응답에 다음 오류가 직접 반환됩니다.

ResourceNotFoundException: 연락처를 찾을 수 없음

일반적인 원인

지정된 `contactId` 존재하지 않거나 다른 AWS 계정에 속합니다.

해결 방법

1. `contactId`가 올바른지 확인합니다.
2. 연락처를 소유한 AWS 계정의 자격 증명을 사용하고 있는지 확인합니다.
3. [ListContacts](#)를 사용하여 올바른 `contactId`를 찾습니다.

ConflictException: 연락처를 업데이트할 수 없음

일반적인 원인

연락처가 업데이트를 허용하지 않는 상태입니다. [UpdateContact](#) API는 연락처가 SCHEDULED, PREPASS 또는 PASS 상태일 때만 호출할 수 있습니다. 이 오류는 다른 업데이트가 이미 진행 중인 경우에도 발생합니다(최신 연락처 버전이 UPDATING 상태임).

해결 방법

1. [DescribeContact](#)를 호출하여 현재 연락처 상태를 확인합니다.
2. 연락처가 터미널 상태(예: COMPLETED FAILED 또는 CANCELLED)인 경우 업데이트할 수 없습니다. 연락처는 SCHEDULED, PREPASS 또는 PASS 상태일 때만 업데이트할 수 있습니다. 터미널 상태의 전체 목록은 섹션을 참조하세요 [AWS Ground Station 고객 응대 상태](#).
3. 다른 업데이트가 진행 중인 경우 다른 업데이트를 제출하기 전에 현재 업데이트가 ACTIVE 또는 FAILED_TO_UPDATE 상태에 도달할 때까지 기다립니다. [DescribeContactVersion](#) API를 폴링하거나 someSDK 및에서 제공하는 ContactUpdated웨이터 편의 유틸리티를 사용할 수 있습니다 AWS Command Line Interface. AWS SDKs

InvalidParameterException: 잘못된 요청 파라미터

일반적인 원인

요청에 잘못된 파라미터가 포함되어 있습니다. 일반적인 사용 사례는 다음과 같습니다.

- 가 누락되었거나 비어 있습니다 `clientToken`.

- 단일 요청에 포함된 여러 유형의 ProgramTrackSettings (azimuth/elevation, OEM 및 TLE) 요청당 하나의 유형만 허용됩니다.
- 고객 응대의 지상국에서 방위 고도 에페메리스에 대한 승인을 받지 않고 satelliteArn null로 설정합니다.
- satelliteArn가 null인 AzElProgramTrackSettings 경우 누락됩니다.
- 지정된와 연결되지 ephemerisId 값을 제공합니다satelliteArn.
- 위성에는 접촉 시간 범위에 대한 지상국의 유효한 가시성 창이 없습니다.
- 위성이 지상국에 온보딩되지 않았거나 미션 프로파일에 필요한 라이선스가 없습니다.
- 미션 프로파일에는 연락처 업데이트에 지원되지 않는 [안테나 다운로드 복조 디코드 구성](#) 구성이 포함되어 있습니다.

해결 방법

1. 응답의 오류 메시지를 검토하여 유효하지 않은 파라미터에 대한 세부 정보를 확인합니다.
2. 요청ProgramTrackSettings당 정확히 한 가지 유형의를 제공해야 합니다.
3. 없이 방위각/상승 각도를 사용하는 경우 해당 계정이 지상국에서이 기능에 대해 승인되었는지 satelliteArn확인합니다. 자세한 내용은 [방위각 고도 에페메리스 데이터 제공](#) 단원을 참조하십시오.
4. 참조하는 에페메리스가 올바른 위성과 연결되어 있고 연락 시간 범위를 포함하는지 확인합니다.

ResourceLimitExceededException: 최대 버전 제한에 도달함

일반적인 원인

고객 응대가 최대 버전 수(128)에 도달했습니다. [UpdateContact](#)에 대한 각 호출은 새 버전을 생성하며 연락처는이 제한을 초과할 수 없습니다.

해결 방법

1. 이 한도는 늘릴 수 없습니다. 추가 변경이 필요한 경우 연락처를 취소하고 새 연락처를 예약합니다.

비동기 장애 코드

다음 실패 코드는 연락처 버전의 failureCodes 필드에 FAILED_TO_UPDATE 상태인 상태로 표시됩니다. [DescribeContactVersion](#)을 사용하여 이러한 세부 정보를 검색합니다. failureMessage 필드는 실패에 대한 추가 컨텍스트를 제공합니다.

실패 코드	일반적인 원인	해결 방법
INTERNAL_ERROR	업데이트를 처리하는 동안 예기치 않은 내부 오류가 발생했습니다.	업데이트를 다시 시도합니다. 문제가 지속되면 AWS Support 에 문의하세요.
INVALID_SATELLITE_ARN	업데이트 요청에 제공된 위성 ARN이 유효하지 않거나 존재하지 않습니다.	위성 ARN을 확인하고 위성이 계정에 등록되어 있는지 확인합니다.
INVALID_UPDATE_CONTACT_REQUEST	업데이트 요청에 동기 검증 중에 포착되지 않은 잘못된 파라미터가 포함되어 있습니다.	에서 failureMessage 세부 정보를 검토하고 요청 파라미터를 수정합니다.
EPHEMERIS_NOT_FOUND	추적 재정의를 참조하는 에페메리스가 존재하지 않습니다.	를 ephemerisId 확인하고 에페메리스가 삭제되지 않았는지 확인합니다.
EPHEMERIS_TIME_RANGE_INVALID	에페메리스는 고객 응대의 시간 범위를 다루지 않습니다.	전체 연락 시간 범위를 포함하는 새 에페메리스를 업로드합니다. 임시 시간 범위를 확장할 수 없는 경우 연락처를 취소하고 임시 시간 범위 동안 새 연락처를 예약합니다. 자세한 내용은 사용자 지정 에페메리스 데이터 제공 단원을 참조하십시오.
EPHEMERIS_NOT_ENABLED	참조된 에페메리스가 ENABLED 상태가 아닙니다.	업데이트를 다시 시도하기 전에 에페메리스 상태를 확인하고 활성화합니다.
SATELLITE_DOES_NOT_MATCH_EPHEMERIS	에페메리스는 업데이트 요청에 지정된 위성과 연결되지 않습니다.	가에 지정된 위성에 ephemerisId 속하는지 확인합니다. satelliteArn .
NOT_ONBOARDED_TO_AZEL_EPHEMERIS	계정이 연락처의 지상국에서 방위각 고도 에페메리스 데이터 를 사용하도록 승인되지 않았습니다.	사용 사례에 방위각 고도 에페메리스가 필요한 경우를 통해 AWS Support 티켓을 열어 액세스를

실패 코드	일반적인 원인	해결 방법
	방위각 고도 에페메리스는 제한된 수의 특수 사용 사례에 사용할 수 있는 제한된 기능입니다.	AWS Support Center Console 요청합니다. 또는 사용 사례에 맞는 경우 TLE 에페메리스 데이터 또는 OEM 에페메리스 데이터 를 사용하는 것이 좋습니다.
AZEL_EPHEMERIS_NOT_FOUND	요청에 참조된 방위각 고도 에페메리스 가 존재하지 않습니다.	를 ephemerisId 확인하고 방위각 고도 에페메리스가 삭제되지 않았는지 확인합니다.
AZEL_EPHEMERIS_WRONG_GROUND_STATION	방위각 고도 에페메리스 는 고객 응대에서 사용하는 것과 다른 지상국에 대해 생성되었습니다.	올바른 지상국에 대한 새 방위각 고도 에페메리스를 업로드하거나 연락처의 지상국과 일치하는 기존 에페메리스를 사용합니다.
AZEL_EPHEMERIS_INVALID_STATUS	방위각 고도 에페메리스 가 유효한 사용 상태가 아닙니다.	에페메리스 상태를 확인합니다. 상태여야 합니다ENABLED. 에페메리스가 검증에 실패한 경우 수정된 버전을 업로드합니다.
AZEL_EPHEMERIS_TIME_RANGE_INVALID	방위각 고도 에페메리스 는 고객 응대의 시간 범위를 다루지 않습니다.	전체 연락 시간 범위를 포함하는 새 방위각 고도 에페메리스를 업로드합니다. 임시 시간 범위를 확장할 수 없는 경우 연락처를 취소하고 임시 시간 범위 동안 새 연락처를 예약합니다.

업데이트 상태 확인

UpdateContact를 호출한 후 새 연락처 버전이 UPDATING 상태로 시작됩니다. 이 기간 동안 [DescribeContact](#)는 이전 활성 버전의 연락처를 계속 반환합니다. 새 버전은 안테나로 전파되고 ACTIVE 상태에 도달할 DescribeContact 때까지에 표시되지 않습니다. 특정 버전의 상태를 확인하려면 [DescribeContactVersion](#)을 사용합니다.

업데이트 성공 또는 실패 여부를 확인하려면:

1. UpdateContact 응답에서 versionId 반환된 contactId 및를 사용하여 [DescribeContactVersion](#)을 호출합니다.
2. version.status 필드를 확인합니다. 상태가 이면 업데이트가 성공적으로 적용되었음을 ACTIVE 의미합니다. 상태가 이면 업데이트가 실패했음을 FAILED_TO_UPDATE 의미합니다.
3. 상태가 인 경우 version.failureCodes 및 version.failureMessage 필드에서 무엇이 잘못 되었는지 자세히 FAILED_TO_UPDATE확인합니다.

Tip

AWS SDKs 및는 버전이 ACTIVE 또는 FAILED_TO_UPDATE 상태에 도달할 DescribeContactVersion 때까지 자동으로 폴링하는 ContactUpdated웨이터를 AWS Command Line Interface 지원합니다. 예를 들어는 [aws groundstation wait contact-updated](#) 명령을 AWS Command Line Interface 제공합니다. 자체 폴링 로직을 구현하는 대신 웨이터를 사용합니다.

FAILED_TO_SCHEDULE 연락처 문제 해결

가 리소스 구성 또는 내부 시스템 내에서 문제를 감지하면 AWS Ground Station 연락이 FAILED_TO_SCHEDULE 상태로 종료됩니다. FAILED_TO_SCHEDULE 상태로 끝나는 연락처는 추가 컨텍스트를 errorMessage 위해 선택적으로를 제공합니다. 연락처 설명에 대한 자세한 내용은 [DescribeContact](#) API를 참조하세요.

FAILED_TO_SCHEDULE 연락을 유발할 수 있는 일반적인 사용 사례는 문제 해결에 도움이 되는 단계와 함께 아래에 나와 있습니다.

Note

이 가이드는 FAILED_TO_SCHEDULE 고객 응대 상태 전용이며 AWS_FAILED, AWS_AWS_CANCELLED 또는 FAILED와 같은 다른 장애 상태에는 사용할 수 없습니다. 연락 상태에 대한 자세한 설명은 [the section called “AWS Ground Station 고객 응대 상태”](#) 섹션을 참조하십시오.

안테나 다운링크 데모 디코딩 구성에 지정된 설정은 지원되지 않습니다.

이 고객 응대를 예약하는 데 사용된 [미션 프로파일](#)에 [antenna-downlink-demod-decode 구성](#)이 유효하지 않았습니다.

기존 AntennaDownlinkDemodDecode 구성

- antenna-downlink-demod-decode 구성이 최근에 변경된 경우 예약을 시도하기 전에 이전 작동 버전으로 롤백합니다.
- 기존 구성 또는 더 이상 성공적으로 예약되지 않는 기존 구성에 대한 의도적인 변경인 경우 새 AntennaDownlinkDemodDecode 구성을 온보딩하는 방법에 대한 다음 단계를 따르세요.

새로 생성된 AntennaDownlinkDemodDecode 구성

에 AWS Ground Station 직접 문의하여 새 구성을 온보딩합니다. FAILED_TO_SCHEDULE 상태로 종료된 contactId를 포함하여 [AWS Support](#)에서 사례 생성

일반 문제 해결 단계

위의 문제 해결 단계에서 문제가 해결되지 않은 경우:

- 고객 응대 예약을 다시 시도하거나 동일한 미션 프로파일을 사용하여 다른 고객 응대를 예약합니다. 연락처를 예약하는 방법에 대한 자세한 내용은 [ReserveContact](#)를 참조하세요.
- 이 미션 프로파일에 대한 FAILED_TO_SCHEDULE 상태를 계속 수신하는 경우 [AWS Support에 문의](#)하세요.

정상 상태가 아닌 DataflowEndpointGroups 문제 해결

다음은 데이터 흐름 엔드포인트 그룹이 특정 HEALTHY 상태에 있지 않을 수 있는 이유와 취해야 할 적절한 수정 조치입니다.

- NO_REGISTERD_AGENT - EC2 인스턴스를 시작하면 에이전트가 등록됩니다. 단, 이 직접적 호출이 성공하려면 유효한 컨트롤러 구성 파일이 있어야 합니다. 해당 파일 구성에 [AWS Ground Station 에이전트 사용](#) 대한 자세한 내용은 를 참조하세요.
- INVALID_IP_OWNERSHIP - DeleteDataFlowEndPointGroup API를 사용하여 데이터플로우 엔드포인트 그룹을 삭제한 다음, CreateDataFlowEndPointGroup API를 사용하여 EC2 인스턴스와 연결된 IP 주소 및 포트를 사용하여 데이터플로우 엔드포인트 그룹을 다시 생성합니다.

- UNVERIFIED_IP_OWNERSHIP - IP 주소는 아직 검증되지 않았습니다. 검증은 주기적으로 이루어지므로 이 문제는 저절로 해결될 것입니다.
- NOT_AUTHORIZED_TO_CREATE_SLR - 계정에 필요한 서비스 연결 역할을 생성할 권한이 없습니다. [Ground Station에 대한 서비스 연결 역할 사용](#)에서 문제 해결 단계를 확인하세요

잘못된 에페메리스 문제 해결

에페메리스 데이터를 업로드하면 비동기 검증 워크플로를 AWS Ground Station 거칩니다. 검증에 실패하면 에페메리스 상태가 로 변경됩니다 INVALID. [DescribeEphemeris](#) 응답의 오류 메시지는 문제를 식별하고 해결하는 데 도움이 되는 자세한 정보를 제공합니다.

에페메리스 검증 오류 이해

에페메리스가 검증에 실패하면 [DescribeEphemeris](#) API 응답에는 문제를 진단하는 데 도움이 되는 두 개의 필드가 포함됩니다.

errorCode

특정 검증 오류를 식별하는 기계 판독 가능 코드입니다. 이는 프로그래밍 방식의 오류 처리에 사용할 수 있습니다.

errorMessage

무엇이 잘못되었는지에 대한 구체적인 세부 정보와 수정 방법에 대한 지침이 포함된 사람이 읽을 수 있는 검증 오류 설명입니다.

잘못된 에페메리스에 대한 [DescribeEphemeris](#) 응답의 예:

```
{
  "ephemerisId": "abc12345-6789-def0-1234-567890abcdef",
  "name": "My Invalid Ephemeris",
  "status": "INVALID",
  "creationTime": 1620254718.765,
  "invalidReason": "METADATA_INVALID",
  "errorCode": "OBJECT_NAME_MISSING",
  "errorMessage": "Metadata field missing: OBJECT_NAME",
  "suppliedData": {
    "tle": {
      "ephemerisData": "[...]"
    }
  }
}
```

```
}
}
```

TLE 에페메리스에 대한 일반적인 검증 오류

다음은 TLE 에페메리스를 업로드할 때 발생하는 일반적인 검증 오류입니다.

일치하지 않는 위성 카탈로그 번호

오류: “에페메리스에 있는 위성 카탈로그 번호가 연결된 위성의 위성 카탈로그 번호와 일치하지 않습니다.”

해결 방법: TLE 라인의 NORAD ID/위성 카탈로그 번호가 위성의 위성 카탈로그 번호와 일치하는지 확인합니다. 카탈로그 번호가 할당되지 않은 위성000000을 사용합니다.

잘못된 평균 모션

오류: "제공된 에페메리스의 평균 모션이 가장 최근 참조 에페메리스와 너무 크게 다름"

해결 방법: TLE 데이터가 올바르고 유효한 궤도를 나타내는지 확인합니다. Ground Station은 검증 중에 스페이스 트랙 에페메리스를 참조로 사용합니다.

OEM 에페메리스의 일반적인 검증 오류

다음은 OEM 에페메리스를 업로드할 때 발생하는 일반적인 검증 오류입니다.

잘못된 참조 프레임

오류: "REF_FRAME은 지원되지 않습니다"

해결 방법: 지원되는 참조 프레임 중 하나인 EME2000 또는 ITRF2000을 사용하도록 OEM 파일을 업데이트합니다.

필수 필드 누락

오류: "메타데이터 필드 누락: 보간"

해결 방법: OEM 메타데이터 섹션에 INTERPOLATION 및 INTERPOLATION_DEGREE 필드를 추가합니다. 이스가 정확한 안테나 조준 각도를 생성하는 AWS Ground Station 데 필요합니다.

지원되지 않는 시간 시스템

오류: "TIME_SYSTEM은 지원되지 않습니다"

해결 방법: OEM 파일이 UTC를 시간 시스템으로 사용하는지 확인합니다.

지원되지 않는 OEM 버전

오류: "CCSDS_OEM_VERS는 지원되지 않습니다."

해결 방법: OEM 파일이 CCSDS OEM 버전 2.0을 사용하는지 확인합니다.

방위각 고도 에페메리스에 대한 일반적인 검증 오류

다음은 방위각 고도 에페메리스를 업로드할 때 발생하는 일반적인 검증 오류입니다.

방위각/상승 데이터 누락

오류: "하나 이상의 AzElSegment에 TimeAzEl 필드가 없습니다." AzElSegment

해결 방법: 방위각 고도 데이터의 각 세그먼트에 시간 태그가 지정된 방위각/상승 쌍이 하나 이상 포함되어 있는지 확인합니다.

잘못된 방위각 범위(도)

오류: "AzEl az는 -180보다 크거나 같아야 하고 360도보다 작거나 같아야 합니다."

해결 방법: 방위각이 $[-180, 360]$ 도 내에 있는지 확인합니다.

잘못된 고도 각도 범위(도)

오류: "AzEl el은 -90보다 크거나 같아야 하고 90도보다 작거나 같아야 합니다."

해결 방법: 고도 각도가 $[-90, 90]$ 도 이내인지 확인합니다.

잘못된 방위각 범위(라디안)

오류: "AzEl az는 $-\pi$ 보다 크거나 같아야 하고 2π 라디안보다 작거나 같아야 합니다."

해결 방법: 방위각이 $[-\pi, 2\pi]$ 라디안 내에 있는지 확인합니다.

잘못된 고도 각도 범위(라디안)

오류: "AzEl el은 $-\pi/2$ 보다 크거나 같아야 하고 $\pi/2$ 라디안보다 작거나 같아야 합니다."

해결 방법: 고도 각도가 $[-\pi/2, \pi/2]$ 라디안 내에 있는지 확인합니다.

단조롭지 않은 시간 값

오류: "AzElSegment 내의 TimeAzEl AzElSegment 항목은 일시적으로 순서대로 되어 있어야 합니다."

해결 방법: 각 세그먼트의 시간 값이 엄격하게 증가하는지 확인합니다.

세그먼트가 순서를 벗어남

오류: "AzElSegments는 일시적으로 순서대로 되어 있어야 합니다"

해결 방법: 세그먼트가 시간순으로 정렬되어 있는지 확인합니다.

세그먼트 겹침

오류: "하나 이상의 세그먼트의 시간 범위가 다른 세그먼트 시간 범위와 겹칩니다"

해결 방법: 각 세그먼트에 중첩되지 않는 고유한 시간 범위가 있는지 확인합니다. 한 세그먼트endTime의는 다음 세그먼트startTime의를 초과해서는 안 됩니다.

문제 해결 단계

에페메리스가 검증에 실패하는 경우 다음 단계에 따라 문제를 해결합니다.

1. 에페메리스 ID로 [DescribeEphemeris](#)를 호출하여 errorCode 및를 검색합니다errorMessage.
2. 오류 메시지를 검토하여 실패한 검증 검사에 대한 구체적인 세부 정보를 확인합니다.
3. 에페메리스 데이터에서 식별된 문제를 수정합니다.
4. [CreateEphemeris](#)를 사용하여 수정된 데이터가 포함된 새 에페메리스를 업로드합니다.
5. ENABLED 상태가 될 때까지 새 에페메리스 상태를 모니터링합니다.
6. 더 이상 필요하지 않은 경우 [DeleteEphemeris](#)를 사용하여 잘못된 에페메리스를 삭제합니다.

오류 코드 참조 완료

다음 섹션에서는 에페메리스 검증이 실패할 때 반환될 수 있는 모든 errorCode 값의 포괄적인 매핑을 상위 수준 invalidReason 범주별로 정리하여 제공합니다.

잘못된 이유: METADATA_INVALID

이러한 오류는 필수 메타데이터 필드가 누락되었거나, 형식이 잘못 지정되었거나, 에페메리스 데이터에 지원되지 않는 값이 포함된 경우에 발생합니다.

오류 코드	오류 메시지
MISMATCHED_SATCAT_ID	TLE 에페메리스에 있는 위성 카탈로그 번호가 관련 위성의 위성 카탈로그 번호와 일치하지 않습니다.
OEM_VERSION_UNSUPPORTED	OEM 에페메리스CCSDS_OEM_VERS 의는 지원되지 않습니다. 지원되는 값: [2.0]
ORIGINATOR_MISSING	OEM 에페메리스에서 ORIGINATOR 헤더 필드가 누락되었습니다.
CREATION_DATE_MISSING	OEM 에페메리스에서 CREATION_DATE 헤더 필드가 누락되었습니다.
OBJECT_NAME_MISSING	OEM 에페메리스에서 OBJECT_NAME 메타데이터 필드가 누락되었습니다.
OBJECT_ID_MISSING	OEM 에페메리스에서 OBJECT_ID 메타데이터 필드가 누락되었습니다.
REF_FRAME_UNSUPPORTED	OEM 에페메리스REF_FRAME 의는 지원되지 않습니다. 지원되는 값: [EME2000, ITRF2000]
REF_FRAME_EPOCH_UNSUPPORTED	OEM 에페메리스의 REF_FRAME_EPOCH 메타데이터 필드는 지원되지 않습니다. 에페메리스에서이 필드를 제거하세요.
TIME_SYSTEM_UNSUPPORTED	OEM 에페메리스TIME_SYSTEM 의는 지원되지 않습니다. 지원되는 값: [UTC]
CENTER_BODY_UNSUPPORTED	OEM 에페메리스CENTER_BODY 의는 지원되지 않습니다. 지원되는 값: [Earth]
인터폴레이션_누락	OEM 에페메리스에서 INTERPOLATION 메타데이터 필드가 누락되었습니다.
보간_정도_무효	OEM 에페메리스의 보간 정도는 보간 방법의 경우 0보다 커야 합니다.

오류 코드	오류 메시지
AZ_EL_SEGMENT_LIST_MISSING	azElSegmentList 필드가 누락되었습니다.
INSUFFICIENT_TIME_AZ_EL	하나 이상 의에 TimeAzEl 필드가 없습니다. azElSegmentList

잘못된 이유: **TIME_RANGE_INVALID**

이러한 오류는 시작/종료 시간, 세그먼트 순서 지정, 세그먼트 중첩 또는 시간 불일치 문제를 포함하여 에페메리스에 잘못된 시간 범위가 포함된 경우 발생합니다.

오류 코드	오류 메시지
START_TIME_IN_FUTURE	에페메리스 시작 시간은 미래이지만 과거여야 합니다.
END_TIME_IN_PAST	에페메리스 종료 시간은 과거이지만 미래여야 합니다.
EXPIRATION_TIME_TOO_EARLY	제공된 만료 시간이 임시 종료 시간보다 빠릅니다.
START_TIME_METADATA_TOO_EARLY	START_TIME 메타데이터 값이 OEM 에페메리스 데이터에 있는 가장 빠른 시간보다 빠름
STOP_TIME_METADATA_TOO_LATE	STOP_TIME 메타데이터 값이 OEM 에페메리스 데이터에 있는 최신 시간보다 늦음
AZ_EL_SEGMENT_END_TIME_BEFORE_START_TIME	하나 이상의 데이터 세그먼트 endTime 의가 세그먼트의 앞에 있음 startTime
AZ_EL_SEGMENT_TIME_S_OVERLAP	하나 이상의 세그먼트의 시간 범위가 다른 세그먼트 시간 범위와 겹칩니다.
AZ_EL_SEGMENTS_OUT_OF_ORDER	세그먼트가 일시적으로 정렬되지 않음
TIME_AZ_EL_ITEMS_OUT_OF_ORDER	내의 TimeAzEl 항목은 시간적 순서여야 AzElSegment 합니다.

오류 코드	오류 메시지
AZ_EL_SEGMENT_REFERENCE_EPOCH_INVALID	세그먼트의 참조 에포크가 잘못되었거나 형식이 잘못되었습니다.
AZ_EL_SEGMENT_START_TIME_INVALID	세그먼트의 유효한 시간 범위의 시작 시간은 첫 번째 세그먼트 이후에 시작되지 않습니다.
AZ_EL_SEGMENT_END_TIME_INVALID	세그먼트의 유효한 시간 범위의 종료 시간이 마지막 세그먼트 이후 종료되지 않음
AZ_EL_SEGMENT_VALID_TIME_RANGE_INVALID	세그먼트의 유효한 시간 범위가 잘못되었습니다.
AZ_EL_SEGMENT_END_TIME_TOO_LATE	세그먼트의 종료 시간이 참조 에포크에서 허용되는 최대 기간을 초과합니다.
AZ_EL_TOTAL_DURATION_EXCEEDED	모든 세그먼트의 총 지속 시간이 허용되는 최대 지시각 지속 시간을 초과합니다.

잘못된 이유: **TRAJECTORY_INVALID**

이러한 오류는 궤도 파라미터, 각도 범위 또는 단위 문제를 포함하여 에페메리스에 잘못된 궤적 데이터가 포함된 경우 발생합니다.

오류 코드	오류 메시지
MEAN_MOTION_INVALID	제공된 TLE 에페메리스의 평균 모션은 가장 최근의 참조 에페메리스와 너무 크게 다릅니다. 참고: Ground Station은 검증 중에 스페이스 트랙 에페메리스를 참조로 사용합니다.
TIME_AZ_EL_AZ_RADIAN_RANGE_INVALID	Az티은 $-\pi$ 보다 크거나 같아 az 야 하며 2π 라디안보다 작거나 같아야 합니다.
TIME_AZ_EL_EL_RADIAN_RANGE_INVALID	Az티은 $-\pi/2$ 보다 크거나 같아 el 야 하며 $\pi/2$ 라디안보다 작거나 같아야 합니다.

오류 코드	오류 메시지
TIME_AZ_EL_AZ_DEGREE_RANGE_INVALID	AzEI은 -180보다 크거나 같아 az 야 하고 360도보다 작거나 같아야 합니다.
TIME_AZ_EL_EL_DEGREE_RANGE_INVALID	AzEI은 -90도 이상, 90도 이하여야 el 합니다.
TIME_AZ_EL_ANGLE_UNITS_INVALID	잘못된 AzEI 각도 단위

잘못된 이유: **KMS_KEY_INVALID**

이러한 오류는 에페메리스 데이터를 암호화하는 데 사용되는 AWS Key Management Service (KMS) 키에 문제가 있을 때 발생합니다.

오류 코드	오류 메시지
INSUFFICIENT_KMS_PERMISSIONS	Ground Station에이 에페메리스의 KMS 키에 액세스할 수 있는 충분한 권한이 없습니다.

잘못된 이유: **VALIDATION_ERROR**

이러한 오류는 다른 특정 범주에 속하지 않는 에페메리스 데이터에 일반적인 검증 문제가 있을 때 발생합니다.

오류 코드	오류 메시지
INTERNAL_ERROR	에페메리스 검증 중에 내부 오류가 발생했습니다.
FILE_FORMAT_INVALID	에페메리스 파일 형식이 잘못되었거나 손상되었습니다. 파일이 에페메리스 유형의 예상 형식을 준수하는지 확인합니다.

데이터를 수신하지 못한 고객 응대 문제 해결

고객 응대는 성공으로 보일 수 있지만 여전히 데이터를 수신하지 못했습니다. 즉, 비어 있는 PCAP 파일을 수신하거나 S3 데이터 전송을 사용하는 경우 PCAP 파일이 전혀 수신되지 않을 수 있습니다. 이는 여러 이유로 발생할 수 있습니다. 다음은 몇 가지 원인과 이를 해결하는 방법을 설명합니다.

잘못된 다운링크 구성

위성에서 데이터를 수신하는 각 연락처에는 [안테나 다운링크 구성](#) 또는가 연결됩니다.[안테나 다운링크 복조 디코드 구성](#). 지정된 구성이 위성에 의해 전송되는 신호와 일치하지 않는 경우 AWS Ground Station 는 전송된 신호를 수신할 수 없습니다. 이렇게 하면에서 데이터를 수신하지 않습니다 AWS Ground Station.

이 문제를 해결하려면 사용 중인 구성이 위성에서 전송되는 신호와 일치하는지 확인하세요. 예를 들어 중심 주파수, 대역폭, 편광 및 필요한 경우 복조 및 디코딩 파라미터를 올바르게 설정했는지 확인합니다.

위성 조작

위성이 일부 통신 시스템을 일시적으로 비활성화하는 조작을 수행할 수 있는 경우가 있습니다. 또한이 조작은 하늘에서 위성의 위치를 크게 변경할 수 있습니다. AWS Ground Station 는 신호를 전송하지 않는 위성으로부터 신호를 수신할 수 없거나, 사용 중인 에페메리스로 인해 AWS Ground Station 안테나가 위성이 없는 하늘의 위치를 가리키는 경우입니다.

NOAA에서 운영하는 퍼블릭 브로드캐스트 위성과 통신하려는 경우 NOAA [위성 알림 메시지](#) 페이지에서 중단 또는 조작을 설명하는 메시지를 찾을 수 있습니다. 메시지에는 데이터 전송이 재개될 것으로 예상되는 타임라인이 포함되거나 후속 메시지에 게시될 수 있습니다.

자체 위성과 통신하는 경우 위성 작업과 이것이 통신에 미치는 영향을 이해하는 것은 사용자의 책임입니다 AWS Ground Station. 위성 궤적에 영향을 미치는 조작을 수행하는 경우 여기에는 업데이트된 사용자 지정 에페메리스 데이터 제공이 포함될 수 있습니다. 사용자 지정 에페메리스 데이터 제공에 대한 자세한 내용은 섹션을 참조하세요 [가 에페메리스를 AWS Ground Station 사용하는 방법 이해](#).

AWS Ground Station 중단

AWS Ground Station 에서 고객 응대가 실패하거나 취소되면 AWS Ground Station 는 고객 응대 상태를 AWS_FAILED 또는 AWS_CANCELLED로 설정합니다. 고객 응대 수명 주기에 대한 자세한 내용은 섹션을 참조하세요 [고객 응대 수명 주기 이해](#). 경우에 따라 AWS Ground Station 에 오류가 발생하여 데이터가 계정으로 전송되지 않지만 고객 응대가 AWS_FAILED 또는 AWS_CANCELLED 상태가 되지

않을 수 있습니다. 이 경우는 계정별 이벤트를 AWS 상태 대시보드에 AWS Ground Station 게시해야 합니다. 상태 대시보드에 AWS 대한 자세한 내용은 [AWS 상태 사용 설명서를 참조하세요](#).

원격 측정 문제 해결

다음 정보를 사용하여 원격 측정과 관련된 일반적인 문제를 해결합니다.

일반적인 설정 문제

IAM 권한 오류

증상

를 호출CreateConfig하여 TelemetrySinkConfig를 생성할 때 오류가 발생합니다.

```
Unable to write to Kinesis Data Streams stream. Ensure that Ground Station has
kinesis:PutRecord permissions for the given stream
```

원인

- TelemetrySinkConfig에 지정된 IAM 역할에는 Kinesis Data Streams 스트림에 쓰는 데 필요한 권한이 없습니다.
- IAM 역할의 신뢰 정책은가 역할을 수임 AWS Ground Station 하도록 허용하지 않습니다.
- TelemetrySinkConfig의 Kinesis Data Streams 스트림 ARN이 잘못되었거나 스트림이 존재하지 않습니다.

솔루션

1. IAM 역할이 존재하고 올바른 권한이 있는지 확인합니다. 모든 단계를 [2단계: TelemetrySinkConfig 생성](#) 검토하고 준수했는지 확인합니다.
2. 가 IAM 역할을 수임 AWS Ground Station 할 수 있는지 확인합니다.

```
aws iam get-role --role-name GroundStationTelemetryRole
```

신뢰 정책에가 신뢰할 수 있는 서비스 보안 주체groundstation.amazonaws.com로 포함되어 있는지 확인합니다.

3. IAM 역할에 필요한 Kinesis 권한이 있는지 확인합니다.

```
aws iam list-attached-role-policies --role-name GroundStationTelemetryRole
```

정책에 스트림에 대한 `kinesis:DescribeStreamkinesis:PutRecord`, 및 `kinesis:PutRecords` 권한이 포함되어 있는지 확인합니다.

4. Kinesis Data Streams 스트림이 존재하고 ARN이 올바른지 확인합니다.

```
aws kinesis describe-stream \
  --stream-name your-stream-name \
  --region us-east-2
```

5. 고객 관리형 암호화를 사용하는 경우 IAM 역할에 AWS KMS 키에 대한 `kms:GenerateDataKey` 권한이 있는지 확인합니다.

PassRole 권한 오류

증상

를 호출CreateConfig할 때 IAM 역할을 전달할 권한이 없다는 오류가 발생합니다.

솔루션

IAM 사용자 또는 역할에 원격 측정 IAM 역할에 대한 `iam:PassRole` 권한이 있는지 확인합니다. 사용자 또는 역할에 다음 정책을 추가합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/your-stream-name"
    }
  ]
}
```

Kinesis Data Streams 스트림 구성 문제

증상

원격 측정 전송이 실패하거나 간헐적입니다.

원인

- Kinesis Data Streams 스트림의 용량이 원격 측정 처리량에 충분하지 않습니다.
- 스트림이 다른 애플리케이션에서 사용 중이므로 쓰기 제한이 발생합니다.

솔루션

1. 스트림 상태를 확인합니다.

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

2. CloudWatch 지표를 사용하여 쓰기 제한을 모니터링합니다.

```
aws cloudwatch get-metric-statistics \  
  --namespace AWS/Kinesis \  
  --metric-name WriteProvisionedThroughputExceeded \  
  --dimensions Name=StreamName,Value=your-stream-name \  
  --start-time 2025-12-08T00:00:00Z \  
  --end-time 2025-12-08T23:59:59Z \  
  --period 60 \  
  --statistics Sum \  
  --region us-east-2
```

3. 제한이 감지되면 다음을 고려하세요.

- 자동 조정을 위해 온디맨드 용량 모드로 전환합니다.
- AWS Ground Station 원격 측정을 위한 전용 스트림 사용.
- 프로비저닝된 모드를 사용하는 경우 샤드 수를 늘립니다.

원격 측정 전송 문제

원격 측정 데이터가 표시되지 않음

증상

원격 측정이 활성화된 미션 프로파일로 고객 응대를 예약하면 Kinesis Data Streams 스트림에 원격 측정 데이터가 표시되지 않습니다.

가능한 원인 및 해결 방법

미션 프로파일에 원격 측정이 활성화되어 있지 않음

연락에 사용되는 미션 프로파일에게 포함되어 있는지 확인합니다 `telemetrySinkConfigArn`.

```
aws groundstation get-mission-profile \  
  --mission-profile-id 12345678-1234-1234-1234-123456789012 \  
  --region us-east-2
```

`telemetrySinkConfigArn` 필드의 출력을 확인합니다. 존재하지 않는 경우 미션 프로파일에 원격 측정이 활성화되어 있지 않습니다.

IAM 역할 권한 문제

의 IAM 권한 문제 해결 단계를 검토합니다 [IAM 권한 오류](#).

Kinesis Data Streams 스트림이 존재하지 않거나 잘못된 리전에 있음

스트림이 올바른 리전에 있는지 확인합니다.

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

연락처가 아직 시작되지 않음

원격 측정 전송은 고객 응대 시작 시간에 시작됩니다. 고객 응대 상태를 확인하여 고객 응대가 시작되었는지 확인합니다.

```
aws groundstation describe-contact \  
  --contact-id 12345678-1234-1234-1234-123456789012 \  
  --region us-east-2
```

디지털 트윈 지상국에 예약된 고객 응대

디지털 트윈 지상국은 원격 측정 전송을 지원하지 않습니다. 디지털 트윈 안테나에 예약된 연락처는 원격 측정 데이터를 생성하지 않습니다. 지상국 이름을 확인하여 고객 응대에 사용되는 지상국이 디지털 트윈이 아닌지 확인합니다. 디지털 트윈 지상국의 이름에는 “디지털 트윈” 접두사가 있습니다. 자세한 내용은 [AWS Ground Station 디지털 트윈 기능 사용](#) 섹션을 참조하세요.

간헐적 원격 측정 데이터

증상

원격 측정 데이터는 격차 또는 누락된 레코드와 일관되지 않게 전송됩니다.

가능한 원인

- Kinesis Data Streams 스트림 용량 문제 또는 제한. [Kinesis Data Streams 스트림 구성 문제](#) 섹션을 참조하세요.
- AWS Ground Station 와 Kinesis Data Streams 스트림 간의 네트워크 연결 문제입니다.

솔루션

- CloudWatch에서 Kinesis Data Streams 스트림 지표의 제한 또는 오류를 모니터링합니다.
- 스트림이 온디맨드 용량 모드를 사용하고 있거나 충분한 프로비저닝 용량이 있는지 확인합니다.
- AWS Ground Station 원격 측정 전용 스트림을 사용하여 다른 애플리케이션과의 경합을 방지합니다.

데이터 형식 문제

JSON 구문 분석 오류

증상

원격 측정 레코드를 JSON으로 구문 분석할 때 애플리케이션에 오류가 발생합니다.

솔루션

- Base64 디코딩 확인 - Kinesis Data Streams 스트림의 데이터는 Base64-encoded. JSON으로 구문 분석하기 전에 데이터를 디코딩해야 합니다. 자세한 내용은 [Kinesis Data Streams 스트림에서 데이터 읽기](#) 섹션을 참조하세요.

- 빈 레코드 확인 - TelemetrySinkConfig를 생성할 때 빈 검증 레코드를 보낼 AWS Ground Station 수 있습니다. 애플리케이션은 비어 있거나 잘못된 형식의 레코드를 정상적으로 처리해야 합니다.
- 버전 인식 구문 분석 구현 - 먼저 telemetryTypeAndVersion, telemetryType 및 telemetryVersion 필드를 구문 분석하여 각 레코드에 적합한 스키마를 결정합니다.

알 수 없는 원격 측정 유형 또는 버전

증상

애플리케이션에서 인식하지 못하는 원격 측정 유형 또는 버전이 발생합니다.

솔루션

이는 시간이 지남에 따라 새로운 원격 측정 유형 및 스키마 버전이 도입될 수 있으므로 예상되는 동작입니다. 애플리케이션은 다음을 수행해야 합니다.

- 모니터링을 위해 알 수 없는 유형 및 버전을 로깅합니다.
- 알려진 유형 및 버전을 계속 처리합니다.
- 알 수 없는 스키마에 대한 정상적인 처리를 구현합니다.

스키마 버전 관리에 대한 자세한 내용은 섹션을 참조하세요 [스키마 버전 관리 및 진화](#).

도움말 가져오기

문제 해결 단계를 수행한 후에도 문제가 계속 발생하면 AWS Support에 문의하세요.

제공할 정보

지원 팀에 문의할 때 다음 정보를 제공합니다.

- 문제가 발생한 연락처 IDs
- 사용된 미션 프로필 ID
- TelemetrySinkConfig ARN
- Kinesis Data Streams 스트림 ARN
- IAM 역할 ARN 및 연결된 정책
- CloudWatch Logs 또는 애플리케이션의 오류 메시지
- 문제가 발생한 시점의 타임스탬프

- 이미 수행된 문제 해결 단계

일반적인 AWS Ground Station 지원은 [AWS Ground Station 사용 설명서를](#) 참조하세요.

할당량 및 제한

지원되는 리전, 관련 엔드포인트, 엔드포인트 및 할당량에서 [AWS Ground Station 할당량을 볼 수 있습니다](#).

[Service Quotas 콘솔](#), [AWS API](#) 및 [AWS CLI](#)를 사용하여 필요한 경우 할당량 증가를 요청할 수 있습니다.

서비스 약관

AWS Ground Station 서비스 약관은 [AWS 서비스 약관](#)을 참조하세요.

AWS Ground Station 사용 설명서의 문서 기록

다음 표에서는 AWS Ground Station 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다.

변경 사항	설명	날짜
새로운 기능	Alpha-5 TLE 위성 번호 지원에 대한 설명서가 추가되었습니다. AWS Ground Station 이제는 100,000~339,999 범위의 위성 카탈로그 번호에 대해 Alpha-5 인코딩 위성 번호가 있는 TLE 에페메리스를 허용합니다. 자세한 내용은 TLE 에페메리스 데이터 제공 을 참조하세요.	2026년 5월 28일
새로운 기능	AWS Ground Station 전용 안테나에 대한 설명서가 추가되었습니다. 자세한 내용은 AWS Ground Station 전용 안테나를 참조하세요 .	2026년 4월 14일
새로운 기능	UpdateContact API 및 연락처 버전 관리에 대한 설명서가 추가되었습니다. 자세한 내용은 연락처 업데이트 및 연락처 버전 관리를 참조하세요 .	2026년 4월 14일
새로운 기능	ListAntennas 및 ListGroundStationReservations APIs에 대한 설명서가 추가되었습니다. 자세한 내용은 AWS Ground Station 위치 및 지상국 예약 보기 를 참조하세요. https://docs.aws.amazon.com/	2026년 4월 14일

[ground-station/latest/ug/locations.reservations.html](https://docs.aws.amazon.com/ground-station/latest/ug/locations.reservations.html)

설명서 업데이트

CancelContact API에 추가 기능이 추가되었으며, 여기에는 해당 기능 및 측정 영향에 대한 정보가 포함됩니다. 자세한 내용은 [고객 응대 측정 이해를 참조하세요](#).

2025년 12월 10일

설명서 업데이트

CloudWatch 지표가 고객 응대의 지상국과 연결된 리전에서 방출됨을 명확히 했습니다. 끊어진 링크를 수정했습니다.

2025년 12월 2일

업데이트된 AWS 관리형 정책

AWS Ground Station 는 작업 응답 URL을 검색AWSGroundStationAgentInstancePolicy 하기 위한 추가 권한을 포함하도록 관리형 정책을 업데이트했습니다. URLs 자세한 내용은 [AWS Ground StationAWS 관리형 정책에 대한 업데이트를 참조하세요](#).

2025년 11월 13일

새로운 기능

방위각 고도 에페메리스를 포함하도록 사용 설명서를 업데이트했습니다. 자세한 내용은 [방위각 고도 에페메리스 데이터 제공을 참조하세요](#).

2025년 10월 22일

설명서 업데이트

리전 간 데이터 전송에는 더 이상 특별한 구성이나 승인이 필요하지 않습니다. 자세한 내용은 [리전 간 데이터 전송 사용을 참조하세요](#).

2025년 9월 11일

설명서 업데이트	구성된 리소스의 고객 응대 사용률에 대한 설명이 추가되었습니다.	2025년 4월 4일
새로운 기능	AWS Ground Station 디지털 트윈을 포함하도록 사용 설명서를 업데이트했습니다.	2024년 8월 6일
설명서 업데이트	새 다이어그램, 예제 등을 포함하여 사용 설명서의 여러 섹션을 업데이트했습니다.	2024년 7월 18일
설명서 업데이트	사용 설명서에 RSS 피드를 추가했습니다.	2024년 7월 18일
설명서 업데이트	AWS Ground Station 에이전트 사용 설명서를 별도의 사용 설명서로 분할합니다.	2024년 7월 18일
새로운 기능	이제 가시성 시간 범위를 벗어나 최대 30초까지 고객 응대를 예약할 수 있습니다. 표시 시간은 DescribeContact 응답에 포함됩니다.	2024년 3월 26일
설명서 업데이트	조직을 개선하고 “EC2 인스턴스 선택 및 CPU 계획” 섹션을 추가했습니다.	2024년 3월 6일
설명서 업데이트	AWS Ground Station 에이전트와 함께 서비스 및 프로세스를 실행하기 위한 새로운 모범 사례를 AWS Ground Station 에이전트 사용 설명서에 추가했습니다.	2024년 2월 23일
설명서 업데이트	에이전트 릴리스 정보 페이지가 추가되었습니다.	2024년 2월 21일

템플릿 업데이트	DirectBroadcastSatelliteWbD igIfEc2DataDelivery 템플릿에 별도의 퍼블릭 서브넷에 대한 지원이 추가되었습니다.	2024년 2월 14일
설명서 업데이트	모니터링 설명서에 AWS 사용 자 알림 에 대한 추천을 추가했 습니다.	2023년 8월 6일
설명서 업데이트	AWS Ground Station 콘솔에 표시할 이름으로 위성에 태그 를 지정하기 위한 지침이 추가 되었습니다.	2023년 7월 26일
새로운 기능	광대역 DigIF 데이터 전송 릴 리스에 대한 AWS Ground Station 에이전트 사용 설명서 가 추가되었습니다.	2023년 8월 12일
새로운 AWS 관리형 정책	AWS Ground Station 에 AWSGroundStationAg entInstancePolicy라는 새 정책 이 추가되었습니다.	2023년 8월 12일
새로운 기능	CPE 프리뷰 릴리스를 위한 사 용 설명서가 업데이트되었습니 다.	2022년 11월 9일
새로운 AWS 관리형 정책	AWS Ground Station 는 AWSServiceRoleForG roundStationDataflowEndpoin tGroup service-linked-rol e(SLR)이라는 새 정책을 포함 하는 AWSServiceRoleForG roundStationDataflowEndpoin tGroupPolicy를 추가했습니다.	2022년 11월 2일

새로운 기능	와의 통합을 포함하도록 사용 설명서를 업데이트했습니다 AWS CLI.	2020년 4월 17일
새로운 기능	CloudWatch 지표와의 통합을 포함하도록 사용 설명서를 업데이트했습니다.	2020년 2월 24일
새 템플릿	AWS Ground Station 사용 설명서에 퍼블릭 브로드캐스트 위성(AquaSnppJpss 템플릿)이 추가되었습니다.	2020년 2월 19일
새로운 기능	교차 리전 데이터 전송을 포함하도록 사용 설명서가 업데이트되었습니다.	2020년 2월 5일
설명서 업데이트	CloudWatch Events를 AWS Ground Station 사용한 모니터링에 대한 예제와 설명이 업데이트되었습니다.	2020년 2월 4일
설명서 업데이트	템플릿 위치가 업데이트되었으며, 시작하기 및 문제 해결 섹션이 수정되었습니다.	2019년 12월 19일
새로운 문제 해결 섹션	문제 해결 섹션이 에 추가되었습니다.	2019년 11월 7일
새로운 시작하기 주제	최신 CloudFormation 템플릿을 포함하는 시작하기 주제가 업데이트되었습니다.	2019년 7월 1일
킨들 버전	AWS Ground Station 사용 설명서의 공개된 Kindle 버전입니다.	2019년 6월 20일

새로운 서비스 및 가이드

및 AWS Ground Station AWS
Ground Station 사용 설명서의
최초 릴리스입니다.

2019년 5월 23일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조의 [AWS 용어집](#)을 참조하세요.

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