



ユーザーガイド

# AWS Ground Station



# AWS Ground Station: ユーザーガイド

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

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

# Table of Contents

とは AWS Ground Station .....	1
一般的なユースケース .....	1
次のステップ .....	2
の AWS Ground Station 仕組み .....	3
衛星オンボーディング .....	3
ミッションプロファイルの構成 .....	3
問い合わせのスケジューリング .....	5
問い合わせの実行 .....	6
デジタルツイン .....	9
AWS Ground Station コアコンポーネントを理解する .....	9
ミッションプロファイル .....	11
設定 .....	14
データフローエンドポイントグループ .....	23
AWS Ground Station エージェント .....	31
はじめに .....	33
にサインアップする AWS アカウント .....	33
管理アクセスを持つユーザーを作成する .....	33
AWS アカウントにアクセス AWS Ground Station 許可を追加する .....	35
衛星搭載 .....	37
顧客のオンボーディングプロセスの概要 .....	37
(オプション) 衛星の名前付け .....	37
パブリックブロードキャスト衛星 .....	40
データフロー通信パスを計画する .....	41
非同期データ配信 .....	41
同期データ配信 .....	42
テレメトリを計画する .....	43
設定の作成 .....	44
データ配信設定 .....	44
テレメトリ設定 (オプション) .....	44
衛星設定 .....	44
ミッションプロファイルを作成する .....	45
次のステップを理解する .....	46
AWS Ground Station ロケーション .....	47
地上局の場所の AWS リージョンの検索 .....	47

AWS Ground Station サポートされている AWS リージョン .....	49
デジタルツインの可用性 .....	49
AWS Ground Station サイトマスク .....	49
お客様固有のマスク .....	50
利用可能なコンタクト時間に対するサイトマスクの影響 .....	50
AWS Ground Station サイト機能 .....	51
がエフェメリス AWS Ground Station を使用する方法を理解する .....	54
デフォルトのエフェメリスデータ .....	54
カスタムエフェメリスデータを提供する .....	55
概要: .....	55
例: お客様提供のエフェメリスを で使用する AWS Ground Station .....	56
TLE エフェメリスデータを提供する .....	56
OEM エフェメリスデータを提供する .....	62
方位高度エフェメリスデータを提供する .....	70
カスタムエフェメリスを使用してコンタクトを予約する .....	81
概要: .....	81
問い合わせ予約ワークフロー .....	81
ワークフロー 1: 利用可能な連絡先を一覧表示し、予約する .....	82
ワークフロー 2: 直接連絡の予約 .....	86
問い合わせ状態の変更のモニタリング .....	90
ベストプラクティスと考慮事項 .....	92
どのエフェメリスが使用されているかを理解する .....	93
TLE および OEM エフェメリス .....	93
方位高度エフェメリス .....	93
以前にスケジュールされたコンタクトに対する新しいエフェメリスの影響 .....	94
衛星の現在のエフェメリスを取得する .....	95
デフォルトのエフェメリスを使用した衛星の GetSatellite リターンの例 .....	95
カスタムエフェメリスを使用した衛星の GetSatellite の例 .....	96
方位高度エフェメリスの一覧表示 .....	96
デフォルトのエフェメリスデータに戻す .....	97
TLE エフェメリスと OEM エフェメリスの元に戻す .....	98
方位高度エフェメリスの管理 .....	98
データフローの使用 .....	99
AWS Ground Station データプレーンインターフェイス .....	99
クロスリージョンデータ配信の使用 .....	100
Amazon S3 のセットアップと設定 .....	100

Amazon VPC のセットアップと設定 .....	100
AWS Ground Station エージェントによる VPC 設定 .....	101
データフローエンドポイントを使用した VPC 設定 .....	104
Amazon EC2 のセットアップと設定 .....	106
提供される一般的なソフトウェア .....	107
AWS Ground Station Amazon マシンイメージ (AMIs) .....	107
テレメトリの操作 .....	108
テレメトリの仕組み .....	108
使用可能なテレメトリタイプ .....	108
リージョナルな可用性 .....	109
テレメトリの設定 .....	109
ステップ 1: 前提条件 AWS となるリソースを作成する .....	110
ステップ 2: TelemetrySinkConfig を作成する .....	111
ステップ 3: ミッションプロファイルにテレメトリを追加する .....	112
ステップ 4: コンタクトをスケジュールする .....	112
次の手順 .....	113
テレメトリデータを理解する .....	113
データ形式の概要 .....	113
ポイントテレメトリ .....	114
テレメトリの追跡 .....	116
Kinesis Data Streams ストリームからのデータの読み取り .....	117
スキーマのバージョンニングと進化 .....	118
連絡先の操作 .....	120
問い合わせのライフサイクルを理解する .....	120
AWS Ground Station 問い合わせステータス .....	123
問い合わせデータの保持 .....	124
問い合わせの請求を理解する .....	125
帯域幅の定義 .....	125
スケジュールモード .....	125
CancelContact .....	125
シナリオ 1: 単一問い合わせ .....	126
シナリオ 2: 単一停止コンタクト .....	127
シナリオ 3: 重複が 1 つ .....	127
シナリオ 4: 短い重複 .....	128
シナリオ 5: 複数の重複 .....	129
シナリオ 6: 複数の停止 .....	131

シナリオ 7: 重複しないマルチアンテナ地上ステーション .....	132
シナリオ 8: コンタクトが重複しているマルチアンテナ地上ステーション .....	133
AWS Ground Station デジタルツイン .....	135
モニタリング .....	136
イベントによる自動化 .....	137
AWS Ground Station イベントタイプ .....	138
問い合わせイベントのタイムライン .....	138
エフェメリスイベント .....	140
CloudTrail を使用した API コールのログ記録 .....	141
AWS Ground Station CloudTrail の情報 .....	142
AWS Ground Station ログファイルエントリについて .....	143
Amazon CloudWatch でメトリクスを表示する .....	144
AWS Ground Station メトリクスとディメンション .....	145
メトリクスの表示 .....	150
セキュリティ .....	157
Identity and Access Management .....	157
オーデイエンス .....	158
アイデンティティを使用した認証 .....	158
ポリシーを使用したアクセスの管理 .....	159
が IAM と AWS Ground Station 連携する方法 .....	161
アイデンティティベースのポリシーの例 .....	167
トラブルシューティング .....	170
AWS マネージドポリシー .....	172
AWSGroundStationAgentInstancePolicy .....	172
AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy .....	173
ポリシーの更新 .....	173
サービスリンクロールを使用する .....	174
Ground Station のサービスにリンクされたロールのアクセス許可 .....	175
Ground Station へのサービスにリンクされたロールの作成 .....	176
Ground Station でのサービスにリンクされたロールの編集 .....	176
Ground Station でのサービスにリンクされたロールの削除 .....	176
Ground Station のサービスにリンクされたロールがサポートされるリージョン .....	177
トラブルシューティング .....	177
の保管中のデータ暗号化 AWS Ground Station .....	177
カスターマネージドキーを作成する .....	179
のカスターマネージドキーの指定 AWS Ground Station .....	180

AWS Ground Station 暗号化コンテキスト .....	181
TLE および OEM エフェメリスデータの保管時の暗号化 .....	181
方位高度エフェメリスの保管時の暗号化 .....	190
の転送中のデータ暗号化 AWS Ground Station .....	199
AWS Ground Station エージェントストリーム .....	199
データフローエンドポイントストリーム .....	200
ミッションプロファイル設定の例 .....	201
JPSS-1 - パブリックブロードキャスト衛星 (PBS) - 評価 .....	201
Amazon S3 データ配信を利用するパブリックブロードキャスト衛星 .....	202
通信パス .....	203
AWS Ground Station 設定 .....	205
AWS Ground Station ミッションプロファイル .....	206
まとめる .....	206
データフローエンドポイントを利用するパブリックブロードキャスト衛星 (ナローバンド) .....	208
通信パス .....	208
AWS Ground Station 設定 .....	215
AWS Ground Station ミッションプロファイル .....	216
まとめる .....	217
データフローエンドポイントを利用するパブリックブロードキャスト衛星 (復調および復号 化) .....	219
通信パス .....	219
AWS Ground Station 設定 .....	226
AWS Ground Station ミッションプロファイル .....	229
まとめる .....	230
AWS Ground Station エージェント (広帯域) を利用するパブリックブロードキャスト衛星 .....	232
通信パス .....	232
AWS Ground Station 設定 .....	244
AWS Ground Station ミッションプロファイル .....	245
まとめる .....	246
トラブルシューティング .....	249
Amazon EC2 にデータを配信する問い合わせのトラブルシューティング .....	249
ステップ 1: EC2 インスタンスが実行されていることを確認する .....	250
ステップ 2: 使用するデータフローアプリケーションのタイプを決定する .....	250
ステップ 3: データフローアプリケーションが実行されていることを確認する .....	250
ステップ 4: データフローアプリケーションストリームが設定されていることを確認する ..	252

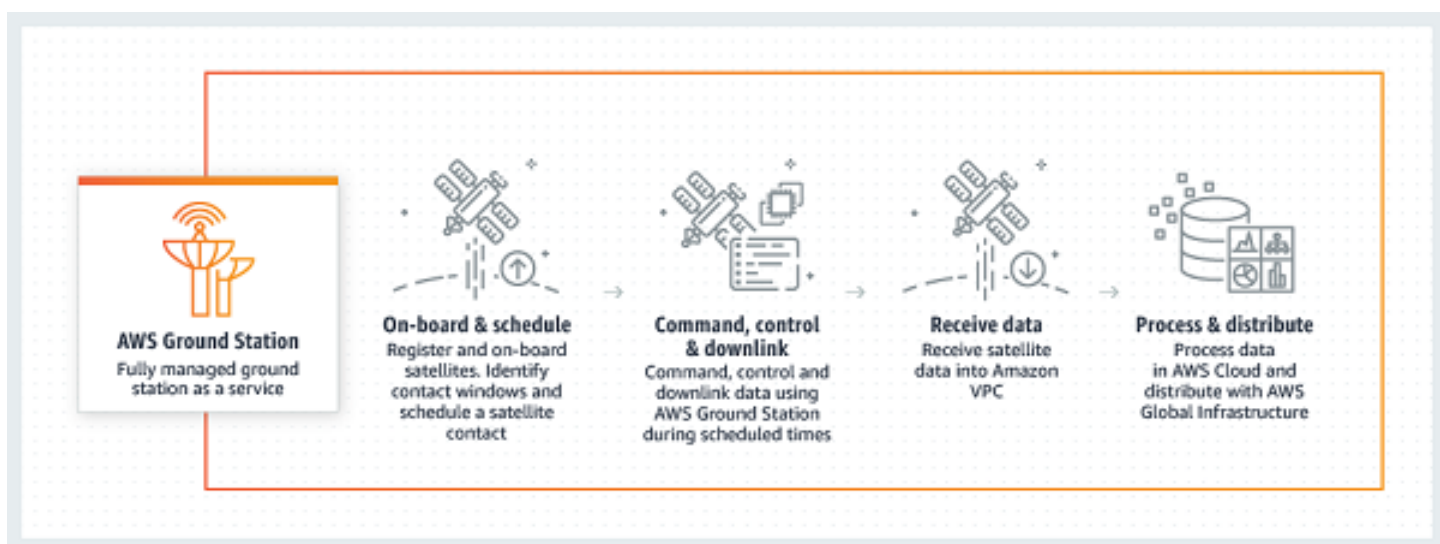
ステップ 5: レシーバーインスタンス (複数可) サブネットに十分な使用可能な IP アドレスがあることを確認する .....	254
失敗した問い合わせのトラブルシューティング .....	255
データフローエンドポイントの失敗のユースケース .....	255
AWS Ground Station エージェントが失敗したユースケース .....	256
FAILED_TO_SCHEDULE コンタクトのトラブルシューティング .....	257
Antenna Downlink Demod Decode Config で指定された設定はサポートされていません .....	257
一般的なトラブルシューティングステップ .....	258
DataflowEndpointGroups が正常でない状態のトラブルシューティング .....	258
無効なエフェメリスのトラブルシューティング .....	259
エフェメリス検証エラーについて .....	259
TLE エフェメリスの一般的な検証エラー .....	260
OEM エフェメリスの一般的な検証エラー .....	260
方位高度エフェメリスの一般的な検証エラー .....	261
トラブルシューティングのステップ .....	262
完全なエラーコードリファレンス .....	262
データを受信しなかった問い合わせのトラブルシューティング .....	267
正しくないダウンリンク設定 .....	267
衛星操作 .....	267
AWS Ground Station 停止 .....	267
テレメトリのトラブルシューティング .....	268
一般的なセットアップの問題 .....	268
テレメトリ配信の問題 .....	271
データ形式の問題 .....	272
ヘルプの利用 .....	273
クォータと制限 .....	275
サービス条件 .....	276
ドキュメント履歴 .....	277
AWS 用語集 .....	282
.....	cclxxxiii

# とは AWS Ground Station

AWS Ground Station は、グローバルインフラストラクチャ全体で安全で高速、予測可能な衛星通信を提供するフルマネージドサービスです。を使用すると AWS Ground Station、独自の地上ステーションインフラストラクチャを構築、管理、スケーリングする必要がなくなります。AWS Ground Station を使用すると、独自の地上ステーションの構築、運用、スケーリングにリソースを費やすのではなく、衛星データを取り込む新しいアプリケーションの革新と迅速な実験に集中できます。

AWS の低レイテンシー、高帯域幅のグローバルファイバーネットワークを使用すると、アンテナシステムでの受信から数秒以内に衛星データの処理を開始できます。これにより、生データを処理された情報または分析された知識に数秒で変換できます。

## 一般的なユースケース



AWS Ground Station では、衛星と双方向に通信でき、次のユースケースをサポートしています。

- **ダウンリンクデータ** – 衛星からデータを受信し、XバンドおよびSバンド周波数を送信して、Amazon EC2 インスタンスにリアルタイムで (VITA-49 形式)、またはアカウントの Amazon S3 バケットに直接 ([PCAP 形式](#)) 配信します。さらに、サポートされているモジュレーションとエンコーディングスキームを使用する衛星の場合、復調およびデコードされたデータを受信するか、生のデジタル中間周波数 (DigIF) サンプル (VITA-49 形式) を受信するかを選択できます。
- **アップリンクデータ** – 送信元の DigIF データ (VITA-49 形式) を送信することで、Sバンド周波数を受信するデータとコマンドを衛星に送信します AWS Ground Station。

- アップリンクエコー – 物理的にコロケーションされたアンテナで送信信号を受信することで、宇宙機に送信されたコマンドを検証し、その他の高度なタスクを実行します。
- Software Defined Radio (SDR)/Front End Processor (FEP) – Amazon EC2 インスタンスで実行できる既存の SDR や FEP を使用して、データをリアルタイムで処理して既存の波形を送受信し、データ製品を生成します。
- Telemetry, Tracking, and Command (TT&C) – 前述のユースケースを組み合わせると TT&C を実行し、衛星フリートを管理します。
- クロスリージョンデータ配信 — 単一の AWS リージョンから AWS Ground Station のグローバルアンテナネットワークを使用して、複数の同時コンタクトを運用します。
- デジタルツイン – 本稼働アンテナ容量を使用せずに、低コストでスケジューリング、設定の検証、適切なエラー処理をテストできます。

## 次のステップ

最初に以下のセクションを読むことをお勧めします。

- 重要な AWS Ground Station 概念については、「」を参照してくださいの [AWS Ground Station 仕組み](#)。
- 使用するアカウントとリソースを設定する方法については AWS Ground Station、「」を参照してください [はじめに](#)。
- プログラムで使用するには AWS Ground Station、[AWS Ground Station API リファレンス](#)を参照してください。API リファレンスでは、のすべての API オペレーション AWS Ground Station について詳しく説明しています。また、サポートされているウェブサービスプロトコルのサンプルリクエスト、レスポンス、エラーも提供します。[AWS CLI](#) または [AWS SDK](#) を任意の言語で使用して、とやり取りするコードを記述できます AWS Ground Station。

## の AWS Ground Station 仕組み

AWS Ground Station は地上アンテナを操作して、衛星との通信を容易にします。アンテナが実行できる物理的な特性は抽象化され、機能と呼ばれます。アンテナの物理的な場所と現在の機能は、[AWS Ground Station ロケーション](#)セクションで参照できます。ユースケース[AWS Support Center Console](#)に追加の機能、追加のロケーションサービス、またはより正確なアンテナロケーションが必要な場合は、[お問い合わせ](#)を通じてお問い合わせください。

AWS Ground Station アンテナのいずれかを使用するには、特定の場所で時間を予約する必要があります。この予約は問い合わせと呼ばれます。問い合わせを正常にスケジュールするには、[お問い合わせ](#)は成功するために追加のデータ AWS Ground Station を必要とします。

- 衛星は 1 つ以上の場所にオンボードする必要があります。これにより、リクエストされた場所でさまざまな機能を運用する承認が得られます。
- 衛星には有効なエフェメリスが必要です。これにより、アンテナの視線が確保され、コンタクト中に衛星を正確に指すことができます。
- 有効なミッションプロファイルが必要です。これにより、衛星へのデータの受信方法や送信方法など、この問い合わせの動作をカスタマイズできます。同じ車両に複数のミッションプロファイルを使用して、発生するさまざまな運用体制やシナリオに合わせて異なるコンタクトを作成できます。

## 衛星オンボーディング

への衛星のオンボーディング AWS Ground Station は、データ収集、技術検証、スペクトルライセンス、統合とテストを含む複数ステップのプロセスです。ガイドの「[衛星オンボーディング](#)」セクションでは、このプロセスについて説明します。

## ミッションプロファイルの構成

衛星周波数情報、[データプレーン](#)情報、およびその他の詳細は、ミッションプロファイルにカプセル化されます。ミッションプロファイルは、設定コンポーネントのコレクションです。これにより、ユースケースに応じて、さまざまなミッションプロファイル間で設定コンポーネントを再利用できます。ミッションプロファイルは個々の衛星を直接参照するのではなく、その技術的機能に関する情報しか持たないため、ミッションプロファイルは同じ設定の複数の衛星でも再利用できます。

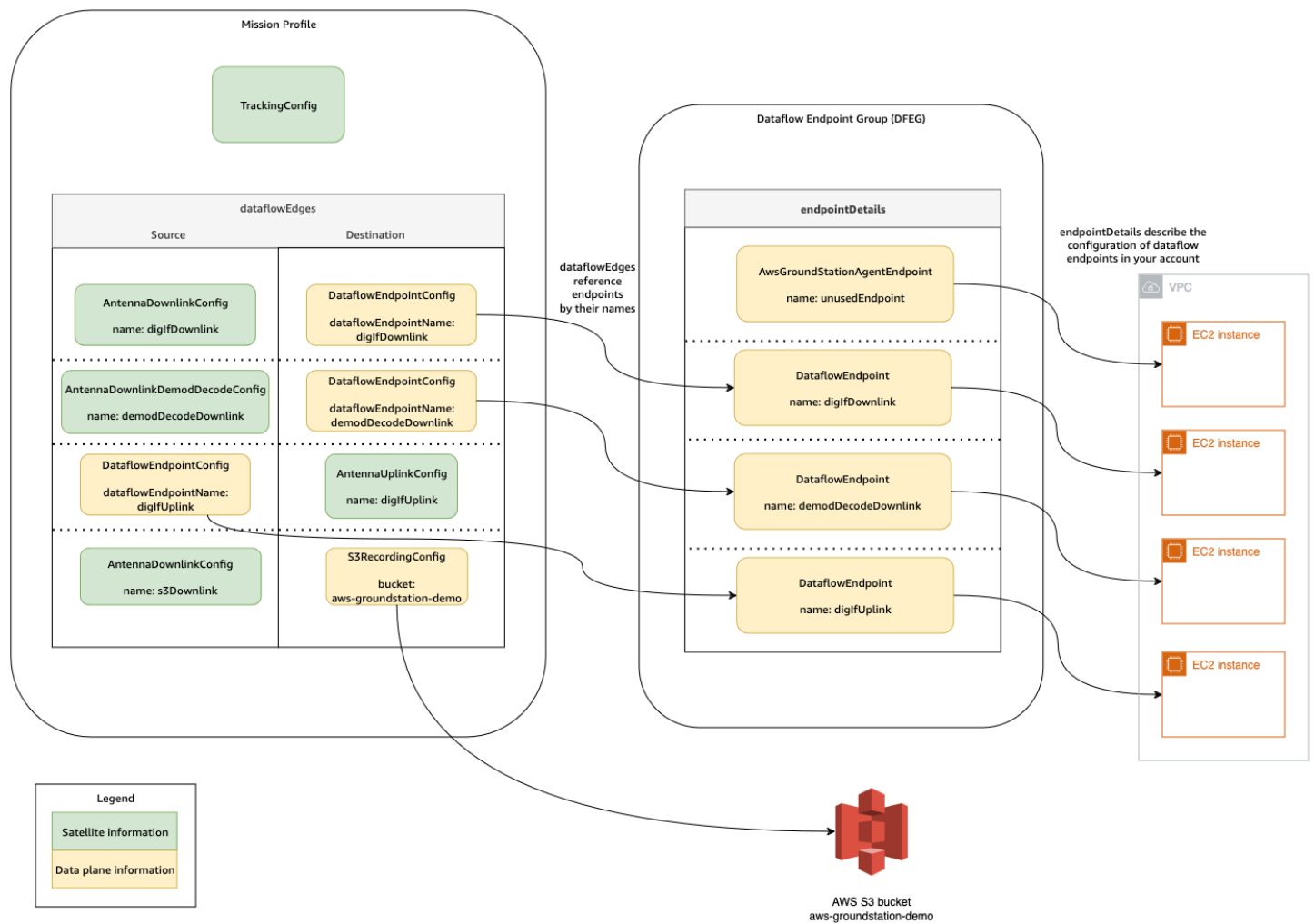
有効なミッションプロファイルには、追跡設定と 1 つ以上のデータフローがあります。追跡設定は、問い合わせ中に追跡する設定を指定します。データフロー内の各設定ペアは、送信元と送信先

を確立します。衛星とその運用モードに応じて、ミッションプロファイルのデータフローの正確な数は、アップリンクおよびダウンリンクの通信パスとデータ処理の側面を表すように異なります。

- 問い合わせ中に使用される Amazon VPC、Amazon S3、Amazon EC2 リソースの設定の詳細については、「」を参照してください[データフローの使用](#)。
- 各設定の動作の詳細については、「」を参照してください[AWS Ground Station 設定を使用する](#)。
- 予想されるすべてのパラメータの詳細については、「」を参照してください[AWS Ground Station ミッションプロファイルを使用する](#)。
- ユースケースをサポートするためにさまざまなミッションプロファイルを作成する方法の例については、「」を参照してください[ミッションプロファイル設定の例](#)。

次の図は、ミッションプロファイルの例と、必要な追加のリソースを示しています。この例では、柔軟性を示すために、このミッションプロファイルに必要なデータフローエンドポイントである `unusedEndpoint` を示しています。この例では、次のデータフローをサポートしています。

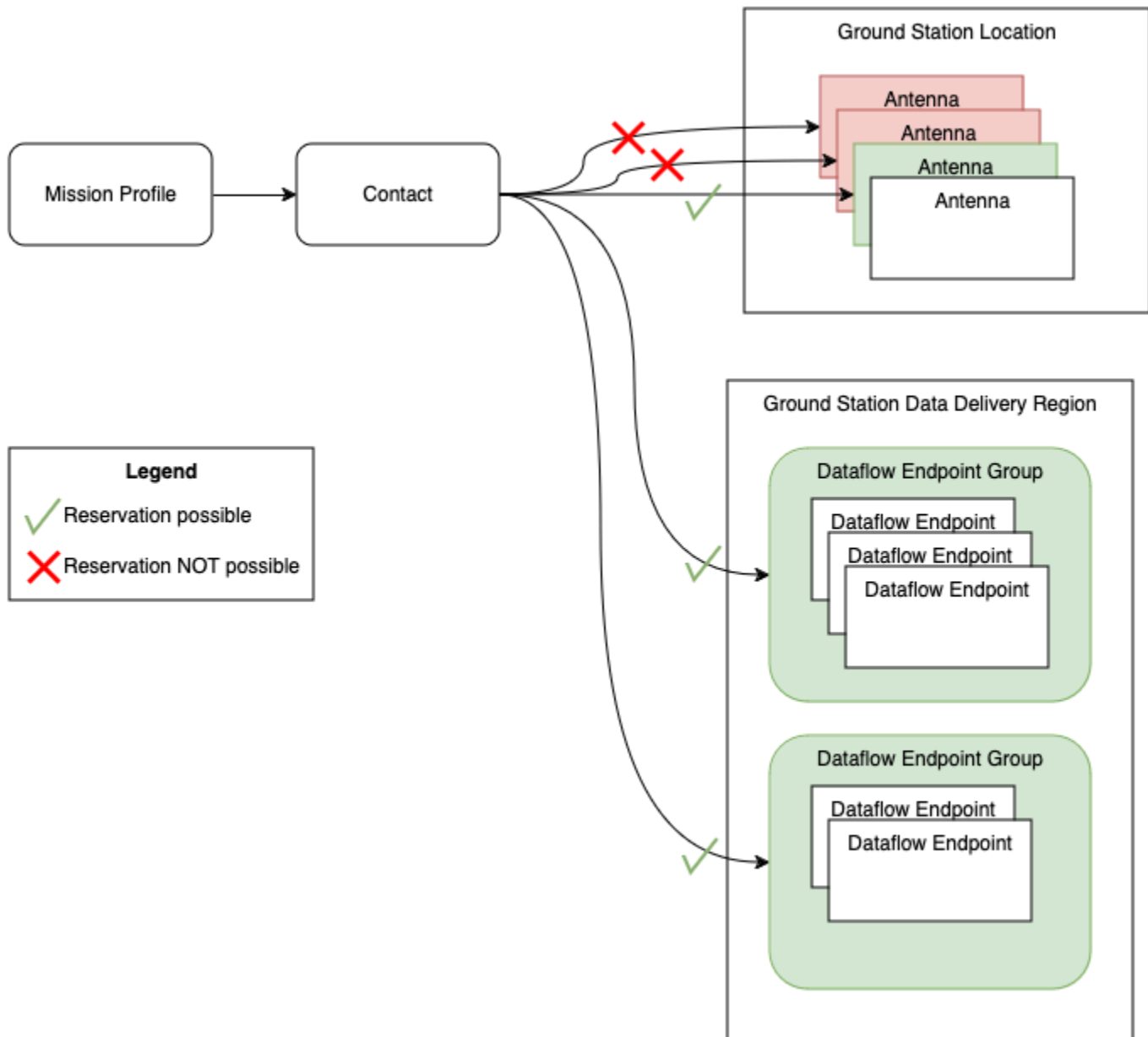
- 管理する Amazon EC2 インスタンスへのデジタル中間周波数データの同期ダウンリンク。 `digIfDownlink` という名前で表されます。
- Amazon S3 バケットへのデジタル中間周波数データの非同期ダウンリンク。バケット名 `aws-groundstation-demo` で表されます。
- 管理する Amazon EC2 インスタンスへの復調および復号化されたデータの同期ダウンリンク。 `demodDecodeDownlink` という名前で表されます。
- 管理する Amazon EC2 インスタンスから AWS Ground Station マネージドアンテナへのデータの同期アップリンク。 `digIfUplink` という名前で表されます。



## 問い合わせのスケジューリング

有効なミッションプロファイルを使用すると、オンボード衛星とのコンタクトをリクエストできます。問い合わせ予約リクエストは非同期であり、グローバルアンテナサービスが関連するすべてのAWSリージョンで一貫したスケジュールを達成するための時間を確保します。このプロセス中、リクエストされた地上局の場所にあるさまざまなアンテナが評価され、それらが使用可能で、問い合わせを処理できるかどうか判断されます。このプロセスでは、設定されたデータフローエンドポイントも評価され、可用性が判断されます。この評価の実行中、問い合わせステータスはスケジュール中になります。

この非同期スケジューリングプロセスはリクエストから5分以内に終了しますが、通常は1分以内に終了します。スケジューリング中にイベントベースのモニタリング [イベント AWS Ground Station による自動化](#)を確認してください。



実行でき、利用可能な問い合わせは、スケジュールされた問い合わせになります。スケジュールされたコンタクトでは、コンタクトの実行に必要なリソースは、ミッションプロファイルで定義されている必要な AWS リージョン全体で予約されています。実行できない問い合わせ、または使用できない部分がある場合、`FAILED_TO_SCHEDULE` 問い合わせが発生します。デバッグの詳細については [FAILED\\_TO\\_SCHEDULE コンタクトのトラブルシューティング](#)、「」を参照してください。

## 問い合わせの実行

AWS Ground Station は、問い合わせ予約中に AWS マネージドリソースを自動的にオーケストレーションします。必要に応じて、ミッションプロファイルでデータフローエンドポイントとして定

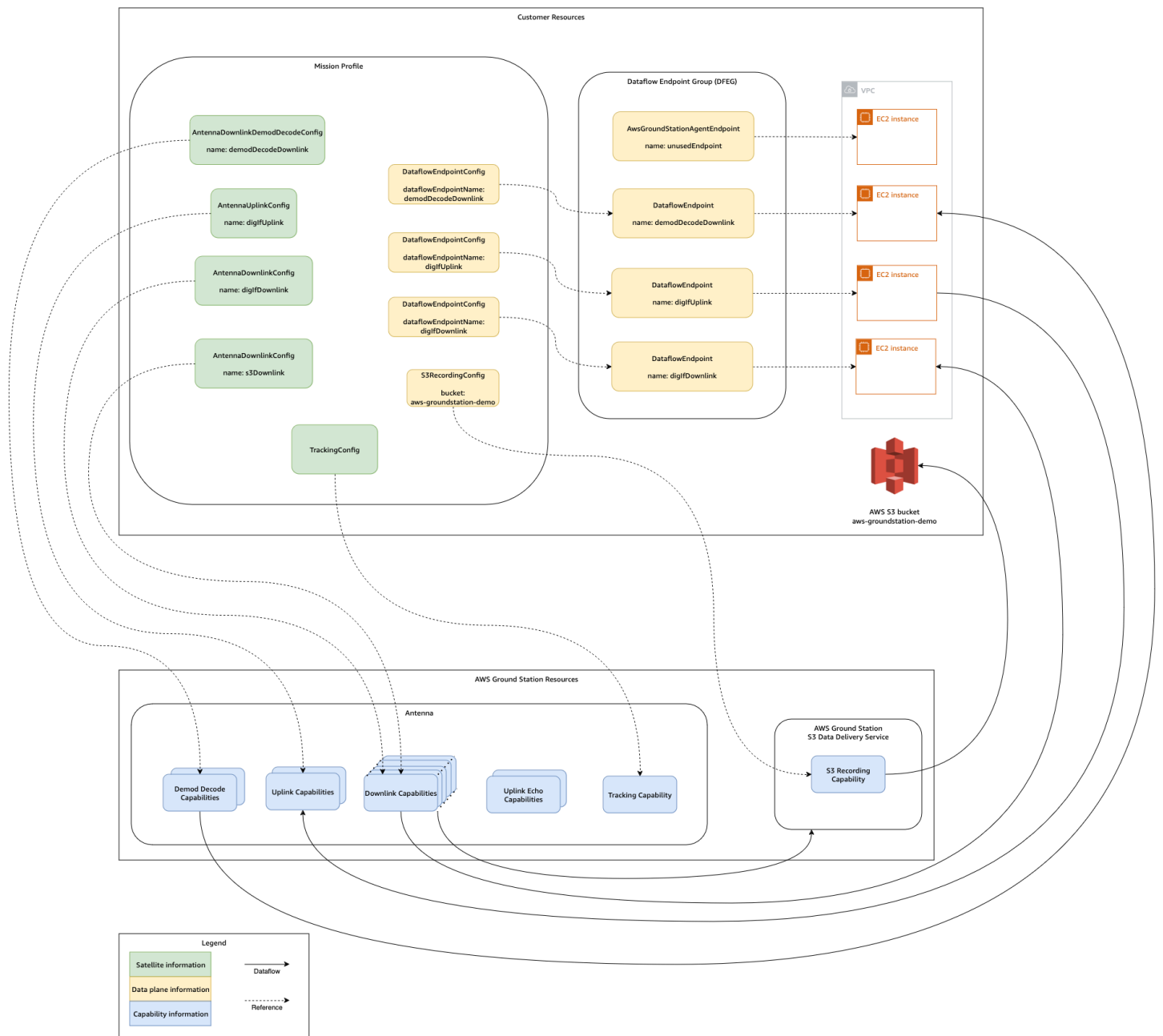
義された EC2 リソースをオーケストレーションする責任があります。は、コストを削減するためにリソースのオーケストレーションを自動化するための [AWS EventBridge イベント](#) AWS Ground Station を提供します。詳細については、「[イベント AWS Ground Station による自動化](#)」を参照してください。

問い合わせ中、問い合わせパフォーマンスに関するテレメトリが AWS CloudWatch に配信されます。実行中に問い合わせをモニタリングする方法については、「」を参照してください。[によるモニタリングを理解する AWS Ground Station。](#)

次の図は、問い合わせ中にオーケストレーションされた同じリソースを表示することで、前の例を続行します。

#### Note

この例では、すべてのアンテナ機能が使用されたわけではありません。例えば、複数の周波数と偏波をサポートする多数のアンテナダウンリンク機能が各アンテナで利用できます。AWS Ground Station アンテナで使用できる各機能タイプの数、およびサポートされている周波数と偏波の詳細については、「」を参照してください。[AWS Ground Station サイト機能](#)。



問い合わせの終了時に、AWS Ground Station は問い合わせのパフォーマンスを評価し、最終的な問い合わせステータスを決定します。エラーが検出されない問い合わせは、完了済みの問い合わせステータスになります。サービスエラーが原因で問い合わせ中にデータ配信の問題が発生した問い合わせは、AWS\_FAILED ステータスになります。クライアントまたはユーザーのエラーによって問い合わせ中にデータ配信の問題が発生した問い合わせは、失敗ステータスになります。コンタクト時間外のエラー、つまりパス前またはパス後のエラーは、判定中に考慮されません。

詳細については「[問い合わせのライフサイクルを理解する](#)」を参照してください。

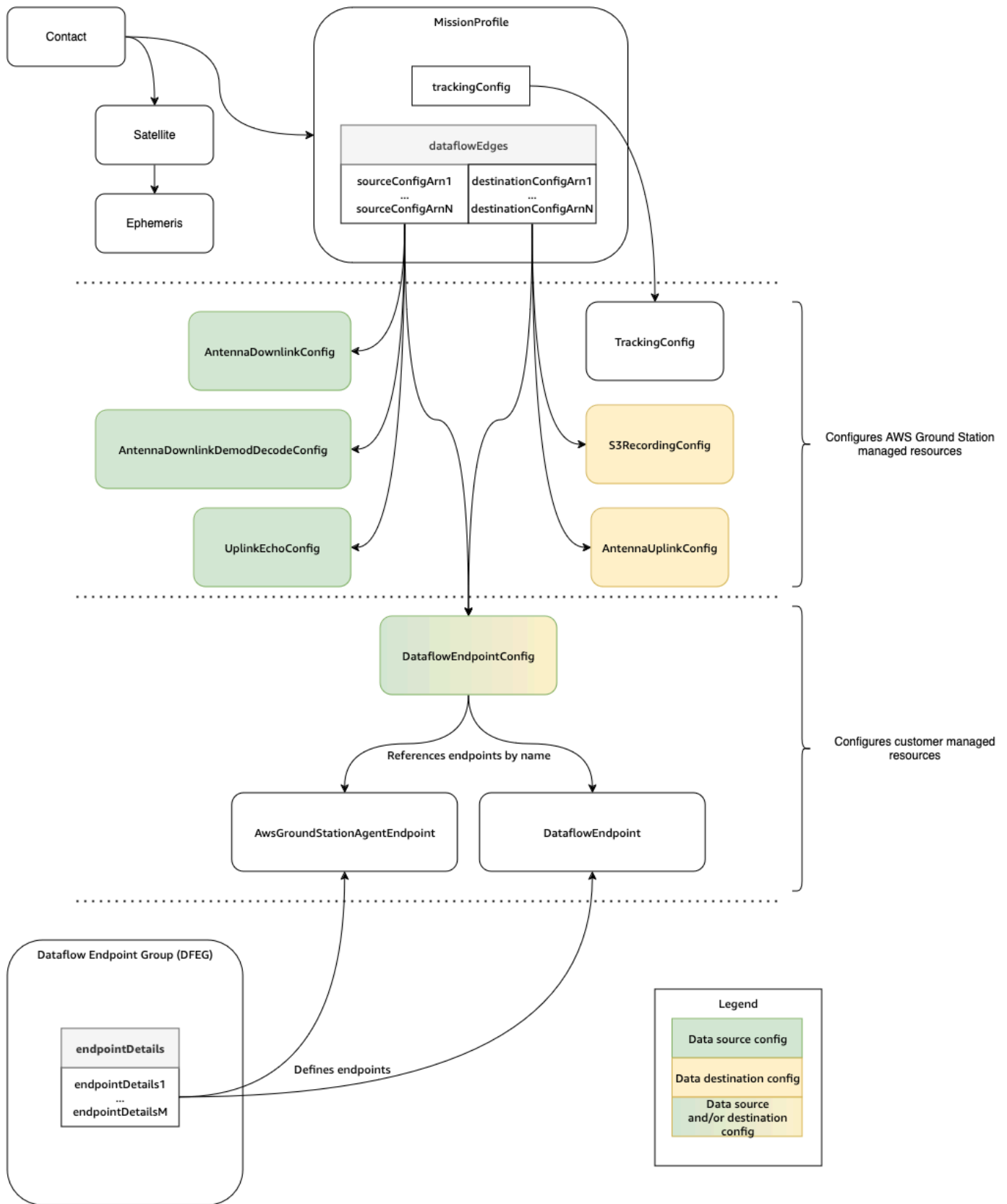
# デジタルツイン

のデジタルツイン機能 AWS Ground Station を使用すると、仮想地上局のロケーションに対してコンタクトをスケジュールできます。これらの仮想地上ステーションは、アンテナ機能、サイトマスク、実際の GPS 座標など、本番稼働用地上ステーションの正確なレプリカです。デジタルツイン機能を使用すると、コンタクトオーケストレーションワークフローを本番環境の地上局と比較してわずかなコストでテストできます。詳細については「[AWS Ground Station デジタルツイン機能を使用する](#)」を参照してください。

## AWS Ground Station コアコンポーネントを理解する

このセクションでは、AWS Ground Station のコアコンポーネントの詳細定義について説明します。

次の図は、 のコアコンポーネント AWS Ground Station とその相互関係を示しています。矢印は、各コンポーネントが依存関係を指すコンポーネント間の依存関係の方向を示します。



以下のトピックでは、AWS Ground Station コアコンポーネントについて詳しく説明します。

## トピック

- [AWS Ground Station ミッションプロファイルを使用する](#)
- [AWS Ground Station 設定を使用する](#)
- [AWS Ground Station Dataflow エンドポイントグループを使用する](#)
- [AWS Ground Station エージェントを使用する](#)

## AWS Ground Station ミッションプロファイルを使用する

ミッションプロファイルには、コンタクトの実行方法に関する設定とパラメータが含まれています。コンタクトを予約したり利用可能なコンタクトを検索したりするときには、使用する予定のミッションプロファイルを指定します。ミッションプロファイルはすべての設定を1つにまとめ、コンタクト中のアンテナの設定方法とデータの配信先を定義します。

ミッションプロファイルは、同じ無線特性を共有する衛星間で共有できます。追加のデータフローエンドポイントグループを作成して、コンステレーションに対して実行する最大同時コンタクト数をバインドできます。

追跡設定は、ミッションプロファイル内の一意のフィールドとして指定されます。追跡設定は、問い合わせ中にプログラム追跡と自動追跡を使用するための設定を指定するために使用されます。詳細については、「[追跡設定](#)」を参照してください。

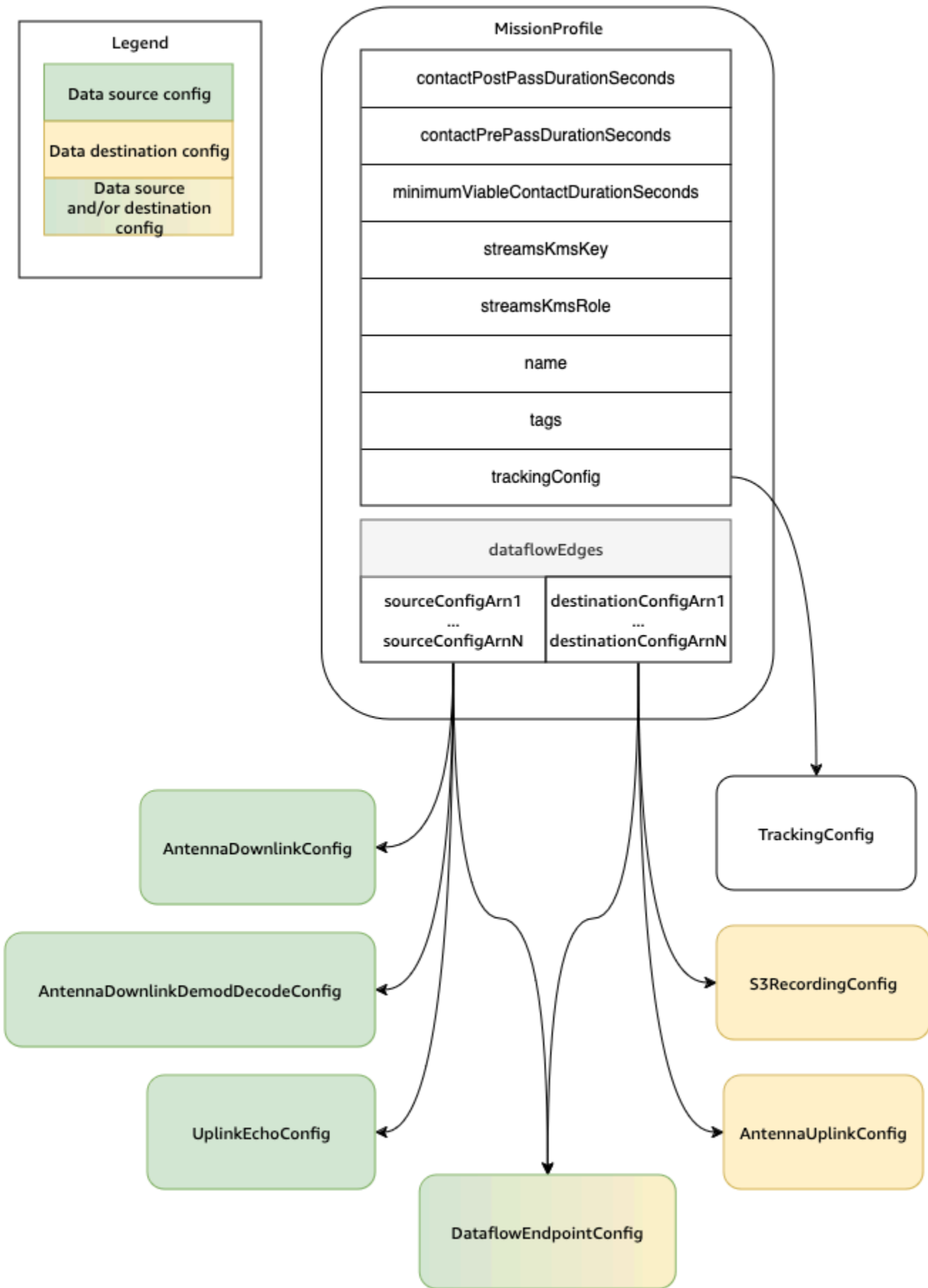
他のすべての設定は、ミッションプロファイルの `dataflowEdges` フィールドに含まれます。これらの設定は、データおよび関連する設定を送受信できる AWS Ground Station マネージドリソースをそれぞれ表すデータフローノードと考えることができます。 `dataflowEdges` フィールドは、必要な送信元と送信先のデータフローノード (設定) を定義します。1つのデータフローエッジは2つの設定 [Amazon リソースネーム \(ARNs\) のリスト](#) です。1つ目はソース設定、2つ目は宛先設定です。2つの設定間でデータフローエッジを指定することで、問い合わせ中にデータがどこ AWS Ground Station からどこへ流れるかをに指示します。詳細については、「[AWS Ground Station 設定を使用する](#)」を参照してください。

`contactPrePassDurationSeconds` と `contactPostPassDurationSeconds` を使用すると、CloudWatch イベント通知を受け取る問い合わせに関連する時間を指定できます。連絡先に関連するイベントのタイムラインについては、「[問い合わせのライフサイクルを理解する](#)」を参照してください。

ミッションプロファイルの name フィールドを確認すれば、作成したミッションプロファイルと区別できます。

streamsKmsRole および streamsKmsKey は、AWS Ground Station エージェントによるデータ配信 AWS Ground Station に が使用する暗号化を定義するために使用されます。[の転送中のデータ暗号化 AWS Ground Station](#) を参照してください。

telemetrySinkConfigArn フィールドはオプションで、コンタクト中に AWS Ground Station テレメトリを有効にできます。指定した場合、 はコンタクトの実行中にほぼリアルタイムのテレメトリデータをアカウントに AWS Ground Station ストリーミングします。テレメトリの設定と使用の詳細については、「」を参照してください[テレメトリの操作](#)。



パラメータと例の完全なリストは、次のドキュメントに含まれています。

- [AWS::GroundStation::MissionProfile CloudFormation リソースタイプ](#)

## AWS Ground Station 設定を使用する

Configs は、AWS Ground Station が問い合わせの各側面のパラメータを定義するために使用するリソースです。希望する設定をミッションプロファイルに追加すると、コンタクトを実行する際にそのミッションプロファイルが使用されます。さまざまなタイプの設定を定義できます。設定は 3 つのカテゴリにグループ化できます。

- 追跡設定
- データフロー設定
- テレメトリ設定

TrackingConfig は、追跡設定の唯一のタイプです。これは、コンタクト中にアンテナの自動トラック設定を構成するために使用され、ミッションプロファイルで必要です。

ミッションプロファイルのデータフローで使用できる設定は、それぞれがデータを送受信できる AWS Ground Station マネージドリソースを表すデータフローノードと考えることができます。ミッションプロファイルには、これらの設定の少なくとも 1 つのペアが必要です。1 つはデータソースを表し、もう 1 つは送信先を表します。これらの設定の概要を次の表に示します。

設定名	データソース/送信先
AntennaDownlinkConfig	ソース
AntennaDownlinkDemodDecodeConfig	ソース
UplinkEchoConfig	ソース
S3RecordingConfig	目的地
AntennaUplinkConfig	目的地
DataflowEndpointConfig	送信元および/または送信先

TelemetrySinkConfig はテレメトリ設定の唯一のタイプです。これは、コンタクト中にテレメトリデータが配信される場所を設定するために使用され、ミッションプロファイルではオプションです。含まれている場合、はコンタクトの実行中にほぼリアルタイムのテレメトリをアカウントに AWS Ground Station ストリーミングします。

CloudFormation、または AWS Ground Station API を使用して設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。特定の設定タイプのドキュメントへのリンクも以下に記載されています。

- [AWS::GroundStation::Config CloudFormation リソースタイプ](#)
- [Config AWS CLI リファレンス](#)
- [CreateConfig API リファレンス](#)

## 追跡設定

ミッションプロファイルの追跡設定を使用して、コンタクト中に自動追跡を有効にする必要があるかどうかを決定できます。この設定には単一のパラメータ、autotrack があります。この autotrack パラメータには以下の値があります。

- REQUIRED -コンタクトに自動追跡が必要。
- PREFERRED -コンタクトに自動追跡が好ましいが、自動追跡がなくてもコンタクトを実行できる。
- REMOVED - コンタクトに自動追跡が使用されるべきではない。

AWS Ground Station は、自動トラックが使用されていない場合にエフェメリスに基づいてポイントするプログラムによる追跡を利用します。エフェメリスの構築方法の詳細については[ガエフェメリス AWS Ground Station を使用する方法を理解する](#)、「」を参照してください。

Autotrack は、予想されるシグナルが見つかるまでプログラム追跡を使用します。これが発生すると、シグナルの強度に基づいて引き続き追跡されます。

CloudFormation、または AWS Ground Station API を使用して 設定の追跡オペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config TrackingConfig CloudFormation プロパティ](#)
- [Config AWS CLI リファレンス](#) (trackingConfig -> (structure)「」セクションを参照)
- [TrackingConfig API リファレンス](#)

## アンテナダウンリンク設定

アンテナダウンリンク設定を使用してコンタクト中のダウンリンク用のアンテナを設定できます。これらは、ダウンリンクコンタクト中に使用すべき周波数、帯域幅、および偏波を指定するスペクトル設定で構成されています。

この設定は、データフロー内のソースノードを表します。無線周波数データのデジタル化を担当します。このノードからストリーミングされたデータは、シグナルデータ/IP 形式に従います。この設定でデータフローを構築する方法の詳細については、「」を参照してください。 [データフローの使用](#)

ダウンリンクのユースケースで復調や復号が必要な場合は、「[アンテナダウンリンク復調デコード設定](#)」を参照してください。

CloudFormation、または AWS Ground Station API を使用してアンテナダウンリンク設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config AntennaDownlinkConfig CloudFormation プロパティ](#)
- [Config AWS CLI リファレンス](#) (antennaDownlinkConfig -> (structure)「」セクションを参照)
- [AntennaDownlinkConfig API リファレンス](#)

## アンテナダウンリンク復調デコード設定

アンテナダウンリンクデモデコード設定は、復調やデコードでダウンリンクコンタクトを実行するために使用できる、より複雑でカスタマイズ可能な設定タイプです。これらのタイプの問い合わせの実行に関心がある場合は、を通じて AWS サポート チケットを開いてください [AWS Support Center Console](#)。ユースケースに適した設定とミッションプロファイルを定義するお手伝いをします。

この設定は、データフロー内のソースノードを表します。無線周波数データをデジタル化し、指定されたとおりに復調とデコードを実行します。このノードからストリーミングされたデータは、復調/復号化されたデータ/IP 形式に従います。この設定でデータフローを構築する方法の詳細については、「」を参照してください。 [データフローの使用](#)

CloudFormation、または AWS Ground Station API を使用してアンテナダウンリンクデモデコード設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config AntennaDownlinkDemodDecodeConfig CloudFormation プロパティ](#)

- [Config AWS CLI リファレンス](#) (antennaDownlinkDemodDecodeConfig -> (structure) 「」セクションを参照)
- [AntennaDownlinkDemodDecodeConfig API リファレンス](#)

## アンテナアップリンク設定

アンテナアップリンク設定を使用してコンタクト中のアップリンクのアンテナを設定できます。これらは、周波数、偏波、および目標実効等方輻射電力 (EIRP) を含むスペクトル設定で構成されています。アップリンクループバックのコンタクトを設定する方法については、「[アンテナアップリンクエコー設定](#)」を参照してください。

この設定は、データフローの送信先ノードを表します。提供されたデジタル無線周波数データ信号をアナログ信号に変換し、衛星が受信できるように出力します。このノードにストリーミングされるデータは、シグナルデータ/IP 形式を満たすことが期待されます。この設定でデータフローを構築する方法の詳細については、「」を参照してください。 [データフローの使用](#)

CloudFormation、または AWS Ground Station API を使用してアンテナアップリンク設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config AntennaUplinkConfig CloudFormation プロパティ](#)
- [Config AWS CLI リファレンス](#) (antennaUplinkConfig -> (structure) 「」セクションを参照)
- [AntAntennaUplinkConfig API リファレンス](#)

## アンテナアップリンクエコー設定

アップリンクエコー設定は、アップリンクエコーを実行する方法をアンテナに伝えます。アップリンクエコーを使用して、宇宙機に送信されたコマンドを検証し、その他の高度なタスクを実行できます。これは、AWS Ground Station アンテナ (アップリンク) によって送信された実際の信号を記録することで実現されます。これにより、アンテナによって送信された信号がデータフローエンドポイントにエコーされ、送信された信号と一致する必要があります。アップリンクエコー設定には、アップリンク設定の ARN が含まれています。アンテナは、アップリンクエコーを実行する際に ARN により指定されたアップリンク設定からのパラメータを使用します。

この設定は、データフロー内のソースノードを表します。このノードからストリーミングされたデータは、シグナルデータ/IP 形式を満たします。この設定でデータフローを構築する方法の詳細については、「」を参照してください。 [データフローの使用](#)

CloudFormation、または AWS Ground Station API を使用してアップリンクエコー設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config UplinkEchoConfig CloudFormation プロパティ](#)
- [Config AWS CLI リファレンス](#) (uplinkEchoConfig -> (structure)「」セクションを参照)
- [UplinkEchoConfig API リファレンス](#)

## データフローエンドポイント設定

### Note

データフローエンドポイント設定は、Amazon EC2 へのデータ配信にのみ使用され、Amazon S3 へのデータ配信には使用されません。

データフローエンドポイント設定を使用して、コンタクト中にデータのフローを行う[データフローエンドポイントグループ](#)内のデータフローエンドポイントを指定します。データフローエンドポイント設定の2つのパラメータは、データフローエンドポイントの名前とリージョンを指定します。コンタクトを予約すると、AWS Ground Station は指定した[ミッションプロファイル](#)を分析し、ミッションプロファイルに含まれるデータフローエンドポイント設定で指定されたすべてのデータフローエンドポイントを含む AWS リージョン内のデータフローエンドポイントグループを検索しようとしてします。適切なデータフローエンドポイントグループが見つかった場合、問い合わせステータスは SCHEDULED になり、それ以外の場合は FAILED\_TO\_SCHEDULE になります。問い合わせのステータスの詳細については、「」を参照してください[AWS Ground Station 問い合わせステータス](#)。

データフローエンドポイント設定の dataflowEndpointName プロパティは、コンタクト中にデータのフローを行うデータフローエンドポイントグループ内のデータフローエンドポイントを指定します。

dataflowEndpointRegion プロパティは、データフローエンドポイントが存在するリージョンを指定します。データフローエンドポイント設定でリージョンが指定されている場合、は指定されたリージョン内のデータフローエンドポイント AWS Ground Station を検索します。リージョンが指定されていない場合、AWS Ground Station はデフォルトで問い合わせの地上局リージョンになります。データフローエンドポイントのリージョンがコンタクトの地上ステーションリージョンと同じでない場合、コンタクトはクロスリージョンデータ配信コンタクトとみなされます。クロスリージョンデータフローの詳細については、[データフローの使用](#)「」を参照してください。

データフローのさまざまな命名スキームがユースケースにどのように役立つかに関するヒント [AWS Ground Station Dataflow エンドポイントグループを使用する](#) については、「」を参照してください。

この設定でデータフローを構築する方法の詳細については、「」を参照してください。 [データフローの使用](#)

CloudFormation、または AWS Ground Station API を使用してデータフローエンドポイント設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config DataflowEndpointConfig CloudFormation プロパティ](#)
- [Config AWS CLI リファレンス](#) (dataflowEndpointConfig -> (structure)「」セクションを参照)
- [DataflowEndpointConfig API リファレンス](#)

## Amazon S3 録画設定

### Note

Amazon S3 記録設定は Amazon S3、Amazon EC2 へのデータ配信には使用されません。

この設定は、データフローの送信先ノードを表します。このノードは、データフローのソースノードからの受信データを pcap データにカプセル化します。この設定でデータフローを構築する方法の詳細については、「」を参照してください。 [データフローの使用](#)

S3 記録設定を使用して、ダウンリンクされたデータを配信する Amazon S3 バケットと、使用する命名規則を指定できます。以下に、これらのパラメータに関する制限と詳細を指定します。

- Amazon S3 バケットの名前は、aws-groundstation で始まる必要があります。
- IAM ロールには、groundstation.amazonaws.com サービスプリンシパルがロールを引き受けることを許可する信頼ポリシーが必要です。例については、以下の「[信頼ポリシーの例](#)」を参照してください。構成の作成時に、構成リソース ID は存在しません。信頼ポリシーでは、*your-config-id* の代わりにアスタリスク (\*) を使用して、設定リソース ID を作成した後に更新することができ

## 信頼ポリシーの例

ロールの信頼ポリシーを更新する方法の詳細については、IAM ユーザーガイドの「[IAM ロールの管理](#)」を参照してください。

### 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 ロールには、バケット上で `s3:GetBucketLocation` アクションとバケットのオブジェクト上での `s3:PutObject` の実行を許可する IAM ポリシーが必要です。Amazon S3 バケットにバケットポリシーがある場合、バケットポリシーは IAM ロールでこれらのアクションの実行を許可する必要があります。例については、以下の「[ロールポリシーの例](#)」を参照してください。

## ロールポリシーの例

ロールポリシーを更新またはアタッチする方法の詳細については、IAM ユーザーガイドの「[IAM ポリシーを管理する](#)」を参照してください。

## 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/*"
      ]
    }
  ]
}
```

- プレフィックスは、S3 データオブジェクトに名前を付けるときに使用されます。代替するオプションのキーを指定できます。これらの値は、連絡先情報の対応する情報に置き換えられます。たとえば、のプレフィックス{satellite\_id}/{year}/{month}/{day}は置き換えられ、次のような出力になります。 fake\_satellite\_id/2021/01/10

置換のオプションキー: {satellite\_id} | {config-name} {config-id} | {year} | {month} | {day} |

CloudFormation、または AWS Ground Station API を使用して S3 記録設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config S3RecordingConfig CloudFormation property](#)
- [Config AWS CLI リファレンス](#) (s3RecordingConfig -> (structure) 「」セクションを参照)
- [S3RecordingConfig API リファレンス](#)

## テレメトリシンク設定

テレメトリシンク設定を使用して、衛星コンタクト中にテレメトリデータを配信する場所を指定できます。テレメトリシンクの設定はオプションであり、テレメトリ対応コンタクトをスケジュールするためにミッションプロファイルに追加されます。以下に、これらのパラメータに関する制限と詳細を指定します。

- IAM ロールには、groundstation.amazonaws.com サービスプリンシパルがロールを引き受けることを許可する信頼ポリシーが必要です。例については、以下の「[信頼ポリシーの例](#)」を参照してください。

### 信頼ポリシーの例

ロールの信頼ポリシーを更新する方法の詳細については、IAM ユーザーガイドの「[IAM ロールの管理](#)」を参照してください。

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

- IAM ロールには、ロールがストリームで kinesis:DescribeStream、kinesis:PutRecord および kinesis:PutRecords アクションを実行できるようにする IAM ポリシーが必要です。例については、以下の「[ロールポリシーの例](#)」を参照してください。

## ロールポリシーの例

ロールポリシーを更新またはアタッチする方法の詳細については、IAM ユーザーガイドの「[IAM ポリシーを管理する](#)」を参照してください。

```
{
  "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"
    }
  ]
}
```

ミッションプロファイルにテレメトリシンク設定を含めると、AWS Ground Station はコンタクト中にテレメトリデータをアカウントにストリーミングします。テレメトリタイプ、データ形式、および必要な AWS リソースの設定の詳細については、「」を参照してください[テレメトリの操作](#)。

CloudFormation、または AWS Ground Station API を使用してテレメトリシンク設定でオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::Config TelemetrySinkConfig CloudFormation プロパティ](#)
- [Config AWS CLI リファレンス](#) (telemetrySinkConfig -> (structure)「」セクションを参照)
- [TelemetrySinkConfig API リファレンス](#)

## AWS Ground Station Dataflow エンドポイントグループを使用する

データフローエンドポイントは、コンタクト中にデータを同期的にストリーミングする場所を定義します。データフローエンドポイントは、常にデータフローエンドポイントグループの一部として作成されます。1つのグループに複数のデータフローエンドポイントを含めることで、1回のコンタクトで指定されたエンドポイントをすべて一緒に使用できることを断定できます。たとえば、コンタクトが3つの別々のデータフローエンドポイントにデータを送信する必要がある場合、1つのデータフ

ローエンドポイントグループに、ミッションプロファイルのデータフローエンドポイント設定と一致するエンドポイントが 3 つ必要です。

## Dataflow エンドポイントグループバージョン

AWS Ground Station は、2 つのバージョンのデータフローエンドポイントグループをサポートしています。

- DataflowEndpointGroup - [データフローエンドポイント](#)を使用したアップリンクとダウンリンク、および[AWS Ground Station エージェントエンドポイント](#)のダウンリンクのみをサポートする元の実装
- DataflowEndpointGroupV2 - AWS Ground Station エージェントエンドポイントのアップリンクデータフローとダウンリンクデータフローの両方をサポートするバージョンを更新し、明確性と機能が向上しました。

### データフローエンドポイントグループの比較

機能	DataflowEndpointGroup	DataflowEndpointGroupV2
サポートされているエンドポイントタイプ	DataflowEndpoint、AwsGroundStationAgentEndpoint	DownlinkAwsGroundStationAgentEndpoint、UplinkAwsGroundStationAgentEndpoint
アップリンクをサポートするエンドポイント	DataflowEndpoint	UplinkAwsGroundStationAgentEndpoint
ダウンリンクをサポートするエンドポイント	DataflowEndpoint、AwsGroundStationAgentEndpoint	DownlinkAwsGroundStationAgentEndpoint

DataflowEndpointGroupV2 は、アップリンクデータフローをサポートし、データフローエンドポイントグループを囲む言語をより明確にするために作成されました。すべての新しいユースケースでは、DataflowEndpointGroupV2 で [UplinkAwsGroundStationAgentEndpoint](#) エンドポイントと [DownlinkAwsGroundStationAgentEndpoint](#) エンドポイントを使用することをお勧めします。[DataflowEndpointGroupV2](#) DataflowEndpointGroup は下位互換性のために引き続きサポートされていますが、DataflowEndpointGroupV2 は強化された機能とより明確な設定オプションを提供します。

**i** Tip

データフローエンドポイントは、問い合わせの実行時に選択した名前で識別されます。これらの名前は、アカウント全体で一意である必要はありません。これにより、同じミッションプロファイルを使用して、異なる衛星とアンテナ間で複数のコンタクトを同時に実行できます。これは、動作特性が同じ衛星の群れがある場合に便利です。データフローエンドポイントグループの数は、衛星の星座が必要とする同時コンタクトの最大数に合わせてスケールアップできます。

データフローエンドポイントグループ内の 1 つ以上のリソースがコンタクトに使用されている場合、グループ全体がそのコンタクトの間リザーブされます。複数のコンタクトを同時に実行できますが、それらのコンタクトは異なるデータフローエンドポイントグループで実行する必要があります。

**A** Important

データフローエンドポイントグループは、それらを使用するコンタクトをスケジュールするために HEALTHY 状態になっている必要があります。HEALTHY 状態ではないデータフローエンドポイントグループのトラブルシューティング方法については、「」を参照してください [DataflowEndpointGroups が正常でない状態のトラブルシューティング](#)。

CloudFormation、または AWS Ground Station API を使用してデータフローエンドポイントグループでオペレーションを実行する方法の詳細については AWS Command Line Interface、次のドキュメントを参照してください。

- [AWS::GroundStation::DataflowEndpointGroup CloudFormation リソースタイプ](#)
- [データフローエンドポイントグループの AWS CLI リファレンス](#)
- [cCreateDataflowEndpointGroup API リファレンス](#)

## データフローエンドポイント

データフローエンドポイントグループのメンバーは、データフローエンドポイントです。サポートされているエンドポイントタイプは、使用するデータフローエンドポイントグループのバージョンによって異なります。

## DataflowEndpointGroup エンドポイント

DataflowEndpointGroup は、[データフローエンドポイント](#)を使用したアップリンクとダウンリンク、および[AWS Ground Station エージェントエンドポイント](#)のダウンリンクのみをサポートします。どちらのタイプのエンドポイントでも、データフローエンドポイントグループを作成する前に、サポートコンストラクト (IP アドレスなど) を作成します。使用するデータフローエンドポイントタイプと、サポートコンストラクトを設定する方法については、[データフローの使用](#)「」を参照してください。

以下のセクションでは、サポートされている両方のエンドポイントタイプについて説明します。

### Important

単一のデータフローエンドポイントグループ内のすべてのデータフローエンドポイントは、同じタイプである必要があります。[AWS Ground Station エージェントエンドポイント](#)を同じグループの [Dataflow エンドポイント](#)と混在させることはできません。ユースケースで両方のタイプのエンドポイントが必要な場合は、タイプごとに個別のデータフローエンドポイントグループを作成する必要があります。

DataflowEndpointGroupV2 の場合、[UplinkAwsGroundStationAgentEndpoint](#) と [DownlinkAwsGroundStationAgentEndpoint](#) を同じグループに混在させることができます。

## AWS Ground Station エージェントエンドポイント

AWS Ground Station エージェントエンドポイントは、AWS Ground Station エージェントをソフトウェアコンポーネントとして使用して接続を終了します。AWS Ground Station エージェントエンドポイントを作成するには、EndpointDetails の AwsGroundStationAgentEndpoint フィールドのみを入力します。AWS Ground Station エージェントの詳細については、[AWS Ground Station 「エージェントユーザーガイド」](#)を参照してください。

AwsGroundStationAgentEndpoint には以下の構成要素があります。

- Name - データフローエンドポイント名。問い合わせがこのデータフローエンドポイントを使用するには、この名前がデータフローエンドポイント設定で使用されている名前と一致する必要があります。
- EgressAddress - エージェントからのデータの出力に使用される IP アドレスとポートアドレス。
- IngressAddress - エージェントへのデータの進入に使用される IP とポートアドレス。

## データフローエンドポイント

Dataflow Endpoint は、ネットワークアプリケーションをソフトウェアコンポーネントとして使用して接続を終了します。デジタル信号データのアップリンク、50MHz 未満のデジタル信号データのダウンリンク、または復調/復号された信号データのダウンリンクを行う場合は、データフローエンドポイントを使用します。Dataflow Endpoint を作成するには、EndpointDetails の Endpoint および Security Details フィールドに入力します。

Endpoint には以下の構成要素があります。

- Name - データフローエンドポイント名。問い合わせがこのデータフローエンドポイントを使用するには、この名前がデータフローエンドポイント設定で使用されている名前と一致する必要があります。
- Address - 使用される IP アドレスとポートアドレス。

SecurityDetails には以下の構成要素があります。

- roleArn - VPC に Elastic Network Interface (ENIs) を作成するために AWS Ground Station が引き受けるロールの Amazon リソースネーム (ARN)。これらの ENI は、コンタクト中にストリーミングされるデータの出入力ポイントとなります。
- securityGroupIds - Elastic Network Interface にアタッチするセキュリティグループ。
- subnetIds - が インスタンスにストリームを送信するために Elastic Network Interface を配置 AWS Ground Station するサブネットのリスト。複数のサブネットを指定する場合は、相互にルーティング可能である必要があります。サブネットが異なるアベイラビリティーゾーン (AZs) にある場合、AZ 間のデータ転送料金が発生する可能性があります。

roleArn に渡される IAM ロールには、groundstation.amazonaws.com サービスプリンシパルがロールを引き受けることを許可する信頼ポリシーが必要です。例については、以下の「[信頼ポリシーの例](#)」を参照してください。エンドポイントの作成時にエンドポイントリソース ID は存在しないため、信頼ポリシーでは、*your-endpoint-id* の代わりにアスタリスク (\*) を使用する必要があります。作成後にエンドポイントリソース ID を使用してこれを更新し、信頼ポリシーをその特定のデータフローエンドポイントグループに絞り込むことができます。

IAM ロールには、 が ENIs AWS Ground Station をセットアップできるようにする IAM ポリシーが必要です。例については、以下の「[ロールポリシーの例](#)」を参照してください。

## 信頼ポリシーの例

ロールの信頼ポリシーを更新する方法の詳細については、IAM ユーザーガイドの「[IAM ロールの管理](#)」を参照してください。

### 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"
        }
      }
    }
  ]
}
```

## ロールポリシーの例

ロールポリシーを更新またはアタッチする方法の詳細については、IAM ユーザーガイドの「[IAM ポリシーを管理する](#)」を参照してください。

### 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 エンドポイント

DataflowEndpointGroupV2 には、より明確な設定と機能拡張を提供する特殊なエンドポイントタイプが導入されています。

- [UplinkAwsGroundStationAgentEndpoint](#) - アップリンクデータフロー用に最適化
- [DownlinkAwsGroundStationAgentEndpoint](#) - ダウンリンクデータフロー用に最適化

これらの特殊なエンドポイントは、汎用 [AwsGroundStationAgentEndpoint](#) を、データフローの設定と管理を容易にする方向固有の設定に置き換えます。

## アップリンク AWS Ground Station エージェントエンドポイント

[UplinkAwsGroundStationAgentEndpoint](#) はアップリンクデータフロー専用に設計されており、より明確な設定オプションを提供します。衛星にアップリンク AWS Ground Station するためにデータを提供する必要がある場合は、このエンドポイントタイプを使用します。

UplinkAwsGroundStationAgentEndpoint には以下の構成要素があります。

- Name - データフローエンドポイント名。問い合わせがこのデータフローエンドポイントを使用するには、この名前がデータフローエンドポイント設定で使用されている名前と一致する必要があります。

- `IngressAddressAndPort` - エージェントへのデータ入力用の単一の IP アドレスとポートアドレス
- `AgentIpAndPortAddress` - エージェント通信のポート範囲

ダウンリンク AWS Ground Station エージェントエンドポイント

[DownlinkAwsGroundStationAgentEndpoint](#) は、ナローバンドダウンリンク、ワイドバンド復調/デコード、アップリンクエコシナリオなどのダウンリンクデータフロー用に最適化されています。

`DownlinkAwsGroundStationAgentEndpoint` には以下の構成要素があります。

- `Name` - データフローエンドポイント名。問い合わせがこのデータフローエンドポイントを使用するには、この名前がデータフローエンドポイント設定で使用されている名前と一致する必要があります。
- `EgressAddressAndPort` - エージェントからのデータ出力用の単一の IP アドレスとポートアドレス
- `AgentIpAndPortAddress` - エージェント通信のポート範囲

## データフローエンドポイントグループの作成

データフローエンドポイントグループは、次のいずれかのバージョンを使用して作成できます。

`CreateDataflowEndpointGroup`

下位互換性のため、または汎用 [AwsGroundStationAgentEndpoint](#) または `DataflowEndpoint` タイプを使用する必要がある場合は、[CreateDataflowEndpointGroup](#) を使用します。 [DataflowEndpoint](#)

`CreateDataflowEndpointGroupV2`

[CreateDataflowEndpointGroupV2](#) を新しい実装に使用して、アップリンクとダウンリンクの両方のデータフローをサポートする特殊なエンドポイントタイプを活用します。この API は、[UplinkAwsGroundStationAgentEndpoint](#) と [DownlinkAwsGroundStationAgentEndpoint](#) のみをサポートします。

## 移行に関する考慮事項

現在 `DataflowEndpointGroup` を使用している場合は、変更することなく既存の設定を引き続き使用できます。 は完全な下位互換性 AWS Ground Station を維持します。

新しい [DataflowEndpointGroupV2](#) を使用するように移行する場合、現在 [DataflowEndpoint](#) と [Dataflow Endpoint Application](#) を使用してデータを受信している場合は、代わりに AWS Ground Station エージェントを使用するように移行する必要があります。ダウンリンクに AWS Ground Station エージェントを既に使用している場合は、アップリンクにも同じエージェントインスタンスを使用できます。追加のエージェントインスタンスは必要ありません。

DataflowEndpointGroupV2 に移行するには:

1. [DataflowEndpoint](#) から移行する場合は、AWS Ground Station 「[エージェント](#)[AWS Ground Station ユーザーガイド](#)」に従ってエージェントを設定します。
2. データフローの方向を特定し、適切なエンドポイントタイプ ([UplinkAwsGroundStationAgentEndpoint](#) または [DownlinkAwsGroundStationAgentEndpoint](#)) を作成します。
3. これらのエンドポイントを参照する [DataflowEndpointGroupV2](#) を作成する
4. 新しい [DataflowEndpointGroupV2](#) を名前参照する新しいデータフローエンドポイント設定を作成する [DataflowEndpointGroupV2](#)
5. データフローエンドポイント設定をデータフローエッジとして参照する新しいミッションプロファイルを作成する
6. 新しいミッションプロファイルを使用してコンタクトをスケジュールする
7. 本番環境にデプロイする前に設定をテストする

ワークフローの詳細については、[AWS Ground Station コアコンポーネントを理解する](#)「」および「」を参照してください[設定の作成](#)。

## AWS Ground Station エージェントを使用する

AWS Ground Station エージェントを使用すると、AWS Ground Station のコンタクト中に同期広帯域デジタル中間周波数 (DigIF) データフローを受信 (ダウンリンク) できます。

### 仕組み

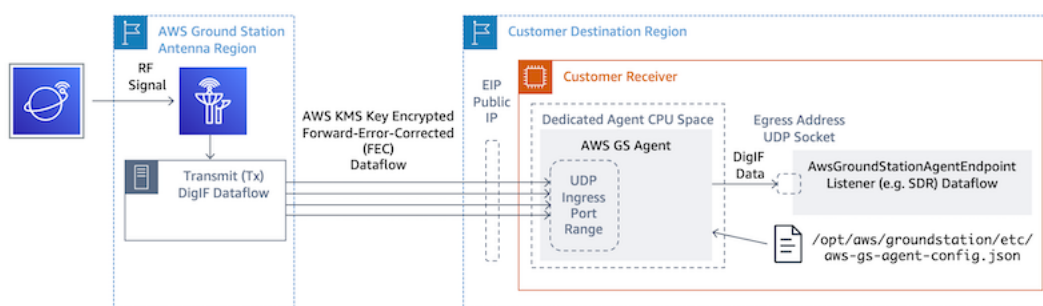
データ配信には 2 つのオプションを選択できます。

1. EC2 インスタンスへのデータ配信 - 所有する EC2 インスタンスへのデータ配信。AWS Ground Station エージェントを管理します。このオプションは、ほぼリアルタイムのデータ処理が必要な場合に最適です。EC2 データ配信の詳細については、[データフローの使用](#)「」セクションを参照してください。

2. S3 バケットへのデータ配信 - AWS S3 バケットへのデータ配信は、[こちら](#)によって完全に管理される AWS Ground Station。S3 データ配信の詳細については、「[はじめにガイド](#)」を参照してください。

どちらのデータ配信モードでも、一連の AWS リソースを作成する必要があります。CloudFormation を使用して AWS リソースを作成することを強くお勧めします。信頼性、正確性、およびサポート性を確保します。各コンタクトは EC2 または S3 のいずれかにのみデータを配信でき、両方に同時に配信することはできません。

次の図は、Software-Defined Radio (SDR) または同様のリスナーを使用して AWS Ground Station、アンテナリージョンから EC2 インスタンスへの DigIF データフローを示しています。



## 追加情報

詳細については、[AWS Ground Station 「エージェントユーザーガイド」](#)を参照してください。

## はじめに

開始する前に、 の基本概念を理解しておく必要があります AWS Ground Station。詳細については、「[の AWS Ground Station 仕組み](#)」を参照してください。

AWS Identity and Access Management (IAM) のベストプラクティスと必要なアクセス許可を以下に示します。適切なロールを設定したら、残りのステップの実行を開始できます。

## にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. <https://portal.aws.amazon.com/billing/signup> を開きます。
2. オンラインの手順に従います。

サインアップ手順の一環として、電話またはテキストメッセージを受け取り、電話キーパッドで検証コードを入力します。

にサインアップすると AWS アカウント、AWS アカウントのルートユーザー が作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティベストプラクティスとして、ユーザーに管理アクセス権を割り当て、[ルートユーザーアクセスが必要なタスク](#)の実行にはルートユーザーのみを使用するようにしてください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。<https://aws.amazon.com/> の [マイアカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

## 管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、日常的なタスクにルートユーザーを使用しないように AWS アカウントのルートユーザー、 のセキュリティを確保し AWS IAM アイデンティティセンター、 を有効にして管理ユーザーを作成します。

## を保護する AWS アカウントのルートユーザー

1. ルートユーザーを選択し、AWS アカウント E メールアドレスを入力して、アカウント所有者[AWS マネジメントコンソール](#)としてサインインします。次のページでパスワードを入力します。

ルートユーザーを使用してサインインする方法については、「AWS サインイン ユーザーガイド」の「[ルートユーザーとしてサインインする](#)」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

手順については、IAM [ユーザーガイドの AWS アカウント「ルートユーザー \(コンソール\) の仮想 MFA デバイス](#)を有効にする」を参照してください。

## 管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「[AWS IAM アイデンティティセンターの有効化](#)」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

を ID ソース IAM アイデンティティセンターディレクトリとして使用する方法的チュートリアルについては、「AWS IAM アイデンティティセンター ユーザーガイド」の「[デフォルトを使用してユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ](#)」を参照してください。

## 管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン「[ユーザーガイド](#)」の [AWS「アクセスポータルにサインインする](#)」を参照してください。

## 追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「[アクセス許可セットを作成する](#)」を参照してください。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「[グループを追加する](#)」を参照してください。

## AWS アカウントにアクセス AWS Ground Station 許可を追加する

管理ユーザーを必要と AWS Ground Station せずに を使用するには、新しいポリシーを作成して AWS アカウントにアタッチする必要があります。

1. にサインイン AWS マネジメントコンソール し、[IAM コンソール](#)を開きます。
2. 新規ポリシーを作成します。以下のステップを使用します。
  - a. ナビゲーションペインで、[Policies (ポリシー)] を選択し、次に [Create Policy (ポリシーの作成)] を選択します。
  - b. [JSON] タブで、次のいずれかの値を使用して JSON を編集します。アプリケーションに最適な JSON を使用します。
    - Ground Station 管理者権限の場合は、次のように [アクション] を [groundstation:\*] に設定します。

JSON

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

```
}

```

- 読み取り専用権限の場合、次に示すように、[アクション] を `groundstation:Get*`、`groundstation:List*`、および `groundstation:Describe*` に設定します。

JSON

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

- 多要素認証を使用したセキュリティの追加では、次に示すように、[アクション] を `groundstation:*` に設定し、[Condition/Bool (条件/ブール)] を `aws:MultiFactorAuthPresent:true` に設定します。

JSON

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

```
    }  
  }  
]  
}
```

3. IAM コンソールで、作成したポリシーを目的のユーザーにアタッチします。

IAM ユーザーの作成とポリシーのアタッチの詳細については、[IAM ユーザーガイド](#)を参照してください。

## 衛星搭載

への衛星のオンボーディング AWS Ground Station は、データ収集、技術検証、スペクトルライセンス、統合とテストを含む複数ステップのプロセスです。非開示契約 (NDAs) も必要です。

### 顧客のオンボーディングプロセスの概要

衛星オンボーディングは、AWS Ground Station コンソールページの[衛星とリソース](#)セクションにある手動プロセスです。次に、プロセス全体について説明します。

1. [AWS Ground Station ロケーション](#) セクションを確認して、衛星が地理的および無線周波数特性を満たしているかどうかを確認します。
2. 衛星のオンボーディングを開始するには AWS Ground Station、AWS Ground Station コンソールページの[衛星とリソースセクション](#)で[衛星](#)オンボーディングアンケートを送信してください。組織名、必要な頻度、衛星がいつ起動されるか、いつ起動されるか、衛星の軌道タイプ、を使用する予定があるかどうかなど、ミッションと衛星のニーズの簡単な概要を含めてください[AWS Ground Station デジタルツイン機能を使用する](#)。
3. リクエストがレビューおよび承認されると、AWS Ground Station は使用する予定の特定の場所で規制ライセンスを申請します。このステップの期間は、場所と既存の規制によって異なります。
4. この承認を取得すると、衛星が使用できるようになります。AWS Ground Station は、正常に更新されたことを通知します。

### (オプション) 衛星の名前付け

オンボーディング後、衛星レコードに名前を追加して、より簡単に認識できます。AWS Ground Station コンソールでは、連絡先ページを使用する際に、衛星のユーザー定義名と Norad ID を表示で

きます。衛星の名前を表示すると、スケジュール設定時に正しい衛星を選択しやすくなります。そのためは、[タグ](#)を使用できます。

AWS Ground Station 衛星のタグ付けは、AWS CLI またはいずれかの AWS SDKs を使用して [tag-resource](#) API を介して行うことができます。このガイドでは、AWS Ground Station CLI を使用してのパブリックブロードキャスト衛星 Aqua (Norad ID 27424) にタグを付ける方法について説明します us-west-2。

## AWS Ground Station CLI

AWS CLI は、とやり取りするために使用できます AWS Ground Station。を使用して衛星 AWS CLI にタグを付ける前に、次の AWS CLI 前提条件を満たす必要があります。

- AWS CLI がインストールされていることを確認します。インストールの詳細については AWS CLI、[「AWS CLI バージョン 2 のインストール」](#)を参照してください。
- AWS CLI が設定されていることを確認します。設定の詳細については AWS CLI、[「AWS CLI バージョン 2 の設定」](#)を参照してください。
- 頻繁に利用される構成設定および認証情報をファイルに保存して AWS CLI によって保守できます。で AWS Ground Station 連絡先を予約および管理するには、これらの設定と認証情報が必要です AWS CLI。設定と認証情報の設定の保存の詳細については、[「設定と認証情報ファイルの設定」](#)を参照してください。

を設定して使用できる状態 AWS CLI になったら、[AWS Ground Station CLI コマンドリファレンス](#) ページを確認して、使用可能なコマンドを理解します。このサービスを使用するときは AWS CLI、コマンド構造に従い、コマンドのプレフィックスを に付け groundstation、使用するサービスとして を指定します AWS Ground Station。AWS CLI コマンド構造の詳細については、[AWS CLI ページの「コマンド構造」](#)を参照してください。コマンド構造の例を以下に示します。

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

### 衛星に名前を付ける

まず、タグ付けする衛星の ARN を取得する必要があります。これは、AWS CLI の [list-satellites](#) API を使用して実行できます。

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

上記の CLI コマンドを実行すると、次のような出力のような出力が返されます。

```
{
  "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"
    }
  ]
}
```

タグ付けする衛星を探して、`satelliteArn` を書き留めます。タグ付けに関する重要な注意点の 1 つは、[タグリソース API](#) にはリージョン ARN が必要であり、[list-satellites](#) によって返される ARN はグローバルであることです。次のステップでは、タグを表示するリージョン (おそらくスケジュールするリージョン) に ARN を拡張する必要があります。この例では、`us-west-2` を使用します。この変更により、ARN は以下のように変更されます。変更元:

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

変更後:

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

コンソールにサテライト名を表示するには、キーとなる `"Name"` があるタグが衛星に必要です。さらに、`jq` を使用しているため AWS CLI、引用符はバックスラッシュでエスケープする必要があります。タグは、次のように表示されます。

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

次に、[タグリソース API](#) を呼び出して衛星にタグを付けます。これは、AWS CLI 次のような方法で実行できます。

```
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"}'
```

これを行うと、衛星に設定した名前が AWS Ground Station コンソールに表示されます。

### 衛星の名前を変更する

衛星の名前を変更する場合は、同じ“Name”キーで、[タグ内の別の値を使用して、衛星 ARN を使用して tag-resource](#) を再度呼び出すことができます。これにより既存のタグが更新され、コンソールに新しい名前が表示されます。この呼び出しの例は、次のようになります：

```
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"}'
```

### 衛星の名前を削除する

衛星の名前セットは、[untag-resource](#) API を使用して削除できます。この API では、タグが存在するリージョンの衛星 ARN とタグキーのリストが必要になります。名前の場合、タグのキーは“Name”です。AWS CLI を使用したこの API 呼び出しの例は、次のようになります。

```
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
```

## パブリックブロードキャスト衛星

独自の衛星のオンボーディングに加えて、パブリックにアクセス可能なダウンリンク通信パスを提供するサポートされているパブリックブロードキャスト衛星のオンボーディングをリクエストできます。これにより、AWS Ground Station を使用してこれらの衛星からデータをダウンリンクできます。

### Note

これらの衛星にアップリンクすることはできません。パブリックにアクセス可能なダウンリンク通信パスのみを使用できます。

AWS Ground Station は、直接ブロードキャストデータをダウンリンクするための次の衛星のオンボーディングをサポートしています。

- Aqua
- SNPP
- JPSS-1/NOAA-20
- Terra

オンボーディングが完了すると、これらの衛星にすぐにアクセスして使用できます。は、サービスの使用を簡単に開始できるように、事前設定された多数の CloudFormation テンプレート AWS Ground Station を維持します。の使用方法の例 [ミッションプロファイル設定の例](#) については、AWS Ground Station 「」を参照してください。

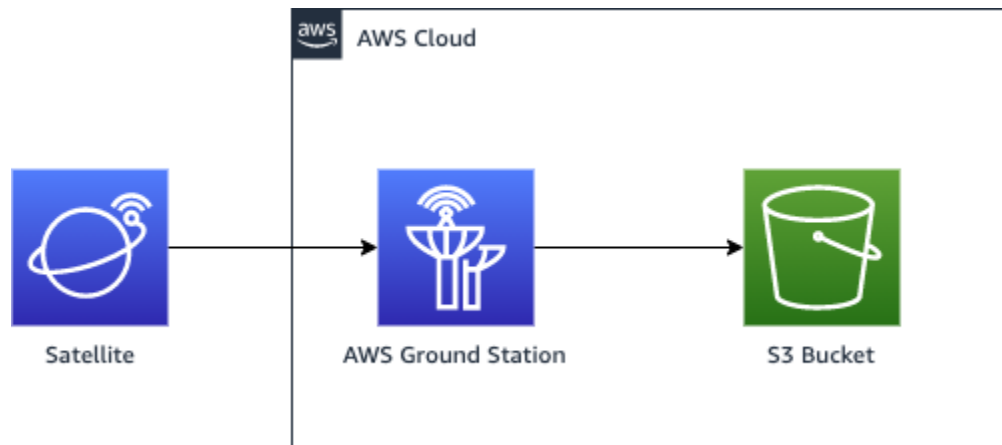
これらの衛星および送信されるデータタイプの詳細については、[Aqua](#)、[JPSS-1/NOAA-20](#) および [SNPP](#)、および [Terra](#) を参照してください。

## データフロー通信パスを計画する

衛星上の各通信パスに対して、同期通信と非同期通信を選択できます。衛星とユースケースによっては、1つまたは両方のタイプが必要になる場合があります。同期通信パスにより、ほぼリアルタイムのアップリンク、ナローバンドおよびワイドバンドのダウンリンク操作が可能になります。非同期通信パスは、ナローバンドおよびワイドバンドのダウンリンクオペレーションのみをサポートします。

### 非同期データ配信

Amazon S3 へのデータ配信では、アカウント内の Amazon S3 バケットにコンタクトデータが非同期に配信されます。コンタクトデータは、コンタクトデータをソフトウェア定義無線 (SDR) に再生するため、または処理を目的としてパケットキャプチャ (pcap) ファイルからペイロードデータを抽出するために pcap ファイルとして配信されます。pcap ファイルは、コンタクトデータがアンテナハードウェアによって受信されると、30 秒ごとに Amazon S3 バケットに配信され、必要に応じてコンタクト中にコンタクトデータを処理できます。受信すると、独自の後処理ソフトウェアを使用してデータを処理したり、Amazon SageMaker AI や Amazon Rekognition などの他の AWS のサービスを使用したりできます。Amazon S3 へのデータ配信は、衛星からのデータのダウンリンクにのみ使用できます。Amazon S3 から衛星にデータをアップリンクすることはできません。



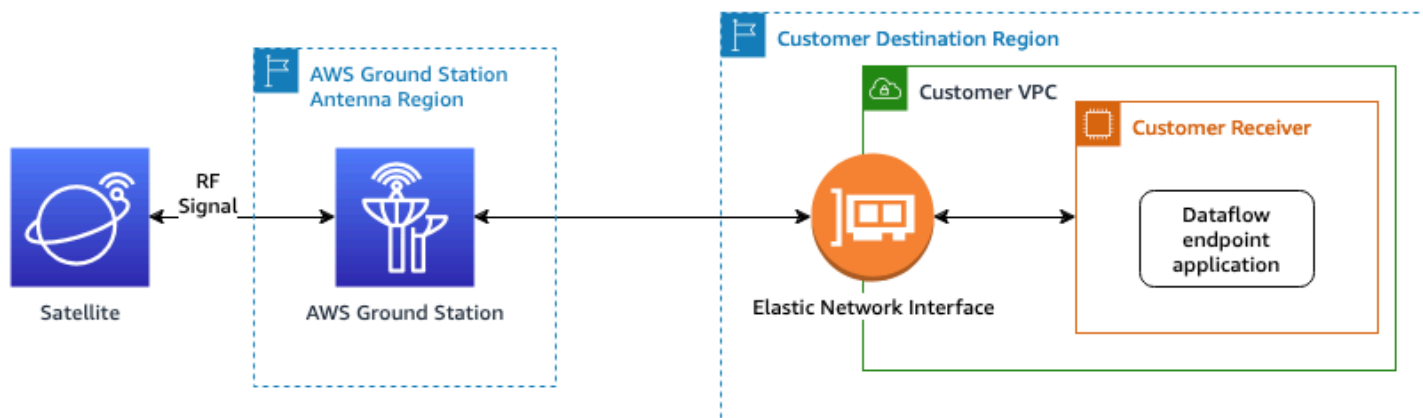
このパスを利用するには、の Amazon S3 バケットを作成してデータを AWS Ground Station 配信する必要があります。次のステップでは、次のステップで S3 Recording Config も作成する必要があります。バケットの命名に関する制限と、ファイルに使用する命名規則を指定する方法については、[Amazon S3 録画設定](#)「」を参照してください。

## 同期データ配信

Amazon EC2 へのデータ配信では、コンタクトデータは Amazon EC2 インスタンスに対してストリーミングされます。Amazon EC2 インスタンスでデータをリアルタイムで処理することや、後処理のためにデータを転送することができます。

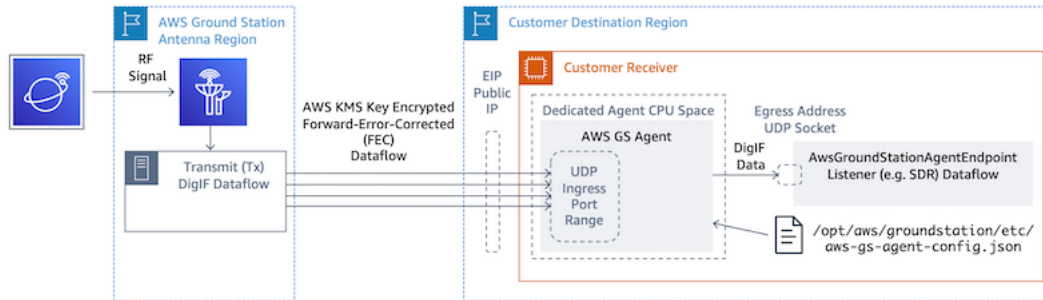
同期パスを利用するには、を使用して Amazon EC2 インスタンスをセットアップおよび設定し、1つ以上の Dataflow Endpoint Groups を作成します。Amazon EC2 インスタンスを設定するには、を参照してください[Amazon EC2 のセットアップと設定](#)。Dataflow エンドポイントグループを作成するには、を参照してください[AWS Ground Station Dataflow エンドポイントグループを使用する](#)。

データフローエンドポイント設定を使用している場合の通信パスを次に示します。



\*End to end data connection is established and maintained only during the scheduled contact duration.

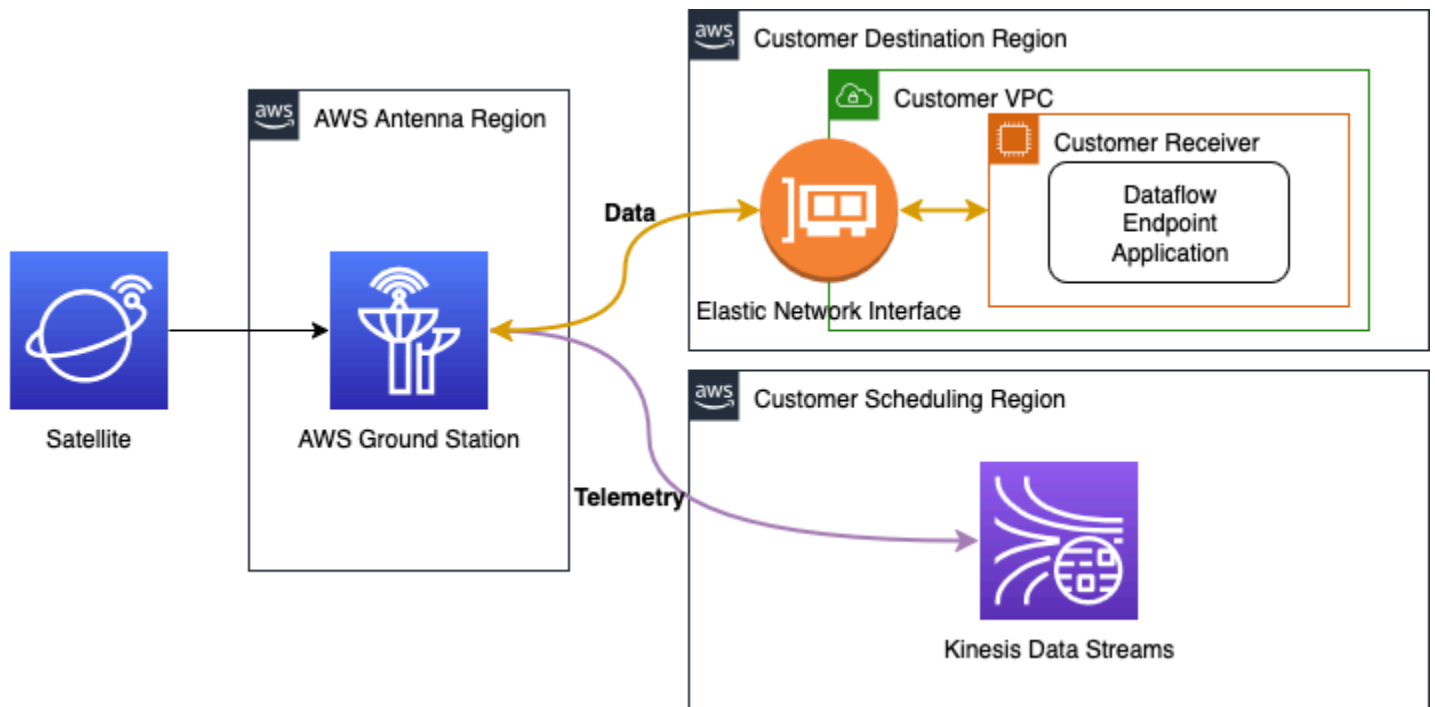
AWS Ground Station エージェント設定を使用している場合の通信パスを次に示します。



## テレメトリを計画する

AWS Ground Station テレメトリは、衛星コンタクト中にアンテナから AWS Ground Station AWS アカウントにメトリクスをストリーミングするオプション機能です。これにより、コンタクトのパフォーマンスをほぼリアルタイムでモニタリングし、カスタムモニタリングソリューションを構築できます。

AWS Ground Station テレメトリを使用すると、AWS Ground Station アンテナからのメトリクスがアカウントに直接ストリーミングされます。テレメトリデータは、問い合わせの開始時にストリーミングを開始し、問い合わせ期間を通じて継続されます。テレメトリデータは、アンテナハードウェアからサンプリングされるため、ほぼリアルタイムでアカウントに配信されます。受信したら、独自の後処理ソフトウェアを使用してデータを処理したり、Amazon Data Firehose や などの他の AWS のサービスを使用したりできます AWS Lambda。



次のステップでは、ミッションプロファイルに必要な設定を作成します。テレメトリを有効にする場合は、追跡設定とデータフロー設定に加えてテレメトリシンク設定を作成します。詳細なセットアップ手順については、「」を参照してください[テレメトリの設定](#)。

TelemetrySinkConfig の詳細については、「」を参照してください[テレメトリシンク設定](#)。

## 設定の作成

このステップでは、必要に応じて衛星、通信パス、IAM、Amazon EC2、Amazon S3 リソースを特定しました。このステップでは、それぞれのパラメータを保存する設定を作成します AWS Ground Station 。

### データ配信設定

作成する最初の設定は、データの配信場所と配信方法に関連しています。前のステップの情報を使用して、以下の設定タイプの多くを構築します。

- [Amazon S3 録画設定](#) - Amazon S3 バケットにデータを配信します。
- [データフローエンドポイント設定](#) - Amazon EC2 インスタンスにデータを配信します。

### テレメトリ設定 (オプション)

コンタクト中にほぼリアルタイムのテレメトリを受信する場合は、TelemetrySinkConfig を作成できます。この設定はオプションであり、AWS Ground Station がテレメトリデータを配信する場所を指定します。

- [テレメトリシンク設定](#) - テレメトリデータをアカウントに配信します。

詳細なセットアップ手順については、「」を参照してください[テレメトリの設定](#)。

### 衛星設定

衛星設定は、AWS Ground Station が衛星と通信する方法に関連しています。で収集した情報を参照します[衛星搭載](#)。

- [追跡設定](#) - コンタクト中に車両を物理的に追跡する方法のプリファレンスを設定します。これは、ミッションプロファイルの構築に必要です。

- [アンテナダウンリンク設定](#) - デジタル化された無線周波数データを配信します。
- [アンテナダウンリンク復調デコード設定](#) - 復調およびデコードされた無線周波数データを配信します。
- [アンテナアップリンク設定](#) - 衛星にデータをアップリンクします。
- [アンテナアップリンクエコー設定](#) - アップリンク信号データのエコーを配信します。

## ミッションプロファイルを作成する

前のステップで作成した設定では、衛星を追跡する方法、衛星と通信する方法、問い合わせの実行中にほぼリアルタイムのテレメトリを有効にする方法を特定しました。このステップでは、1つ以上のミッションプロファイルを作成します。ミッションプロファイルは、可能な設定を予想される動作に集約し、それをスケジュールして運用します。

最新のパラメータについては、[AWS::GroundStation::MissionProfile CloudFormation リソースタイプ](#)を参照してください。

1. ミッションプロファイルに名前を付けます。これにより、システム内での使用状況をすばやく把握できます。例えば、緊急オペレーション用の別のナローバンドキャリアがある場合は、satellite-wideband-narrowband-nominal-operationsオペレーションとsatellite-narrowband-emergency-operationsがあります。
2. 追跡設定を設定します。
3. 有効な最小コンタクト期間を設定します。これにより、潜在的な問い合わせをフィルタリングしてミッションのニーズを満たすことができます。
4. 転送中のデータの暗号化に使用される streamsKmsKey と streamsKmsRole を設定します。これは、すべての AWS Ground Station エージェントデータフローに使用されます。
5. データフローを設定します。前のステップで作成した設定を使用して、キャリアシグナルと一致するデータフローを作成します。
6. [オプション] パス前とパス後のコンタクト継続時間を秒単位で設定します。これは、コンタクト前とコンタクト後にそれぞれコンタクトごとのイベントを出力するために使用されます。詳細については「[イベント AWS Ground Station による自動化](#)」を参照してください。
7. [オプション] telemetrySinkConfigArn を設定して、コンタクト中のテレメトリを有効にします。これにより、モニタリングと分析のために、アカウントでほぼリアルタイムのテレメトリを直接受信できます。詳細については「[テレメトリの操作](#)」を参照してください。
8. [オプション] タグをミッションプロファイルに関連付けることができます。これらは、ミッションプロファイルをプログラムで区別するのに役立ちます。

を参照して[ミッションプロファイル設定の例](#)、潜在的な設定の一部のみを表示できます。

## 次のステップを理解する

これで、オンボードされた衛星と有効なミッションプロファイルが作成され、コンタクトをスケジュールし、衛星と通信する準備が整いました AWS Ground Station。

次のいずれかの方法でコンタクトをスケジュールできます。

- [AWS Ground Station コンソール](#)。
- AWS CLI [リザーブコンタクト](#) コマンド。
- AWS SDK。 [ReserveContact](#) API。

が衛星の軌道 AWS Ground Station を追跡する方法とその情報の使用方法については、「」を参照してください [がエフェメリス AWS Ground Station を使用する方法を理解する](#)。

AWS Ground Station では、サービスの使用を簡単に開始できるように、事前設定された CloudFormation テンプレートが多数用意されています。の使用方法の例 [ミッションプロファイル設定の例](#) については、AWS Ground Station 「」を参照してください。

デジタル中間周波数データ、または から提供される復調およびデコードされたデータは AWS Ground Station 、特定のユースケースによって異なります。以下のブログ記事は、利用可能なオプションの一部を理解するのに役立ちます。

- [AWS Ground Station Amazon S3 データ配信を使用した自動地球観測](#) (および関連付けられた GitHub リポジトリ [awslabs/aws-groundstation-eos-pipeline](#) )
- [を使用して衛星地上セグメントを仮想化する AWS](#)
- [を使用した地球観測 AWS Ground Station: ガイド方法](#)
- [AWS Ground Station WideBand DigIF と Amphinicy Blink SDR](#) (および関連付けられた GitHub リポジトリ [aws-samples/aws-groundstation-wbdigif-snpp](#)) を使用した高スループットの衛星データダウンロードアーキテクチャの構築 GitHub [aws-samples/aws-groundstation-wbdigif-snpp](#)

# AWS Ground Station ロケーション

AWS Ground Station は、AWS インフラストラクチャリージョンのグローバルネットワークに近接した地上局のグローバルネットワークを提供します。これらの場所の使用は、サポートされている任意の AWS リージョンから設定できます。これには、データが配信される AWS リージョンが含まれます。



## 地上ステーションの場所の AWS リージョンの検索

AWS Ground Station グローバルネットワークには、接続先の [AWS リージョン](#) に物理的に配置されていない地上ステーションの場所が含まれます。アクセスできる地上ステーションのリストは、AWS SDK [ListGroundStation](#) レスポンスを介して取得できます。地上ステーションの場所の完全なリストを以下に示します。詳細は近日公開予定です。衛星のサイト承認を追加または変更するには、オンボーディングガイドを参照してください。

Ground Station 名	Ground Station の場所	AWS リージョン 名	AWS リージョン コード	メモ
アラスカ 1	アラスカ、米国	米国西部 (オレゴン)	us-west-2	物理的に AWS リージョンに配置されていない
バーレーン 1	バーレーン	中東 (バーレーン)	me-south-1	
ケープタウン 1	ケープタウン、南アフリカ	アフリカ (ケープタウン)	af-south-1	
Dubbo 1	オーストラリア、ダボ	アジアパシフィック (シドニー)	ap-southeast-2	物理的に AWS リージョンに配置されていない
ハワイ 1	米国、ハワイ	米国西部 (オレゴン)	us-west-2	物理的に AWS リージョンに配置されていない
アイルランド 1	アイルランド	欧州 (アイルランド)	eu-west-1	
オハイオ 1	米国オハイオ州	米国東部 (オハイオ)	us-east-2	
オレゴン 1	オレゴン、米国	米国西部 (オレゴン)	us-west-2	
プンタアリーナ 1	プンタアリーナス、チリ	南米 (サンパウロ)	sa-east-1	物理的に AWS リージョンに配置されていない
ソウル 1	ソウル、韓国	アジアパシフィック (ソウル)	ap-northeast-2	

Ground Station 名	Ground Station の場所	AWS リージョン 名	AWS リージョン コード	メモ
シンガポール 1	シンガポール	アジアパシフィック (シンガポール)	ap-southeast-1	
ストックホルム 1	ストックホルム、スウェーデン	欧州 (ストックホルム)	eu-north-1	

## AWS Ground Station サポートされている AWS リージョン

AWS SDK またはサポートされている AWS リージョンの AWS Ground Station コンソールを使用して、データを配信し、連絡先を設定できます。サポートされているリージョンおよび関連するエンドポイントは、[AWS Ground Station エンドポイントとクォータ](#)で表示できます。

## デジタルツインの可用性

[AWS Ground Station デジタルツイン機能を使用する](#) は、AWS Ground Station が利用可能なすべての [AWS リージョン](#) で利用できます。デジタルツイン地上ステーションは、「デジタルツイン」というプレフィックスが付けられた本番稼働用地上ステーションの正確なコピーです。たとえば、「デジタルツインオハイオ 1」は、「オハイオ 1」本番稼働用地上ステーションの正確なコピーであるデジタルツイン地上ステーションです。

## AWS Ground Station サイトマスク

各 AWS Ground Station [アンテナロケーション](#) には、サイトマスクが関連付けられています。これらのマスクは、その場所のアンテナが特定の方向 (通常は地平線に近い) を指しているときに、送信または受信をブロックします。マスクには以下の事項が考慮されます。

- アンテナを囲む地理的地形の特徴 – 例えば、無線周波数 (RF) 信号をブロックしたり、送信を妨げたりする山や建物などがあります。
- 無線周波数干渉 (RFI) – これは、受信機能 (AWS Ground Station アンテナへのダウンリンク信号に影響を与える外部 RFI ソース) と送信機能 (外部レシーバーに悪影響を与える AWS Ground Station アンテナによって送信される RF 信号) の両方に影響します。

- 法的認可 – 各リージョンで AWS Ground Station を運用するためのローカルサイトの認可には、送信の最小標高角度などの特定の制限が含まれる場合があります。

これらのサイトマスクは時間の経過とともに変化する可能性があります。例えば、アンテナの場所の付近に新しい建物が建設されたり、RFI ソースが変更されたり、法的許可が更新されて異なる制限が適用される場合があります。AWS Ground Station サイトマスクは、秘密保持契約 (NDA) に基づいて利用できます。

## お客様固有のマスク

各サイトの AWS Ground Station サイトマスクに加えて、特定のリージョンの衛星と通信するための独自の法的認可の制限により、追加のマスクがある場合があります。このようなマスクは AWS Ground Station でケースバイケースで設定できるため、AWS Ground Station を使用してこれらの衛星と通信する際のコンプライアンスを確保できます。詳細については、AWS Ground Station チームにお問い合わせください。

## 利用可能なコンタクト時間に対するサイトマスクの影響

サイトマスクには、アップリンク (送信) サイトマスクとダウンリンク (受信) サイトマスクの 2 種類があります。

ListContacts オペレーションを使用して利用可能なコンタクト時間を一覧表示すると、AWS Ground Station は、衛星がダウンリンクマスクを超えて設定されるタイミングに基づいて可視性時間を返します。使用可能なコンタクト時間は、このダウンリンクマスクの可視性ウィンドウに基づいています。これにより、衛星がダウンリンクマスクを下回る時間を確保できなくなります。

ミッションプロファイル内のデータフローエッジに [Antenna Uplink Config](#) が含まれていても、アップリンクサイトマスクは利用可能なコンタクト時間には適用されません。これにより、アップリンクサイトマスクが原因でその時間の一部でアップリンクが利用できない場合でも、ダウンリンクに使用できるすべてのコンタクト時間を使用できます。ただし、アップリンク信号は、衛星コンタクト用に予約された時間の一部または全部の間は送信されない場合があります。アップリンク送信をスケジューリングするときは、提供されたアップリンクマスクを考慮する必要があります。

コンタクトのうち、アップリンクに使用できない部分は、アンテナの場所のアップリンクサイトマスクを基準としたコンタクト中の衛星軌道によって異なります。アップリンクサイトマスクとダウンリンクサイトマスクが類似しているリージョンでは、通常、この期間は短くなります。それ以外の、アップリンクマスクがダウンリンクサイトマスクよりもかなり長いリージョンでは、コンタクト時間の相当部分または全部がアップリンク用として利用できなくなる可能性があります。予約時間の一部がアップリンクに利用できない場合でも、完全な問い合わせ時間が請求されます。

## AWS Ground Station サイト機能

エクスペリエンスを簡素化するために、はアンテナタイプの一般的な機能セット AWS Ground Station を決定し、複数のアンテナを地上局の場所にデプロイします。オンボーディングステップの一部により、衛星が特定の場所のアンテナタイプと互換性があります。コンタクトを予約するときは、使用するアンテナタイプを間接的に決定します。これにより、どのアンテナが使用されているかに関係なく、特定の地上局の場所でのエクスペリエンスが時間の経過とともに同じままになります。問い合わせの特定のパフォーマンスは、サイトの天気など、さまざまな環境上の懸念により異なります。

現在、すべてのサイトは次の機能をサポートしています。

### Note

次の表の各行は、特に明記されていない限り、独立した通信パスを示しています。複数の通信パスを同時に使用できるマルチチャネル機能を反映するために、重複した行があります。

機能タイプ	頻度範囲	帯域幅範囲	Polarization	共通名	メモ
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	RHCP	Xバンドワイドバンドダウンリンク	この機能を使用するには、 <a href="#">AWS Ground Station エージェント</a> を使用する必要があります。
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	RHCP		
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	RHCP		
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	RHCP		
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	RHCP		
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	LHCP		

この機能は、Alaska 1 または Punta Arenas 1 ではサポートされていません。

総帯域幅は、各場所の偏

機能タイプ	頻度範囲	帯域幅範囲	Polarization	共通名	メモ
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	LHCP		波ごとに400MHzを超えることはできません。  使用するすべての頻度範囲は重複しない必要があります。
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	LHCP		
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	LHCP		
アンテナダウンリンク	7750 ~ 8500 MHz	50 ~ 400 MHz	LHCP		
アンテナダウンリンク	2200 ~ 2290 MHz	最大 40 MHz	RHCP	S バンドダウンリンク	一度に使用できる偏波は1つだけです
アンテナダウンリンク	2200 ~ 2290 MHz	最大 40 MHz	LHCP		
アンテナダウンリンク	7750 ~ 8500 MHz	最大 40 MHz	RHCP	X バンドナローバンドダウンリンク	一度に使用できる偏波は1つだけです
アンテナダウンリンク	7750 ~ 8500 MHz	最大 40 MHz	LHCP		
アンテナアップリンク	2025 ~ 2110 MHz	最大 40 MHz	RHCP	S バンドアップリンク	一度に使用できる偏波は1つだけです
アンテナアップリンク	2025 ~ 2110 MHz	最大 40 MHz	LHCP		
antenna-uplink-echo	2025 ~ 2110 MHz	2 MHz	RHCP	アップリンクエコー	アンテナアップリンクの制限に一致
antenna-uplink-echo	2025 ~ 2110 MHz	2 MHz	LHCP		

機能タイプ	頻度範囲	帯域幅範囲	Polarization	共通名	メモ
antenna-d ownlink-d emod-decode	7750 ~ 8500 MHz	最大 500 MHz	RHCP	Xバンドの復 調およびデ コードされた ダウンリンク	
antenna-d ownlink-d emod-decode	7750 ~ 8500 MHz	最大 500 MHz	LHCP		
追跡	N/A	N/A	N/A	N/A	自動追跡とプ ログラム追跡 のサポート

\* RHCP = 右回りの円偏波、LHCP = 左回りの円偏波。偏波の詳細については、[「円偏波」](#)を参照してください。

# がエフェメリス AWS Ground Station を使用する方法を理解する

[エフェメリス](#) (単数形: ephemeris、複数形: ephemerides) は、天体の軌道を提供するファイルまたはデータ構造です。従来、このファイルは表形式のデータのみを参照していましたが、次第に、宇宙機の軌道を示すさまざまなデータファイルを参照するようになりました。

Ephemeris API では、衛星 AWS Ground Station で使用するカスタムエフェメリスを にアップロードできます。これらのエフェメリスは、[Space-Track](#) からのデフォルトのエフェメリスを上書きします (「」を参照[デフォルトのエフェメリスデータ](#))。エフェメリスデータの受信は、オービットエフェメリスメッセージ (OEM)、2 行要素 (TLE)、および方位標高形式でサポートされています。

AWS Ground Station はエフェメリスデータを使用して、提供されたエフェメリスに基づいてコンタクトがいつ利用可能になるかを判断し、AWS Ground Station ネットワーク内のアンテナを正しくコマンドします。デフォルトでは、衛星に [NORAD ID](#) が割り当てられている場合、AWS Ground Station にエフェメリスを提供するアクションは必要ありません。

カスタムエフェメリスをアップロードすると、追跡の品質が向上し、[Space-Track](#) エフェメリスが利用できない早期オペレーションを処理し AWS Ground Station、操作を考慮できます。

または、 は方位標高形式 AWS Ground Station をサポートしているため、衛星軌道情報を提供せずにアンテナの向きを直接指定できます。これは、衛星軌道情報が不正確または不明であるため、正確なアンテナポイントが必要なシナリオに役立ちます。

## トピック

- [デフォルトのエフェメリスデータ](#)
- [カスタムエフェメリスデータを提供する](#)
- [カスタムエフェメリスを使用してコンタクトを予約する](#)
- [どのエフェメリスが使用されているかを理解する](#)
- [衛星の現在のエフェメリスを取得する](#)
- [デフォルトのエフェメリスデータに戻す](#)

## デフォルトのエフェメリスデータ

デフォルトでは、 は [Space-Track](#) から公開されているデータ AWS Ground Station を使用し、これらのデフォルトのエフェメリスを指定 AWS Ground Station するためのアクションは必要ありません。

せん。これらのエフェメリスは、衛星の [NORAD ID](#) に関連付けられた [2 行の要素セット \(TLEs\)](#) です。すべてのデフォルトのエフェメリスの優先度は 0 です。その結果、有効期限が切れていないカスタムエフェメリスは、エフェメリス API を介してアップロードされ、常に 1 以上の優先度を持つ必要があります。

NORAD ID のない衛星は、カスタムエフェメリスデータを にアップロードする必要があります AWS Ground Station。たとえば、起動したばかりの衛星や、意図的に [Space-Track](#) カタログから除外された衛星には NORAD ID がいないため、カスタムエフェメリスをアップロードする必要があります。カスタムエフェメリスデータの提供の詳細については、[「カスタムエフェメリスデータの提供」](#) を参照してください。

## カスタムエフェメリスデータを提供する

### Important

エフェメリス API は現在プレビュー状態です。

エフェメリス API へのアクセスは、必要な場合にのみ提供されます。カスタムエフェメリスデータをアップロードする必要がある場合は、[AWS Support Center Console](#) を通じて AWS サポート チケットを開いてください。AWS [Support Center Console](#)。当社のチームがお客様と協力して、特定の要件に応じてこの機能を有効にします。

### 概要:

Ephemeris API では、衛星 AWS Ground Station で使用するカスタムエフェメリスを にアップロードできます。これらのエフェメリスは、[Space-Track](#) のデフォルトのエフェメリスを上書きします (「[デフォルトのエフェメリスデータ](#)」を参照)。エフェメリスデータの受信は、オービットエフェメリスメッセージ (OEM)、2 行要素 (TLE)、および方位標高形式でサポートされています。

AWS Ground Station はエフェメリスを [個別の使用状況データ](#) として扱います。このオプション機能を使用する場合、AWS はエフェメリスデータを使用してトラブルシューティングサポートを提供します。

カスタムエフェメリスをアップロードすると、追跡の品質が向上し、[Space-Track](#) エフェメリスが利用できないオペレーションを処理し AWS Ground Station、操作を考慮できます。

無効なエフェメリスのトラブルシューティングを行うには、以下を参照してください。 [無効なエフェメリスのトラブルシューティング](#)

## 例: お客様提供のエフェメリスを で使用する AWS Ground Station

で顧客提供のエフェメリスを使用する詳細な手順については AWS Ground Station、 「[で顧客提供のエフェメリスを使用する AWS Ground Station](#)」 および 「[関連付けられた GitHub リポジトリ `aws-samples/aws-groundstation-cpe`](#)」 を参照してください。

### TLE エフェメリスデータを提供する

#### Important

エフェメリス API は現在プレビュー状態です。

エフェメリス API へのアクセスは、必要な場合にのみ提供されます。カスタムエフェメリスデータをアップロードする必要がある場合は、 を通じて AWS サポート チケットを開いてください [AWS Support Center Console](#)。当社のチームがお客様と協力して、特定の要件に応じてこの機能を有効にします。

#### 概要:

2 行要素 (TLE) セットは、衛星軌道を記述するための標準化された形式です。Ephemeris API では、衛星 AWS Ground Station で使用する TLE エフェメリスを にアップロードできます。これらのエフェメリスは、[Space-Track](#) のデフォルトのエフェメリスを上書きします (「」を参照 [デフォルトのエフェメリスデータ](#))。

AWS Ground Station はエフェメリスを [個別の使用状況データ](#) として扱います。このオプション機能を使用する場合、AWS はエフェメリスデータを使用してトラブルシューティングサポートを提供します。

カスタム TLE エフェメリスをアップロードすると、追跡の品質が向上し、[Space-Track](#) エフェメリスが利用できない早期オペレーションを処理し AWS Ground Station、操作を考慮できます。

#### Note

衛星に衛星カタログ番号が割り当てられる前にカスタムエフェメリスを提供する場合、TLE の 00000 衛星カタログ番号フィールドと TLE の国際指定子フィールドの 000 起動番号部分 (2024 年に起動された車両など) 24000A に を使用できます。

TLEs」 を参照してください。 [https://en.wikipedia.org/wiki/Two-line\\_element\\_set](https://en.wikipedia.org/wiki/Two-line_element_set)

## TLE エフェメリスの作成

TLE エフェメリスは、API の [CreateEphemeris](#) アクション AWS Ground Station を使用して作成できます。このアクションによって、リクエスト本文または指定された S3 バケットのデータを使用してエフェメリスがアップロードされます。

エフェメリスがアップロードされると、エフェメリスが VALIDATING に設定されて非同期ワークフローが開始されることに注意してください。このワークフローでは、エフェメリスを検証し、そのエフェメリスを基に潜在的なコンタクトを生成します。エフェメリスは、このワークフローを実施して ENABLED になって初めて、コンタクトに使用されます。 [DescribeEphemeris](#) をポーリングしてエフェメリスステータスを確認するか、CloudWatch イベントを使用してエフェメリスのステータスの変化を追跡する必要があります。

無効なエフェメリスのトラブルシューティングを行うには、以下を参照してください。 [無効なエフェメリスのトラブルシューティング](#)

### 例: API を使用して 2 行要素 (TLE) セットエフェメリスを作成する

AWS SDKs と CLI を使用して、[CreateEphemeris](#) 呼び出し AWS Ground Station を介して 2 行要素 (TLE) セットのエフェメリスを にアップロードできます。このエフェメリスは、衛星のデフォルトのエフェメリスデータの代わりに使用されます (「」を参照[デフォルトのエフェメリスデータ](#))。この例では、[AWS SDK for Python \(Boto3\)](#) を使用してこれを行う方法を示します。

TLE セットは JSON 形式のオブジェクトで、1 つ以上の TLE をつなぎ合わせて連続した軌道を構築します。TLE セット内の TLE は、軌道の構築に使用できる連続したセットを形成する (つまり、TLE セット内の TLE 間に時間的なギャップがない) 必要があります。TLE セットの例を以下に示します。

```
[
  {
    "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": 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 セット内の TLE の時間範囲は、有効で連続的な軌道になるためには正確に一致する必要があります。

TLE セットは、次のように boto3 AWS Ground Station クライアントを介してアップロードできます。

```

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']}")

```

この呼び出しは、今後エフェメリスを参照するために使用できる `ephemerisId` を返します。たとえば、上記の呼び出しから提供された `ephemerisId` を使用して、エフェメリスのステータスをポーリングできます。

```

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 .00000075 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 = 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](#) アクションからのレスポンスの例を以下に示します。

```

{
  "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](#) ルートをポーリングするか、CloudWatch イベントを使用して、アップロードされたエフェメリスのステータスを追跡することをお勧めします。アップロードされたエフェメリスは、に設定ENABLEDされ、問い合わせのスケジュールと実行に使用できるようになる前に非同期検証ワークフローを通過する必要があるためです。

上記の例では、TLEs セット内のすべての TLE の NORAD ID 25994 は、衛星が [Space-Track](#) データベースで割り当てられている NORAD ID と一致する必要があります。

### 例: S3 バケットから TLE エフェメリスデータをアップロードする

バケットとオブジェクトキーを指すことで、S3 バケットから直接 TLE エフェメリスファイルをアップロードすることもできます。AWS Ground Station はユーザーに代わってオブジェクトを取得します。の保管中のデータの暗号化の詳細については AWS Ground Station、「」を参照してください [保管中のデータ暗号化 AWS Ground Station](#)。

以下は、S3 バケットから TLE エフェメリスファイルをアップロードする例です。

```
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 .00000075 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 .00000075 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 エフェメリスデータを提供する

### Important

エフェメリス API は現在プレビュー状態です。

エフェメリス API へのアクセスは、必要な場合にのみ提供されます。カスタムエフェメリスデータをアップロードする必要がある場合は、[AWS](#) を通じて AWS サポート チケットを開いてください

[Support Center Console](#)。当社のチームがお客様と協力して、特定の要件に応じてこの機能を有効にします。

## 概要:

軌道エフェメリスメッセージ (OEM) は、宇宙機の軌道データを表すための標準化された形式です。Ephemeris API を使用すると、OEM エフェメリスをアップロードして AWS Ground Station にアップロードして衛星で使用できます。これらのエフェメリスは、[Space-Track](#) からのデフォルトのエフェメリスを上書きします (「」を参照[デフォルトのエフェメリスデータ](#))。

AWS Ground Station はエフェメリスを[個別の使用状況データ](#)として扱います。このオプション機能を使用する場合、AWS はエフェメリスデータを使用してトラブルシューティングサポートを提供します。

カスタム OEM エフェメリスをアップロードすると、追跡の品質が向上し、[Space-Track](#) エフェメリスが利用できない早期オペレーションを処理し AWS Ground Station、操作を考慮できます。

### Note

衛星に衛星カタログ番号が割り当てられる前にカスタムエフェメリスを提供する場合、OEM の satelliteId OBJECT\_ID 部分にを使用できます。OEMs 「」を参照してください[OEM エフェメリス形式](#)。

## OEM エフェメリス形式

AWS Ground Station は、[CCSDS 標準](#)に従って OEM カスタマー提供のエフェメリスを処理しますが、追加の制限があります。OEM ファイルは KVN 形式である必要があります。次の表は、OEM のさまざまなフィールドと、と CCSDS 標準 AWS Ground Station の違いの概要を示しています。

セクション	フィールド	CCSDS が必要	AWS Ground Station 必須	注意事項
ヘッダー	CCSDS_OEM_VERS	はい	はい	必須値: 2.0
	COMMENT	いいえ	いいえ	
	分類	いいえ	いいえ	

セクション	フィールド	CCSDS が必要	AWS Ground Station 必須	注意事項
	CREATION_DATE	はい	はい	
	発信者	はい	はい	
	メッセージ ID	いいえ	いいえ	
メタデータ	META_START	はい	はい	
	COMMENT	いいえ	いいえ	
	オブジェクト名	はい	はい	
	オブジェクト ID	はい	はい	
	CENTER_NAME	はい	はい	必須値: Earth
	REF_FRAME	はい	はい	許容値: EME2000, ITRF2000
	REF_FRAME_EPOCH	いいえ	サポートされていない*	承認された REF_FRAMEs には暗黙的なエポックがあるため不要
	TIME_SYSTEM	はい	はい	必須値: UTC
	START_TIME	はい	はい	
	USEABLE_START_TIME	いいえ	いいえ	
	USEABLE_TOP_TIME	いいえ	いいえ	

セクション	フィールド	CCSDS が必要	AWS Ground Station 必須	注意事項
	STOP_TIME	はい	はい	
	解釈	いいえ	はい	AWS Ground Station がコンタクトの正確なポインティング角度を生成できるようにするために必要です。
	INTERPOLATION_DEGREE	いいえ	はい	AWS Ground Station がコンタクトの正確なポインティング角度を生成できるようにするために必要です。
	META_STOP	はい	はい	
データ	X	はい	はい	で表される km
	はい	はい	はい	で表される km
	Z	はい	はい	で表される km
	X_DOT	はい	はい	で表される km/s
	Y_DOT	はい	はい	で表される km/s
	Z_DOT	はい	はい	で表される km/s
	X_DDOT	いいえ	いいえ	で表される km/s <sup>2</sup>

セクション	フィールド	CCSDS が必要	AWS Ground Station 必須	注意事項
	Y_DDOT	いいえ	いいえ	で表される km/s <sup>2</sup>
	Z_DDOT	いいえ	いいえ	で表される km/s <sup>2</sup>
共分散行列	COVARIANCE_START	いいえ	いいえ	
	EPOCH	いいえ	いいえ	
	COV_REF_FRAME	いいえ	いいえ	
	COVARIANCE_STOP	いいえ	いいえ	

\* でサポートされていない行 AWS Ground Station が提供された OEM に含まれている場合、OEM は検証に失敗します。

の CCSDS 標準からの重要な逸脱 AWS Ground Station は次のとおりです。

- CCSDS\_OEM\_VERS は である必要があります 2.0。
- REF\_FRAME は、EME2000 または のいずれかである必要があります ITRF2000。
- REF\_FRAME\_EPOCH は ではありません AWS Ground Station。
- CENTER\_NAME は である必要があります Earth。
- TIME\_SYSTEM は である必要があります UTC。
- INTERPOLATION と INTERPOLATION\_DEGREE はどちらも、AWS Ground Station お客様が提供したエフェメリスに必要です。

## KVN 形式の OEM エフェメリスの例

以下は、JPSS-1 パブリックブロードキャスト衛星の KVN 形式の OEM エフェメリスの切り捨てられた例です。

```
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 エフェメリスの作成

OEM エフェメリスは、API の [CreateEphemeris](#) アクション AWS Ground Station を使用して作成できます。このアクションによって、リクエスト本文または指定された S3 バケットのデータを使用してエフェメリスがアップロードされます。

エフェメリスがアップロードされると、エフェメリスが VALIDATING に設定されて非同期ワークフローが開始されることに注意してください。このワークフローでは、エフェメリスを検証し、そのエフェメリスを基に潜在的なコンタクトを生成します。エフェメリスは、このワークフローを実施して ENABLED になって初めて、コンタクトに使用されます。 [DescribeEphemeris](#) をポーリングしてエフェメリスステータスを確認するか、CloudWatch イベントを使用してエフェメリスのステータスの変化を追跡する必要があります。

無効なエフェメリスのトラブルシューティングについては、以下を参照してください。 [無効なエフェメリスのトラブルシューティング](#)

### 例: S3 バケットから OEM エフェメリスデータをアップロードする

バケットとオブジェクトキーを指すことで、S3 バケットから直接 OEM エフェメリスファイルをアップロードすることもできます。AWS Ground Station はユーザーに代わってオブジェクトを取得します。の保管中のデータの暗号化の詳細については AWS Ground Station、「」を参照してくださいの [保管中のデータ暗号化 AWS Ground Station](#)。

以下は、S3 バケットから OEM エフェメリスファイルをアップロードする例です。

```
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']}")
```

以下は、サンプルコードの前のブロックでアップロードされた OEM エフェメリスに対して呼び出される [DescribeEphemeris](#) アクションから返されたデータの例です。

```
{
  "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"
      }
    }
  }
}
```

## 方位高度エフェメリスデータを提供する

### Important

方位高度エフェメリス機能は、現在プレビュー状態であり、明示的なオンボーディングが必要です。

方位標高エフェメリス機能は、事前に決定された特殊なユースケースの数に限り、厳格なアクセスコントロール下にあります。アクセスは、お客様が用意した標準のエフェメリス機能よりも大幅に制限されます。承認されたユースケースとアクセスリクエストプロセスの詳細については、 から AWS サポート チケットを開いてください [AWS Support Center Console](#)。当社のチームが、特殊なユースケースの承認プロセスを案内します。

## 概要:

方位高度エフェメリスは、衛星軌道情報を提供せずにアンテナの方向を直接指定する方法を提供します。衛星の軌道を記述するエフェメリスデータをアップロードする代わりに、コンタクト全体でどこを指すかをアンテナに正確に指示する、時間タグ付けされた方位角と標高角度を提供します。

AWS Ground Station はエフェメリスを [個別の使用状況データ](#) として扱います。このオプション機能を使用する場合、AWS はエフェメリスデータを使用してトラブルシューティングサポートを提供します。

このアプローチは、以下のシナリオで特に役立ちます。

- 早期オペレーションのサポート: 正確な軌道データが利用できない場合、または軌道パラメータが急速に変化している場合、起動フェーズと早期軌道フェーズ (LEOP) 中。
- カスタムポイントパターン: アンテナテストまたは非標準オペレーションに特定のポイントシーケンスを実装します。

### Note

方位高度エフェメリスを使用する場合、衛星 ARN は問い合わせ予約リクエストから省略できます。衛星 ARN を省略しない場合でも、コンタクトデータの一部として含まれますが、方位高度エフェメリスは、エフェメリス優先度解決を実行するのではなく、アンテナポイントに使用されます。方位高度エフェメリスは、特定の地上局に関連付けられ、その場所のアンテナ指向方向を定義します。

## 方位高度エフェメリスデータ形式

方位高度エフェメリスデータは、時間タグ付けされた方位値と、セグメントに整理された高度値で構成されます。各セグメントには、特定の時間範囲をカバーする一連の方位角と角度が含まれます。

方位高度エフェメリスデータの主なコンポーネントは次のとおりです。

- Ground Station: この方位高度エフェメリスを使用する特定の地上局。
- Angle Unit: 角度の測定単位 (DEGREE\_ANGLE または RADIAN)。
- セグメント: 方位角と標高角度の 1 つ以上の時間制限付きコレクション。
- タイムタグ付き角度: 関連するタイムスタンプを持つ個々の方位値と標高値。

各セグメントには以下が必要です。

- リファレンスエポック (セグメントの基本時間)
- 有効な時間範囲 (セグメントの開始時刻と終了時刻)
- 少なくとも 5 つの時間タグ付けされた方位/高さのペア

方位標高の制約:

- 方位度:  $-180^{\circ} \sim 360^{\circ}$
- ラジアンでの方位数:  $-\pi \sim 2\pi$
- 角度での高さ:  $-90^{\circ} \sim 90^{\circ}$
- ラジアンでの上昇:  $-\pi/2$  から  $\pi/2$
- 時間値は各セグメント内で昇順である必要があります
- セグメントは時間的に重複してはいけません

詳細については、[CreateEphemeris](#) API ドキュメントと [TimeAzEl](#) データ型を参照してください。

## 方位高度エフェメリスの作成

方位高度エフェメリスは、同じ [CreateEphemeris](#) API アクションを使用して作成されますが、`azEl` エフェメリスタイプを使用します。TLE と OEM エフェメリスの主な違いは次のとおりです。

- `groundStation` パラメータを指定する必要があります
- `satelliteId` パラメータはリクエストから省略する必要があります
- 優先度設定は適用されません (各方位高度エフェメリスは地上局に固有です)
- 4 次 Lagrange 補間をサポートするには、各セグメントに少なくとも 5 つの方位/上昇ポイントが含まれている必要があります

- その他の制限と要件については、[CreateEphemeris](#) API ドキュメントを参照してください。

エフェメリスがアップロードされると、エフェメリスが VALIDATING に設定されて非同期ワークフローが開始されることに注意してください。このワークフローでは、エフェメリスを検証し、そのエフェメリスを基に潜在的なコンタクトを生成します。エフェメリスは、このワークフローに合格し、そのステータスが になった後の問い合わせにのみ使用されますENABLED。 [DescribeEphemeris](#) をポーリングしてエフェメリスステータスを確認するか、CloudWatch イベントを使用してエフェメリスのステータスの変化を追跡する必要があります。

無効なエフェメリスのトラブルシューティングについては、以下を参照してください。 [無効なエフェメリスのトラブルシューティング](#)

### 例: API を使用して方位高度エフェメリスを作成する

次の例は、AWS SDK for Python (Boto3) を使用して方位高度エフェメリスを作成する方法を示しています。

```
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']}")

```

この例では、以下のようにになっています：

- 方位高度データは「オハイオ 1」地上局に関連付けられています
- 角度は度数で指定します
- セグメントは 15 分間を対象とします。
- dt 値は参照エポックからのアトミック秒オフセットです
- 6 つの方位/標高ペアが用意されています (最小は 5)

### 例: S3 から方位標高データをアップロードする

大規模なデータセットの場合、S3 バケットから方位高度データをアップロードできます。

```

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 オブジェクトには、直接アップロードの例に示すのと同じ形式の方位標高データを含む JSON 構造が含まれている必要があります。

## 方位高度エフェメリスを使用したコンタクトの予約

方位高度エフェメリスを使用してコンタクトを予約する場合、プロセスは TLE および OEM エフェメリスとは異なります。

1. [CreateEphemeris](#) を使用して方位高度エフェメリスを作成する
2. エフェメリスが ENABLED ステータスになるまで待ちます。
3. [ReserveContact](#) と追跡オーバーライドを使用して問い合わせを予約する

方位高度エフェメリスとのコンタクトを予約する例:

```
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

方位角エフェメリスを含むコンタクトを予約するときは、`satelliteArn`パラメータを省略できます。アンテナは、接触中に指定された方位角と標高角度に従います。

## 利用可能な連絡先の一覧表示

方位高度エフェメリスを使用する場合、[ListContacts](#) API には特定のパラメータが必要です。

- `satelliteArn` パラメータはリクエストから省略できます
- 使用するエフェメリスを指定するには、方位高度エフェメリス ID を持つ `ephemeris` パラメータを指定する必要があります
- 使用可能なコンタクトウィンドウは、指定された方位角と標高角度が、リクエストされた地上局の[サイトマスク](#)を超えると表示されます。
- `groundStation` と `missionProfileArn` を指定する必要があります。

方位高度エフェメリスを作成し、それで利用可能な連絡先を一覧表示する例:

```
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

コンタクトを一覧表示するとき、アジマス標高 ID を持つ ephemeris パラメータを指定して、コンタクトウィンドウの決定に使用するアジマス標高エフェメリスを指定する必要があります。satelliteArn が含まれている場合、問い合わせデータに関連付けられますが、方位高度エフェメリスは、エフェメリスの優先度解決を実行するのではなく、アンテナポイントに使用されます。

## カスタムエフェメリスを使用してコンタクトを予約する

### 概要:

カスタムエフェメリス (TLE、OEM、または方位標高) を使用する場合は、[ReserveContact](#) API を使用してコンタクトを予約できます。このセクションでは、コンタクトを予約するための 2 つの一般的なワークフローと、コンタクトスケジューリングを成功させるための重要な考慮事項について説明します。

AWS Ground Station アンテナは、複数のお客様間で共有されるリソースです。つまり、問い合わせを一覧表示するとき問い合わせウィンドウが使用可能になっても、別の顧客が事前に予約する場合があります。したがって、予約後に問い合わせが SCHEDULED 状態になったことを確認し、問い合わせの状態の変化を適切にモニタリングすることが重要です。

### Important

方位高度エフェメリスの場合、satelliteArn パラメータを ReserveContact リクエストから省略できます。エフェメリス ID trackingOverrides を指定する必要があります。TLE および OEM エフェメリスの場合は、引き続き を指定する必要があります satelliteArn。

## 問い合わせ予約ワークフロー

カスタムエフェメリスでコンタクトを予約するには、主に 2 つのワークフローがあります。

1. List-then-reserve ワークフロー: [ListContacts](#) を使用して利用可能な問い合わせウィンドウを最初に一覧表示し、次に特定のウィンドウを選択して予約します。このアプローチは、選択する前に利用可能なすべての機会を確認する場合に便利です。
2. 直接予約ワークフロー: 利用可能な連絡先を最初に一覧表示することなく、特定の時間枠の連絡先を直接予約します。このアプローチは、希望するコンタクト時間が既にわかっている場合や、事前に決められたスケジュールで作業している場合に便利です。

どちらのワークフローも有効であり、選択は運用要件によって異なります。以下のセクションでは、各アプローチの例を示します。

## ワークフロー 1: 利用可能な連絡先を一覧表示し、予約する

このワークフローは、最初に利用可能な問い合わせウィンドウをクエリし、次に特定のウィンドウを予約します。これは、選択する前に利用可能なすべての機会を表示する場合に便利です。

### 例: 方位高度エフェメリスを使用した一覧表示と予約

```
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 = ephemeras_response["ephemerisId"]
print(f"Created ephemeras: {ephemeris_id}")

# Wait for ephemeras to become ENABLED
while True:
    status = ground_station_client.describe_ephemeras(ephemerisId=ephemeris_id)[
        "status"
    ]
    if status == "ENABLED":
        print("Ephemeras is ENABLED")
        break
    elif status in ["INVALID", "ERROR"]:
        raise RuntimeError(f"Ephemeras 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 ephemeras
    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"],
    ephemeras={"azEl": {"id": ephemeras_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")
```

## 例: TLE エフェメリスを使用した一覧表示と予約

```
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")
```

## ワークフロー 2: 直接連絡の予約

このワークフローは、利用可能なウィンドウを最初に一覧表示することなく、問い合わせを直接予約します。このアプローチは、希望する問い合わせ時間がすでにわかっている場合や、自動スケジューリングを実装している場合に便利です。

### 例: 方位高度エフェメリスを使用した直接予約

```
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']}")
```

## 例: TLE エフェメリスによる直接予約

```
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']}")
```

## 問い合わせ状態の変更のモニタリング

コンタクトを予約したら、その状態をモニタリングして、問題が正常に に移行SCHEDULEDし、通知されるようにすることが重要です。 は、すべてのコンタクト状態の変更について Amazon EventBridge にイベント AWS Ground Station を発行します。

問い合わせの状態は次のライフサイクルに従います。

- SCHEDULING - コンタクトはスケジューリングのために処理中です
- SCHEDULED - 問い合わせは正常にスケジュールされ、実行されます
- FAILED\_TO\_SCHEDULE - コンタクトをスケジュールできませんでした (終了状態)

問い合わせの状態とライフサイクルの詳細については、「」を参照してください[問い合わせのライフサイクルを理解する](#)。

## EventBridge を使用した問い合わせ状態モニタリングの実装

問い合わせの状態の変化をリアルタイムでモニタリングするには、Ground Station の問い合わせの状態が変わるたびに Lambda 関数をトリガーする Amazon EventBridge ルールを設定できます。このアプローチは、問い合わせステータスをポーリングするよりも効率的でスケーラブルです。

### 実装手順

1. 問い合わせ状態変更イベントを処理する Lambda 関数を作成する
2. Ground Station の問い合わせ状態変更イベントに一致する EventBridge ルールを作成する
3. ルールのターゲットとして Lambda 関数を追加する

### Lambda 関数ハンドラーの例

問い合わせ状態の変更イベントを処理する Lambda 関数の完全な例については、AquaSnppJpssTerraDigIF.yml CloudFormation テンプレートの GroundStationCloudWatchEventHandlerLambdaリソースを参照してください。このテンプレートは、AWS Ground Station カスタマーオンボーディング Amazon S3 バケットで使用できま

す。このテンプレートにアクセスする手順については、データフローエンドポイントの例の[まとめ](#)「」セクションを参照してください。

## EventBridge ルール設定

EventBridge ルールは、次のイベントパターンを使用して、Ground Station のすべての問い合わせ状態の変化と一致する必要があります。

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

特定の状態 (障害など) のみをフィルタリングするには、詳細フィルターを追加できます。

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

Lambda ターゲットを使用して EventBridge ルールを作成する詳細な手順については、「Amazon EventBridge ユーザーガイド」の[「イベントに対応するルールの作成」](#)を参照してください。

## 自動化のための EventBridge ルールの設定

EventBridge ルールを作成して、問い合わせ状態の変更に自動的に対応できます。例えば、次のようになります。

- 問い合わせがスケジュールに失敗したときに通知を送信する
- 問い合わせが入ったときに Lambda 関数をトリガーしてリソースを準備する PREPASS
- 監査目的での問い合わせ完了のログ記録

AWS Ground Station イベントの EventBridge ルールの設定の詳細については、「」を参照してください [イベント AWS Ground Station による自動化](#)。

## ベストプラクティスと考慮事項

### スケジュールの競合の処理

AWS Ground Station アンテナは共有リソースであるため、に表示される問い合わせウィンドウ ListContacts は、予約する前に別のお客様が予約する場合があります。これを処理するには:

1. 予約後に常に問い合わせステータスを確認する
2. 代替時間枠を使用して再試行ロジックを実装する
3. 可能な場合は、事前に連絡先を予約することを検討してください。
4. EventBridge イベントを使用して FAILED\_TO\_SCHEDULE 状態をモニタリングする

### エフェメリス検証のタイミング

エフェメリスを使用してコンタクトを予約する前に、エフェメリスが ENABLED 状態になっている必要があることに注意してください。検証プロセスは通常、エフェメリスのタイプとサイズに応じて数秒から数分かかります。コンタクトを予約する前に、必ずエフェメリスのステータスを確認してください。

### 問い合わせタイミングに関する考慮事項

カスタムエフェメリスを使用する場合:

- エフェメリスが問い合わせ期間全体をカバーしていることを確認します。
- 方位高度エフェメリスの場合、コンタクト全体を通して角度がアンテナを [サイトマスク](#) の上に維持していることを確認します。
- 将来のコンタクトをスケジュールするときにエフェメリスの有効期限を考慮する

### エフェメリスタイプ別の API の違い

ReserveContact API の動作はエフェメリスタイプによって異なります。

エフェメリスタイプ	satelliteArn 必須	trackingOverrides 必須
TLE	はい	いいえ (オプション)

エフェメリスタイプ	satelliteArn 必須	trackingOverrides 必須
OEM	はい	いいえ (オプション)
方位高度	いいえ (オプション)	はい

## どのエフェメリスが使用されているかを理解する

エフェメリスには、優先度、有効期限、有効フラグがあります。これらを組み合わせることで、問い合わせ中の追跡に使用されるエフェメリスが決まります。

### TLE および OEM エフェメリス

OEM および TLE エフェメリスの場合、衛星ごとにアクティブにできるエフェメリスは 1 つだけです。使用されるエフェメリスは、有効期限が今後のもので優先度が最も高く、有効になっているエフェメリスです。優先度の値が大きいほど、優先度が高いことを示します。[ListContacts](#) によって返される利用可能なコンタクト時間は、このエフェメリスに基づいています。複数の ENABLED エフェメリスの優先度が同じ場合は、最後に作成または更新されたエフェメリスが使用されます。

#### Note

AWS Ground Station には、ENABLED お客様が用意した衛星あたりのエフェメリスの数に対するサービスクォータがあります ([Service Quotas](#) を参照)。このクォータに達した後にエフェメリスデータをアップロードするには、お客様が作成したエフェメリスの優先度が最も低い/最も早いものを削除 ([DeleteEphemeris](#) を使用) または無効化 ([UpdateEphemeris](#) を使用) します。

エフェメリスが作成されていない場合、または ENABLED ステータスを持つエフェメリスがない場合、AWS Ground Station は利用可能な場合は衛星にデフォルトのエフェメリス ([Space-Track](#) から) を使用します。このデフォルトのエフェメリスの優先度は 0 です。

### 方位高度エフェメリス

方位高度エフェメリスは、OEM および TLE エフェメリスとは動作が異なります。各方位高度エフェメリスは、特定の地上局に関連付けられており、優先順位はありません。方位高度エフェメリスを使用してコンタクトを予約する場合、`trackingOverrides` パラメータを使用して使用する方位高度エフェメリスを明示的に指定します。

## 方位高度エフェメリスの主な違い:

- 優先度システムなし - 問い合わせごとにエフェメリスを明示的に選択します
- 地上局固有 - 各エフェメリスは特定の地上局に関連付けられます
- 自動フォールバックなし - 指定されたエフェメリスが使用できない場合、問い合わせは失敗します

### Note

方位高度エフェメリスは OEM および TLE エフェメリスと競合しません。これらは問い合わせを予約するときに明示的に選択され、追跡オーバーライドが指定されている場合にのみ使用されます。

## 以前にスケジュールされたコンタクトに対する新しいエフェメリスの影響

[DescribeContact API](#) を使用して、アクティブな可視性時間を返すことで、以前にスケジュールされたコンタクトに対する新しいエフェメリスの影響を表示します。

OEM および TLE エフェメリスの場合、新しいエフェメリスをアップロードする前にスケジュールされたコンタクトは、最初にスケジュールされたコンタクト時間を保持し、アンテナ追跡はアクティブなエフェメリスを使用します。アクティブなエフェメリスに基づいて宇宙機の位置が以前のエフェメリスと大きく異なる場合、宇宙機が送受信サイトマスクの外部で動作するため、アンテナとの衛星接触時間が短くなる可能性があります。したがって、以前のエフェメリスと大きく異なる新しいエフェメリスをアップロードした後、今後の連絡先をキャンセルして再スケジュールすることをお勧めします。

[DescribeContact API](#) を使用すると、スケジュールされたコンタクトと、endTime返された startTime および visibilityStartTime visibilityEndTime を比較することで、宇宙機が送受信サイトマスクの外部で動作するために、将来のコンタクトの使用できない部分を判断できます。visibilityStartTime visibilityEndTime。今後の問い合わせをキャンセルして再スケジュールすることを選択した場合、問い合わせ時間範囲は可視性時間範囲から 30 秒を超えて外れてはなりません (複数可)。キャンセルされた問い合わせは、問い合わせ時刻に近づきすぎるとコストが発生する可能性があります。キャンセルされたコンタクトの詳細については、「[Ground Station FAQ](#)」を参照してください。

方位高度エフェメリスの場合、スケジュールされたコンタクトは、コンタクトが予約されたときに選択された特定のエフェメリスを使用します。スケジュールされたコンタクトの方位高度データを更新する必要がある場合は、コンタクトをキャンセルして新しいエフェメリスで再スケジュールできます。

## 衛星の現在のエフェメリスを取得する

特定の衛星 AWS Ground Station に対して が現在使用しているエフェメリスは、[GetSatellite](#) または [ListSatellites](#) アクションを呼び出すことで取得できます。これらのメソッドはいずれも、現在使用中のエフェメリスのメタデータを返します。このエフェメリスメタデータは、 にアップロードされたカスタムエフェメリス AWS Ground Station とデフォルトのエフェメリスでは異なります。

### Note

方位高度エフェメリスは衛星に関連付けられていないため、[GetSatellite](#) または [ListSatellites](#) によって返されません。方位高度エフェメリスに関する情報を取得するには、特定のエフェメリス ID で [DescribeEphemeris](#) API を使用するか、[ListEphemerides](#) を使用してアカウントで使用可能なすべてのエフェメリスを表示します。

デフォルトのエフェメリスには、`source` と `epoch` のフィールドのみが含まれます。`epoch` は、[Space-Track](#) からプルされた [2 行の要素セット](#) の [エポック](#) であり、現在衛星の軌道の計算に使用されています。

カスタムエフェメリスは、`source` 値が `CUSTOMER_PROVIDED` となり、`ephemerisId` フィールドには一意の識別子が含まれます。この一意の識別子は、[DescribeEphemeris](#) アクションを介してエフェメリスをクエリするために使用できます。[CreateEphemeris](#) アクション AWS Ground Station を介してへのアップロード中にエフェメリスに名前が割り当てられた場合、オプションの `name` フィールドが返されます。

エフェメリスは によって動的に更新 AWS Ground Station されるため、返されるデータは API の呼び出し時に使用されているエフェメリスのスナップショットにすぎません。

## デフォルトのエフェメリスを使用した衛星の [GetSatellite](#) リターンの例

```
{
  "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
}
```

## カスタムエフェメリスを使用した衛星の [GetSatellite](#) の例

```
{
  "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"
  }
}
```

## 方位高度エフェメリスの一覧表示

方位高度エフェメリスは衛星に関連付けられていないため、異なる APIs を使用してそれらに関する情報を検出および取得する必要があります。

1. [ListEphemerides](#) を使用して、方位高度エフェメリスを含む、アカウント内のすべてのエフェメリスを一覧表示します。ステータスとエフェメリスタイプでフィルタリングできます。
2. 特定のエフェメリス ID で [DescribeEphemeris](#) を使用して、方位高度エフェメリスに関する詳細情報を取得します。
3. 特定の問い合わせ ID で [DescribeContact](#) を使用して、問い合わせに使用されるエフェメリスに関する詳細情報を取得します。

方位高度エフェメリスを含む [ListEphemerides](#) レスポンスの例:

```
{
```

```
"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](#) レスポンスでは、方位高度エフェメリスには `groundStation` フィールドではなく `satelliteId` フィールドがあるため、識別が容易になります。

## デフォルトのエフェメリスデータに戻す

カスタムエフェメリスデータをアップロードすると、その特定の衛星で AWS Ground Station が使用するデフォルトのエフェメリスが上書きされます。現在有効で有効期限が切れていないエフェメリスが使用できなくなるまで、はデフォルトのエフェメリスを再度使用 AWS Ground Station しません。AWS Ground Station また、は、その有効期限を過ぎたデフォルトのエフェメリスが使用可能であっても、現在のお客様提供のエフェメリスの有効期限を過ぎたコンタクトを一覧表示しません。

#### Note

方位高度エフェメリスにはデフォルト値がなく、衛星エフェメリスを上書きしません。これらは、`trackingOverrides` パラメータを使用してコンタクトを予約するときに明示的に選択されます。方位高度エフェメリスを使用しない場合は、追跡オーバーライドを指定せずにコンタクトを予約するだけで、システムは代わりにアクティブな衛星エフェメリスを使用します。

## TLE エフェメリスと OEM エフェメリスの元に戻す

衛星のデフォルトの [Space-Track](#) エフェメリスに戻すには、次のいずれかを実行する必要があります。

- お客様が用意したエフェメリスをすべて削除 ([DeleteEphemeris](#) を使用) または無効化 ([UpdateEphemeris](#) を使用) します。 [ListEphemerides](#) を使用して、お客様が用意した衛星のエフェメリスを一覧表示できます。
- 既存のお客様提供のエフェメリスがすべて期限切れになるまで待ちます。

[GetSatellite](#) を呼び出して、衛星の現在のエフェメリスsourceの がであることを確認しますSPACE\_TRACK。デフォルトのエフェメリスの詳細については、[デフォルトのエフェメリスデータ](#)「」を参照してください。

## 方位高度エフェメリスの管理

方位高度エフェメリスはコンタクトごとに明示的に選択され、衛星に関連付けられていないため、「デフォルトに戻す」という概念はありません。代わりに、方位高度エフェメリスを次のように管理できます。

- 方位標高エフェメリスの使用を停止するには: を指定trackingOverridesおよび指定せずに、単に新しい連絡先を予約しますsatelliteArn。コンタクトは、代わりに指定された衛星にアクティブなエフェメリスを使用します。
- 未使用の方位高度エフェメリスを削除するには: [DeleteEphemeris](#) を使用して、不要になった方位高度エフェメリスを削除します。スケジュールされたコンタクトで現在使用されているエフェメリスは削除できません。

アカウント内のすべての方位高度エフェメリスを一覧表示するには、[ListEphemerides](#) を使用します。方位高度エフェメリスは、ephemerisTypeフィールドによって、またはレスポンスのgroundStation フィールドではなく satelliteIdフィールドの存在によって識別できます。

# データフローの使用

AWS Ground Station はノードとエッジの関係を使用してデータフローを構築し、データのストリーム処理を有効にします。各ノードは、予想される処理を記述する設定で表されます。この概念を説明するには、 から antenna-downlink へのデータフローを検討してください s3-recording。 antenna-downlink ノードは、設定で定義されたパラメータごとの無線周波数スペクトルのアナログからデジタルへの変換を表します。は、受信データを受信して S3 バケットに保存するコンピューティングノード s3-recording を表します。結果のデータフローは、仕様に基づいて、デジタル化された RF データを S3 バケットに非同期で配信します。

ミッションプロファイル内では、ニーズに合わせて多くのデータフローを作成できます。以下のセクションでは、 で使用する他の AWS リソースを設定する方法について説明 AWS Ground Station し、データフローを構築するための推奨事項を提供します。ソースノードまたは宛先ノードと見なされるかどうかなど、各ノードの動作の詳細については、「」を参照してください [AWS Ground Station 設定を使用する](#)。

## トピック

- [AWS Ground Station データプレーンインターフェイス](#)
- [クロスリージョンデータ配信を使用する](#)
- [Amazon S3 のセットアップと設定](#)
- [Amazon VPC のセットアップと設定](#)
- [Amazon EC2 のセットアップと設定](#)

## AWS Ground Station データプレーンインターフェイス

選択したデータフローの結果のデータ構造は、データフローのソースによって異なります。これらの形式の詳細は、衛星のオンボーディング中に提供されます。以下は、各タイプのデータフローに使用される形式をまとめたものです。

- アンテナダウンリンク
  - (帯域幅 less-than-or-equal-to) のデータは、 [VITA-49 Signal Data/IP Format](#) パケットとして配信されます。 40MHz
  - (帯域幅が 40MHz より大きい) データは、 AWS Ground Station クラス 2 パケットとして配信されます。
- antenna-downlink-demod-decode

- データは、復調/復号化されたデータ/IP 形式のパケットとして配信されます。
- アンテナアップリンク
  - データは [VITA-49 Signal Data/IP Format](#) パケットとして配信する必要があります。
- antenna-uplink-echo
  - データは [VITA-49 Signal Data/IP Format](#) パケットとして配信されます。

## クロスリージョンデータ配信を使用する

AWS Ground Station クロスリージョンデータ配信機能により、アンテナから AWS Ground Station サポートされている任意の AWS リージョンにデータを柔軟に送信できます。つまり、インフラストラクチャを単一の AWS リージョンに維持し、オンボーディング先の任意の で連絡先をスケジュール [AWS Ground Station ロケーション](#) できます。

Amazon S3 バケットで問い合わせデータを受信すると、AWS Ground Station はすべての配信面を管理します。

Amazon EC2 インスタンスへのクロスリージョンデータ配信 (AWS Ground Station エージェントまたはデータフローエンドポイントを使用) を使用するには、データフローエンドポイントを現在の AWS リージョンに作成し、dataflow-endpoint-config で同じリージョンを指定する必要があります。AWS Ground Station は、データクロスリージョンの配信を管理します。

## Amazon S3 のセットアップと設定

Amazon S3 バケットを使用して、 を使用してダウンロードシグナルを受信できます AWS Ground Station。送信先 s3-recording-config を作成するには、Amazon S3 バケットと、バケット AWS Ground Station へのファイルの書き込みを に許可する IAM ロールを指定する必要があります。

Amazon S3 バケット、IAM ロール、または AWS Ground Station 設定の作成に関する制限 [Amazon S3 録画設定](#) については、「」を参照してください。

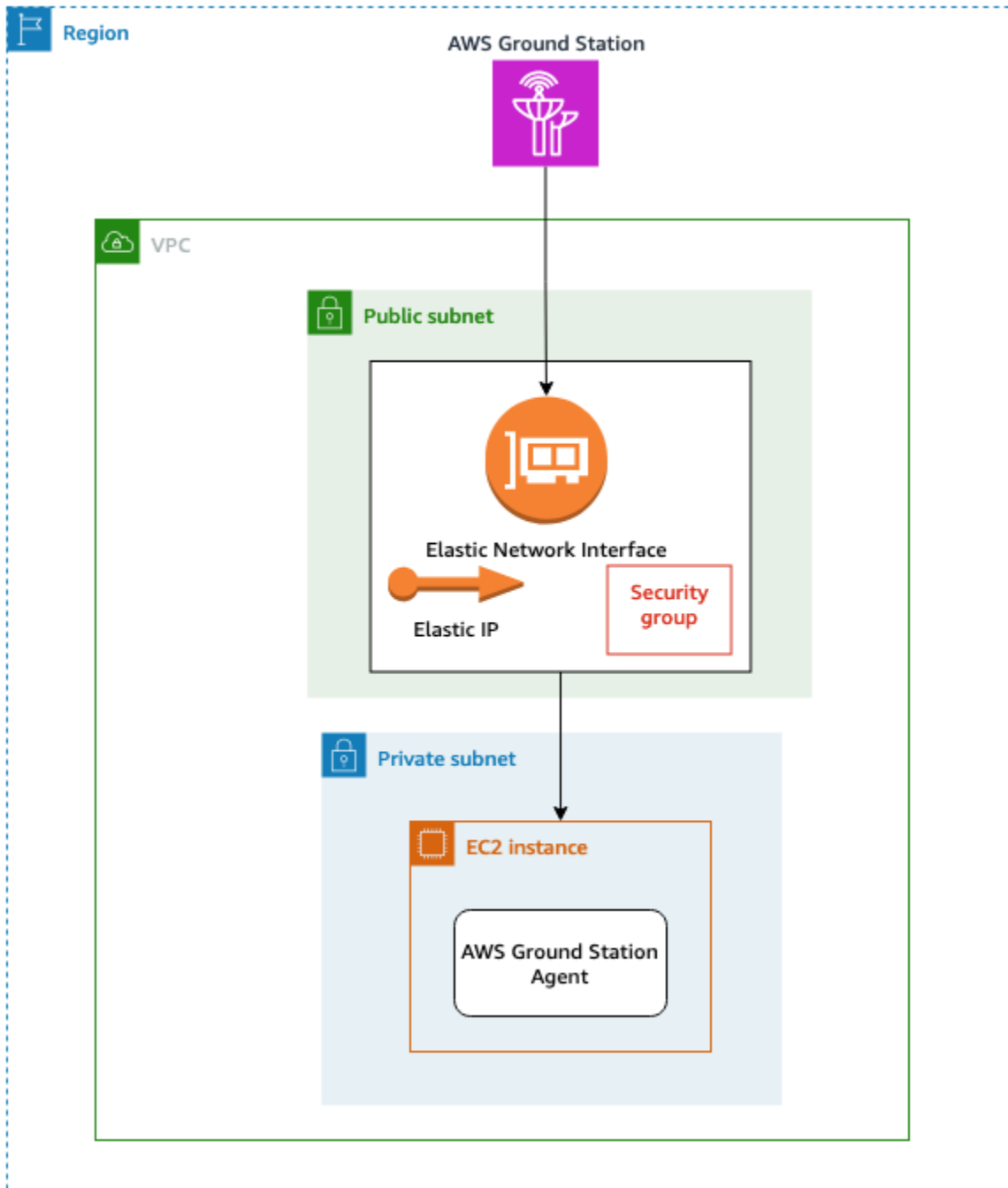
## Amazon VPC のセットアップと設定

VPC をセットアップするための完全なガイドは、このガイドの範囲外です。詳細については、「[Amazon VPC ユーザーガイド](#)」を参照してください。

このセクションでは、Amazon EC2 とデータフローエンドポイントが VPC 内に存在する方法について説明します。 は、特定のデータフローに対して複数の配信ポイントをサポート AWS Ground

Station していません。各データフローは 1 つの EC2 レシーバーに終了することが予想されます。単一の EC2 レシーバーを想定しているため、設定はマルチ AZ 冗長ではありません。VPC を使用する完全な例については、「」を参照してください [ミッションプロファイル設定の例](#)。

## AWS Ground Station エージェントによる VPC 設定



衛星データは、アンテナに近接する AWS Ground Station エージェントインスタンスに提供されます。AWS Ground Station エージェントはストライピングし、指定した AWS KMS キーを使用し

てデータを暗号化します。各ストライプは、AWS Network バックボーン全体のソースアンテナから [Amazon EC2 Elastic IP \(EIP\)](#) に送信されます。データは、アタッチされた Amazon EC2 Elastic Network Interface (ENI) を介して EC2 インスタンスに到着します。 [Amazon EC2](#) EC2 インスタンスで、インストールされた AWS Ground Station エージェントはデータを復号し、前方エラー修正 (FEC) を実行してドロップされたデータを復元し、セットアップで指定した IP とポートに転送します。

次のリストでは、エージェントの配信用に AWS Ground Station VPC を設定するとき、一意のセットアップに関する考慮事項について説明します。

セキュリティグループ - AWS Ground Station トラフィック専用のセキュリティグループを設定することをお勧めします。このセキュリティグループは、Dataflow Endpoint Group で指定したのと同じポート範囲で UDP 進入トラフィックを許可する必要があります。は AWS マネージドプレフィックスリスト AWS Ground Station を維持し、アクセス許可を AWS Ground Station IP アドレスのみに制限します。デプロイリージョンの [PrefixListId を置き換える方法の詳細については、「AWS マネージドプレフィックスリスト」](#) を参照してください。 PrefixListId

Elastic Network Interface (ENI) - 上記のセキュリティグループをこの ENI に関連付け、パブリックサブネットに配置する必要があります。

#### Note

ENI ごとにアタッチされるセキュリティグループの数のデフォルトのクォータは 5 です。これは最大 16 個の調整可能な制限です。 [「Amazon VPC クォータ」](#) を参照してください。

次の CloudFormation テンプレートは、このセクションで説明するインフラストラクチャを作成する方法を示しています。

#### *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."
```

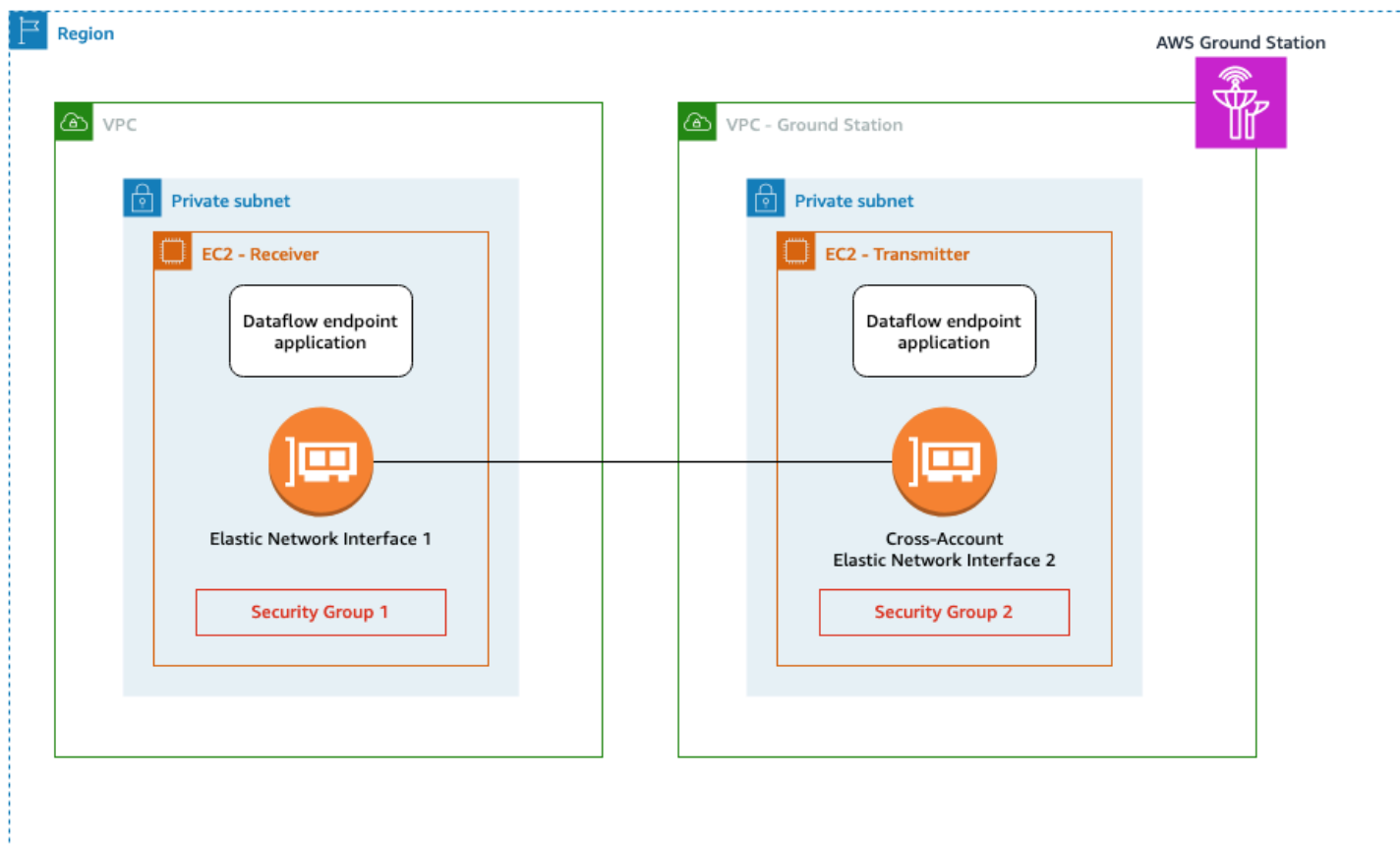
*InstanceNetworkInterface:*

```
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
```

## データフローエンドポイントを使用した VPC 設定



衛星データは、アンテナに近接するデータフローエンドポイントアプリケーションインスタンスに提供されます。その後、が所有する VPC からクロスアカウント [Amazon EC2 Elastic Network Interface \(ENI\)](#) を介してデータが送信されます AWS Ground Station。その後、データは Amazon EC2 インスタンスにアタッチされた ENI を介して Amazon EC2 インスタンスに到着します。インストールされたデータフローエンドポイントアプリケーションは、セットアップで指定した IP とポートに転送します。このフローの逆は、アップリンク接続で発生します。

次のリストでは、データフローエンドポイント配信用に VPC を設定するときに、固有の設定上の考慮事項について説明します。

### Note

ENI ごとにアタッチされるセキュリティグループの数のデフォルトのクォータは 5 です。これは最大 16 個の調整可能な制限です。[「Amazon VPC クォータ」](#) を参照してください。

IAM ロール - IAM ロールはデータフローエンドポイントの一部であり、図には示されていません。クロスアカウント ENI を作成して AWS Ground Station Amazon EC2 インスタンスにアタッチするために使用される IAM ロール。

セキュリティグループ 1 - このセキュリティグループは、アカウントの Amazon EC2 インスタンスに関連付けられる ENI にアタッチされます。dataflow-endpoint-groupグループで指定されたポートで、セキュリティグループ 2 からの UDP トラフィックを許可する必要があります。

Elastic Network Interface (ENI) 1 - セキュリティグループ 1 をこの ENI に関連付け、サブネットに配置する必要があります。

サブネット - アカウント内の Amazon EC2 インスタンスのデータフローごとに少なくとも 1 つの使用可能な IP アドレスがあることを確認する必要があります。サブネットのサイズ設定の詳細については、「[サブネット CIDR ブロック](#)」を参照してください。

セキュリティグループ 2 - このセキュリティグループは Dataflow エンドポイントで参照されます。このセキュリティグループは、AWS Ground Station がアカウントにデータを配置するために使用する ENI にアタッチされます。

リージョン - クロスリージョン接続でサポートされているリージョンの詳細については、「」を参照してください。[クロスリージョンデータ配信を使用する](#)。

次の CloudFormation テンプレートは、このセクションで説明するインフラストラクチャを作成する方法を示しています。

**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 のセットアップと設定

AWS Ground Station エージェントまたはデータフローエンドポイントを介して配信される VITA-49 Signal/IP データまたは VITA-49 Extension データ/IP の同期配信には、Amazon EC2 インスタンスを適切に設定する必要があります。特定のニーズに応じて、フロントエンド (FE) プロセッサまたは Software Defined Radio (SDR) を同じインスタンスで直接実行したり、追加の EC2 インスタンスを利用したりすることができます。FE または SDR の選択とインストールは、このユーザーガイドの範囲外です。特定のデータ形式の詳細については、「」を参照してください[AWS Ground Station データプレーンインターフェイス](#)。

サービス条件の詳細については、[AWS 「サービス条件」](#) を参照してください。

## 提供される一般的なソフトウェア

AWS Ground Station は、Amazon EC2 インスタンスのセットアップを容易にする一般的なソフトウェアを提供します。

### AWS Ground Station エージェント

AWS Ground Station エージェントは、デジタル中間周波数 (DigIF) ダウンリンクデータを受信し、以下を可能にする復号化されたデータを出力します。

- 40 MHz から 400 MHz の帯域幅までの DigIF ダウンリンク機能。
- AWS ネットワーク上のパブリック IP (AWS Elastic IP) への高レート、低ジッターの DigIF データ配信。
- 前方誤り訂正 (FEC) による信頼性の高いデータ配信。
- 暗号化にカスタマーマネージド AWS KMS キーを使用した安全なデータ配信。

詳細については、[AWS Ground Station 「エージェントユーザーガイド」](#) を参照してください。

### データフローエンドポイントアプリケーション

AWS Ground Station アンテナの場所と Amazon EC2 インスタンス間でデータを送受信 AWS Ground Station するために使用されるネットワークアプリケーション。データのアップリンクとダウンリンクに使用できます。

### Software Defined Radio (SDR)

衛星との通信に使用される信号を調整/復調するために使用できるソフトウェア定義無線 (SDR)。

### AWS Ground Station Amazon マシンイメージ (AMIs)

これらのインストールのビルドおよび設定時間を短縮するために、には事前設定された AMI AWS Ground Station も用意されています。AMIs データフローエンドポイントネットワークアプリケーションと Software defined radio (SDR) を備えた AMIs は、オンボーディングが完了した後にアカウントで利用できるようになります。これらは、プライベート Amazon EC2 コンソールで確認できます。 [AMIs](#) AWS Ground Station エージェントのある AMIs はパブリックであり、パブリック Amazon マシンイメージ (AMI) でグラウンドステーションを検索することで Amazon EC2 コンソールで確認できます。 [AMIs](#)

# テレメトリの操作

AWS Ground Station テレメトリは、衛星コンタクト中に AWS Ground Station アンテナからほぼリアルタイムのメトリクスを提供します。テレメトリデータを使用して、コンタクトパフォーマンスのモニタリング、異常の検出、衛星通信に関する情報に基づいた意思決定を行うことができます。

## テレメトリの仕組み

テレメトリを使用するには、テレメトリデータを AWS Ground Station 配信する場所を指定する `TelemetrySinkConfig` を設定します。次に、`telemetrySinkConfigArn` フィールドを使用して、この設定をミッションプロファイルに追加します。テレメトリが有効なミッションプロファイルを使用する問い合わせ中、はテレメトリデータをアカウントに AWS Ground Station ストリーミングします。

テレメトリ配信プロセスは次のように機能します。

1. テレメトリデータを受信するには、AWS アカウントに Kinesis Data Streams ストリームを作成します。ストリームは、連絡先をスケジュールするのと同じアカウントとリージョンで作成する必要があります。
2. ストリームにデータを書き込む AWS Ground Station アクセス許可を付与する IAM ロールを作成します。
3. ストリームと IAM ロールを参照する `TelemetrySinkConfig` を作成します。
4. `TelemetrySinkConfig` をミッションプロファイルに追加します。
5. 新しいテレメトリ対応ミッションプロファイルを使用して、連絡先を一覧表示および予約します。
6. このミッションプロファイルを使用する問い合わせ中、AWS Ground Station はテレメトリデータをほぼリアルタイムで Kinesis Data Streams ストリームにストリーミングします。
7. AWS サービスまたは独自のアプリケーションを使用して、ストリームのテレメトリデータを消費して処理します。

## 使用可能なテレメトリタイプ

AWS Ground Station は、問い合わせ中に次のテレメトリタイプを提供します。

**Note**

AWS Ground Station は、サポートされているテレメトリタイプの拡張に取り組んでいます

## ポイントテレメトリ

衛星コンタクト中のアンテナの向きに関する情報を提供します。このテレメトリタイプは常にコンタクト中に送信され、実際の方角とコマンドされた方角が含まれます。詳細については、「[ポイントテレメトリ](#)」を参照してください。

## テレメトリの追跡

アンテナ追跡ステータスと追跡エラーに関する情報を提供します。このテレメトリタイプは、追跡設定で自動追跡が有効になっている場合に送信されます。詳細については、「[テレメトリの追跡](#)」を参照してください。

## リージョナルな可用性

テレメトリは、が AWS Ground Station 動作するすべての AWS リージョンで使用できます。コンタクトの実行中、テレメトリは AWS Ground Station アンテナからコンタクトをスケジュールしたリージョンに配信され、クロスリージョンサポートが提供されます。

AWS Ground Station リージョンと地上局の場所の完全なリストについては、「」を参照してください [AWS Ground Station ロケーション](#)。

### トピック

- [テレメトリの設定](#)
- [テレメトリデータを理解する](#)

## テレメトリの設定

AWS Ground Station 連絡先のテレメトリを設定するには、次の手順に従います。この設定が完了すると、テレメトリ対応ミッションプロファイルを使用するコンタクト中に、テレメトリデータが Kinesis Data Streams ストリームに配信されます。Kinesis Data Streams の詳細については、[Kinesis Data Streams ユーザーガイド](#)を参照してください。

## ステップ 1: 前提条件 AWS となるリソースを作成する

次の CloudFormation スニペットは、テレメトリ配信の前提条件 AWS となるリソースを作成する方法を示しています。このスニペットは、Kinesis Data Streams ストリームと、ストリームにテレメトリデータを書き込む AWS Ground Station アクセス許可を付与する IAM ロールを作成します。

### *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
```

以下のリストでは、テレメトリ配信を設定する際の一意的設定に関する考慮事項を示します AWS Ground Station。

Kinesis Data Streams ストリーム - ストリームはオンデマンドキャパシティモードを使用し、スループットに基づいて自動的にスケールされます。これは、ほとんどのユースケースで推奨されま

す。ストリームは、データを 24 時間保持するように設定されています。デフォルトでは、ストリームは AWS マネージド暗号化を使用します。カスタマーマネージド暗号化を使用するには AWS Key Management Service、StreamEncryptionプロパティを追加し、IAM ロールポリシーを更新して `アクセスkms:GenerateDataKey` 許可を含めます。詳細については、[Amazon Kinesis Data Streams でのデータ保護](#) を参照してください。

**IAM ロール** - IAM ロールは、`groundstation.amazonaws.com` サービスプリンシパルがロールを引き受け、テレメトリデータを Kinesis Data Streams ストリームに書き込むことを許可します。ロールポリシーは `kinesis:DescribeStream`、ストリーム上の、`kinesis:PutRecord`、および `kinesis:PutRecords` アクションのアクセス許可を付与します。信頼ポリシーとロールポリシーの設定に関するガイダンス [テレメトリシンク設定](#) については、「」を参照してください。

**追加設定** - AWS Ground Station API コールに使用する IAM ユーザーまたはロールに `iam:PassRole` 許可を追加します。これにより、`TelemetrySinkConfig` の作成 AWS Ground Station 時にテレメトリロールを に渡すことができます。

### PassRole ポリシーの例

ロールポリシーを更新またはアタッチする方法の詳細については、IAM ユーザーガイドの「[IAM ポリシーの管理](#)」を参照してください。アクセス `iam:PassRole` 許可の詳細については、「[AWS サービスにロールを渡すアクセス許可をユーザーに付与する](#)」を参照してください。

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

## ステップ 2: TelemetrySinkConfig を作成する

AWS Ground Station がテレメトリデータを Kinesis Data Streams ストリームに配信する方法を定義する `TelemetrySinkConfig` を作成します。ステップ 1 の CloudFormation スタック出力からストリーム ARN とロール ARN を使用します。

**Note**

TelemetrySinkConfig を作成すると、AWS Ground Station はパーティションキーが の空のテストレコードを配信することで、Kinesis Data Streams ストリームへのアクセスを検証しますtest。

TelemetrySinkConfig の作成の詳細については、「」を参照してください[テレメトリシンク設定](#)。

## ステップ 3: ミッションプロファイルにテレメトリを追加する

ミッションプロファイルを作成します。ミッションプロファイルの作成の詳細については、「」を参照してください[AWS Ground Station ミッションプロファイルを使用する](#)。をミッションプロファイルに追加telemetrySinkConfigArnして、コンタクト中のテレメトリ配信を有効にします。ステップ 2 で作成した TelemetrySinkConfig の ARN を使用します。

## ステップ 4: コンタクトをスケジュールする

テレメトリが有効なミッションプロファイルを使用してコンタクトをスケジュールします。問い合わせ中、AWS Ground Station はテレメトリデータを Kinesis Data Streams ストリームにストリーミングします。

問い合わせ中に予想されること

- テレメトリ開始 - 問い合わせが開始されると、データがストリーミングを開始します。
- ほぼリアルタイムの配信 - テレメトリはほぼリアルタイムで Kinesis Data Streams ストリームに到着します。
- 問い合わせ期間 - データは問い合わせ全体を通じて継続されます。
- 自動停止 - コンタクトが終了すると、テレメトリはストリーミングを停止します。

### 配信のモニタリング

テレメトリ配信は、以下を使用してモニタリングできます。

- Kinesis Data Streams ストリームメトリクス - CloudWatch で受信レコードを確認します。詳細については、[Amazon Kinesis Data Streams のモニタリング](#)」を参照してください。
- アプリケーションログ - ストリームから消費するアプリケーションのデータ処理を検証します。

- Kinesis Data Viewer - Kinesis Data Streams ストリームコンソールを使用して、ストリームのサンプルレコードを表示します。

## 次の手順

セットアップが完了したら、次のことができます。

- テレメトリデータ形式と使用可能なテレメトリタイプについて説明します。「[テレメトリデータを理解する](#)」を参照してください。
- Kinesis Data Streams ストリームからテレメトリデータを処理するためのアプリケーションを構築します。詳細については、[Amazon Kinesis Data Streams のコンシューマーの構築](#)」を参照してください。
- CloudWatch およびその他の AWS サービスを使用してダッシュボードとアラートを作成します。
- 問題が発生した場合は、トラブルシューティングガイダンスを確認してください。「[テレメトリのトラブルシューティング](#)」を参照してください。

## テレメトリデータを理解する

テレメトリデータは Base64-encodedされた JSON レコードとして Kinesis Data Streams ストリームに配信されます。各レコードには、コンタクトに関するメタデータやサンプリングされたテレメトリ測定値など、衛星コンタクト中に収集された情報が含まれます。

### データ形式の概要

各テレメトリレコードには、次のコンポーネントが含まれています。

#### テレメトリのタイプとバージョン

テレメトリデータの特定のタイプとそのスキーマバージョンを識別します。これにより、さまざまなテレメトリタイプを適切に解析できます。スキーマのバージョンニングの詳細については、「[スキーマのバージョンニングと進化](#)」を参照してください。

#### スコープ ID

テレメトリの範囲の一意的識別子。これにより、テレメトリデータを特定の連絡先に関連付けることができます。

#### メタデータ

テレメトリに関するコンテキスト情報。

## データ

テレメトリタイプに固有のサンプリングされたテレメトリ測定値。

## パーティションキー

テレメトリレコードは、形式のパーティションキーを使用して Kinesis Data Streams ストリームに配信されます。

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

このパーティションキーにより、1つのコンタクトに対する特定のタイプのすべてのテレメトリが Kinesis Data Streams ストリーム内の同じシャードに配信され、そのコンタクトのテレメトリストリームの最適な順序付けが可能になります。

## ポイントテレメトリ

指向テレメトリは、衛星コンタクト中のアンテナの指向方向に関する情報を提供します。このテレメトリタイプは、常に問い合わせ中に送信されます。

### データフィールド

#### sampleTimestamp

テレメトリデータがサンプリングされた時刻。ISO-8601 形式では UTC、ミリ秒精度。

#### 方位

アンテナの実際の方位角を度数で表したものです。

#### 標高

アンテナの実際の上昇角度を度数で表します。

#### commandedAzimuth

角度単位のコマンドされた方位角。これは、アンテナが達成しようとしているターゲット方位角です。

#### commandedElevation

角度単位のコマンドされた標高角度。これは、アンテナが達成しようとしているターゲットの標高角度です。

**Note**

実際のアンテナ位置は、物理的な制限や接触中の機械的な遅延により、コマンドされた位置とは異なる場合があります。

## メタデータフィールド

## groundStation

地上局の名前（「オハイオ 1」など）。

## satelliteId

の衛星リソースの識別子 AWS Ground Station。

## contactId

問い合わせの識別子。

## JSON の例

```
{
  "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
  }
}
```

## テレメトリの追跡

追跡テレメトリは、アンテナ追跡ステータスと追跡エラーに関する情報を提供します。このテレメトリタイプは、追跡設定で自動追跡が有効になっているとき、およびアンテナが自動追跡をアクティブに使用しているときに送信されます。

### Note

TrackingConfig の autotrackパラメータが に設定されている場合REMOVED、追跡テレメトリは配信されません。設定の追跡の詳細については、「」を参照してください[追跡設定](#)。

### データフィールド

#### sampleTimestamp

テレメトリデータがサンプリングされた時刻。精度がミリ秒の ISO-8601 形式の UTC。

#### trackingStatus

アンテナの現在の追跡ステータス。指定できる値には、TRACKING、ACQUIRING、MASKED などがあります。

#### trackingErrorAzimuth

方位軸の追跡エラー。度数で測定されます。

#### trackingErrorElevation

高度軸の追跡エラー。度数で測定されます。

### Note

追跡エラー値は、信号強度を最大化するために自動追跡中 AWS Ground Station に適用されるエフェメリスベースのプログラムトラックの調整を表します。

### メタデータフィールド

追跡テレメトリには、ポイントテレメトリと同じメタデータフィールド

groundStation、satelliteId、および が含まれますcontactId。

### JSON の例

```
{
  "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 ストリームからのデータの読み取り

テレメトリデータは Kinesis Data Streams ストリームに配信され、標準のストリーム消費パターンを使用して消費できます。ストリームからデータを読み取るときは、次の考慮事項に注意してください。

### Base64 デコード

Kinesis Data Streams ストリームのデータは Base64-encoded されています。データを JSON として解析する前に、データをデコードする必要があります。詳細については、[Amazon Kinesis Data Streams の使用](#)」を参照してください。

### Kinesis Data Viewer の使用

テレメトリデータにすばやくアクセスするために、Kinesis Data Streams ストリームコンソールにはデータビューワー機能が用意されています。この機能を使用する場合:

- テレメトリ配信は、ストリーム内の任意のシャードに発生する可能性があります。
- デフォルトの開始位置は、シャード内の最新のレコードから読み取ります。
- 選択したシャードを調整し、「タイムスタンプ時」の開始位置を使用して、受信したレコードを表示する必要がある場合があります。

## Kinesis Client Library の使用

Kinesis Client Library (KCL) は、シャード管理、チェックポイント、ロードバランシングなど、Kinesis Data Streams ストリームからのデータの消費に関連する多くの複雑さを管理します。本番環境のテレメトリ消費アプリケーションには KCL を使用することをお勧めします。

詳細については、[「Kinesis Client Library を使用したコンシューマーの開発」](#)を参照してください。

### 使用のベストプラクティス

- レイテンシーの最小化 - 拡張ファンアウトを使用して Kinesis Data Streams ストリームから読み取ると、ポーリングに比べて専用のスループットとレイテンシーが低くなります。詳細については、[「拡張ファンアウトコンシューマーの開発」](#)を参照してください。
- 専用ストリーム - AWS Ground Station テレメトリ統合に専用の Kinesis Data Streams ストリームを使用します。ストリームを他のアプリケーションと共有すると、書き込みスループットの飽和やテレメトリ配信の失敗が発生する可能性があります。
- オンデマンドキャパシティ - Kinesis Data Streams ストリームをオンデマンドプロビジョニングモードでデプロイして、スループットに基づいてシャードを自動的にスケールリングできるようにします。
- スループットのモニタリング - CloudWatch メトリクスを使用してストリームのスロットリングをモニタリングします。詳細については、[Amazon Kinesis Data Streams のモニタリング](#)」を参照してください。

## スキーマのバージョンニングと進化

テレメトリスキーマは、時間の経過に伴う進化をサポートするようにバージョン管理されています。各レコードの `telemetryVersion` フィールドはスキーマバージョンを示します。

### スキーマの変更の処理

- 今後、新しいテレメトリタイプが導入される可能性があります。
- 既存のテレメトリタイプは、重大な変更を含む新しいバージョンを受け取る場合があります。
- アプリケーションは、不明なテレメトリタイプとバージョンに対して寛容である必要があります。
- `telemetryTypeAndVersion`、`telemetryType`、および `telemetryVersion` フィールドを解析して、各レコードの処理方法を決定します。

複数のスキーマバージョンを適切に処理できるバージョン対応ペイロードのシリアル化を実装し、新しいバージョンが導入されてもアプリケーションが引き続き機能できるようにすることをお勧めします。

## 連絡先の操作

コンソール、または任意の言語で AWS SDK を使用して、衛星データの入力、アンテナの位置の特定、通信 AWS Ground Station AWS CLI、選択した衛星のアンテナ時間のスケジュールを行うことができます。問い合わせ予約は、問い合わせ開始\* の 15 分前に確認、キャンセル、再スケジュールできます。さらに、リザーブド分料金モデルを使用している場合は、AWS Ground Station リザーブド分料金プランの詳細を表示できます。

AWS Ground Station は、クロスリージョンデータ配信をサポートしています。選択したミッションプロファイルの一部であるデータフローエンドポイント設定によって、データの配信先のリージョンが決まります。クロスリージョンデータ配信の使用の詳細については、「」を参照してください[クロスリージョンデータ配信を使用する](#)。

コンタクトをスケジュールするには、リソースを設定する必要があります。リソースを設定していない場合は、「」を参照してください[はじめに](#)。 [ReserveContact](#) が呼び出されると、はミッションプロファイルのスナップショット AWS Ground Station を取得し、問い合わせのライフサイクル全体で使用するリソースを設定します。 [UpdateMissionProfile](#) API と [UpdateConfig](#) APIs を使用したこれらのリソースへの変更は、更新前に予約された問い合わせに反映されません。スケジュール済みの問い合わせにリソースの変更を適用する必要がある場合は、まず [CancelContact](#) を使用して問い合わせをキャンセルしてから、[ReserveContact](#) を使用して再スケジュールする必要があります。

\* キャンセルされた問い合わせは、キャンセルが問い合わせ時刻に近すぎるとコストが発生する可能性があります。キャンセルされたコンタクトの詳細については、「[Ground Station FAQ](#)」を参照してください。

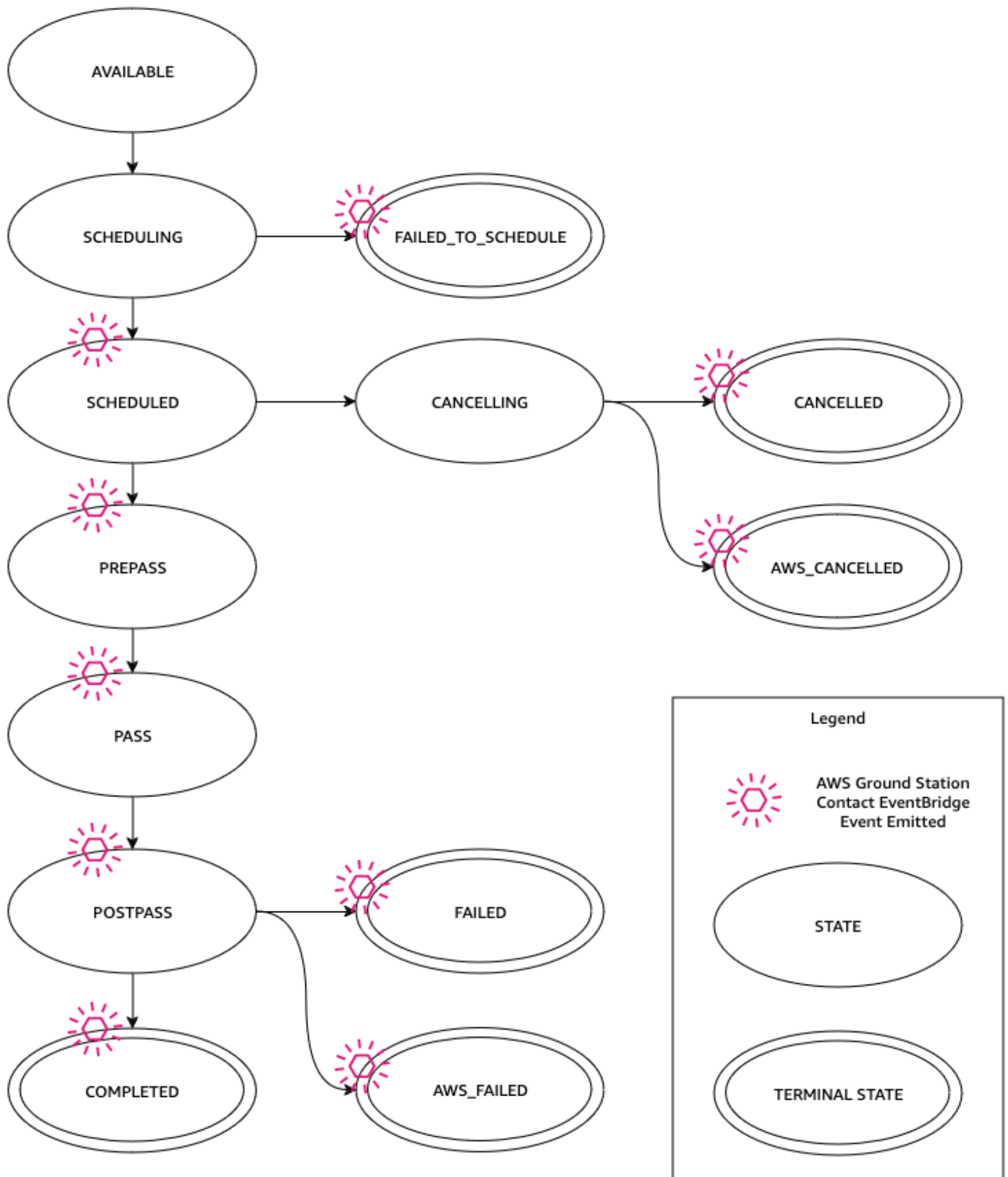
### トピック

- [問い合わせのライフサイクルを理解する](#)
- [問い合わせの請求を理解する](#)

## 問い合わせのライフサイクルを理解する

問い合わせのライフサイクルを理解することは、 の使用中にさまざまな問題を自動化およびトラブルシューティングするのに役立ちます AWS Ground Station。次の図は、 AWS Ground Station 問い合わせのライフサイクルと、ライフサイクル中に出力される Event Bridge Events を示しています。COMPLETED、FAILED、FAILED\_TO\_SCHEDULE、CANCELLED、AWS\_CANCELLED、および AWS\_FAILED は終了状態であることを注意してください。コンタクトは終了状態から移行しませ

ん。各ステータスが示す内容と [CancelContact](#) を使用して停止可能かキャンセル可能かの詳細については、[AWS Ground Station 問い合わせステータス「」](#)を参照してください。



## AWS Ground Station 問い合わせステータス

AWS Ground Station 問い合わせのステータスは、特定の時点でその問い合わせに何が起こっているかについてのインサイトを提供します。

### 問い合わせステータス

次の表は、問い合わせのステータスを示しています。

ステータス	説明	Terminal	キャンセル可能	停止可能
利用可能	問い合わせは予約できます。	いいえ	該当なし	該当なし
スケジュール	問い合わせはスケジューリング中です。	いいえ	あり	なし
SCHEDULED	問い合わせは正常にスケジュールされました。	いいえ	あり	なし
FAILED_TO_SCHEDULE	問い合わせはスケジュールに失敗しました。	はい	なし	いいえ
PREPASS	問い合わせは間もなく開始され、リソースは準備中です。	いいえ	あり	なし
PASS	コンタクトは現在実行中で、衛星は通信中です。	いいえ	なし	はい
POSTPASS	通信が完了し、使用されているリソースがクリーンアップされています。	いいえ	なし	いいえ
COMPLETED	問い合わせはエラーなしで完了しました。	はい	なし	いいえ
FAILED	リソース設定に問題があるため、問い合わせに失敗しました。	はい	なし	いいえ

ステータス	説明	Terminal	キャンセル可能	停止可能
AWS_FAILED	AWS Ground Station サービスに問題があるため、問い合わせに失敗しました。	はい	なし	いいえ
キャンセル中	問い合わせはキャンセル中です。	いいえ	なし	いいえ
AWS_CANCELLED	問い合わせは AWS Ground Station サービスによってキャンセルされました。アンテナまたはサイトのメンテナンス、およびエフェメリスドリフトは、このような状況が発生する場合の例です。	はい	なし	いいえ
CANCELLED	問い合わせがキャンセルされました。	はい	なし	いいえ

#### Note

キャンセルまたは停止された問い合わせの請求への影響については、「」を参照してください。  
[問い合わせの請求を理解する](#)。

## 問い合わせデータの保持

AWS Ground Station は、[ReserveContact](#) リクエストが実行されてから 1 年間、問い合わせデータを保持します。1 年が経過すると、問い合わせデータは削除されます。

1 年を超えて問い合わせデータを保持する必要がある場合は、保持期間が終了する前にデータをエクスポートすることをお勧めします。連絡先データにアクセスしてエクスポートする方法の詳細については、以下を参照してください。

- [AWS Ground Station API リファレンス](#)
- [AWS Ground Station CLI コマンドリファレンス](#)

## 問い合わせの請求を理解する

では AWS Ground Station、アンテナ時間に対してのみ料金が発生します。AWS Ground Station メーターの問い合わせ使用量は 1 分あたりです。問い合わせごとに、サービスは開始時刻から終了時刻までの問い合わせ期間を計算し、最も近い分に切り上げます。この計測期間によって、その問い合わせの料金が決まります。

レートは 2 つの主な要因によって異なります。

- 帯域幅 – 問い合わせ用に予約されている帯域幅の量 (狭帯域または広帯域)
- 地上ステーションの場所 – 料金は地上ステーションの場所によって異なります

### 帯域幅の定義

AWS Ground Station は、即時帯域幅に基づいてコンタクトを 2 つの帯域幅階層に分類します。

- ナローバンド – 即時帯域幅が 40 MHz 以下のコンタクト
- 広帯域 – 瞬時帯域幅が 40 MHz を超えるコンタクト

### スケジュールモード

AWS Ground Station には 2 つのスケジューリングモードがあります。

- オンデマンド – 長期契約なしでアンテナアクセスの料金を支払う
- リザーブド – 割引料金を提供し、オンデマンドと比較してスケジューリングを改善します。また、毎月のコミットメントがあります。リザーブド分単位の料金は、一定期間毎月の使用をコミットするお客様にご利用いただけます。

アカウントの特定の料金情報、またはリザーブドスケジューリングモードの詳細については、AWS 担当者にお問い合わせください。

### CancelContact

[CancelContact](#) API の使用は、呼び出し時の問い合わせの状態によって異なります。

- 問い合わせを開始する前に - 問い合わせを完全にキャンセルします

- 問い合わせの開始後および問い合わせの終了前 - 進行中の問い合わせを停止します

問い合わせをキャンセルする場合、請求はスケジュールモードとキャンセル時期によって異なります。詳細については、AWS 担当者にお問い合わせください。

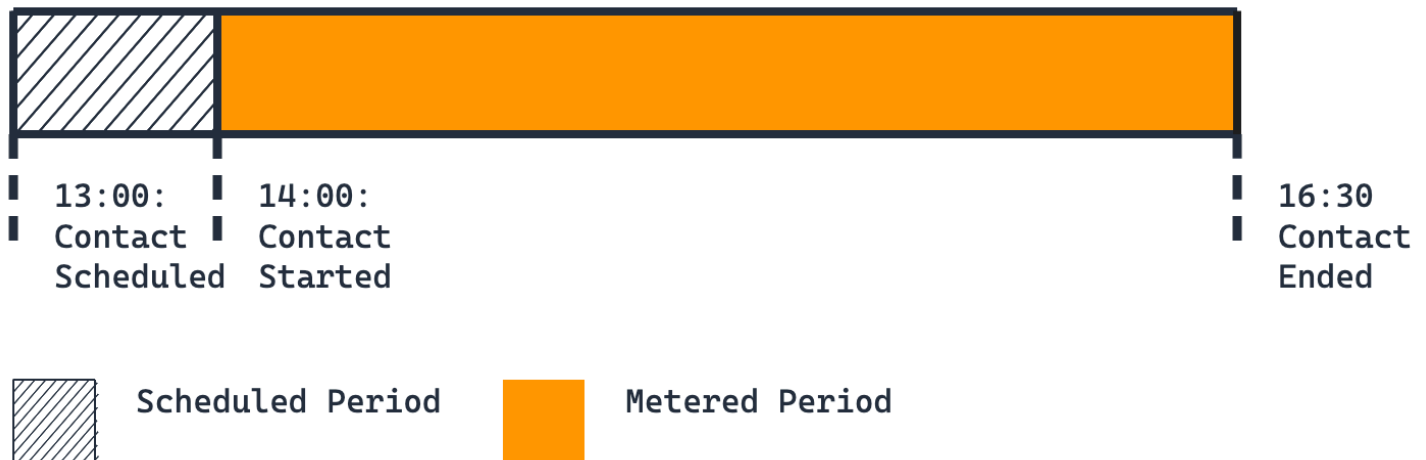
コンタクトを停止すると、実行されたコンタクトの部分と、重複したコンタクトでカバーされない残りの時間に対して課金されます。このコンテキストでの重複した問い合わせは次のとおりです。

- 元の停止したコンタクトと同じ地上局でスケジュールされている
- 元の停止した問い合わせと同じ AWS アカウント ID でスケジュールされている
- 元の問い合わせを停止するためにコマンドが発行された後に予約されます

以下のシナリオは、この計測が実際にどのように機能するかを示しています。

## シナリオ 1: 単一問い合わせ

Ground Station Anytown 1 で 150 分のコンタクトをスケジュールして、14:00 に開始し、16:30 に終了します。



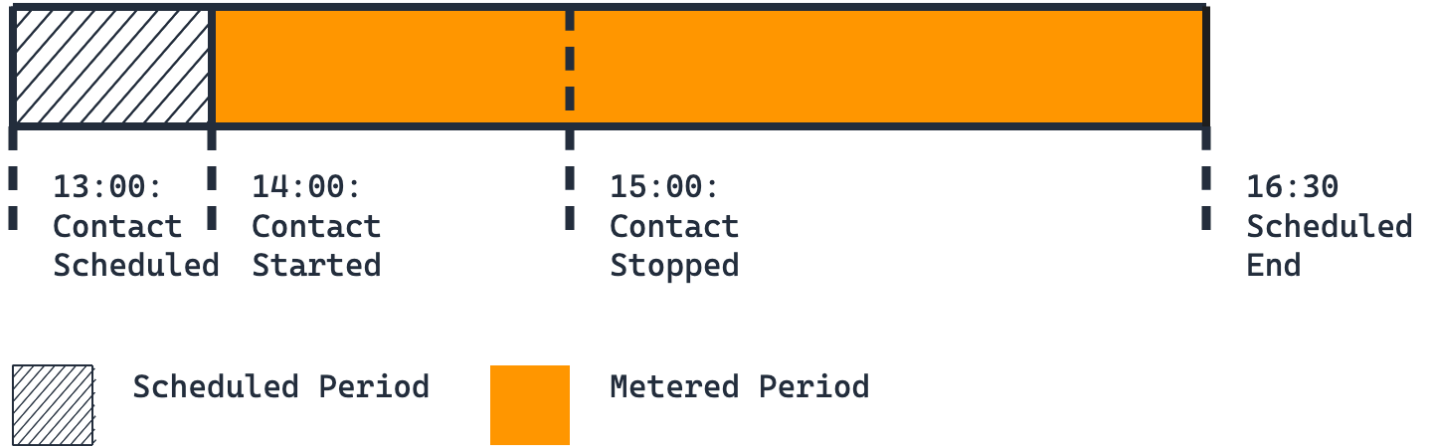
請求内訳:

- 最初の問い合わせ: 150 分 (全期間)

150 分間請求されます。これは、コンタクトが停止やキャンセルなしでスケジュールされた完了まで実行されるベースラインシナリオです。

## シナリオ 2: 単一停止コンタクト

Ground Station Anytown 1 で 150 分のコンタクトをスケジュールして、14:00 に開始し、16:30 に終了します。15:00 に CancelContact API を呼び出して問い合わせを停止します。



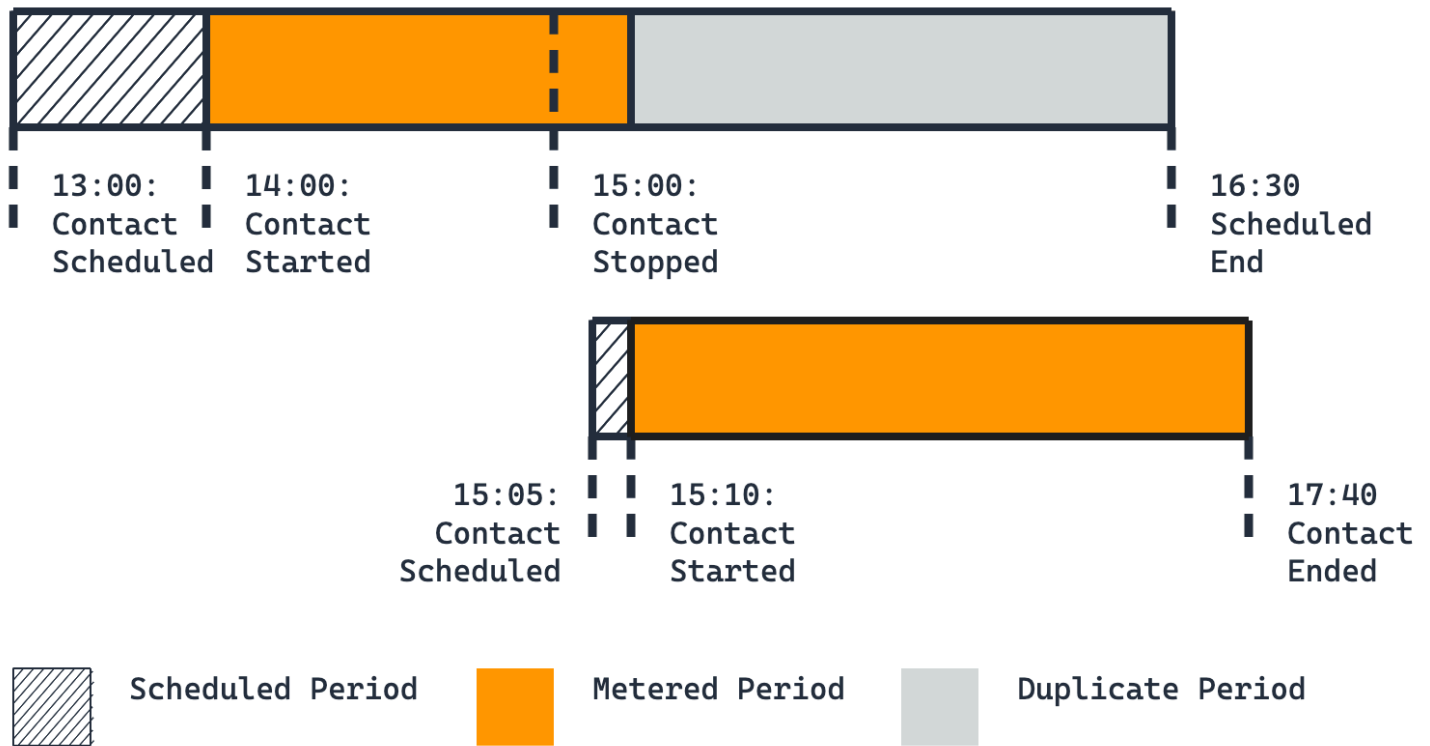
請求内訳:

- 最初の問い合わせ: 150 分 (完全な元の期間)

問い合わせを停止したが、残りの時間 (15:00 ~ 16:30) をカバーするように重複する問い合わせをスケジュールしなかったため、150 分分の料金が請求されます。重複をスケジュールせずに問い合わせを停止しても、最初にスケジュールされた期間全体は引き続きお客様の責任となります。

## シナリオ 3: 重複が 1 つ

Ground Station Anytown 1 で 150 分のコンタクトをスケジュールして、14:00 に開始し、16:30 に終了します。15:00 に CancelContact API を呼び出して、最初の問い合わせを停止します。CancelContact を呼び出した後、同じ Ground Station で別の連絡先を 15:10 から 150 分間スケジュールします。



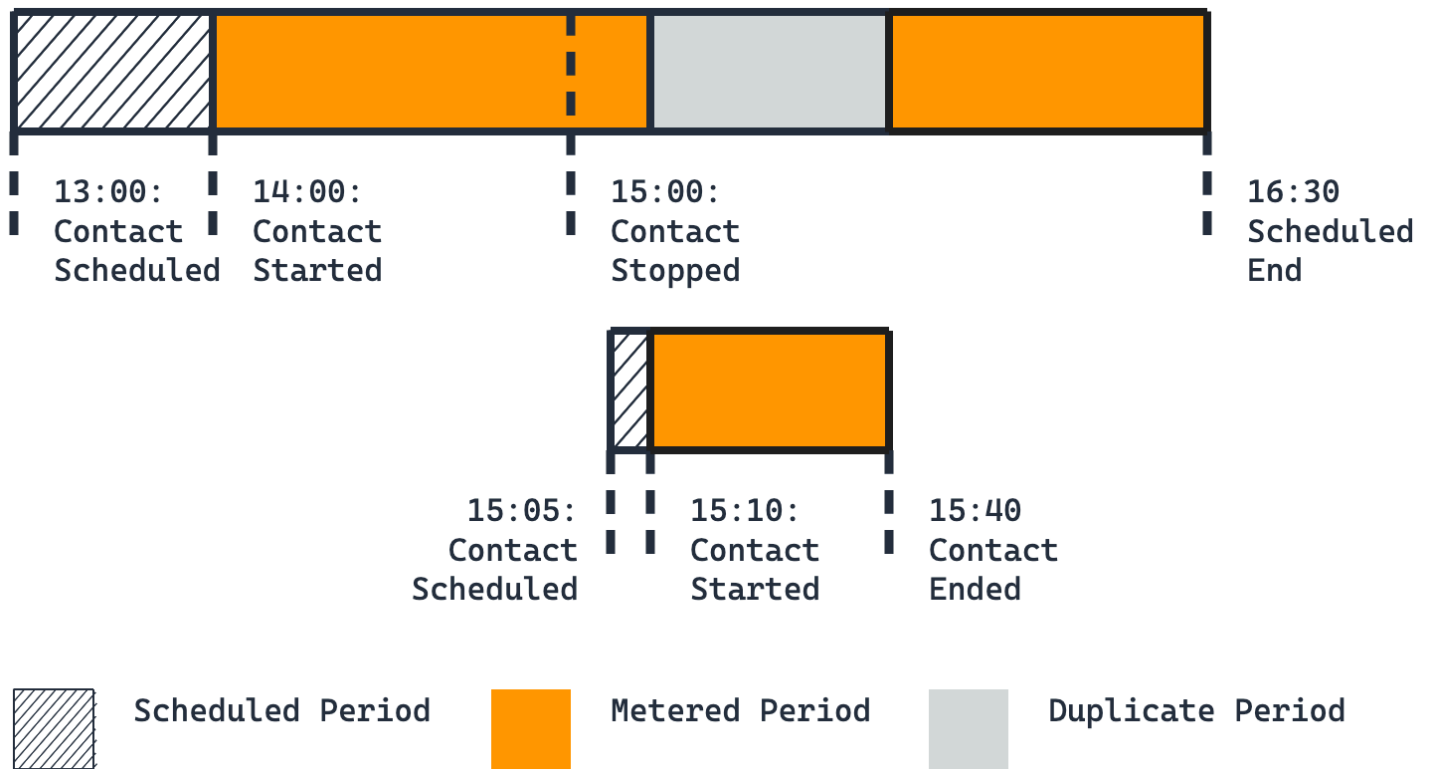
#### 請求内訳:

- 最初の問い合わせ: 70 分 (実行時間 60 分 + 2 番目の問い合わせが始まる前のダウンタイム 10 分)
- 2 回目の問い合わせ: 150 分 (全期間)

最初の問い合わせを停止した後にスケジュールしたため、2 番目の問い合わせは重複しています。重複は 15:10 から 16:30 までの残り時間をカバーしているため、最初の問い合わせが実際に実行された時刻と、停止と再起動の間の 10 分のギャップに対してのみ課金されます。

#### シナリオ 4: 短い重複

Ground Station Anytown 1 で 150 分のコンタクトをスケジュールして、14:00 に開始し、16:30 に終了します。15:00 に CancelContact API を呼び出して、最初の問い合わせを停止します。CancelContact を呼び出した後、15:10 から同じ Ground Station で 30 分のコンタクトをスケジュールします。



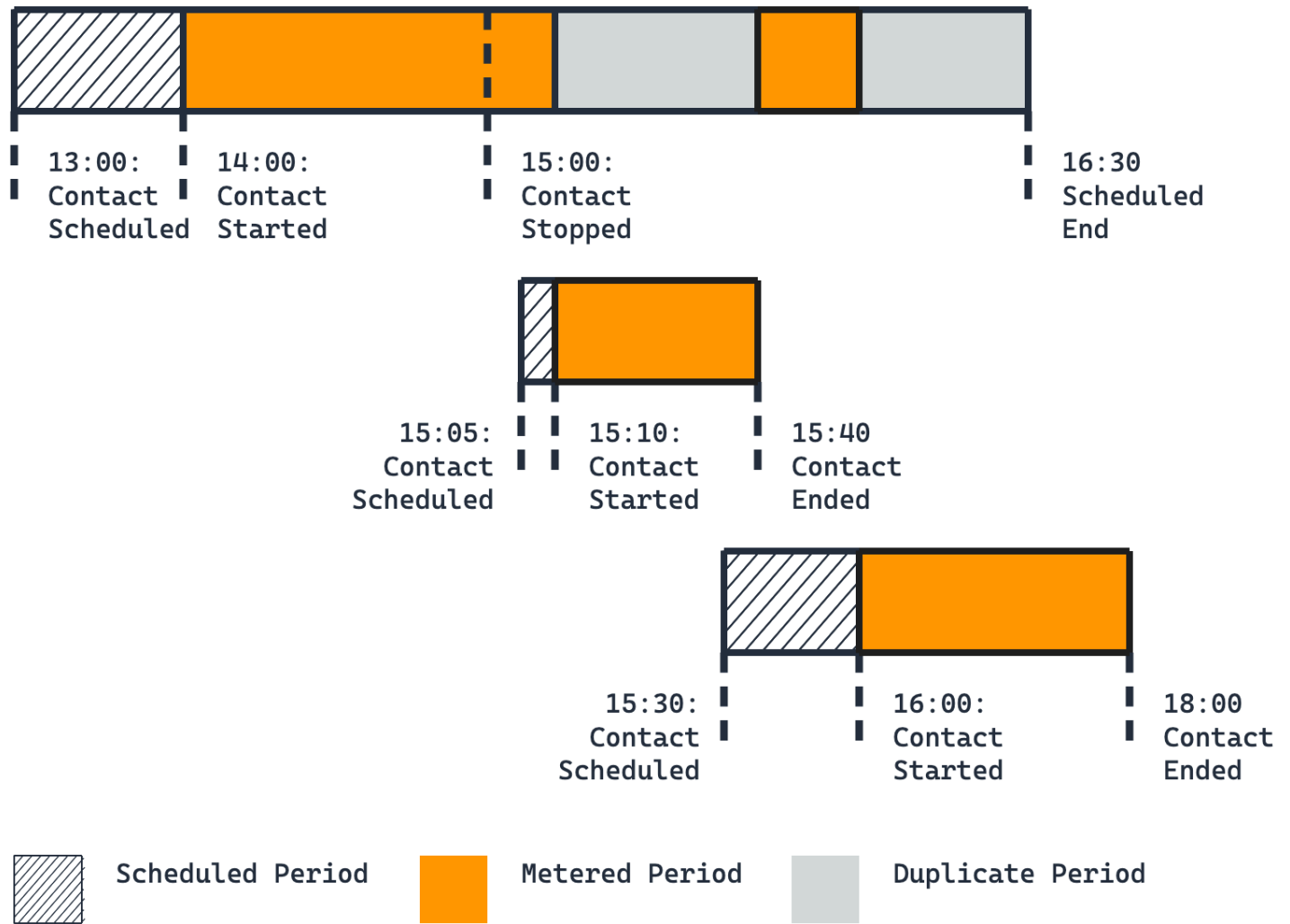
請求内訳:

- 最初の問い合わせ: 120 分 (実行時間 60 分 + 2 番目の問い合わせが開始される前のダウンタイム 10 分 + 重複がカバーしなかった残り時間 50 分)
- 2 回目の問い合わせ: 30 分 (全期間)

重複した問い合わせは、最初の問い合わせを停止してから残り 90 分のうち 30 分 (15:10 ~ 15:40) のみを対象とします。重複開始の 10 分のギャップと、重複終了の 50 分の時間 (15:40 ~ 16:30) の両方に対して課金されます。

## シナリオ 5: 複数の重複

Ground Station Anytown 1 で 150 分のコンタクトをスケジュールして、14:00 に開始し、16:30 に終了します。15:00 に CancelContact API を呼び出して、最初の問い合わせを停止します。CancelContact を呼び出した後、15:10 から同じ Ground Station で 30 分のコンタクトをスケジュールします。その後、15:30 に、120 分間 16:00 から別の問い合わせをスケジュールします。



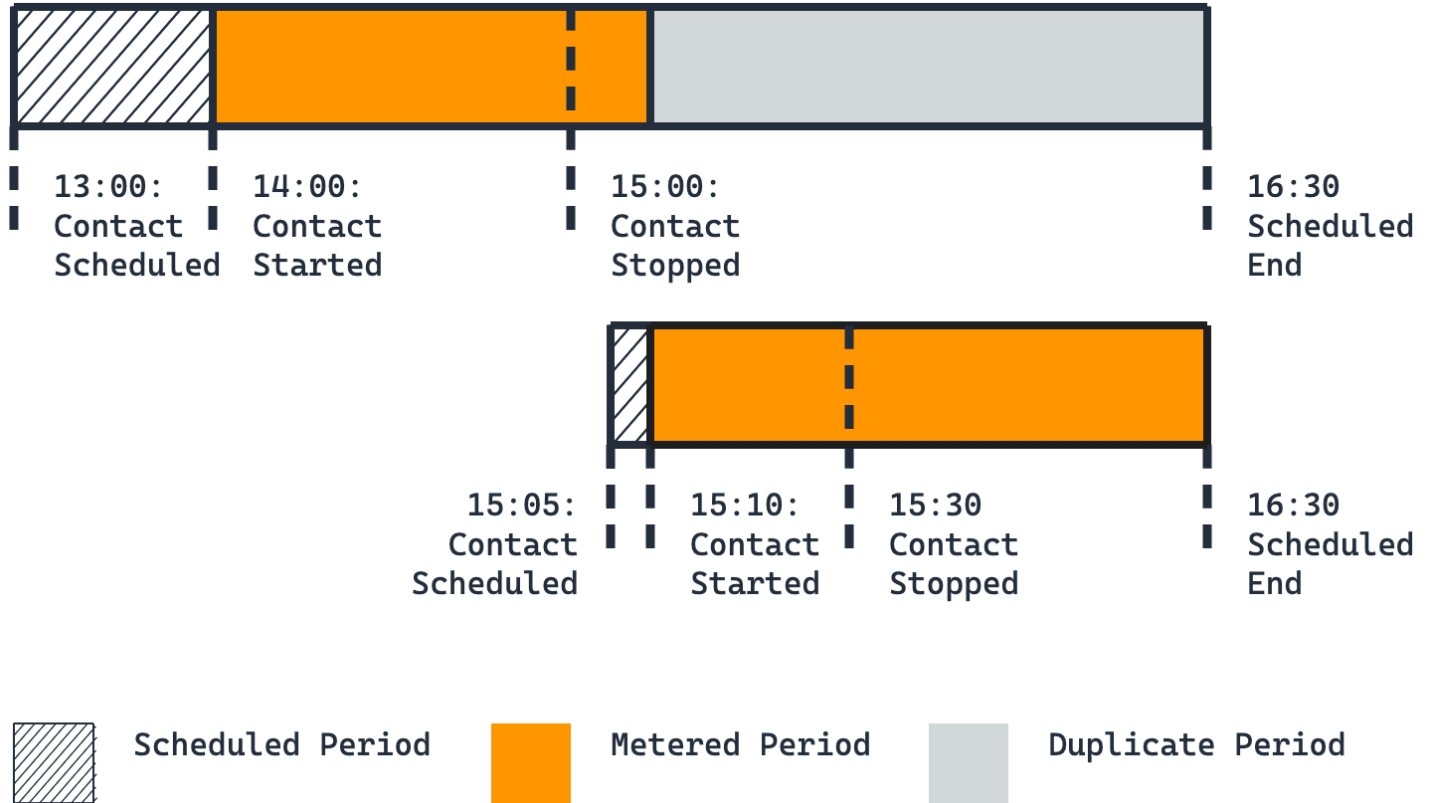
## 請求内訳:

- 最初の問い合わせ: 90 分 (実行時間 60 分 + 2 番目の問い合わせが始まる前のダウンタイム 10 分 + 2 番目と 3 番目の問い合わせの間のダウンタイム 20 分)
- 2 回目の問い合わせ: 30 分 (全期間)
- 3 回目の問い合わせ: 120 分 (全期間)

最初の問い合わせを停止した後にスケジュールしたため、2 番目と 3 番目の問い合わせの両方が重複としてカウントされます。ただし、1 回目の停止 (15:00) から 2 回目の開始 (15:10) までは 10 分、2 回目の終了 (15:40) から 3 回目の開始 (16:00) までは 20 分という問い合わせ間のギャップに対して課金されます。

## シナリオ 6: 複数の停止

Ground Station Anytown 1 で 150 分のコンタクトをスケジュールして、14:00 に開始し、16:30 に終了します。15:00 に CancelContact API を呼び出して、最初の問い合わせを停止します。CancelContact を呼び出した後、Ground Station Anytown 1 で 80 分のコンタクトをスケジュールします。このコンタクトは 15:10 に開始され、16:30 に終了します。15:30 に CancelContact API を再度呼び出し、重複する問い合わせを停止します。



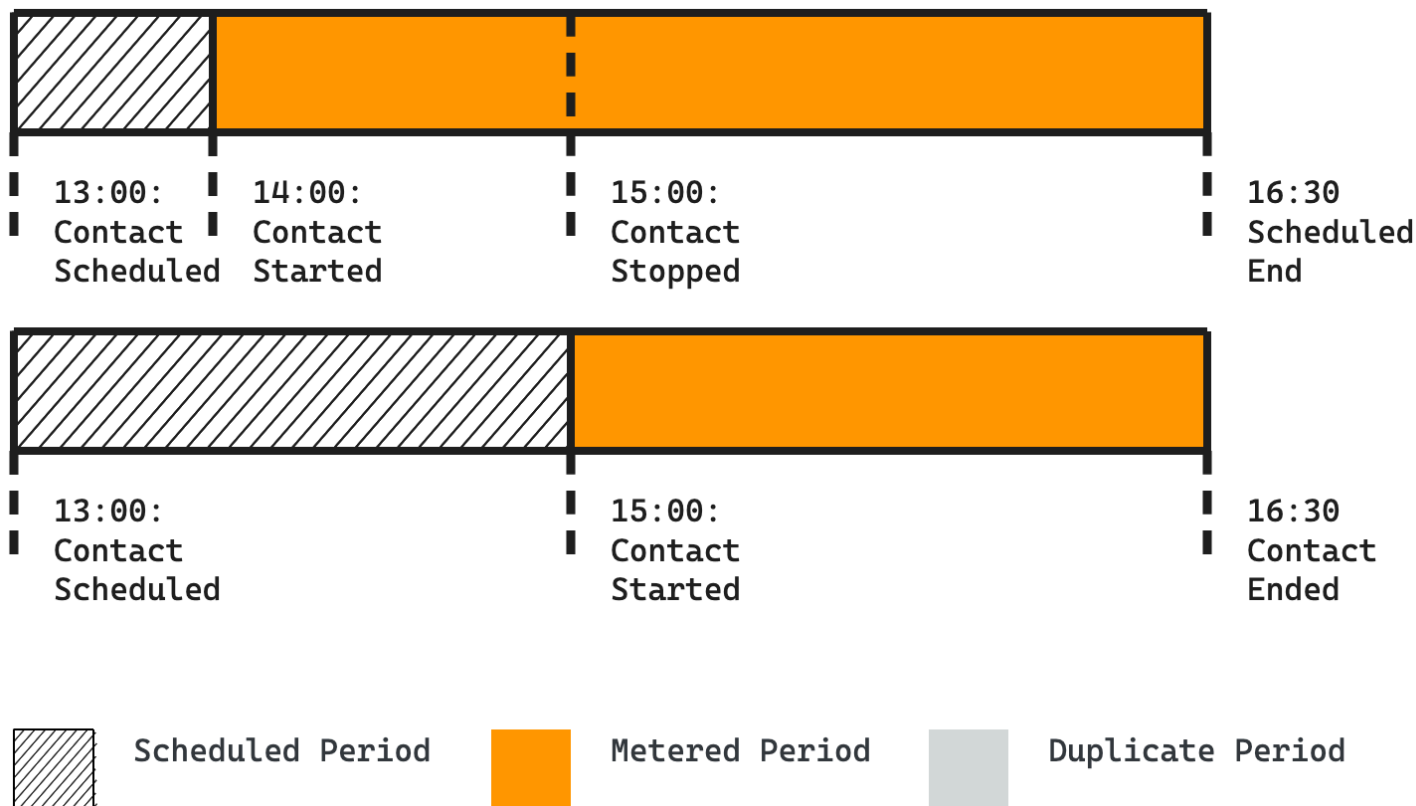
### 請求内訳:

- 最初の問い合わせ: 70 分 (実行時間 60 分 + 2 番目の問い合わせが始まる前のダウンタイム 10 分)
- 2 回目の問い合わせ: 80 分 (完全な元の期間)

2 番目の問い合わせは、最初にスケジュールされた時刻 (15:30 ~ 16:30) の 60 分を未入力のままにして 15:30 に停止したため、80 分の全期間に対して請求されます。残りの時間をカバーするために別の重複した問い合わせをスケジュールしない限り、停止した問い合わせの期間全体はお客様の責任となります。

## シナリオ 7: 重複しないマルチアンテナ地上ステーション

13:00 に、Ground Station Anytown 1 で 2 つのコンタクトをスケジュールします。1 つ目は、14:00 から 16:30 までの 150 分の問い合わせです。2 つ目は、15:00 に開始し、16:30 に終了する 90 分の問い合わせです。15:00 に CancelContact API を呼び出して、最初の問い合わせを停止します。Ground Station Anytown 1 はマルチアンテナの地上ステーションで、両方のコンタクトを同時に実行できます。



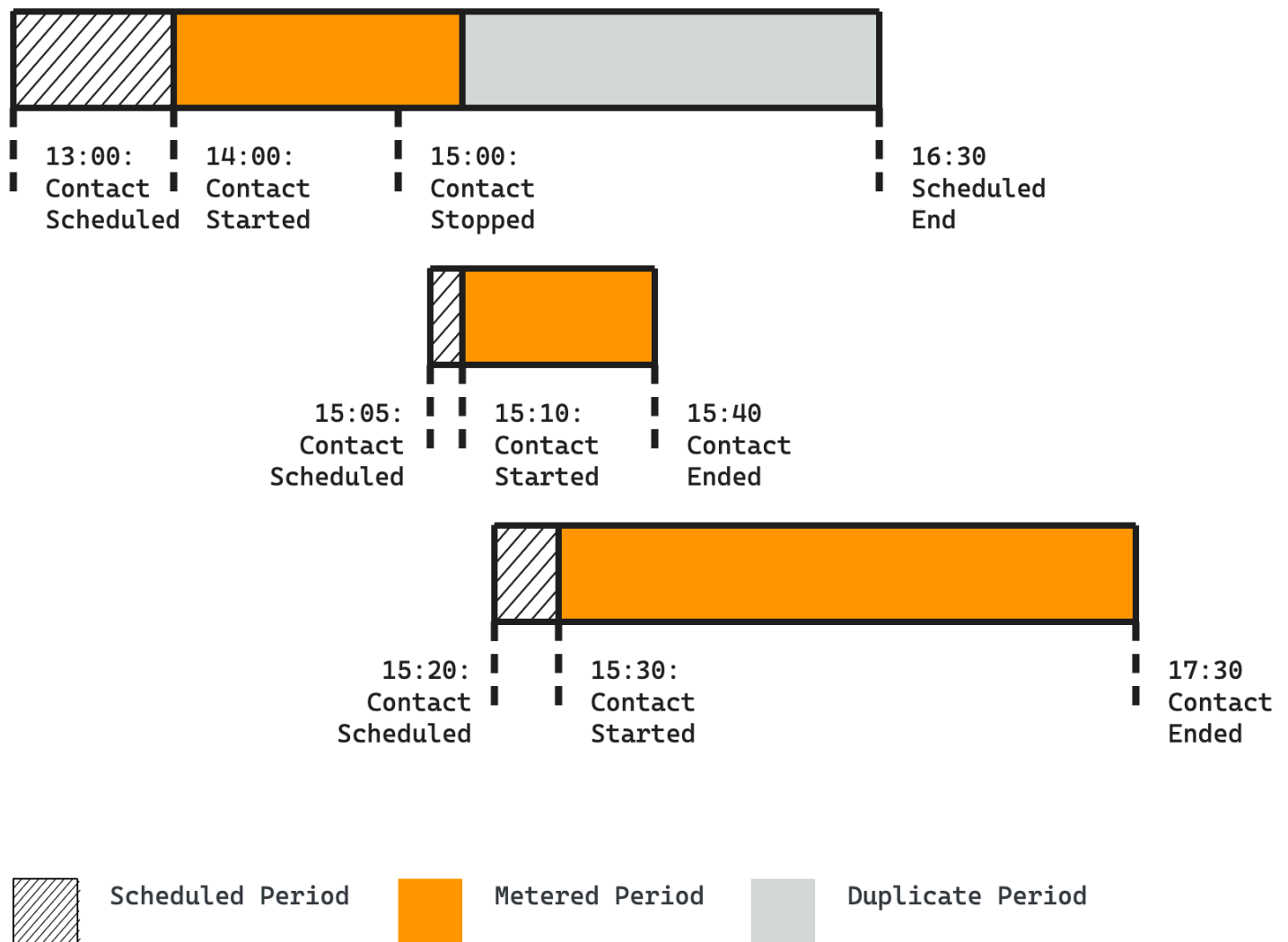
請求内訳:

- 最初の問い合わせ: 150 分 (完全な元の期間)
- 2 回目の問い合わせ: 90 分 (全期間)

2 番目の問い合わせは最初の問い合わせの停止部分と重複しますが、重複としてカウントされません。2 番目の問い合わせは、重複の最初の基準を満たしていません。最初の問い合わせを 15:00 に停止する前に、13:00 にスケジュールされました。重複していないため、いつ停止したかにかかわらず、最初の問い合わせの元の期間全体に対して課金されます。

## シナリオ 8: コンタクトが重複しているマルチアンテナ地上ステーション

Ground Station Anytown 1 で 150 分のコンタクトをスケジュールして、14:00 に開始し、16:30 に終了します。15:00 に CancelContact API を呼び出して、最初の問い合わせを停止します。CancelContact を呼び出した後、Ground Station Anytown 1 で 30 分のコンタクトをスケジュールします。開始時刻は 15:10 で、終了時刻は 15:40 です。その後、Ground Station Anytown 1 で、15:30 から 17:00 まで、さらに 90 分のコンタクトをスケジュールします。Ground Station Anytown 1 はマルチアンテナの地上局であり、重複するコンタクトの両方を重複時間で同時に実行できます。



請求内訳:

- 最初の問い合わせ: 70 分 (実行時間 60 分 + 2 番目の問い合わせが開始されるまでのダウンタイム 10 分)

- 2 回目の問い合わせ: 30 分 (全期間)
- 3 回目の問い合わせ: 90 分 (全期間)

最初の問い合わせを停止した後にスケジュールしたため、2 番目と 3 番目の問い合わせの両方が重複としてカウントされます。最初の問い合わせの停止 (15:00) と 2 番目の問い合わせの開始 (15:10) の間の 10 分のギャップは、元の問い合わせに対して請求されるダウンタイムを表します。

## AWS Ground Station デジタルツイン機能を使用する

のデジタルツイン機能は、衛星ミッション管理およびコマンドアンドコントロールソフトウェアをテストして統合できる環境 AWS Ground Station を提供します。デジタルツイン機能を使用すると、本番アンテナ容量を使用せずに、スケジューリング、設定の検証、適切なエラー処理をテストできます。デジタルツイン機能と AWS Ground Station の統合をテストすることで、衛星オペレーションをスムーズに管理するシステムの能力に対する信頼を高めることができます。また、本番稼働用容量を使用したり、スペクトルライセンスを必要とせずに AWS Ground Station APIs をテストすることもできます。

開始するには、「」に従い[衛星搭載](#)、デジタルツイン機能へのオンボーディングをリクエストします。衛星がデジタルツイン機能にオンボードされると、デジタルツイン地上ステーションに対してコンタクトをスケジュールできます。アクセスできる地上ステーションのリストは、AWS SDK [ListGroundStations](#) レスポンスを介して取得できます。デジタルツイン地上ステーションは、「デジタルツイン」というプレフィックスが付けられた[AWS Ground Station ロケーション](#)、「」に記載されている地上ステーションの正確なコピーです。これには、サイトマスクや実際の GPS 座標など、アンテナの機能やメタデータが含まれますが、これらに限定されません。現時点では、デジタルツイン機能は、「」で説明されているようにデータ配信をサポートしていません[データフローの使用](#)。

オンボードされると、デジタルツイン機能は、「」で説明されているように、本番稼働用サービスと同じ Amazon EventBridge イベントと API レスポンスを出力します[イベント AWS Ground Station による自動化](#)。これらのイベントにより、設定とデータフローエンドポイントグループを微調整できます。

# によるモニタリングを理解する AWS Ground Station

モニタリングは、AWS Ground Stationの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。AWS には、監視 AWS Ground Station、問題発生時の報告、必要に応じて自動アクションを実行するための以下のモニタリングツールが用意されています。

- Amazon EventBridge Events は、AWS リソースの変更を記述するシステムイベントのほぼリアルタイムのストリームを提供します。EventBridge Events は、特定のイベントを監視し、これらのイベントが発生したときに他の AWS サービスで自動アクションをトリガーするルールを記述できるため、イベント駆動型の自動コンピューティングを有効にします。EventBridge イベントの詳細については、「[Amazon EventBridge イベントユーザーガイド](#)」を参照してください。
- AWS CloudTrail は、AWS アカウントによって、またはアカウントに代わって行われた API コールおよび関連イベントをキャプチャし、指定した Amazon S3 バケットにログファイルを配信します。が呼び出したユーザーとアカウント AWS、呼び出し元のソース IP アドレス、呼び出しの発生日時を特定できます。詳細については AWS CloudTrail、[AWS CloudTrail 「ユーザーガイド」](#)を参照してください。
- Amazon CloudWatch Metrics は、の使用時にスケジュールされた問い合わせのメトリクスをキャプチャします AWS Ground Station。CloudWatch メトリクスを使用すると、チャンネル、偏波、および人工衛星 ID に基づいてデータを分析し、コンタクトの信号強度とエラーを識別できます。詳細については、「[Amazon CloudWatch メトリクスを使用する](#)」を参照してください。
- [AWS User Notifications](#) を使用して、AWS Ground Station イベントに関する通知を受け取る配信チャンネルを設定できます。指定したルールにイベントが一致すると、通知を受け取ります。イベントの通知は、Eメール、[チャットアプリケーション内の Amazon Q Developer](#)のチャット通知、[AWS Console Mobile Application](#)のプッシュ通知などの複数のチャンネルで受け取ることができます。Console AWS [Notification Center](#). User Notifications support 集計で通知を表示することもできます。これにより、特定のイベント中に受信する通知の数を減らすことができます。

AWS Ground Station をモニタリングするには、次のトピックを参照してください。

## トピック

- [イベント AWS Ground Station による自動化](#)
- [を使用した AWS Ground Station API コールのログ記録 AWS CloudTrail](#)
- [Amazon CloudWatch でメトリクスを表示する](#)

# イベント AWS Ground Station による自動化

## Note

このドキュメントでは、全体を通して「イベント」という用語を使用しています。CloudWatch Events と EventBridge は同じ基盤となるサービスと API です。いずれかのサービスを使用することで、受信イベントを一致させ、処理のためにターゲットにルーティングするルールを作成できます。

イベントを使用すると、AWS サービスを自動化し、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。AWS サービスからのイベントはほぼリアルタイムで配信されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。自動的にトリガーできるアクションには、次のようなものがあります。

- AWS Lambda 関数の呼び出し
- Amazon EC2 Run Command の呼び出し
- Amazon Kinesis Data Streams へのイベントの中継
- AWS Step Functions ステートマシンのアクティブ化
- Amazon SNS トピックまたは Amazon SQS キューの通知

でイベントを使用する例 AWS Ground Station には、次のようなものがあります。

- イベント状態に基づいて Amazon EC2 インスタンスの開始と停止を自動化する Lambda 関数を呼び出す。
- コンタクトの状態が変化するたびに Amazon SNS トピックを発行する。これらのトピックは、コンタクトの最初または最後に E メール通知を送信するように設定できます。

詳細については、[「Amazon EventBridge Events ユーザーガイド」](#)を参照してください。

## AWS Ground Station イベントタイプ

### Note

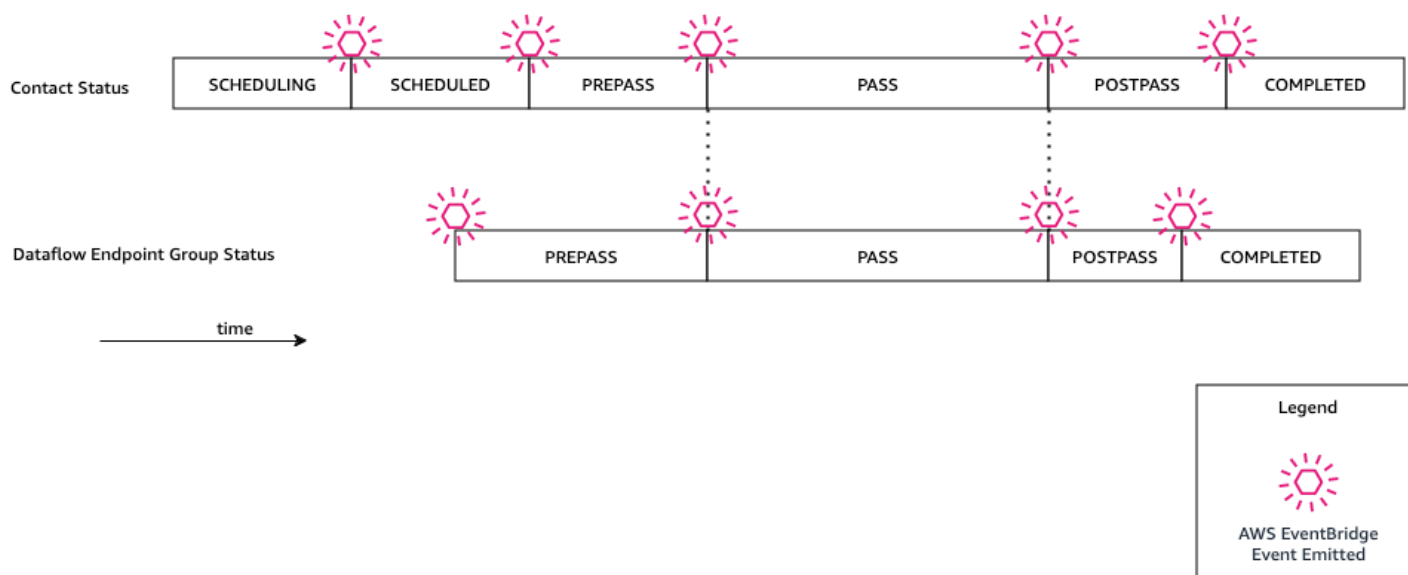
AWS Ground Station によって生成されるすべてのイベントには、「ソース」の値として「aws.groundstation」が含まれます。

AWS Ground Station は、オートメーションをカスタマイズできるように、状態変更に関連するイベントを発行します。現在、は問い合わせ状態変更イベント、データフローエンドポイントグループ変更イベント、エフェメリス状態変更イベント AWS Ground Station をサポートしています。以下のセクションでは、各タイプに関する詳細情報を提供します。

### 問い合わせイベントのタイムライン

AWS Ground Station は、問い合わせの状態が変更されたときにイベントを発行します。これらの状態の変化とは何か、および状態自体の意味の詳細については、[問い合わせのライフサイクルを理解する](#)「」を参照してください。問い合わせで使用されているデータフローエンドポイントグループには、独立した一連のイベントも出力されます。同じ期間に、データフローエンドポイントグループのイベントも出力されます。パス前およびパス後のイベントの正確な時間は、ミッションプロファイルとデータフローエンドポイントグループを設定するときに設定できます。

次の図は、名目上の問い合わせおよび関連するデータフローエンドポイントグループに対して出力されるステータスとイベントを示しています。



## Ground Station のコンタクト状態の変化

今後の問い合わせの状態が変化しているときに特定のアクションを実行する場合は、このアクションを自動化するルールを設定できます。これは、コンタクトの状態変更に関する通知を受信する場合に役立ちます。これらのイベントを受信したときに を変更する場合は、ミッションプロファイルの [contactPrePassDurationSeconds](#) と [contactPostPassDurationSeconds](#) を変更できます。イベントは、コンタクトのスケジュール元のリージョンに送信されます。

イベントの例を以下に示します。

```
{
  "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"
  }
}
```

`contactStatus` に指定できる値は、[the section called “AWS Ground Station 問い合わせステータス”](#) で定義されています。

## Ground Station データフローエンドポイントグループの状態変更

データフローエンドポイントグループをデータ受信に使用しているときにアクションを実行する場合は、このアクションを自動化するルールを設定できます。これにより、データフローエンドポイントグループステータスの状態変更に応じて、さまざまなアクションを実行できます。これ

らのイベントを受信したときに を変更する場合は、異なる [contactPrePassDurationSeconds](#) と [contactPostPassDurationSeconds](#) を持つデータフローエンドポイントグループを使用します。このイベントは、データフローエンドポイントグループのリージョンに送信されます。

以下に例を示します。

```
{
  "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"
  }
}
```

`dataflowEndpointGroupState` の状態として、PREPASS、PASS、POSTPASS、および COMPLETED が考えられます。

## エフェメリスイベント

### Ground Station エフェメリス状態の変化

エフェメリスの状態が変わったときにアクションを実行する場合は、このアクションを自動化するルールを設定できます。これにより、エフェメリスの状態変化に応じてさまざまなアクションを実行できます。例えば、エフェメリスの検証が完了し、ENABLED になっているときにアクションを実行できます。このイベントの通知は、エフェメリスがアップロードされたリージョンに送信されます。

以下に例を示します。

```
{
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "Ground Station Ephemeric 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"
  }
}
```

ephemerisStatus の状態として、ENABLED、VALIDATING、INVALID、ERROR、DISABLED、および EXPIRED が考えられます。

## を使用した AWS Ground Station API コールのログ記録 AWS CloudTrail

AWS Ground Station は、ユーザー AWS CloudTrail、ロール、または のサービスによって実行されたアクションを記録する AWS サービスであると統合されています AWS Ground Station。CloudTrail は、AWS Ground Station のすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、AWS Ground Station コンソールからの呼び出しと AWS Ground Station API オペレーションへのコード呼び出しが含まれます。証跡を作成する場合は、イベ

ントを含む Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます AWS Ground Station。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、リクエストの実行元の IP アドレス AWS Ground Station、リクエストの実行者、リクエストの実行日時などの詳細を確認できます。

CloudTrail の詳細については、「[AWS CloudTrail ユーザーガイド](#)」を参照してください。

## AWS Ground Station CloudTrail の情報

CloudTrail は、AWS アカウントの作成時にアカウントで有効になります。アクティビティが発生すると AWS Ground Station、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、[CloudTrail イベント履歴でのイベントの表示](#)を参照してください。

のイベントなど、AWS アカウントのイベントの継続的な記録については AWS Ground Station、証跡を作成します。追跡により、CloudTrail はログファイルを Simple Storage Service (Amazon S3) バケットに配信できます。デフォルトでは、コンソールで作成した証跡がすべての AWS リージョンに適用されます。証跡は、AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように他の AWS サービスを設定できます。詳細については、次を参照してください:

- [証跡の作成のための概要](#)
- [CloudTrail がサポートするサービスと統合](#)
- [CloudTrail 用 Amazon SNS 通知の構成](#)
- [複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る](#)

すべての AWS Ground Station アクションは CloudTrail によってログに記録され、[AWS Ground Station API リファレンス](#)に記載されています。例えば、ReserveContact、CancelContact、ListConfigs の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報を使用して行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「[CloudTrail userIdentity 要素](#)」を参照してください。

## AWS Ground Station ログファイルエントリについて

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは、任意の出典からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

以下の例は、ReserveContact アクションを示す CloudTrail ログエントリです。

例: 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 CloudWatch でメトリクスを表示する

問い合わせ中、はデータ AWS Ground Station を自動的にキャプチャし、分析のために CloudWatch に送信します。データは Amazon CloudWatch コンソールで表示できます。CloudWatch メトリクスのアクセス方法と詳細については、「[Amazon CloudWatch メトリクスの使用](#)」を参照してください。

AWS Ground Station テレメトリ機能を使用して、コンタクト中にほぼリアルタイムのメトリクスを受信することもできます。CloudWatch メトリクスはほぼリアルタイムでは利用できないため、配信が遅れる可能性があります。また、CloudWatch は 1 秒間にわたってメトリクスを集約するため、データの詳細度が低下する可能性があります。テレメトリ機能は、個々のメトリクスを提供し、ほぼリアルタイムで AWS アカウントに直接配信します。詳細については、「[テレメトリの操作](#)」を参照してください。

**⚠ Important**

AWS Ground Station は、問い合わせがスケジュールされた AWS リージョンではなく、問い合わせの地上局の場所に関連付けられた AWS リージョンに CloudWatch メトリクスを出力します。問い合わせのメトリクスを表示するには、地上局リージョンの CloudWatch にアクセスする必要があります。各地上局のロケーションに関連付けられているリージョンについては AWS、「」を参照してください[地上ステーションの場所の AWS リージョンの検索](#)。コンタクトをスケジュールするリージョンでテレメトリデータを受信するには、AWS Ground Station テレメトリ機能を使用できます。詳細については、「[テレメトリの操作](#)」を参照してください。

## AWS Ground Station メトリクスとディメンション

### 利用可能なメトリクス

次のメトリクスは から入手できます AWS Ground Station。

**i Note**

出力される特定のメトリクスは、使用されている AWS Ground Station 機能によって異なります。設定によっては、以下のメトリクスのサブセットのみが出力される場合があります。

メトリクス	メトリクスディメンション	説明
AzimuthAngle	SatelliteId	アンテナの方位角。真北は 0 度、東は 90 度です。  単位: 度
BitErrorRate	チャンネル、分極化、SatelliteId	伝送したビットのうち、エラーが発生したビットの割合。ビットエラーは、ノ

メトリクス	メトリクスディメンション	説明
		<p>イズ、ゆがみ、または干渉によって発生します。</p> <p>単位: 単位時間あたりのビットエラー数</p>
BlockErrorRate	チャンネル、分極化、Satelliteld	<p>受信したブロックのうち、エラーが発生したブロックの割合。ブロックエラーは干渉によって発生します。</p> <p>単位: エラーが発生したブロック数/ブロック総数</p>
CarrierFrequencyRecovery_Cn0	カテゴリ、Config、Satelliteld	<p>単位帯域幅あたりのキャリア対ノイズ密度の比率。</p> <p>単位: デシベルヘルツ (dB-Hz)</p>

メトリクス	メトリクスディメンション	説明
CarrierFrequencyRecovery_Locked	カテゴリ、Config、Satelliteld	<p>復調器のキャリア周波数回復ループがロックされている場合は 1 に設定され、ロックが解除されている場合は 0 に設定されます。</p> <p>単位: 単位なし</p>
CarrierFrequencyRecovery_OffsetFrequency_Hz	カテゴリ、Config、Satelliteld	<p>推定された信号中心周波数と理想的な中心周波数の間のオフセット。この原因は、宇宙機とアンテナシステム間のドップラーシフトと局部発振器のオフセットです。</p> <p>単位: ヘルツ (Hz)</p>
ElevationAngle	Satelliteld	<p>アンテナの仰角。水平線は 0 度、天頂は 90 度です。</p> <p>単位: 度</p>

メトリクス	メトリクスディメンション	説明
Es/N0	チャンネル、分極化、SatelliteId	シンボルあたりのエネルギーとノイズパワースペクトル密度の比率。  単位: デシベル (dB)
ReceivedPower	分極化、SatelliteId	復調器/デコーダで測定された信号強度。  単位: ミリワットを基準値とするデシベル (dBm)
SymbolTimingRecovery_ErrorVectorMagnitude	カテゴリ、Config、SatelliteId	受信したシンボルと理想的なコンスタレーション点の間の誤差ベクトルの大きさ。  単位: パーセント

メトリクス	メトリクスディメンション	説明
SymbolTimingRecovery_Locked	カテゴリ、Config、SatelliteId	復調器シンボルのタイミング回復ループがロックされている場合は 1 に設定され、ロックが解除されている場合は 0 に設定されます。  単位: 単位なし
SymbolTimingRecovery_OffsetSymbolRate	カテゴリ、Config、SatelliteId	推定シンボレートと理想的な信号シンボレートとの間のオフセット。この原因は、宇宙機とアンテナシステム間のドップラーシフトと局部発振器のオフセットです。  単位: シンボル/秒

## どのディメンションが使用されています AWS Ground Stationが？

次のディメンションを使用して AWS Ground Station データをフィルタリングできます。

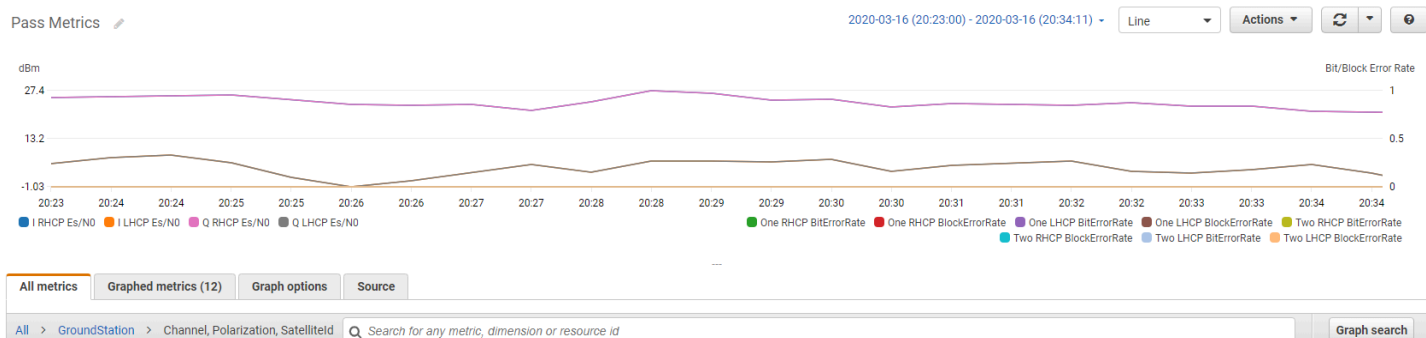
ディメンション	説明
Category	復調またはデコード。

ディメンション	説明
Channel	各コンタクトのチャンネルには、1、2、I (同相)、Q (直交) があります。
Config	アンテナダウンリンクデモデコード設定 ARN。
Polarization	各コンタクトの偏波には、LHCP (左円偏波) または RHCP (右円偏波) があります。
SatelliteId	人工衛星 ID には、コンタクトの人工衛星の ARN が含まれます。

## メトリクスの表示

グラフ化されたメトリクスを表示する場合、集計の時間帯によってメトリクスの表示方法が変わることに注意する必要があります。データの受信後 3 時間の間は、コンタクトの各メトリクスが 1 秒あたりのデータとして表示されます。データは、3 時間の経過後に CloudWatch Metrics によって 1 分あたりのデータとして集計されます。1 秒あたりのデータ測定値のメトリクスを表示する必要がある場合は、データを受信してから 3 時間以内にデータを表示するか、CloudWatch Metrics の外部に保持することをお勧めします。CloudWatch 保持の詳細については、[Amazon CloudWatch の概念 - メトリクス保持](#)」を参照してください。

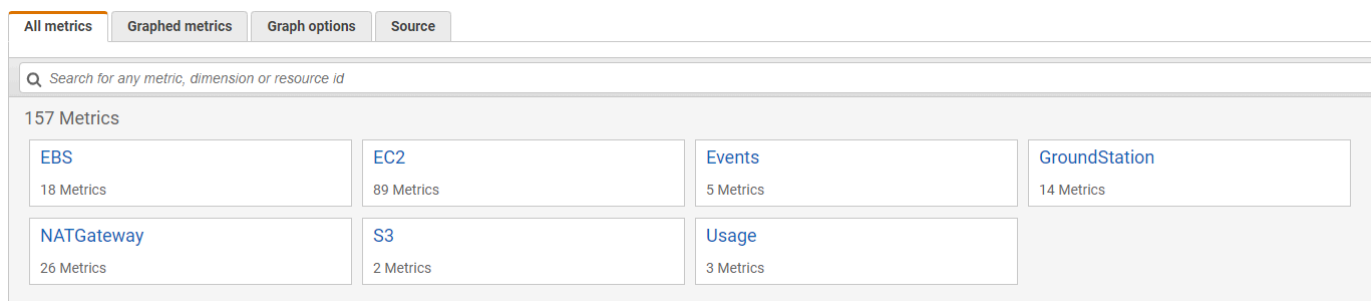
さらに、最初の 60 秒以内にキャプチャされたデータには、意味のあるメトリクスを生成するための十分な情報が含まれていないため、データが表示されない可能性があります。意味のあるメトリクスを表示するには、60 秒が経過した後でデータを表示することをお勧めします。



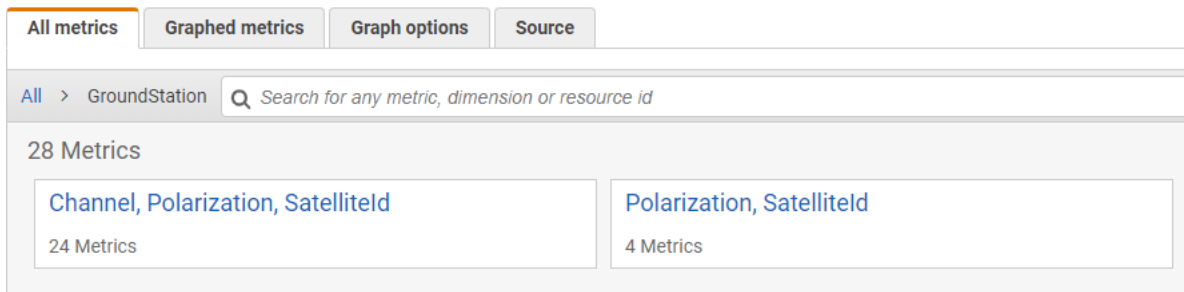
CloudWatch での AWS Ground Station メトリクスのグラフ化の詳細については、[「メトリクスのグラフ化](#)」を参照してください。

## コンソールを使用してメトリクスを表示するには

1. 地上局の場所に関連付けられた AWS リージョンを決定します。AWS Ground Station は、連絡先の地上局の場所に関連付けられたリージョンに CloudWatch メトリクスを出力します。地上局の場所とそれに関連する AWS リージョンのリストについては、「」を参照してください [地上ステーションの場所の AWS リージョンの検索](#)。
2. [CloudWatch コンソール](#)を開きます。
3. ナビゲーションペインで [Metrics (メトリクス)] を選択してください。
4. GroundStation 名前空間を選択します。



5. 目的のメトリクスディメンション (チャンネル、分極化、SatelliteId など) を選択します。



6. [All metrics] タブには、名前空間内のそのディメンションのメトリクスがすべて表示されます。以下の操作を行うことができます。
  - a. テーブルを並べ替えるには、列見出しを使用します。
  - b. メトリクスをグラフ化するには、メトリクスに関連付けられたチェックボックスをオンにします。すべてのメトリクスを選択するには、テーブルの見出し行のチェックボックスをオンにします。
  - c. リソースでフィルタするには、リソース ID を選択し、[Add to search] を選択します。
  - d. メトリクスでフィルタするには、メトリクスの名前を選択し、[Add to search] を選択します。

## を使用してメトリクスを表示するには AWS CLI

AWS Ground Station は、問い合わせの地上局の場所に関連付けられた リージョンに CloudWatch メトリクスを出力します。地上局の関連 AWS リージョンのリストについては、「」を参照してください [地上ステーションの場所の AWS リージョンの検索](#)。 *ground-station-region-code* を、地上局の場所の AWS リージョンコード (us-west-2オレゴン 1、ハワイ 1、アラスカ 1 など) に置き換えます。この手順の後続の AWS CLI コマンドはすべて、同じリージョンを使用する必要があります。

1. AWS CLI がインストールされていることを確認します。インストールの詳細については AWS CLI、[「AWS CLI バージョン 2 のインストール」](#)を参照してください。
2. 地上局の場所に関連付けられた AWS リージョンを特定します。
3. CloudWatch CLI の [get-metric-data](#) メソッドを使用して、変更して目的のメトリクスを指定し、それらのメトリクスのクエリに使用できるファイルを生成します。

これを行うには、 を実行します `aws cloudwatch get-metric-data --region ground-station-region-code --generate-cli-skeleton`。これにより、次のような出力が生成されます。

```
{
  "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. を実行して、使用可能な CloudWatch `aws cloudwatch list-metrics --region ground-station-region-code` メトリクスを一覧表示します。

最近 を使用した場合 AWS Ground Station、メソッドは次のようなエントリを含む出力を返す必要があります。

```

...
{
  "Namespace": "AWS/GroundStation",
  "MetricName": "ReceivedPower",
  "Dimensions": [
    {
      "Name": "Polarization",
      "Value": "LHCP"
    },
    {
      "Name": "SatelliteId",
      "Value": "arn:aws:groundstation::111111111111:satellite/aaaaaaaa-
bbbb-cccc-dddd-eeeeeeeeeeee"
    }
  ]
},
...

```

**Note**

最後に を使用してから 2 週間以上が経過している場合は AWS Ground Station、[使用可能なメトリクスのテーブル](#)を手動で検査して、メトリクス名前空間で AWS/GroundStationメトリクス名とディメンションを見つける必要があります。CloudWatch の制限の詳細については、[「View available metrics」](#)を参照してください。

5. ステップ 2 で作成した JSON ファイルを変更して、ステップ 3 の SatelliteId やメトリクスの必要な値と一致Polarizationさせます。また、連絡先に合わせて StartTime、EndTime の値を必ず更新してください。例えば、次のようになります。

```
{
  "MetricDataQueries": [
    {
      "Id": "receivedPowerExample",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/GroundStation",
          "MetricName": "ReceivedPower",
          "Dimensions": [
            {
              "Name": "SatelliteId",
              "Value":
"arn:aws:groundstation::111111111111:satellite/aaaaaaaa-bbbb-cccc-dddd-
eeeeeeeeeeeeee"
            },
            {
              "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 は、メトリクスに応じて 1~60 秒ごとにメトリクスを発行します。Period フィールドの値がメトリクスの発行期間より小さい場合、メトリクスは返されません。

6. 前のステップで作成した設定ファイル `aws cloudwatch get-metric-data` を使用して を実行します。以下に例を示します。

```
aws cloudwatch get-metric-data --region ground-station-region-code --cli-input-json  
file://<nameOfConfigurationFileCreatedInStep2>.json
```

メトリクスには、コンタクトのタイムスタンプが表示されます。AWS Ground Station メトリクスの出力例を次に示します。

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

## のセキュリティ AWS Ground Station

でのクラウドセキュリティが最優先事項 AWS です。お客様は AWS、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。AWS は、セキュリティの目標達成に役立つセキュリティ固有のツールと機能を提供しています。これらのツールと機能には、ネットワークセキュリティ、設定管理、アクセスコントロール、およびデータセキュリティが含まれます。

を使用する場合は AWS Ground Station、業界のベストプラクティスに従い、end-to-end暗号化を実装することをお勧めします。AWS では、暗号化とデータ保護を統合するための API を提供しています。AWS セキュリティの詳細については、[AWS セキュリティ入門](#) ホワイトペーパーを参照してください。

以下のトピックでは、 のリソースをセキュリティで保護する方法について説明します。

### トピック

- [の Identity and Access Management AWS Ground Station](#)
- [AWS の 管理ポリシー AWS Ground Station](#)
- [Ground Station のサービスにリンクされたロールを使用する](#)
- [の保管中のデータ暗号化 AWS Ground Station](#)
- [の転送中のデータ暗号化 AWS Ground Station](#)

## の Identity and Access Management AWS Ground Station

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つです。IAM 管理者は、誰を認証 (サインイン) し、誰に AWS Ground Station リソースの使用を許可する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

### トピック

- [オーディエンス](#)
- [アイデンティティを使用した認証](#)
- [ポリシーを使用したアクセスの管理](#)

- [が IAM と AWS Ground Station 連携する方法](#)
- [のアイデンティティベースのポリシーの例 AWS Ground Station](#)
- [AWS Ground Station ID とアクセスのトラブルシューティング](#)

## オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、ロールによって異なります。

- サービスユーザー - 機能にアクセスできない場合は、管理者にアクセス許可をリクエストします (「[AWS Ground Station ID とアクセスのトラブルシューティング](#)」を参照)。
- サービス管理者 - ユーザーアクセスを決定し、アクセス許可リクエストを送信します (「[が IAM と AWS Ground Station 連携する方法](#)」を参照)
- IAM 管理者 - アクセスを管理するためのポリシーを作成します (「[のアイデンティティベースのポリシーの例 AWS Ground Station](#)」を参照)

## アイデンティティを使用した認証

認証は、ID 認証情報 AWS を使用してサインインする方法です。、IAM ユーザー AWS アカウントのルートユーザー、または IAM ロールを引き受けることで認証される必要があります。

AWS IAM アイデンティティセンター (IAM Identity Center)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーテッド ID としてサインインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の「[AWS アカウントにサインインする方法](#)」を参照してください。

プログラムによるアクセスの場合、は SDK と CLI AWS を提供してリクエストを暗号化して署名します。詳細については、「IAM ユーザーガイド」の「[API リクエストに対するAWS 署名バージョン 4](#)」を参照してください。

## AWS アカウント ルートユーザー

を作成するときは AWS アカウント、すべての AWS のサービス および リソースへの完全なアクセス権を持つ AWS アカウント ルートユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザー認証情報を必要とするタスクについては、「IAM ユーザーガイド」の「[ルートユーザー認証情報が必要なタスク](#)」を参照してください。

## フェデレーテッドアイデンティティ

ベストプラクティスとして、人間のユーザーが一時的な認証情報 AWS のサービス を使用して にアクセスするには、ID プロバイダーとのフェデレーションを使用する必要があります。

フェデレーテッド ID は、エンタープライズディレクトリ、ウェブ ID プロバイダー、または ID Directory Service ソースの認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッドアイデンティティは、一時的な認証情報を提供するロールを引き受けます。

アクセスを一元管理する場合は、AWS IAM アイデンティティセンターをお勧めします。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「[IAM アイデンティティセンターとは](#)」を参照してください。

## IAM ユーザーとグループ

[IAM ユーザー](#)は、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの代わりに一時的な認証情報を使用することをお勧めします。詳細については、IAM ユーザーガイドの「[ID プロバイダーとのフェデレーションを使用してにアクセスする必要がある AWS](#)」を参照してください。

[IAM グループ](#)は、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「[IAM ユーザーに関するユースケース](#)」を参照してください。

## IAM ロール

[IAM ロール](#)は、特定のアクセス許可を持つアイデンティティであり、一時的な認証情報を提供します。ユーザーから [IAM ロール \(コンソール\)](#) に切り替えるか、または [API オペレーション](#) を呼び出すことで、[ロール](#) を引き受けることができます。AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「[ロールを引き受けるための各種方法](#)」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。詳細については、IAM ユーザーガイドの [IAM でのクロスアカウントリソースアクセス](#) を参照してください。

## ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられている場合のアクセス許可を定義します。は、プリンシパルがリクエストを行うときにこれらのポリシー AWS を評価します。ほとんど

のポリシーは JSON ドキュメント AWS としてに保存されます。JSON ポリシードキュメントの詳細については、「IAM ユーザーガイド」の「[JSON ポリシー概要](#)」を参照してください。

管理者は、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

## アイデンティティベースのポリシー

アイデンティティベースのポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイドの「[カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する](#)」を参照してください。

アイデンティティベースのポリシーは、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーとインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「[管理ポリシーとインラインポリシーのいずれかを選択する](#)」を参照してください。

## リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

## その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の上限を設定できる追加のポリシータイプをサポートしています。

- **アクセス許可の境界** – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。詳細については、「IAM ユーザーガイド」の「[IAM エンティティのアクセス許可境界](#)」を参照してください。

- サービスコントロールポリシー (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「[サービスコントロールポリシー](#)」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「[リソースコントロールポリシー \(RCP\)](#)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。詳細については、「IAM ユーザーガイド」の「[セッションポリシー](#)」を参照してください。

## 複数のポリシータイプ

1つのリクエストに複数のタイプのポリシーが適用されると、結果として作成されるアクセス許可を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、「IAM ユーザーガイド」の「[ポリシー評価ロジック](#)」を参照してください。

## が IAM と AWS Ground Station 連携する方法

IAM を使用して へのアクセスを管理する前に AWS Ground Station、 で使用できる IAM 機能を確認してください AWS Ground Station。

で使用できる IAM 機能 AWS Ground Station

IAM 機能	AWS Ground Station サポート
<a href="#">アイデンティティベースのポリシー</a>	あり
<a href="#">リソースベースのポリシー</a>	なし
<a href="#">ポリシーアクション</a>	あり
<a href="#">ポリシーリソース</a>	はい
<a href="#">ポリシー条件キー (サービス固有)</a>	はい
<a href="#">ACL</a>	なし

IAM 機能	AWS Ground Station サポート
<a href="#">ABAC (ポリシー内のタグ)</a>	あり
<a href="#">一時的な認証情報</a>	あり
<a href="#">プリンシパルアクセス権限</a>	あり
<a href="#">サービスロール</a>	いいえ
<a href="#">サービスリンクロール</a>	はい

AWS Ground Station およびその他の AWS のサービスがほとんどの IAM 機能とどのように連携するかの概要については、「IAM ユーザーガイド」の[AWS 「IAM と連携する のサービス」](#)を参照してください。

## のアイデンティティベースのポリシー AWS Ground Station

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「[カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する](#)」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「[IAM JSON ポリシーの要素のリファレンス](#)」を参照してください。

## のアイデンティティベースのポリシーの例 AWS Ground Station

AWS Ground Station アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS Ground Station](#)。

## 内のリソースベースのポリシー AWS Ground Station

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーテッドユーザー、またはを含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。詳細については、IAM ユーザーガイドの[IAM でのクロスアカウントリソースアクセス](#)を参照してください。

## のポリシーアクション AWS Ground Station

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

AWS Ground Station アクションのリストを確認するには、「サービス認可リファレンス」の「[で定義されるアクション AWS Ground Station](#)」を参照してください。

のポリシーアクションは、アクションの前に次のプレフィックス AWS Ground Station を使用します。

```
groundstation
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

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

AWS Ground Station アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS Ground Station](#)。

## のポリシーリソース AWS Ground Station

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、[Amazon リソースネーム \(ARN\)](#) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (\*) を使用します。

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

AWS Ground Station リソースタイプとその ARNs 「[で定義されるリソース AWS Ground Station](#)」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、「[AWS Ground Stationで定義されるアクション](#)」を参照してください。

AWS Ground Station アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS Ground Station](#)。

## のポリシー条件キー AWS Ground Station

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの[条件演算子](#)を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」の[AWS 「グローバル条件コンテキストキー」](#)を参照してください。

AWS Ground Station 条件キーのリストを確認するには、「サービス認可リファレンス」の「[の条件キー AWS Ground Station](#)」を参照してください。条件キーを使用できるアクションとリソースについては、「[で定義されるアクション AWS Ground Station](#)」を参照してください。

AWS Ground Station アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいの[アイデンティティベースのポリシーの例 AWS Ground Station](#)。

## ACLs AWS Ground Station

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

## を使用した ABAC AWS Ground Station

ABAC (ポリシー内のタグ) のサポート: あり

属性ベースのアクセス制御 (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの[条件要素](#)でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「[ABAC 認可でアクセス許可を定義する](#)」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「[属性ベースのアクセスコントロール \(ABAC\) を使用する](#)」を参照してください。

## での一時的な認証情報の使用 AWS Ground Station

一時的な認証情報のサポート: あり

一時的な認証情報は、AWS リソースへの短期的なアクセスを提供し、フェデレーションまたは切り替えロールを使用する場合に自動的に作成されます。AWS では、長期的なアクセスキーを使用する

代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「IAM ユーザーガイド」の「[IAM の一時的な認証情報](#)」および「[AWS のサービスと IAM との連携](#)」を参照してください。

## のクロスサービスプリンシパルのアクセス許可 AWS Ground Station

転送アクセスセッション (FAS) のサポート: あり

転送アクセスセッション (FAS) は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする を使用します。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。

## のサービスロール AWS Ground Station

サービスロールのサポート: なし

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける [IAM ロール](#)です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの [AWS のサービスに許可を委任するロールを作成する](#) を参照してください。

### Warning

サービスロールのアクセス許可を変更すると、AWS Ground Station 機能が破損する可能性があります。AWS Ground Station が指示する場合にのみ、サービスロールを編集します。

## のサービスにリンクされたロール AWS Ground Station

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「[IAM と提携するAWS のサービス](#)」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つ

けます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

## のアイデンティティベースのポリシーの例 AWS Ground Station

デフォルトでは、ユーザーおよびロールには、AWS Ground Station リソースを作成または変更する権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーを作成する \(コンソール\)](#)」を参照してください。

各リソースタイプの ARN の形式など AWS Ground Station、で定義されるアクションとリソースタイプの詳細については、「サービス認可リファレンス」の「[のアクション、リソース、および条件キー AWS Ground Station](#)」を参照してください。ARNs

### トピック

- [ポリシーに関するベストプラクティス](#)
- [AWS Ground Station コンソールの使用](#)
- [自分の権限の表示をユーザーに許可する](#)

### ポリシーに関するベストプラクティス

ID ベースのポリシーは、アカウント内の AWS Ground Station リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションでは、AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらはで使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイドの [AWS マネージドポリシー](#) または [ジョブ機能のAWS マネージドポリシー](#) を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアク

ションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイドの [IAM でのポリシーとアクセス許可](#) を参照してください。

- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションがなどの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイドの [IAM JSON ポリシー要素:条件](#) を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイドの [IAM Access Analyzer でポリシーを検証する](#) を参照してください。
- 多要素認証 (MFA) を要求する - IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイドの [MFA を使用した安全な API アクセス](#) を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイドの [IAM でのセキュリティのベストプラクティス](#) を参照してください。

## AWS Ground Station コンソールの使用

AWS Ground Station コンソールにアクセスするには、最小限のアクセス許可のセットが必要です。これらのアクセス許可により、 の AWS Ground Station リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーとロールが AWS Ground Station 引き続きコンソールを使用できるようにするには、エンティティに AWS Ground Station *ConsoleAccess* または *ReadOnly* AWS 管理ポリシーもアタッチ

します。詳細については、「IAM ユーザーガイド」の「[ユーザーへのアクセス許可の追加](#)」を参照してください。

## 自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
  "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 ID とアクセスのトラブルシューティング

次の情報は、 および IAM の使用時に発生する可能性がある一般的な問題の診断 AWS Ground Station と修復に役立ちます。

### トピック

- [でアクションを実行する権限がありません AWS Ground Station](#)
- [iam:PassRole を実行する権限がありません](#)
- [自分の 以外のユーザーに自分の AWS Ground Station リソース AWS アカウント へのアクセスを許可したい](#)

### でアクションを実行する権限がありません AWS Ground Station

アクションを実行する権限がないというエラーが表示された場合は、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要な `groundstation:GetWidget` アクセス許可を持っていない場合に発生するものです。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
groundstation:GetWidget on resource: my-example-widget
```

この場合、`groundstation:GetWidget` アクションを使用して *my-example-widget* リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

### iam:PassRole を実行する権限がありません

`iam:PassRole` アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して AWS Ground Station にロールを渡すことができるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、既存のロールをそのサービスに渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して AWS Ground Station でアクションを実行しようとする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

### 自分の 以外のユーザーに自分の AWS Ground Station リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- がこれらの機能 AWS Ground Station をサポートしているかどうかを確認するには、「」を参照してください [IAM と AWS Ground Station 連携する方法](#)。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、「[IAM ユーザーガイド](#)」の「[所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する](#)」を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの「[サードパーティー AWS アカウント が所有する へのアクセスを提供する](#)」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの [外部で認証されたユーザー \(ID フェデレーション\) へのアクセスの許可](#) を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用法の違いについては、「IAM ユーザーガイド」の「[IAM でのクロスアカウントのリソースへのアクセス](#)」を参照してください。

## AWS の 管理ポリシー AWS Ground Station

AWS 管理ポリシーは、によって作成および管理されるスタンドアロンポリシーです AWS。AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できるように、多くの一般的なユースケースにアクセス許可を付与するように設計されています。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有の [カスタマー管理ポリシー](#) を定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、AWS マネージドポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「[AWS マネージドポリシー](#)」を参照してください。

### AWS マネージドポリシー: AWSGroundStationAgentInstancePolicy

AWSGroundStationAgentInstancePolicy ポリシーを IAM アイデンティティにアタッチできます。

このポリシーは、Amazon EC2 インスタンスに AWS Ground Station エージェントアクセス許可を付与し、インスタンスが Ground Station の問い合わせ中にデータを送受信できるようにします。このポリシーのすべてのアクセス許可は、Ground Station サービスからのものです。

#### アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `groundstation` — データフローエンドポイントインスタンスが Ground Station Agent API を呼び出すことを許可します。

JSON ポリシードキュメントの最新バージョンを表示するには、AWS [AWSGroundStationAgentInstancePolicy](#)」を参照してください。

## AWS マネージドポリシー:

### AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy

AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy を IAM エンティティにアタッチすることはできません。このポリシーは、がユーザーに代わってアクションを実行できるようにするサービスにリンクされたロール AWS Ground Station にアタッチされます。詳細については、「[サービスにリンクされたロールの使用](#)」を参照してください。

このポリシーは、がパブリック IPv4 アドレスを検索できるようにする EC2 アクセス許可を付与 AWS Ground Station します。

#### アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `ec2:DescribeAddresses` – がユーザーに代わって EIP に関連付けられているすべての IPs を一覧表示 AWS Ground Station できるようにします。EIPs
- `ec2:DescribeNetworkInterfaces` – ユーザーに代わって EC2 インスタンスに関連付けられたネットワークインターフェイスに関する情報の取得 AWS Ground Station を許可します。

JSON ポリシードキュメントの最新バージョンを確認するには、AWS [AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy](#)」を参照してください。

## AWS Ground Station AWS 管理ポリシーの更新

このサービスがこれらの変更の追跡を開始 AWS Ground Station してからの AWS の管理ポリシーの更新に関する詳細を表示します。このページの変更に関する自動アラートについては、AWS Ground Station ドキュメント履歴ページの RSS フィードにサブスクライブしてください。

変更	説明	日付
<a href="#">AWSGroundStationAgentInstancePolicy</a> – 既存のポリシーの更新	AWS Ground Station は、エージェントが拡張 Ground Station オペレーションのタスクレスポンス URLs を取得できるようにする新しいアクセス許可を追加しました。	2025 年 11 月 13 日
<a href="#">AWSGroundStationAgentInstancePolicy</a> — 新しいポリシー	AWS Ground Station は、AWS Ground Station エージェントを使用するためのアクセス許可をデータフローエンドポイントインスタンスに提供する新しいポリシーを追加しました。	2023 年 4 月 12 日
<a href="#">AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy</a> — 新しいポリシー	AWS Ground Station は、が EIP に関連付けられたパブリック IPv4 アドレスと EC2 インスタンスに関連付けられたネットワークインターフェイスを検索 AWS Ground Station できるようにする EC2 アクセス許可を付与する新しいポリシーを追加しました。 EIPs	2022 年 11 月 2 日
AWS Ground Station が変更の追跡を開始しました	AWS Ground Station は、AWS 管理ポリシーの変更の追跡を開始しました。	2021 年 3 月 1 日

## Ground Station のサービスにリンクされたロールを使用する

AWS Ground Station は AWS Identity and Access Management (IAM) [サービスにリンクされたロール](#)を使用します。サービスにリンクされたロールは、Ground Station に直接リンクされた一意のタイ

プの IAM ロールです。サービスにリンクされたロールは Ground Station によって事前定義されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用することで、必要なアクセス許可を手動で追加する必要がなくなるため、Ground Station の設定が簡単になります。Ground Station は、サービスにリンクされたロールのアクセス許可を定義し、他の定義がされている場合を除き、Ground Station のみがそのロールを引き受けることができます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスにリンクされたロールをサポートする他のサービスの詳細については、[AWS 「IAM と連携するサービス」](#)を参照し、「サービスにリンクされたロール」列で「はい」があるサービスを探します。サービスリンクロールに関するドキュメントをサービスで表示するには、リンクで [はい] を選択します。

## Ground Station のサービスにリンクされたロールのアクセス許可

Ground Station は、AWSServiceRoleForGroundStationDataflowEndpointGroup というサービスにリンクされたロールを使用します – AWS GroundStation は、このサービスにリンクされたロールを使用して EC2 を呼び出し、パブリック IPv4 アドレスを検索します。

AWSServiceRoleForApplicationDiscoveryServiceContinuousExport という、サービスにリンクされたロールは、以下のサービスを信頼してロールを引き受けます。

- `groundstation.amazonaws.com`

AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy というロールのアクセス許可ポリシーでは、Ground Station は、指定されたリソースで次のアクションを完了することができます。

- アクション: `ec2:DescribeAddresses`。対象リソース: `all AWS resources (*)`

アクションにより、Ground Station は EIP に関連付けられているすべての IP を一覧表示できます。

- アクション: `ec2:DescribeNetworkInterfaces`。対象リソース: `all AWS resources (*)`

アクションにより、Ground Station は EC2 インスタンスに関連付けられたネットワークインターフェイスに関する情報を取得できます

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するにはアクセス許可を設定する必要があります。詳細については IAM ユーザーガイドの「[サービスにリンクされた役割のアクセス許可](#)」を参照してください。

## Ground Station へのサービスにリンクされたロールの作成

サービスリンクロールを手動で作成する必要はありません。AWS CLI または AWS API で `DataflowEndpointGroup` を作成すると、Ground Station によってサービスにリンクされたロールが作成されます。

このサービスリンクロールを削除した後で再度作成する必要がある場合は同じ方法でアカウントにロールを再作成できます。`DataflowEndpointGroup` を作成すると、Ground Station によってサービスにリンクされたロールが再作成されます。

IAM コンソールを使用して、Amazon EC2 へのデータ配信ユースケースで、サービスにリンクされたロールを作成することもできます。AWS CLI または AWS API で、サービス名を使用して `groundstation.amazonaws.com` サービスにリンクされたロールを作成します。詳細については、IAM ユーザーガイドの「[サービスリンクロールの作成](#)」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

## Ground Station でのサービスにリンクされたロールの編集

Ground Station では、サービスにリンクされたロールである `AWSServiceRoleForGroundStationDataflowEndpointGroup` を編集できません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「[サービスリンクロールの編集](#)」を参照してください。

## Ground Station でのサービスにリンクされたロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、モニタリングや保守が積極的に行われていない未使用のエンティティを排除できます。

サービスにリンクされたロールは、サービスにリンクされたロールを使用して `DataflowEndpointGroups` を削除した後でしか削除できません。これにより、`DataflowEndpointGroups` に対するアクセス許可を誤って取り消すことがなくなります。サービスにリンクされたロールが複数の `DataflowEndpointGroups` で使用されている場合、サービスにリンクされたロールを削除する前に、そのロールを使用するすべての `DataflowEndpointGroups` を削除する必要があります。

**Note**

リソースを削除する際に、Ground Station のサービスでそのロールが使用されている場合、削除は失敗することがあります。失敗した場合は数分待ってから操作を再試行してください。

AWSServiceRoleForGroundStationDataflowEndpointGroup によって使用される Ground Station リソースを削除するには

- AWS CLI または AWS API を使用して DataflowEndpointGroups を削除します。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、AWS CLI または AWS API を使用して、AWSServiceRoleForGroundStationDataflowEndpointGroup サービスにリンクされたロールを削除します。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの削除](#)」を参照してください。

## Ground Station のサービスにリンクされたロールがサポートされるリージョン

Ground Station は、サービスが利用可能なすべてのリージョンで、サービスにリンクされたロールの使用をサポートします。詳細については、「[リージョン表](#)」を参照してください。

## トラブルシューティング

NOT\_AUTHORIZED\_TO\_CREATE\_SLR - これは、CreateDataflowEndpointGroup API の呼び出しに使用されているアカウントのロールに iam:CreateServiceLinkedRole のアクセス許可がないことを示しています。iam:CreateServiceLinkedRole のアクセス許可を持つ管理者は、アカウントのサービスにリンクされたロールを手動で作成する必要があります。

## の保管中のデータ暗号化 AWS Ground Station

AWS Ground Station はデフォルトで暗号化を提供し、AWS 所有の暗号化キーを使用して保管中の機密データを保護します。

- AWS 所有キー - デフォルトでは、これらのキー AWS Ground Station を使用して、個人が直接識別可能なデータとエフェメリスを自動的に暗号化します。AWS所有キーを表示、管理、または使用したり、その使用を監査したりすることはできません。ただし、データを暗号化するキーを保護するためにアクションを実行したりプログラムを変更したりする必要はありません。詳細については、「[AWS Key Management Service デベロッパーガイド](#)」の[AWS「所有キー」](#)を参照してください。

保管中のデータをデフォルトで暗号化することで、機密データの保護におけるオーバーヘッドと複雑な作業を減らすのに役立ちます。同時に、セキュリティを重視したアプリケーションを構築して、暗号化のコンプライアンスと規制の厳格な要件を満たすことができます。

AWS Ground Station は、すべての機密性の高い保管時のデータに対して暗号化を適用しますが、エフェメリスなど一部の AWS Ground Station リソースでは、デフォルトのマネージドキーの代わりにカスタマー AWS マネージドキーを使用することを選択できます。

- カスタマー マネージドキー -- 既存の AWS 所有暗号化の代わりに作成、所有、管理する対称カスタマー マネージドキーの使用 AWS Ground Station をサポートします。この暗号化層はユーザーが完全に制御できるため、次のようなタスクを実行できます。
  - キーポリシーの策定と維持
  - IAM ポリシーとグラントの策定と維持
  - キーポリシーの有効化と無効化
  - キー暗号化マテリアルのローテーション
  - タグを追加する
  - キーエイリアスの作成
  - キー削除のスケジュール設定

詳細については、「[AWS Key Management Service デベロッパーガイド](#)」の「[カスタマー マネージドキー](#)」を参照してください。

次の表は、がカスタマー マネージドキーの使用 AWS Ground Station をサポートしているリソースをまとめたものです。

データ型	AWS 所有キーの暗号化	カスタマーマネージドキーの暗号化 (オプション)
衛星の軌跡の計算に使用されるエフェメリスデータ	有効	有効
アンテナのコマンドに使用される方位高度エフェメリス	有効	有効

### Note

AWS Ground Station は、を使用して保管時の暗号化を自動的に有効に AWS 所有のキーし、個人を特定できるデータを無償で保護します。ただし、カスタマーマネージドキーの使用には AWS KMS 料金が適用されます。料金の詳細については、「[AWS Key Management Service 料金](#)」を参照してください。

詳細については AWS KMS、「[AWS Key Management Service デベロッパーガイド](#)」を参照してください。

各リソースタイプに固有の情報については、以下を参照してください。

- [TLE および OEM エフェメリスデータの保管時の暗号化](#)
- [方位高度エフェメリスの保管時の暗号化](#)

## カスタマーマネージドキーを作成する

対称カスタマーマネージドキーは AWS マネジメントコンソール、または AWS KMS APIs を使用して作成できます。

対称カスタマーマネージドキーを作成するには

「[AWS Key Management Service デベロッパーガイド](#)」の「対称カスタマーマネージドキーの作成」の手順に従います。

## キーポリシーの概要

キーポリシーは、カスタマーマネージドキーへのアクセスを制御します。すべてのカスタマーマネージドキーには、キーポリシーが1つだけ必要です。このポリシーには、そのキーを使用できるユーザーとその使用方法を決定するステートメントが含まれています。キーポリシーは、カスタマーマネージドキーの作成時に指定できます。詳細については、「AWS Key Management Service デベロッパーガイド」の「[カスタマーマネージドキーへのアクセスの管理](#)」を参照してください。

AWS Ground Station リソースでカスタマーマネージドキーを使用するには、AWS Ground Station サービスに適切なアクセス許可を付与するようにキーポリシーを設定する必要があります。特定のアクセス許可とポリシー設定は、暗号化するリソースのタイプによって異なります。

- TLE および OEM エフェメリスデータ - 特定のキーポリシーの要件と例[TLE および OEM エフェメリスデータの保管時の暗号化](#)については、「」を参照してください。
- 方位標高エフェメリスデータ - 特定のキーポリシーの要件と例[方位高度エフェメリスの保管時の暗号化](#)については、「」を参照してください。

### Note

キーポリシーの設定はエフェメリスタイプによって異なります。TLE および OEM エフェメリスデータはキーアクセスに許可を使用し、アジマス昇格エフェメリスは直接キーポリシー許可を使用します。暗号化する特定のリソースタイプに従ってキーポリシーを設定してください。

[ポリシーでのアクセス許可の指定とキーアクセスのトラブルシューティングの詳細については、「AWS Key Management Service デベロッパーガイド」を参照してください。](#) <https://docs.aws.amazon.com/kms/latest/developerguide/policy-evaluation.html#example-no-iam>

## のカスタマーマネージドキーの指定 AWS Ground Station

カスタマーマネージドキーを指定して、次のリソースを暗号化できます。

- エフェメリス (TLE、OEM、方位高度)

リソースを作成するときに kmsKeyArn を提供することでデータキーを指定できます。

- kmsKeyArn - AWS KMS カスタマーマネージド[キーのキー識別子](#)

## AWS Ground Station 暗号化コンテキスト

[暗号化コンテキスト](#)とは、データに関する追加のコンテキスト情報を含むために、使用する (オプションの) キーと値のペアのセットです。AWS KMS は、暗号化コンテキストを追加の認証済みデータとして使用し、暗号化の認証をサポートします。データを暗号化するリクエストに暗号化コンテキストを含めると、は暗号化コンテキストを暗号化されたデータに AWS KMS バインドします。データを復号化するには、そのリクエストに (暗号化時と) 同じ暗号化コンテキストを含めます。

AWS Ground Station は、暗号化されるリソースに応じて異なる暗号化コンテキストを使用し、作成されたキー許可ごとに特定の暗号化コンテキストを指定します。

リソース固有の暗号化コンテキストの詳細については、以下を参照してください。

- [TLE および OEM エフェメリスデータの保管時の暗号化](#)
- [方位高度エフェメリスの保管時の暗号化](#)

### TLE および OEM エフェメリスデータの保管時の暗号化

#### TLE および OEM エフェメリスの主要なポリシー要件

エフェメリスデータでカスターマネージドキーを使用するには、キーポリシーが AWS Ground Station サービスに次のアクセス許可を付与する必要があります。

- [kms:CreateGrant](#) - カスターマネージドキーにアクセス許可を作成します。暗号化されたデータを読み取って保存するための、カスターマネージドキーに対する[許可オペレーション](#)を実行する AWS Ground Station アクセス許可を付与します。
- [kms:DescribeKey](#) - カスターマネージドキーの詳細を提供し、提供されたキーを使用する前に AWS Ground Station がキーを検証できるようにします。

Grants [の使用](#)の詳細については、「AWS Key Management Service デベロッパーガイド」を参照してください。

#### カスターマネージドキーを使用してエフェメリスを作成するための IAM ユーザーアクセス許可

が暗号化オペレーションでカスターマネージドキー AWS Ground Station を使用する場合、エフェメリスリソースを作成するユーザーに代わって動作します。

カスタマーマネージドキーを使用してエフェメリスリソースを作成するには、ユーザーはカスタマーマネージドキーで次のオペレーションを呼び出すアクセス許可を持っている必要があります。

- [kms:CreateGrant](#) - ユーザーが に代わってカスタマーマネージドキーに許可を作成できるようにします AWS Ground Station。
- [kms:DescribeKey](#) - ユーザーがカスタマーマネージドキーの詳細を表示してキーを検証できるようにします。

これらの必要なアクセス許可は、キーポリシーか、キーポリシーで許可されている場合は IAM ポリシーで指定できます。これらのアクセス許可により、ユーザーはユーザーに代わって暗号化オペレーションにカスタマーマネージドキーを使用すること AWS Ground Station を に許可できます。

## がエフェメリス AWS KMS の で許可 AWS Ground Station を使用する方法

AWS Ground Station には、カスタマーマネージド [キーを使用するためのキー許可](#) が必要です。

カスタマーマネージドキーで暗号化されたエフェメリスをアップロードすると、 は [CreateGrant](#) リクエストを送信して、ユーザーに代わってキー許可 AWS Ground Station を作成します AWS KMS。の権限 AWS KMS は、アカウントの AWS KMS キーへのアクセスを許可する AWS Ground Station ために使用されます。

これにより、AWS Ground Station は以下を実行できます。

- [GenerateDataKey](#) を呼び出して暗号化されたデータキーを生成して保存します。
- [Decrypt](#) を呼び出して、保存された暗号化されたデータキーを使用して暗号化されたデータにアクセスします。
- [Encrypt](#) を呼び出して、データキーを使用してデータを暗号化します。
- サービスに [RetireGrant](#) を許可するには、リタイアプリンシパルを設定します。

許可へのアクセスはいつでも取り消すことができます。その場合、カスタマーマネージドキーによって暗号化されたデータにはアクセス AWS Ground Station できず、そのデータに依存するオペレーションに影響します。たとえば、コンタクトに現在使用されているエフェメリスからキー許可を削除すると、AWS Ground Station はコンタクト中にアンテナをポイントするために提供されたエフェメリスデータを使用できなくなります。これにより、コンタクトは FAILED 状態で終了します。

## Ephemeris 暗号化コンテキスト

エフェメリスリソースを暗号化するためのキー許可は、特定の衛星 ARN にバインドされます。

```
"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

キーグ許可は同じキーと衛星のペアに再利用されます。

## 暗号化コンテキストによるモニタリングに暗号化コンテキストを使用する

対称カスターマネージドキーを使用してエメリフィスを暗号化する場合は、監査レコードとログで暗号化コンテキストを使用して、カスターマネージドキーがどのように使用されているかを特定することもできます。暗号化コンテキストは、[AWS CloudTrail](#) または [Amazon CloudWatch Logs](#) によって生成されたログにも表示されます。

## 暗号化コンテキストを使用してカスターマネージドキーへのアクセスを制御する

対称カスターマネージドキー (CMK) へのアクセスを制御するための conditions として、キーポリシーと IAM ポリシー内の暗号化コンテキストを使用することができます。グラントに暗号化コンテキストの制約を使用することもできます。

AWS Ground Station は、権限の暗号化コンテキスト制約を使用して、アカウントまたはリージョンのカスターマネージドキーへのアクセスを制御します。グラントの制約では、指定された暗号化コンテキストの使用をグラントが許可するオペレーションが必要です。

次に、特定の暗号化コンテキストのカスターマネージドキーへのアクセスを付与するキーポリシーステートメントの例を示します。このポリシーステートメントの条件では、暗号化コンテキストを指定する暗号化コンテキスト制約がグラントに必要です。

次の例は、衛星にバインドされたエフェメリスデータのキーポリシーを示しています。

### 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"
    }
  }
}
]
```

## エフェメリスの暗号化キーのモニタリング

エフェメリスリソースで AWS Key Management Service カスタマーマネージドキーを使用する場合、[AWS CloudTrail](#)または [Amazon CloudWatch logs](#)を使用して、が AWS Ground Station に送信するリクエストを追跡できます AWS KMS。次の例は、[CreateGrant](#)、[GenerateDataKey](#)、[Decrypt](#)、[DescribeKey](#) の CloudTrail イベントで、カスタマーマネージドキーによって暗号化されたデータにアクセス AWS Ground Station するために によって呼び出されるオペレーションをモニタリング AWS KMS します。

### CreateGrant

AWS KMS カスタマーマネージドキーを使用してエフェメリスリソースを暗号化すると、はユーザーに代わって [CreateGrant](#) リクエスト AWS Ground Station を送信して AWS 、アカウントの AWS KMS キーにアクセスします。が AWS Ground Station 作成する権限は、AWS KMS カス

タマネーagementドキーに関連付けられたリソースに固有です。さらに、AWS Ground Station リソースを削除すると、は [RetireGrant](#) オペレーションを使用して許可を削除します。

次のイベント例では、エフェメリスの [CreateGrant](#) オペレーションを記録します。

```
{
  "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 カスタマーマネージドキーを使用してエフェメリスリソースを暗号化すると、はユーザーに代わって [DescribeKey](#) リクエスト AWS Ground Station を送信し、リクエストされたキーがアカウントに存在することを確認します。

次のイベント例では、[DescribeKey](#) オペレーションを記録します。

```

{
  "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 カスタマーマネージドキーを使用してエフェメリスリソースを暗号化すると、は [GenerateDataKey](#) リクエストを AWS Ground Station に送信して、データを暗号化するデータキーを生成します。

次のイベント例では、エフェメリスの [GenerateDataKey](#) オペレーションを記録します。

```
{
  "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 カスタマーマネージドキーを使用してエフェメリスリソースを暗号化する場合、は [Decrypt](#) オペレーション AWS Ground Station を使用して、提供されたエフェメリスが既に同じカスタマーマネージドキーで暗号化されている場合に復号します。例えば、エフェメリスが S3 バケットからアップロードされ、そのバケット内で特定のキーで暗号化されているとします。

次のイベント例では、エフェメリスの [Decrypt](#) オペレーションを記録します。

```

{
  "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"
}
```

## 方位高度エフェメリスの保管時の暗号化

### 方位高度エフェメリスの主要なポリシー要件

方位標高エフェメリスデータでカスタマーマネージドキーを使用するには、キーポリシーが AWS Ground Station サービスに次のアクセス許可を付与する必要があります。許可を使用する TLE および OEM エフェメリスデータとは異なり、方位高度エフェメリスは暗号化オペレーションにダイレクトキーポリシーアクセス許可を使用します。これは、のアクセス許可を管理し、キーを使用するためのより簡単な方法です。

- [kms:GenerateDataKey](#) - 方位高度エフェメリスデータを暗号化するためのデータキーを生成します。
- [kms:Decrypt](#) - 方位角エフェメリスデータにアクセスするときに、暗号化されたデータキーを復号します。

カスタマーマネージドキー AWS Ground Station へのアクセスを許可するキーポリシーの例

#### Note

方位高度エフェメリスでは、キーポリシーでこれらのアクセス許可を直接設定する必要があります。リージョン AWS Ground Station サービスプリンシパル (など `groundstation.region.amazonaws.com`) には、キーポリシーステートメントでこれ

らのアクセス許可を付与する必要があります。これらのステートメントをキーポリシーに追加しないと AWS Ground Station、カスタム方位高度エフェメリスを保存またはアクセスできなくなります。

## 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": "*"
    }
  ]
}
```

## カスタマーマネージドキーを使用して方位高度エフェメリスを作成するための IAM ユーザーアクセス許可

は、暗号化オペレーションでカスタマーマネージドキー AWS Ground Station を使用する場合、方位角エフェメリスリソースを作成するユーザーに代わって動作します。

カスタマーマネージドキーを使用して方位角エフェメリスリソースを作成するには、ユーザーはカスタマーマネージドキーで次のオペレーションを呼び出すアクセス許可を持っている必要があります。

- [kms:GenerateDataKey](#) - ユーザーが方位高度エフェメリスデータを暗号化するためのデータキーを生成できるようにします。
- [kms:Decrypt](#) - 方位角エフェメリスデータにアクセスするときに、ユーザーがデータキーを復号化できるようにします。
- [kms:DescribeKey](#) - ユーザーがカスタマーマネージドキーの詳細を表示してキーを検証できるようにします。

これらの必要なアクセス許可は、キーポリシーか、キーポリシーで許可されている場合は IAM ポリシーで指定できます。これらのアクセス許可により、ユーザーはユーザーに代わって暗号化オペレーションにカスタマーマネージドキーを使用すること AWS Ground Station を に許可できます。

## が方位高度エフェメリスのキーポリシー AWS Ground Station を使用する方法

カスタマーマネージドキーを使用して方位角エフェメリスデータを提供すると、はキーポリシー AWS Ground Station を使用して暗号化キーにアクセスします。アクセス許可は、TLE または OEM エフェメリスデータと同様に、グラントではなく、AWS Ground Station キーポリシーステートメントを通じてに直接付与されます。

カスタマーマネージドキー AWS Ground Station へのアクセスを削除した場合、そのキーによって暗号化されたデータにはアクセス AWS Ground Station できず、そのデータに依存するオペレーションに影響します。例えば、コンタクトに現在使用されている方位標高エフェメリスのキーポリシー許可を削除すると、AWS Ground Station はコンタクト中にアンテナをコマンドするために提供された方位標高データを使用できなくなります。これにより、コンタクトは FAILED 状態で終了します。

## 方位高度エフェメリス暗号化コンテキスト

が AWS KMS キー AWS Ground Station を使用して方位高度エフェメリスデータを暗号化する場合、サービスは[暗号化コンテキスト](#)を指定します。暗号化コンテキストは、データの整合性を確保するために が AWS KMS 使用する追加の認証データ (AAD) です。暗号化オペレーションで暗号化コンテキストを指定すると、サービスは復号オペレーションでも同じ暗号化コンテキストを指定する必要があります。そうしないと、復号は失敗します。暗号化コンテキストは CloudTrail ログにも書き込まれ、特定の AWS KMS キーが使用された理由を理解するのに役立ちます。CloudTrail ログには、AWS KMS キーの使用を説明するエントリが多数含まれている場合がありますが、各ログエントリの暗号化コンテキストは、その特定の使用理由を判断するのに役立ちます。

AWS Ground Station は、方位角エフェメリスでカスターマネージドキーを使用して暗号化オペレーションを実行するときに、次の暗号化コンテキストを指定します。

```
{
  "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"
  }
}
```

暗号化コンテキストには以下が含まれます。

`aws:groundstation:ground-station-id`

方位高度エフェメリスに関連付けられた地上局の名前。

`aws:groundstation:arn`

エフェメリスリソースの ARN。

`aws:s3:arn`

Amazon S3 に保存されているエフェメリスの ARN。

## 暗号化コンテキストを使用してカスターマネージドキーへのアクセスを制御する

IAM 条件ステートメントを使用して、カスターマネージドキー AWS Ground Station へのアクセスを制御できます。および `kms:Decrypt` アクションに条件ステートメントを追加する `kms:GenerateDataKey` と、AWS KMS を使用できる地上局が制限されます。

以下は、特定の地上局の特定のリージョンのカスターマネージドキー AWS Ground Station へのアクセスを許可するためのキーポリシーステートメントの例です。このポリシーステートメントの条件では、キーポリシーの条件に一致する暗号化コンテキストを指定するキーへのすべての暗号化および復号アクセスが必要です。

## 特定の地上局のカスタマーマネージドキー AWS Ground Station へのアクセスを許可するキーポリシーの例

### 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"
        }
      }
    }
  ]
}
```

## 複数の地上局のカスタマーマネージドキー AWS Ground Station へのアクセスを許可するキーポリシーの例

### 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"
            ]
        }
      }
    }
  ]
}
```

## 方位高度エフェメリスの暗号化キーのモニタリング

方位高度エフェメリスリソースで AWS KMS カスタマーマネージドキーを使用する場合、[CloudTrail](#) または [CloudWatch ログ](#) を使用してが AWS Ground Station 送信するリクエストを追跡できます AWS KMS。次の例は、[GenerateDataKey](#) と [Decrypt](#) の CloudTrail イベントで、カスタマーマネージドキーによって暗号化されたデータにアクセス AWS Ground Station するために によって呼び出されるオペレーションをモニタリング AWS KMS します。

### GenerateDataKey

AWS KMS カスタマーマネージドキーを使用して方位高度エフェメリスリソースを暗号化すると、は [GenerateDataKey](#) リクエストを AWS Ground Station に送信 AWS KMS して、データを暗号化するデータキーを生成します。

次のイベント例では、方位高度エフェメリスの [GenerateDataKey](#) オペレーションを記録します。

```
{
  "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 カスタマーマネージドキーを使用して方位角エフェメリスリソースを暗号化する場合、は [Decrypt](#) オペレーション AWS Ground Station を使用して、同じカスタマーマネージドキーで既に暗号化されている場合に提供された方位角エフェメリスデータを復号します。

次のイベント例では、方位高度エフェメリスの [Decrypt](#) オペレーションを記録します。

```
{
  "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"
}
```

## の転送中のデータ暗号化 AWS Ground Station

AWS Ground Station は、転送中に機密データを保護するための暗号化をデフォルトで提供します。データは、ミッションプロファイルの設定に応じて、2つの方法で AWS Ground Station アンテナロケーションと Amazon EC2 インスタンス間でストリーミングできます。

- AWS Ground Station エージェント
- データフローエンドポイント

ストリーミングデータの各方法は、転送中のデータの暗号化を異なる方法で処理します。以降のセクションでは、それぞれの方法について説明します。

### AWS Ground Station エージェントストリーム

AWS Ground Station エージェントは、カスタマーマネージド AWS KMS キーを使用してストリームを暗号化します。Amazon EC2 インスタンスで実行されている AWS Ground Station エージェントは、ストリームを自動的に復号化して、復号化されたデータを提供します。

ストリームの暗号化に使用される AWS KMS キーは、[streamsKmsKey](#) パラメータ MissionProfile で作成するときに指定されます。キー AWS Ground Station へのアクセスを許可するすべてのアクセス許可は、 にアタッチされた AWS KMS キーポリシーを通じて処理されず streamsKmsKey。

## データフローエンドポイントストリーム

データフローエンドポイントストリームは、[Datagram Transport Layer Security \(DTLS\)](#) を使用して暗号化されます。これは自己署名証明書を使用して行われ、追加の設定は必要ありません。

## ミッションプロファイル設定の例

提供された例は、パブリックブロードキャスト衛星を取得し、それをサポートするミッションプロファイルを作成する方法を示しています。結果のテンプレートは、パブリックブロードキャスト衛星コンタクトを取りやすくし、衛星に関する意思決定に役立つように提供されています。

### トピック

- [JPSS-1 - パブリックブロードキャスト衛星 \(PBS\) - 評価](#)
- [Amazon S3 データ配信を利用するパブリックブロードキャスト衛星](#)
- [データフローエンドポイントを利用するパブリックブロードキャスト衛星 \(ナローバンド\)](#)
- [データフローエンドポイントを利用するパブリックブロードキャスト衛星 \(復調および復号化\)](#)
- [AWS Ground Station エージェント \(広帯域\) を利用するパブリックブロードキャスト衛星](#)

## JPSS-1 - パブリックブロードキャスト衛星 (PBS) - 評価

この例のセクションは、と一致します [顧客のオンボーディングプロセスの概要](#)。との簡単な互換性分析を提供し AWS Ground Station、以下の特定の例のステージを設定します。

[パブリックブロードキャスト衛星](#) 「」セクションで説明したように、公開されている一部の衛星、または衛星の通信パスを使用できます。このセクションでは、[JPSS-1](#) を AWS Ground Station 用語で説明します。参考までに、[Joint Polar Satellite System 1 \(JPSS-1\) Spacecraft High Rate Data \(HRD\) to Direct Broadcast Stations \(DBS\) Radio Frequency \(RF\) Interface Control Document \(ICD\)](#) を使用して例を完成させます。また、JPSS-1 は NORAD ID 43013 に関連付けられていることに注意してください。

JPSS-1 衛星は、ICD の図 1-1 に示すように、1つのアップリンクと3つのダイレクトダウンリンク通信パスを提供します。これら4つの通信パスのうち、パブリックに使用できるのは1つのハイレートデータ (HRD) ダウンリンク通信パスのみです。これに基づいて、このパスにはより具体的なデータも関連付けられることがわかります。4つのパスは次のとおりです。

- 2067.27 MHz の中心周波数で、データレートが 2~128 kbps のコマンドパス (アップリンク)。このパスはパブリックにアクセスできません。
- データレートが 1~524 kbps で、中心周波数 2247.5 MHz のテレメトリパス (ダウンリンク)。このパスはパブリックにアクセスできません。
- データレートが 150~300 Mbps の 26.7034 GHz 中心周波数の SMD パス (ダウンリンク)。このパスはパブリックにアクセスできません。

- 7812 MHz の中心周波数で、データレートが 15 Mbps の HRD パス (ダウンリンク) の RF。帯域幅は 30 MHz で、right-hand-circular-polarized。JPSS-1 を にオンボードすると AWS Ground Station、これはアクセス可能な通信パスです。この通信パスには、計測科学データ、計測エンジニアリングデータ、計測テレメトリデータ、リアルタイム宇宙船ハウスキーピングデータが含まれます。

潜在的なデータパスを比較すると、コマンド (アップリンク)、テレメトリ (ダウンリンク)、および HRD (ダウンリンク) パスが、の頻度、帯域幅、およびマルチチャンネル同時使用機能を満たしていることがわかります AWS Ground Station。中心周波数が既存のレシーバーの範囲外であるため、SMD パスには互換性がありません。サポートされている機能の詳細については、「」を参照してください [AWS Ground Station サイト機能](#)。

#### Note

SMD パスは互換性がないため AWS Ground Station、設定例には表示されません。

#### Note

コマンド (アップリンク) パスとテレメトリ (ダウンリンク) パスは ICD で定義されておらず、パブリックに使用できないため、使用されるときに提供される値は概念的 です。

## Amazon S3 データ配信を利用するパブリックブロードキャスト衛星

この例では、ユーザーガイドの [JPSS-1 - パブリックブロードキャスト衛星 \(PBS\) - 評価](#) セクションで行った分析に基づいています。

この例では、シナリオを想定する必要があります。つまり、HRD 通信パスをデジタル中間周波数としてキャプチャし、将来のバッチ処理のために保存する必要があります。これにより、デジタル化後の未加工の無線周波数 (RF) のフェーズ内クワドラチャ (I/Q) サンプルが節約されます。Amazon S3 バケットにデータが格納されたら、任意のソフトウェアを使用してデータを復調およびデコードできます。処理の詳細な例については、[MathWorks チュートリアル](#) を参照してください。この例を使用した後、Amazon EC2 スポット料金コンポーネントを追加してデータを処理し、全体的な処理コストを削減することを検討できます。

## 通信パス

このセクションでは[データフロー通信パスを計画する](#)、開始方法を示します。

次のテンプレートスニペットはすべて、テンプレートのリソースセクションに属します CloudFormation。

Resources:

```
# Resources that you would like to create should be placed within the Resources section.
```

### Note

CloudFormation テンプレートの内容の詳細については、「[テンプレート](#)」[セクション](#)を参照してください。

Amazon S3 に単一の通信パスを配信するシナリオを考えると、単一の非同期配信パスがあることがわかります。[非同期データ配信](#) セクションごとに、Amazon S3 バケットを定義する必要があります。

```
# 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]]]]]]
```

さらに、ガバケット AWS Ground Station を使用できるようにするには、適切なロールとポリシーを作成する必要があります。

```
# 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 設定

このセクションでは[設定の作成](#)、開始方法を示します。

自動トラックの使用に関する設定を行うには、tracking-config が必要です。オートトラックとして PREFERRED を選択すると、シグナル品質が向上しますが、JPSS-1 エフェメリスの品質が十分であるため、シグナル品質を満たす必要はありません。

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"
```

通信パスに基づいて、衛星部分を表すアンテナダウンリンク設定と、先ほど作成した Amazon S3 バケットを参照する s3 録画を定義する必要があります。

```
# 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 ミッションプロファイル

このセクションでは[ミッションプロファイルを作成する](#)、開始方法を示します。

関連付けられた設定ができたので、それらを使用してデータフローを構築できます。残りのパラメータにはデフォルトを使用します。

```
# 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
```

## まとめ

上記のリソースを使用すると、オンボーディングされたのいずれかから非同期データ配信のために JPSS-1 コンタクトをスケジュールできるようになりました AWS Ground Station [AWS Ground Station ロケーション](#)。

以下は、このセクションで説明されているすべてのリソースを 1 つの CloudFormation テンプレートにまとめた完全なテンプレートです。このテンプレートは [AWS CloudFormation](#) で直接使用できます CloudFormation。

という名前の CloudFormation テンプレートには、Amazon S3 バケットと、問い合わせをスケジュールし、VITA-49 Signal/IP ダイレクトブロードキャストデータを受信するために必要な AWS Ground Station リソース `AquaSnppJpss-1TerraDigIfS3DataDelivery.yml` が含まれています。

Aqua、SNPP、JPSS-1/NOAA-20、Terra がアカウントにオンボードされていない場合は、「」を参照してください [衛星搭載](#)。

#### Note

テンプレートにアクセスするには、有効な AWS 認証情報を使用して Amazon S3 バケットをオンボーディングするカスタマーにアクセスします。以下のリンクでは、リージョン Amazon S3 バケットを使用しています。us-west-2 リージョンコードを変更して、スタックを作成する CloudFormation 対応するリージョンを表します。

さらに、次の手順では YAML を使用します。ただし、テンプレートは YAML 形式と JSON 形式の両方で使用できます。JSON を使用するには、テンプレートをダウンロードする .json ときに .yaml ファイル拡張子を に置き換えます。

を使用してテンプレートをダウンロードするには AWS CLI、次のコマンドを使用します。

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml .
```

ブラウザで以下の URL に移動して、テンプレートをコンソールで表示およびダウンロードできます。

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml
```

次のリンク CloudFormation を使用して、 でテンプレートを直接指定できます。

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml
```

# データフローエンドポイントを利用するパブリックブロードキャスト衛星 (ナローバンド)

この例では、ユーザーガイドの [JPSS-1 - パブリックブロードキャスト衛星 \(PBS\) - 評価](#) セクションで行った分析に基づいています。

この例を完了するには、シナリオを想定する必要があります。つまり、HRD 通信パスをデジタル中間周波数 (DigIF) としてキャプチャし、SDR を使用して Amazon EC2 インスタンス上のデータフローエンドポイントアプリケーションで受信したとおりに処理する必要があります。

## 通信パス

このセクションでは [データフロー通信パスを計画する](#)、開始方法を示します。この例では、CloudFormation テンプレートにパラメータセクションとリソースセクションの 2 つのセクションを作成します。

### Note

CloudFormation テンプレートの内容の詳細については、「[テンプレート](#)」[セクション](#)を参照してください。

パラメータセクションでは、次のパラメータを追加します。CloudFormation コンソールを使用してスタックを作成するときに、これらの値を指定します。

#### 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

キーペアを作成し、Amazon EC2 EC2Keyパラメータの名前を指定する必要があります。

[Amazon EC2 インスタンスのキーペアを作成する](#)を参照してください。

さらに、CloudFormation スタックの作成時に、正しいリージョン固有の AMI ID を指定する必要があります。「[AWS Ground Station Amazon マシンイメージ \(AMIs\)](#)」を参照してください。

残りのテンプレートスニペットは、テンプレートのリソースセクションに属します  
CloudFormation。

Resources:

```
# Resources that you would like to create should be placed within the resource section.
```

EC2 インスタンスに単一の通信パスを配信するシナリオを考えると、単一の同期配信パスがあります。[同期データ配信](#) セクションごとに、データフローエンドポイントアプリケーションで Amazon EC2 インスタンスを設定および設定し、1 つ以上のデータフローエンドポイントグループを作成する必要があります。

```
# 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

    echo "Configuring dataflow endpoint application streams"
    python "${GROUND_STATION_BIN_DIR}/configure_streams.py" --configFileName
"${STREAM_CONFIG_PATH}"
    sleep 2

```

```
python "${GROUND_STATION_BIN_DIR}/save_default_config.py"

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
```

さらに、AWS Ground Station がアカウントに Elastic Network Interface (ENI) を作成できるように、適切なポリシーとロールを作成する必要があります。

```
# 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 設定

このセクションでは[設定の作成](#)、開始方法を示します。

自動トラックの使用に関する設定を行うには、tracking-config が必要です。オートトラックとして PREFERRED を選択すると信号品質が向上しますが、JPSS-1 エフェメリスの品質が十分であるため、信号品質を満たす必要はありません。

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"
```

通信パスに基づいて、衛星部分を表すアンテナダウンリンク設定と、エンドポイントの詳細を定義するデータフローエンドポイントグループを参照するデータフローエンドポイント設定を定義する必要があります。

```
# 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 ミッションプロファイル

このセクションでは[ミッションプロファイルを作成する](#)、開始方法を示します。

関連付けられた設定ができたので、それを使用してデータフローを構築できます。残りのパラメータにはデフォルトを使用します。

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnpjMissionProfile:
  Type: AWS::GroundStation::MissionProfile
  Properties:
    Name: "37849 SNPP And 43013 JPSS"
    ContactPrePassDurationSeconds: 120
    ContactPostPassDurationSeconds: 60
    MinimumViableContactDurationSeconds: 180
    TrackingConfigArn: !Ref TrackingConfig
    DataflowEdges:
      - Source: !Ref SnpjDownlinkDigIfAntennaConfig
        Destination: !Ref DownlinkDigIfEndpointConfig
```

## まとめる

上記のリソースを使用すると、オンボーディングされた のいずれかからの同期データ配信のために JPSS-1 コンタクトをスケジュールできるようになりました AWS Ground Station [AWS Ground Station ロケーション](#)。

以下は、このセクションで説明するすべてのリソースを 1 つの CloudFormation テンプレートにまとめた完全なテンプレートです。このテンプレートを直接使用できます CloudFormation。

という名前の CloudFormation テンプレート

トAquaSnppJpssTerraDigIF.ymlは、Aqua、SNPP、JPSS-1/NOAA-20、Terra 衛星のデジタル中間周波数 (DigIF) データの受信をすばやく開始できるように設計されています。これには、Amazon EC2 インスタンスと、生の DigIF ダイレクトブロードキャストデータを受信するために必要な CloudFormation リソースが含まれています。

Aqua、SNPP、JPSS-1/NOAA-20、Terra がアカウントにオンボードされていない場合は、「」を参照してください[衛星搭載](#)。

### Note

テンプレートにアクセスするには、有効な AWS 認証情報を使用して顧客オンボーディング Amazon S3 バケットにアクセスします。以下のリンクでは、リージョンの Amazon S3 バケットを使用しています。us-west-2 リージョンコードを変更して、スタックを作成する CloudFormation 対応するリージョンを表します。

さらに、次の手順では YAML を使用します。ただし、テンプレートは YAML 形式と JSON 形式の両方で使用できます。JSON を使用するには、テンプレートをダウンロードする .json ときに .yaml ファイル拡張子を に置き換えます。

を使用してテンプレートをダウンロードするには AWS CLI、次のコマンドを使用します。

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpssTerraDigIF.yml .
```

ブラウザで以下の URL に移動して、テンプレートをコンソールで表示およびダウンロードできます。

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpssTerraDigIF.yml
```

次のリンク CloudFormation を使用して、でテンプレートを直接指定できます。

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpssTerraDigIF.yml
```

テンプレートはどのような追加リソースを定義しますか？

AquaSnppJpssTerraDigIF テンプレートには、以下の追加リソースが含まれています。

- (オプション) CloudWatch Event Triggers - AWS Lambda 問い合わせ AWS Ground Station の前後に よって送信された CloudWatch Events を使用してトリガーされる関数。AWS Lambda 関数は、レシーバーインスタンスを起動し、オプションで停止します。
- (オプション) コンタクトの EC2 検証 - Lambda を使用して SNS 通知でコンタクトに Amazon EC2 インスタンスの検証システムをセットアップするオプション。現在の使用状況によっては、料金が発生する可能性があることに注意してください。
- Ground Station Amazon マシンイメージ取得 Lambda - インスタンスにインストールされているソフトウェアと任意の AMI を選択するオプション。ソフトウェアのオプションは、DDX 2.6.2 Only と DDX 2.6.2 with qRadio 3.6.0 です。これらのオプションは、追加のソフトウェア更新プログラムおよび機能がリリースされるにつれて引き続き拡張されます。
- 追加のミッションプロファイル - 追加のパブリックブロードキャスト衛星 (Aqua、SNPP、Terra) のミッションプロファイル。
- 追加のアンテナダウンリンク設定 - 追加のパブリックブロードキャスト衛星 (Aqua、SNPP、Terra) のアンテナダウンリンク設定。

このテンプレートでは衛星の値とパラメータが入力済みです。これらのパラメータを使用すると、これらの衛星で AWS Ground Station すぐに を簡単に使用できます。このテンプレートを使用する AWS Ground Station ときに を使用するには、独自の値を設定する必要はありません。ただし、値をカスタマイズして、ユースケースに合わせてテンプレートを使用することもできます。

データはどこで受信できますか？

データフローエンドポイントグループは、テンプレートで作成されるレシーバーインスタンスのネットワークインターフェイスを使用するように設定されます。レシーバーインスタンスは、データフローエンドポイントアプリケーションを使用して、データフローエンドポイントで定義されたポート AWS Ground Station で からデータストリームを受信します。受信すると、受信側インスタンスのループバックアダプターの UDP ポート 50000 を介してデータ

を消費できるようになります。データフローエンドポイントグループの設定の詳細については、「[AWS::GroundStation::DataflowEndpointGroup](#)」を参照してください。

## データフローエンドポイントを利用するパブリックブロードキャスト衛星 (復調および復号化)

この例では、ユーザーガイドの [JPSS-1 - パブリックブロードキャスト衛星 \(PBS\) - 評価](#) セクションで行った分析に基づいています。

この例を完了するには、シナリオを想定する必要があります。つまり、データフローエンドポイントを使用して、HRD 通信パスを復調およびデコードされたダイレクトブロードキャストデータとしてキャプチャする必要があります。この例は、NASA Direct Readout Labs ソフトウェア (RT-STPS および IPOPP) を使用してデータを処理する場合の出発点として最適です。

### 通信パス

このセクションでは [データフロー通信パスを計画する](#)、開始方法を示します。この例では、CloudFormation テンプレートにパラメータとリソースセクションの 2 つのセクションを作成します。

#### Note

CloudFormation テンプレートの内容の詳細については、「[テンプレート](#)」[セクション](#)を参照してください。

Parameters セクションでは、次のパラメータを追加します。CloudFormation コンソールを使用してスタックを作成するときに、これらの値を指定します。

#### 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

キーペアを作成し、Amazon EC2 EC2Keyパラメータの名前を指定する必要があります。

[Amazon EC2 インスタンスのキーペアを作成する](#)を参照してください。

さらに、CloudFormation スタックの作成時に、正しいリージョン固有の AMI ID を指定する必要があります。「[AWS Ground Station Amazon マシンイメージ \(AMIs\)](#)」を参照してください。

残りのテンプレートスニペットは、テンプレートのリソースセクションに属します  
CloudFormation。

Resources:

```
# Resources that you would like to create should be placed within the resource section.
```

EC2 インスタンスに単一の通信パスを配信するシナリオを考えると、単一の同期配信パスがあります。[同期データ配信](#) セクションごとに、データフローエンドポイントアプリケーションを使用して Amazon EC2 インスタンスをセットアップおよび設定し、1 つ以上のデータフローエンドポイントグループを作成する必要があります。

```
# 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)

    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

```

2&gt;&amp;1

```
    echo "Configuring dataflow endpoint application streams"
    python "${GROUND_STATION_BIN_DIR}/configure_streams.py" --configFileName
"${STREAM_CONFIG_PATH}"
    sleep 2
    python "${GROUND_STATION_BIN_DIR}/save_default_config.py"

    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: tcp
        FromPort: 22
        ToPort: 22
        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
```

また、AWS Ground Station がアカウントに Elastic Network Interface (ENI) を作成できるようにするには、適切なポリシー、ロール、プロファイルも必要です。

```
# 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 設定

このセクションでは[設定の作成](#)、ユーザーガイドについて説明します。

自動トラックの使用に関する設定を行うには、tracking-config が必要です。自動トラックとして PREFERRED を選択すると、シグナルの品質が向上しますが、JPSS-1 エフェメリスの品質が十分であるため、シグナルの品質を満たす必要はありません。

```

TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"

```

通信パスに基づいて、衛星部分を表す antenna-downlink-demod-decode 設定と、エンドポイントの詳細を定義するデータフローエンドポイントグループを参照する dataflow-endpoint 設定を定義する必要があります。

### Note

DemodulationConfig、および の値を設定する方法の詳細についてはDecodeConfig、「」を参照してください[アンテナダウンリンク復調デコード設定](#)。

```

# 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 ミッションプロファイル

このセクションでは[ミッションプロファイルを作成する](#)、ユーザーガイドについて説明します。

関連付けられた設定ができたので、それらを使用してデータフローを構築できます。残りのパラメータにはデフォルトを使用します。

```

# 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
```

## まとめる

上記のリソースを使用すると、オンボーディングされた のいずれかからの同期データ配信のために JPSS-1 コンタクトをスケジュールできるようになりました AWS Ground Station [AWS Ground Station ロケーション](#)。

以下は、このセクションで説明されているすべてのリソースを 1 つの CloudFormation テンプレートにまとめた完全なテンプレートです。このテンプレートは で直接使用できません CloudFormation。

という名前の CloudFormation テンプレート AquaSnppJpss.yml は、Aqua、SNPP、および JPSS-1/NOAA-20 衛星のデータ受信を開始するためのクイックアクセスを提供するように設計されています。これには、Amazon EC2 インスタンスと、問い合わせをスケジュールし、復調およびデコードされたダイレクトブロードキャストデータを受信するために必要な AWS Ground Station リソースが含まれています。

Aqua、SNPP、JPSS-1/NOAA-20、Terra がアカウントにオンボードされていない場合は、「」を参照してください [衛星搭載](#)。

### Note

テンプレートにアクセスするには、有効な AWS 認証情報を使用して Amazon S3 バケットをオンボーディングするカスタマーにアクセスします。以下のリンクでは、リージョン Amazon S3 バケットを使用しています。us-west-2 リージョンコードを変更して、スタックを作成する CloudFormation 対応するリージョンを表します。

さらに、次の手順では YAML を使用します。ただし、テンプレートは YAML 形式と JSON 形式の両方で使用できます。JSON を使用するには、テンプレートをダウンロードする .json ときに .yaml ファイル拡張子を に置き換えます。

を使用してテンプレートをダウンロードするには AWS CLI、次のコマンドを使用します。

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yml .
```

ブラウザで以下の URL に移動して、テンプレートをコンソールで表示およびダウンロードできます。

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yml
```

次のリンク CloudFormation を使用して、 でテンプレートを直接指定できます。

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss.yml
```

テンプレートではどのような追加リソースが定義されていますか？

AquaSnppJpss テンプレートには、以下の追加リソースが含まれています。

- ( オプション) CloudWatch Event Triggers - AWS Lambda 問い合わせの AWS Ground Station 前後に よって送信された CloudWatch Events を使用してトリガーされる関数。AWS Lambda 関数は、レシーバーインスタンスを起動し、オプションで停止します。
- (オプション) コンタクトの EC2 検証 - Lambda を使用して SNS 通知でコンタクトに Amazon EC2 インスタンスの検証システムをセットアップするオプション。現在の使用状況によっては、料金が発生する可能性があることに注意してください。
- Ground Station Amazon マシンイメージ取得 Lambda - インスタンスにインストールされているソフトウェアと任意の AMI を選択するオプション。ソフトウェアのオプションは、DDX 2.6.2 Only と DDX 2.6.2 with qRadio 3.6.0 です。広帯域 DigIF データ配信と AWS Ground Station エージェントを使用する場合は、「」を参照してください [AWS Ground Station エージェント \(広帯域\) を利用するパブリックブロードキャスト衛星](#)。これらのオプションは、追加のソフトウェア更新プログラムおよび機能がリリースされるにつれて引き続き拡張されます。
- 追加のミッションプロファイル - 追加のパブリックブロードキャスト衛星 (Aqua、SNPP、Terra) のミッションプロファイル。
- 追加のアンテナダウンリンク設定 - 追加のパブリックブロードキャスト衛星 (Aqua、SNPP、Terra) のアンテナダウンリンク設定。

このテンプレートでは衛星の値とパラメータが入力済みです。これらのパラメータを使用すると、これらの衛星で AWS Ground Station すぐに を簡単に使用できます。このテンプレートを使用する

AWS Ground Station ときに を使用するには、独自の値を設定する必要はありません。ただし、値をカスタマイズして、ユースケースに合わせてテンプレートを使用することもできます。

データはどこで受信できますか？

データフローエンドポイントグループは、テンプレートで作成されるレシーバーインスタンスのネットワークインターフェイスを使用するように設定されます。レシーバーインスタンスは、データフローエンドポイントアプリケーションを使用して、データフローエンドポイントで定義されたポート AWS Ground Station からデータストリームを受信します。受信すると、受信側インスタンスのループバックアダプターの UDP ポート 50000 を介してデータを消費できるようになります。データフローエンドポイントグループの設定の詳細については、「[AWS::GroundStation::DataflowEndpointGroup](#)」を参照してください。

## AWS Ground Station エージェント (広帯域) を利用するパブリックブロードキャスト衛星

この例では、ユーザーガイドの [JPSS-1 - パブリックブロードキャスト衛星 \(PBS\) - 評価](#) セクションで行った分析に基づいています。

この例を完了するには、シナリオを想定する必要があります。つまり、HRD 通信パスを広帯域デジタル中間周波数 (DigIF) としてキャプチャし、SDR を使用して Amazon EC2 インスタンスの AWS Ground Station エージェントが受信したとおりに処理する必要があります。

### Note

実際の JPSS HRD 通信パス信号の帯域幅は 30 MHz ですが、アンテナダウンリンク設定を 100 MHz 帯域幅の信号として扱うように設定して、この例で AWS Ground Station エージェントによって受信される正しいパスを通過できるようにします。

## 通信パス

このセクションでは [データフロー通信パスを計画する](#)、開始方法を示します。この例では、他の例で使用されていない追加のセクションであるマッピングセクションを CloudFormation テンプレートに追加する必要があります。

**Note**

CloudFormation テンプレートの内容の詳細については、[「テンプレートセクション」](#)を参照してください。

まず、テンプレートでリージョン別の AWS Ground Station プレフィックスリスト CloudFormation のマッピングセクションを設定します。これにより、Amazon EC2 インスタンスのセキュリティグループがプレフィックスリストを簡単に参照できるようになります。プレフィックスリストの使用の詳細については、「」を参照してください[AWS Ground Station エージェントによる VPC 設定](#)。

**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

Parameters セクションでは、次のパラメータを追加します。CloudFormation コンソールを使用してスタックを作成するときに、これらの値を指定します。

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

キーペアを作成し、Amazon EC2 EC2Keyパラメータの名前を指定する必要があります。

[Amazon EC2 インスタンスのキーペアを作成する](#)」を参照してください。

さらに、CloudFormation スタックの作成時に、正しいリージョン固有の AMI ID を指定する必要があります。「[AWS Ground Station Amazon マシンイメージ \(AMIs\)](#)」を参照してください。

残りのテンプレートスニペットは、テンプレートのリソースセクションに属します CloudFormation 。

Resources:

# Resources that you would like to create should be placed within the Resources section.

Amazon EC2 インスタンスに単一の通信パスを配信するシナリオを考えると、単一の同期配信パスがあることがわかります。[同期データ配信](#) セクションごとに、AWS Ground Station エージェントで Amazon EC2 インスタンスをセットアップして設定し、1 つ以上のデータフローエンドポイントグループを作成する必要があります。まず、AWS Ground Station エージェントの Amazon VPC を設定します。

```
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::VPCEGatewayAttachment
Properties:
  VpcId: !Ref ReceiverVPC
  InternetGatewayId: !Ref InternetGateway
```

**Note**

AWS Ground Station エージェントでサポートされている VPC 設定の詳細については、[AWS Ground Station 「エージェントの要件 - VPC 図」](#)を参照してください。

次に、レシーバー Amazon EC2 インスタンスを設定します。

```
# 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

```

また、AWS Ground Station がアカウントに Elastic Network Interface (ENI) を作成できるようにするには、適切なポリシー、ロール、プロファイルも必要です。

```

# 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 設定

このセクションでは[設定の作成](#)、開始方法を示します。

自動トラックの使用設定を行うには、tracking-config が必要です。オートトラックとして PREFERRED を選択すると、シグナル品質が向上しますが、JPSS-1 エフェメリスの品質が十分であるため、シグナル品質を満たす必要はありません。

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"
```

通信パスに基づいて、衛星部分を表すアンテナダウンリンク設定と、エンドポイントの詳細を定義するデータフローエンドポイントグループを参照するデータフローエンドポイント設定を定義する必要があります。

```
# The AWS Ground Station Antenna Downlink Config that defines the frequency spectrum
used to
# downlink data from your satellite.
SnpJPSSDownlinkDigIfAntennaConfig:
  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 ミッションプロファイル

このセクションでは[ミッションプロファイルを作成する](#)、開始方法を示します。

関連付けられた設定ができたので、それらを使用してデータフローを構築できます。残りのパラメータにはデフォルトを使用します。

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnpjpsMissionProfile:
  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 SnpjpsDownlinkDigIfAntennaConfig
        Destination: !Ref DownlinkDigIfEndpointConfig
    StreamsKmsKey:
      KmsKeyArn: !GetAtt GroundStationDataDeliveryKmsKey.Arn
    StreamsKmsRole: !GetAtt GroundStationKmsKeyRole.Arn
```

## まとめる

上記のリソースを使用すると、オンボーディングされた のいずれかからの同期データ配信のために JPSS-1 コンタクトをスケジュールできるようになりました [AWS Ground Station 口ケーション](#)。

以下は、このセクションで説明するすべてのリソースを 1 つの CloudFormation テンプレートにまとめた完全なテンプレートです。このテンプレートは [直接](#)使用できます CloudFormation。

という名前の CloudFormation テンプレート

トDirectBroadcastSatelliteWbDigIfEc2DataDelivery.ymlは、Aqua、SNPP、JPSS-1/NOAA-20、Terra 衛星のデジタル中間周波数 (DigIF) データをすばやく受信できるように設計されています。これには、Amazon EC2 インスタンスと、AWS Ground Station エージェントを使用して生の DigIF ダイレクトブロードキャストデータを受信するために必要な CloudFormation リソースが含まれています。

Aqua、SNPP、JPSS-1/NOAA-20、Terra がアカウントにオンボードされていない場合は、「」を参照してください[衛星搭載](#)。

### Note

テンプレートにアクセスするには、有効な AWS 認証情報を使用して Amazon S3 バケットをオンボーディングするカスタマーにアクセスします。以下のリンクでは、リージョン Amazon S3 バケットを使用しています。us-west-2 リージョンコードを変更して、スタックを作成する CloudFormation 対応するリージョンを表します。

さらに、次の手順では YAML を使用します。ただし、テンプレートは YAML 形式と JSON 形式の両方で使用できます。JSON を使用するには、テンプレートをダウンロードする .json ときに .yaml ファイル拡張子を に置き換えます。

を使用してテンプレートをダウンロードするには AWS CLI、次のコマンドを使用します。

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml .
```

ブラウザで以下の URL に移動して、テンプレートをコンソールで表示およびダウンロードできます。

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

次のリンク CloudFormation を使用して、 でテンプレートを直接指定できます。

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

テンプレートではどのような追加リソースが定義されていますか？

DirectBroadcastSatelliteWbDigIfEc2DataDelivery テンプレートには、以下の追加リソースが含まれています。

- レシーバーインスタンスの Elastic Network Interface - (条件付き) Elastic Network Interface は、提供されている場合、PublicSubnetId で指定されたサブネットに作成されます。これは、レシーバーインスタンスがプライベートサブネットにある場合に必要です。Elastic Network Interface は EIP に関連付けられ、レシーバーインスタンスにアタッチされます。
- レシーバーインスタンスの Elastic IP - AWS Ground Station が接続する Elastic IP。これにより、レシーバーインスタンスまたは Elastic Network Interface にアタッチされます。
- 次のいずれかの Elastic IP 関連付け：
  - レシーバーインスタンスと Elastic IP の関連付け - PublicSubnetId が指定されていない場合の、Elastic IP とレシーバーインスタンスの関連付け。そのためには、SubnetId がパブリックサブネットを参照する必要があります。
  - レシーバーインスタンスの Elastic Network Interface と Elastic IP の関連付け - PublicSubnetId が指定されている場合、Elastic IP とレシーバーインスタンスの Elastic Network Interface の関連付け。
- ( オプション) CloudWatch Event Triggers - AWS Lambda 問い合わせ AWS Ground Station の前後によって送信された CloudWatch Events を使用してトリガーされる関数。AWS Lambda 関数は、レシーバーインスタンスを起動し、オプションで停止します。
- ( オプション) Amazon EC2 Verification for Contacts - Lambda を使用して、SNS 通知を持つ連絡先の Amazon EC2 インスタンス (複数可) の検証システムを設定するオプション。現在の使用状況によっては、料金が発生する可能性があることに注意してください。
- 追加のミッションプロファイル - 追加のパブリックブロードキャスト衛星 (Aqua、SNPP、Terra) のミッションプロファイル。
- 追加のアンテナダウンリンク設定 - 追加のパブリックブロードキャスト衛星 (Aqua、SNPP、Terra) のアンテナダウンリンク設定。

このテンプレートでは衛星の値とパラメータが入力済みです。これらのパラメータを使用すると、これらの衛星で AWS Ground Station [すぐに](#) を簡単に使用できます。このテンプレートを使用する AWS Ground Station [ときに](#) を使用するには、独自の値を設定する必要はありません。ただし、値をカスタマイズして、ユースケースに合わせてテンプレートを使用することもできます。

データはどこで受信できますか？

データフローエンドポイントグループは、テンプレートで作成されるレシーバーインスタンスのネットワークインターフェイスを使用するように設定されます。レシーバーインスタンスは AWS Ground Station エージェントを使用して、データフローエンドポイントで定義されたポート AWS Ground Station からデータストリームを受信します。データフローエンドポイントグループの設定の詳細については、[「AWS::GroundStation::DataflowEndpointGroup」](#) を参照してください。エージェントの詳細については、AWS Ground Station [AWS Ground Station 「エージェントとは」](#) を参照してください。

# トラブルシューティング

次のドキュメントは、 の使用中に発生する可能性のある問題のトラブルシューティングに役立ちます AWS Ground Station。

## トピック

- [Amazon EC2 にデータを配信する問い合わせのトラブルシューティング](#)
- [失敗した問い合わせのトラブルシューティング](#)
- [FAILED\\_TO\\_SCHEDULE コンタクトのトラブルシューティング](#)
- [DataflowEndpointGroups が正常でない状態のトラブルシューティング](#)
- [無効なエフェメリスのトラブルシューティング](#)
- [データを受信しなかった問い合わせのトラブルシューティング](#)
- [テレメトリのトラブルシューティング](#)

## Amazon EC2 にデータを配信する問い合わせのトラブルシューティング

AWS Ground Station 問い合わせを正常に完了できない場合は、Amazon EC2 インスタンスが実行されていること、データフローエンドポイントアプリケーションが実行されていること、およびデータフローエンドポイントアプリケーションのストリームが正しく設定されていることを確認する必要があります。

### Note

DataDefender (DDX) は、 で現在サポートされているデータフローエンドポイントアプリケーションの例です。 AWS Ground Station

## 前提条件

次の手順では、Amazon EC2 インスタンスがすでにセットアップされていることを前提としています。で Amazon EC2 インスタンスをセットアップするには AWS Ground Station、 [「開始方法」](#) を参照してください。

## ステップ 1: EC2 インスタンスが実行されていることを確認する

次の手順は、コンソールで Amazon EC2 インスタンスを検索し、実行されていない場合に起動する方法を示しています。

1. トラブルシューティングする連絡先に使用された Amazon EC2 インスタンスを見つけます。以下のステップを使用します。
  - a. CloudFormation ダッシュボードで、Amazon EC2 インスタンスを含むスタックを選択します。
  - b. [リソース] タブをクリックし、Amazon EC2 インスタンスを [論理 ID] 列でロードバランサーの ID をクリックします。[状況] 列でインスタンスが作成されていることを確認します。
  - c. [物理 ID] 列で、Amazon EC2 インスタンスのリンクを選択します。Amazon EC2 マネジメントコンソールが表示されます。
2. Amazon EC2 マネジメントコンソールで、Amazon EC2 の [インスタンスの状態] が [実行中] になっていることを確認します。
3. インスタンスが実行中の場合は、次のステップに進みます。インスタンスが実行されていない場合は、次の手順を使用してインスタンスを起動します。
  - Amazon EC2 インスタンスを選択した状態で、[アクション] > [インスタンスの状態] > [開始] の順に選択します。

## ステップ 2: 使用するデータフローアプリケーションのタイプを決定する

データ配信に AWS Ground Station エージェントを使用している場合は、[AWS Ground Station 「エージェントのトラブルシューティング」](#) セクションにリダイレクトしてください。それ以外の場合は、DataDefender (DDX) アプリケーションを使用している場合は、[に進みます the section called “ステップ 3: データフローアプリケーションが実行されていることを確認する”](#)。

## ステップ 3: データフローアプリケーションが実行されていることを確認する

DataDefender のステータスを検証するには、Amazon EC2 のインスタンスに接続する必要があります。インスタンスへの接続の詳細については、[「Linux インスタンスへの接続」](#) を参照してください。

次の手順では、SSH クライアントでコマンドを使用したトラブルシューティングの手順を示します。

1. ターミナルまたはコマンドプロンプトを開き、SSH を使用して Amazon EC2 インスタンスに接続します。DataDefender ウェブ UI を表示するために、リモートホストのポート 80 を転送します。以下のコマンドは、SSH を使用して、ポート転送が有効になっている踏み台を介して Amazon EC2 インスタンスに接続する方法を示しています。

#### Note

<SSH KEY>、<BASTION HOST>、および <HOST> は、特定の ssh キー、踏み台ホスト名、および Amazon EC2 インスタンスホスト名に置き換える必要があります。

### Windows の場合

```
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>
```

### Mac の場合

```
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>
```

2. DataDefender (DDX と呼ばれる) が実行されていることを確認するには、出力で ddx という名前の実行プロセスの grepping (チェック) を実行します。実行中のプロセスをグレッピング (チェック) するためのコマンドと成功した出力例を以下に示します。

```
[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
```

DataDefender が実行されている場合は、「[それ以外のthe section called “ステップ 4: データフローアプリケーションストリームが設定されていることを確認する”](#)場合は、次のステップに進みます。

- 以下のコマンドを使用して DataDefender を起動します。

```
sudo service rtlogic-ddx start
```

コマンドの使用後に DataDefender が実行されている場合は、「それ以外の[the section called “ステップ 4: データフローアプリケーションストリームが設定されていることを確認する”](#)場合は、次のステップに進みます。

- 以下のコマンドを使用して以下のファイルを検査し、DataDefender のインストールと設定中にエラーが発生したかどうかを確認します。

```
cat /var/log/user-data.log
cat /opt/aws/groundstation/.startup.out
```

#### Note

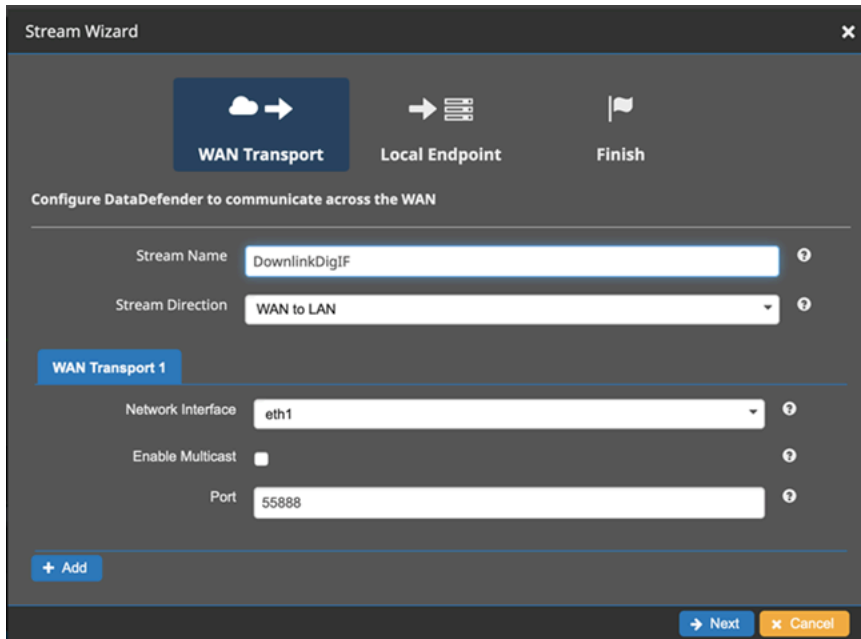
これらのファイルを検査したときに発見される一般的な問題は、Amazon EC2 インスタンスが実行されている Amazon VPC に Amazon S3 へのアクセス許可がないためにインストールファイルをダウンロードできないことです。これが問題であることがログで判明した場合は、EC2 インスタンスの Amazon VPC とセキュリティグループの設定をチェックして、Amazon S3 へのアクセスがブロックされていないことを確認します。

Amazon VPC 設定を確認した後に DataDefender が実行されている場合は、「」に進みます[the section called “ステップ 4: データフローアプリケーションストリームが設定されていることを確認する”](#)。問題が解決しない場合は、[AWS サポートに連絡](#)し、問題の説明を添えてログファイルを送信してください。

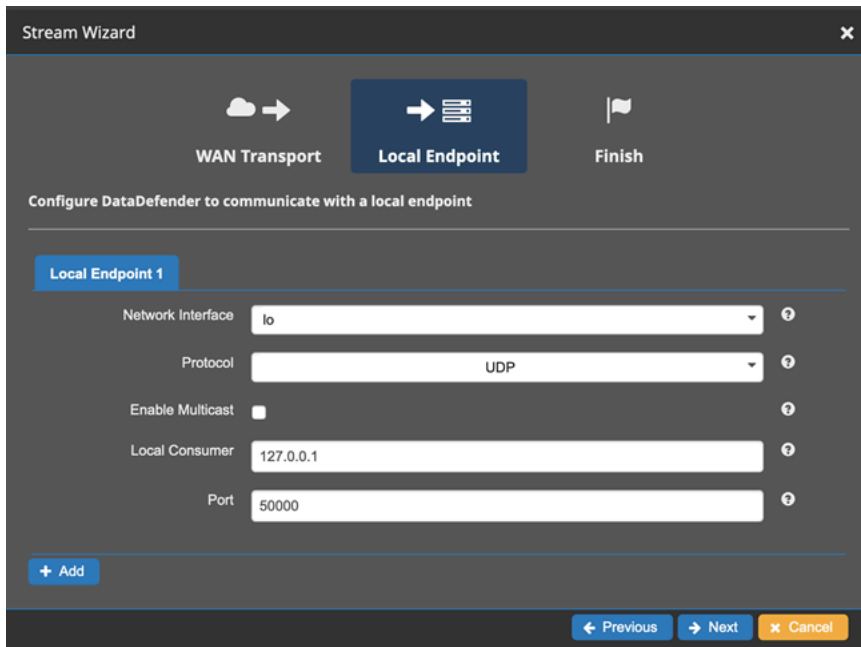
## ステップ 4: データフローアプリケーションストリームが設定されていることを確認する

- ウェブブラウザで、アドレスバーに localhost:8080 というアドレスを入力して、DataDefender ウェブユーザーインターフェイスにアクセスします。次に、Enter キーを押します。
- [DataDefender] ダッシュボードで、[Go to Details (詳細へ移動)] を選択します。
- ストリームのリストからストリームを選択し、[Edit Stream (ストリームを編集)] を選択します。
- [Stream Wizard (ストリームウィザード)] ダイアログボックスで、次の操作を行います。

- a. [WAN Transport] (WAN トランスポート) ペインで、[Stream Direction] (ストリーム方向) が [WAN to LAN] (WAN から LAN) に設定されていることを確認します。
- b. [Port (ポート)] ボックスで、データフローエンドポイントグループ用に選択した WAN ポートが存在することを確認します。デフォルトでは、このポートは 55888 です。その後、[Next] を選択します。



- c. [Local Endpoint (ローカルエンドポイント)] ペインで、[Port (ポート)] ボックスに有効なポートがあることを確認します。デフォルトでは、このポートは 50000 です。これは、DataDefender が AWS Ground Station サービスからデータを受信した後にデータを受信するポートです。その後、[Next] を選択します。



- d. 値を変更した場合は、残りのメニューで [Finish (完了)] を選択します。それ以外の場合は、[Stream Wizard (ストリームウィザード)] メニューからキャンセルできます。

これで、Amazon EC2 インスタンスと DataDefender の両方が実行され、データを受信するように適切に設定されていることが確認されました AWS Ground Station。 [「the section called “ステップ 5: レシーバーインスタンス \(複数可\) サブネットに十分な使用可能な IP アドレスがあることを確認する”」に進みます。](#)

## ステップ 5: レシーバーインスタンス (複数可) サブネットに十分な使用可能な IP アドレスがあることを確認する

次の手順は、コンソールの Amazon EC2 受信者インスタンスで使用可能な IP アドレスの数を確認する方法を示しています。

1. トラブルシューティングする問い合わせに使用された Amazon EC2 レシーバーインスタンスごとに。以下のステップを使用します。
  - a. CloudFormation ダッシュボードで、Amazon EC2 インスタンスを含むスタックを選択します。
  - b. [リソース] タブをクリックし、Amazon EC2 インスタンスを [論理 ID] 列でロードバランサーの ID をクリックします。[状況] 列でインスタンスが作成されていることを確認します。

- c. [物理 ID] 列で、Amazon EC2 インスタンスのリンクを選択します。Amazon EC2 マネジメントコンソールが表示されます。
2. Amazon EC2 マネジメントコンソールで、Amazon EC2 レシーバーインスタンスのインスタンス概要のサブネット ID リンクを見つけてクリックします。これにより、対応する Amazon VPC 管理コンソールに移動します。
3. Amazon VPC 管理コンソールで一致するサブネットを選択し、サブネットの詳細で使用可能な IPv4 アドレスを確認します。この数が、この Amazon EC2 レシーバーインスタンスを使用するデータフローエンドポイントと同じ数でない場合は、次の操作を行います。
  - a. CloudFormation テンプレートの対応するサブネット CidrBlock のサイズを正しく更新します。サブネットのサイズ設定の詳細については、[「サブネット CIDR ブロック」](#)を参照してください。
  - b. 更新された CloudFormation テンプレートを使用してスタックを再デプロイします。

問題が解決しない場合は、[AWS サポートにお問い合わせください](#)。

## 失敗した問い合わせのトラブルシューティング

がリソース設定の問題 AWS Ground Station を検出すると、コンタクトのターミナルコンタクトステータスは FAILED になります。コンタクトが FAILED になる原因となる一般的な使用例と、トラブルシューティングに役立つ手順を以下に示します。

### Note

このガイドは、特にコンタクトの FAILED ステータスを対象としており、AWS\_FAILED、AWS\_CANCELED、FAILED\_TO\_SCHEDULE などの他の失敗ステータスを対象としたものではありません。コンタクトのステータスの詳細については、「[the section called “AWS Ground Station 問い合わせステータス”](#)」を参照してください。

## データフローエンドポイントの失敗のユースケース

以下は、データフローエンドポイントベースのデータフローの問い合わせステータスが失敗する可能性がある一般的なユースケースのリストです。

- データフローエンドポイントは接続されません - 1 つ以上のデータフローの AWS Ground Station アンテナとデータフローエンドポイントグループ間の接続は確立されませんでした。

- データフローエンドポイントの接続が遅れる - 1 つ以上のデータフローの AWS Ground Station アンテナとデータフローエンドポイントグループ間の接続が、問い合わせの開始時間後に確立されませんでした。
- データフローエンドポイントのサブネットに使用可能な IP アドレスがありません。AWS Ground Station のデータ配信ソリューションは、受信者インスタンスのサブネットに使用可能な IP アドレスがないため、プライベートネットワークに ENI を作成できません。
- Dataflow エンドポイントのサブネットが無効です。Dataflow エンドポイントグループで指定された指定されたサブネットにアクセスできないため、AWS Ground Station のデータ配信ソリューションはプライベートネットワークに ENI を作成できません。

データフローエンドポイントの障害が発生した場合は、以下を確認することをお勧めします。

- コンタクト開始時刻より前に、受信側の Amazon EC2 インスタンスが正常に起動したことを確認します。
- 問い合わせ中にデータフローエンドポイントソフトウェアが稼働していることを確認します。
- レシーバーインスタンスサブネットごとに、データフローエンドポイントごとに少なくとも 1 つの使用可能な IP アドレスがあることを確認します。
- で設定されたデータフローを通じて、Dataflow Endpoint Group に関連付けられたサブネットがアクティブで使用可能 [Amazon VPC のセットアップと設定](#) であることを確認します AWS Ground Station。

より具体的なトラブルシューティング手順については、「[Amazon EC2 にデータを配信する問い合わせのトラブルシューティング](#)」のセクションを参照してください。

## AWS Ground Station エージェントが失敗したユースケース

Agent ベースのデータフローでコンタクトステータスが FAILED になることがある一般的なユースケースを、以下に示します。

- AWS Ground Station エージェント未報告ステータス - 1 つ以上のデータフローのデータフローエンドポイントグループのデータ配信をオーケストレーションするエージェントは、ステータスを正常に報告しません AWS Ground Station。このステータスの更新は、コンタクト終了時刻の数秒以内に行われます。
- AWS Ground Station エージェント開始遅延 - 1 つ以上のデータフローのデータ配信を Dataflow Endpoint Group でオーケストレーションするエージェントは、問い合わせの開始時刻より遅く開始されました。

AWS Ground Station エージェントデータフローの障害が発生した場合は、以下を確認することをお勧めします。

- コンタクト開始時刻より前に、受信側の Amazon EC2 インスタンスが正常に起動したことを確認します。
- コンタクトの開始時とコンタクト中に、Agent アプリケーションが起動して実行中であったことを確認します。
- Agent アプリケーションと Amazon EC2 インスタンスが、コンタクト終了から 15 秒以内にシャットダウンされていないことを確認します。これにより、Agent は AWS Ground Station にステータスを報告するのに十分な時間を確保できます。

より具体的なトラブルシューティング手順については、「[Amazon EC2 にデータを配信する問い合わせのトラブルシューティング](#)」のセクションを参照してください。

## FAILED\_TO\_SCHEDULE コンタクトのトラブルシューティング

ガリソース設定または内部システム内の問題を検出すると、問い合わせは FAILED\_TO\_SCHEDULE 状態で AWS Ground Station 終了します。FAILED\_TO\_SCHEDULE 状態で終了するコンタクトは、オプションで追加のコンテキストerrorMessage用に を提供します。連絡先の説明については、[DescribeContact](#) API を参照してください。

FAILED\_TO\_SCHEDULE コンタクトを引き起こす可能性がある一般的なユースケースを、トラブルシューティングに役立つ手順とともに以下に示します。

### Note

このガイドは、特に FAILED\_TO\_SCHEDULE 問い合わせステータスを対象としており、AWS\_FAILEDAWS\_CANCELLED、FAILED などの他の障害ステータスを対象としていません。コンタクトのステータスの詳細については、「[the section called “AWS Ground Station 問い合わせステータス”](#)」を参照してください。

## Antenna Downlink Demod Decode Config で指定された設定はサポートされていません

この問い合わせをスケジュールするために使用された[ミッションプロファイル](#)には、有効でない [antenna-downlink-demod-decode](#) 設定がありました。

## 既存の AntennaDownlinkDemodDecode 設定

- antenna-downlink-demod-decode 設定が最近変更された場合は、スケジュールする前に以前の動作バージョンにロールバックします。
- これが既存の設定の意図的な変更である場合、または以前に既存の設定が正常にスケジュールされなくなった場合は、新しい AntennaDownlinkDemodDecode 設定をオンボードする方法に関する次のステップに従います。

## 新しく作成された AntennaDownlinkDemodDecode 設定

新しい設定をオンボードするには、AWS Ground Station に直接お問い合わせください。FAILED\_TO\_SCHEDULE 状態でcontactId終了した を含む [AWS サポート](#) でケースを作成する

## 一般的なトラブルシューティングステップ

上記のトラブルシューティング手順で問題が解決しなかった場合:

- 同じミッションプロファイルを使用して、コンタクトのスケジュールを再実行するか、別のコンタクトをスケジュールします。連絡先の予約方法については、「[ReserveContact](#)」を参照してください。
- このミッションプロファイルの FAILED\_TO\_SCHEDULE ステータスが引き続き表示される場合は、[AWS サポートにお問い合わせください](#)。

## DataflowEndpointGroups が正常でない状態のトラブルシューティング

データフローエンドポイントグループが HEALTHY の状態にならない可能性がある理由と、取るべき適切な是正措置を以下に示します。

- NO\_REGISTERED\_AGENT - EC2 インスタンスを起動します。これにより、エージェントが登録されます。この呼び出しが成功するには、有効なコントローラー設定ファイルが必要であることに注意してください。このファイルの設定の詳細については、「[AWS Ground Station エージェントを使用する](#)」を参照してください。
- INVALID\_IP\_OWNERSHIP - DeleteDataflowEndpointGroup API を使用してデータフローエンドポイントグループを削除し、CreateDataflowEndpointGroup API を使用して EC2 インスタンスに関連付けられた IP アドレスとポートをでデータフローエンドポイントグループを再作成します。

- UNVERIFIED\_IP\_OWNERSHIP - IP アドレスはまだ検証されていません。検証は定期的に行われるため、これは、自動的に解決するはずです。
- NOT\_AUTHORIZED\_TO\_CREATE\_SLR - アカウントに、必要なサービスにリンクされたロールを作成する権限がありません。「[Ground Station のサービスにリンクされたロールを使用する](#)」のトラブルシューティングの手順を確認してください。

## 無効なエフェメリスのトラブルシューティング

エフェメリスデータを にアップロードすると AWS Ground Station、非同期検証ワークフローが実行されます。検証が失敗すると、エフェメリスステータスは に変わります INVALID。 [DescribeEphemeris](#) レスポンスのエラーメッセージには、問題の特定と解決に役立つ詳細情報が表示されます。

### エフェメリス検証エラーについて

エフェメリスが検証に失敗すると、 [DescribeEphemeris](#) API レスポンスには、問題の診断に役立つ 2 つのフィールドが含まれます。

#### errorCode

特定の検証エラーを識別する機械読み取り可能なコード。これは、プログラムによるエラー処理に使用できます。

#### errorMessage

問題の詳細と修正方法に関するガイダンスを含む、人間が読める検証エラーの説明。

無効なエフェメリスに対する [DescribeEphemeris](#) レスポンスの例:

```
{
  "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 エフェメリスの一般的な検証エラー

TLE エフェメリスのアップロード時に発生する一般的な検証エラーを次に示します。

### 一致しない衛星のカタログ番号

エラー: 「エフェメリスに存在する衛星カタログ番号が、関連付けられた衛星の衛星カタログ番号と一致しません」

解決策: TLE 行の NORAD ID/衛星カタログ番号が衛星の衛星カタログ番号と一致していることを確認します。カタログ番号が割り当てられていない衛星00000に を使用します。

### 無効な平均モーション

エラー: 「提供されたエフェメリスの平均モーションが最新のリファレンスエフェメリスと大きく異なる」

解決策: TLE データが正しく、有効な軌道を表していることを確認します。Ground Station は、検証中に Space-Track エフェメリスをリファレンスとして使用します。

## OEM エフェメリスの一般的な検証エラー

OEM エフェメリスのアップロード時に発生する一般的な検証エラーを次に示します。

### 無効な参照フレーム

エラー: 「REF\_FRAME はサポートされていません」

解決策: サポートされているリファレンスフレームの 1 つを使用するように OEM ファイルを更新します: EME2000 または ITRF2000。

### 必須フィールドがありません

エラー: 「メタデータフィールドがありません: INTERPOLATION」

解決策: INTERPOLATION フィールドと INTERPOLATION\_DEGREE フィールドを OEM メタデータセクションに追加します。これらは、 が正確なアンテナポイント角度を生成する AWS Ground Station ために必要です。

## サポートされていない時間システム

エラー: 「TIME\_SYSTEM はサポートされていません」

解決策: OEM ファイルがタイムシステムとして UTC を使用していることを確認します。

## サポートされていない OEM バージョン

エラー: 「CCSDS\_OEM\_VERS はサポートされていません」

解決策: OEM ファイルが CCSDS OEM バージョン 2.0 を使用していることを確認します。

## 方位高度エフェメリスの一般的な検証エラー

以下は、方位高度エフェメリスをアップロードするときに発生する一般的な検証エラーです。

### 方位/上昇データの欠落

エラー: 「少なくとも 1 つの AzElSegment に TimeAzEl フィールドが存在しませんでした  
AzElSegment 」

解決策: 方位標高データ内の各セグメントに、少なくとも 1 つの時間タグ付けされた方位/標高ペアが含まれていることを確認します。

### 無効な方位角度範囲 (度)

エラー: AzEl az は -180 以上、360 度以下である必要があります」

解決策: 方位角が [-180, 360] 度以内であることを確認します。

### 無効な上昇角度範囲 (度)

エラー: AzEl el は -90 以上、90 度以下である必要があります」

解決策: 上昇角度が [-90, 90] 度以内であることを確認します。

### 無効な方位角範囲 (ラジアン)

エラー: AzEl az は  $-\pi$  以上、 $2\pi$  ラジアン以下である必要があります」

解決策: 方位角が  $[-\pi, 2\pi]$  ラジアン以内であることを確認します。

### 無効な標高角度範囲 (ラジアン)

エラー: AzEl el は  $-\pi/2$  以上、 $\pi/2$  ラジアン以下である必要があります」

解決策: 標高角度が  $[-\pi/2, \pi/2]$  ラジアン以内であることを確認します。

## 非単調時間値

エラー: 「AzElSegment 内の TimeAzEl AzElSegment 項目は一時的な順序である必要があります」

解決策: 各セグメントの時間値が厳密に増加していることを確認します。

セグメントの順序が間違っている

エラー: "AzElSegments must be temporally in order"

解決策: セグメントが時系列順に配置されていることを確認します。

セグメントの重複

エラー: 「少なくとも 1 つのセグメントの時間範囲が他のセグメントの時間範囲と重複しています」

解決策: 各セグメントに重複しない一意の時間範囲があることを確認します。1 つのセグメント endTime のは、次のセグメント startTime の を超えることはできません。

## トラブルシューティングのステップ

エフェメリスが検証に失敗した場合は、以下の手順に従って問題を解決します。

1. エフェメリス ID を使用して [DescribeEphemeris](#) を呼び出し、errorCode と を取得し、errorMessage を取得します。
2. 失敗した検証チェックに関する具体的な詳細については、エラーメッセージを確認してください。
3. エフェメリスデータで特定された問題を修正します。
4. [CreateEphemeris](#) を使用して、修正されたデータを含む新しいエフェメリスをアップロードします。
5. 新しいエフェメリスのステータスが ENABLED 状態になるまでモニタリングします。
6. 不要になった場合は、[DeleteEphemeris](#) を使用して無効なエフェメリスを削除します。

## 完全なエラーコードリファレンス

以下のセクションでは、エフェメリス検証が失敗した場合に返される可能性のあるすべての errorCode 値を高レベル invalidReason カテゴリ別に包括的にマッピングします。

## 無効な理由: METADATA\_INVALID

これらのエラーは、必須メタデータフィールドが欠落しているか、形式が正しくないか、エフェメリスデータにサポートされていない値が含まれている場合に発生します。

エラーコード	エラーメッセージ
MISMATCHED_SATCAT_ID	TLE エフェメリスに存在する衛星カタログ番号が、関連付けられた衛星の衛星カタログ番号と一致しません
OEM_VERSION_UNSUPPORTED	OEM エフェメリスCCSDS_OEM_VERS のはサポートされていません。サポートされている値: [2.0]
ORIGINATOR_MISSING	ORIGINATOR ヘッダーフィールドが OEM エフェメリスにありません
CREATION_DATE_MISSING	CREATION_DATE ヘッダーフィールドが OEM エフェメリスにありません
OBJECT_NAME_MISSING	OEM エフェメリスにOBJECT_NAME メタデータフィールドがありません
オブジェクト ID 欠落	OEM エフェメリスにOBJECT_ID メタデータフィールドがありません
REF_FRAME_UNSUPPORTED	OEM エフェメリスREF_FRAME のはサポートされていません。サポートされている値: [EME2000, ITRF2000]
REF_FRAME_EPOCH_UNSUPPORTED	OEM エフェメリスのREF_FRAME_EPOCH メタデータフィールドはサポートされていません。このフィールドをエフェメリスから削除してください
TIME_SYSTEM_UNSUPPORTED	OEM エフェメリスTIME_SYSTEM のはサポートされていません。サポートされている値: [UTC]
CENTER_BODY_UNSUPPORTED	OEM エフェメリスCENTER_BODY のはサポートされていません。サポートされている値: [Earth]
INTERPOLATION_MISSING	OEM エフェメリスにINTERPOLATION メタデータフィールドがありません

エラーコード	エラーメッセージ
INTERPOLATION_DEGREE_INVALID	補間方法の場合、OEM エフェメリスの補間度は 0 より大きい必要があります
AZ_EL_SEGMENT_LIST_MISSING	<a href="#">azElSegmentList</a> フィールドがありません
INSUFFICIENT_TIME_AZ_EL	少なくとも 1 つの <a href="#">TimeAzEl</a> フィールドが存在しませんでした <a href="#">azElSegmentList</a>

## 無効な理由: **TIME\_RANGE\_INVALID**

これらのエラーは、開始/終了時刻の問題、セグメントの順序付け、セグメントの重複、一時的な不整合など、エフェメリスに無効な時間範囲が含まれている場合に発生します。

エラーコード	エラーメッセージ
START_TIME_IN_FUTURE	エフェメリスの開始時刻は将来のものですが、過去のものである必要があります
END_TIME_IN_PAST	エフェメリスの終了時刻は過去のものですが、将来のものである必要があります
EXPIRATION_TIME_TOO_EARLY	指定された有効期限がエフェメリス終了時間より前です
START_TIME_METADATA_TOO_EARLY	START_TIME メタデータ値が OEM エフェメリスデータに存在する最も早い時刻より前である
STOP_TIME_METADATA_TOO_LATE	STOP_TIME メタデータ値が OEM エフェメリスデータに存在する最新の時刻より後である
AZ_EL_SEGMENT_END_TIME_BEFORE_START_TIME	少なくとも 1 つのデータセグメント <a href="#">endTime</a> のは、セグメントの <a href="#">startTime</a>
AZ_EL_SEGMENT_TIMES_OVERLAP	少なくとも 1 つのセグメントの時間範囲が他のセグメントの時間範囲と重複している
AZ_EL_SEGMENTS_OUT_OF_ORDER	セグメントが一時的に順序付けられていない

エラーコード	エラーメッセージ
TIME_AZ_EL_ITEMS_OUT_OF_ORDER	内の <a href="#">TimeAzEl</a> 項目は、一時的な順序に <a href="#">AzElSegment</a> する必要があります
AZ_EL_SEGMENT_REFERENCE_EPOCH_INVALID	セグメントのリファレンスエポックが無効または形式が正しくない
AZ_EL_SEGMENT_START_TIME_INVALID	セグメントの有効な時間範囲の開始時刻が最初のセグメントの後に開始されない
AZ_EL_SEGMENT_END_TIME_INVALID	セグメントの有効な時間範囲の終了時刻が最後のセグメントより後に終了しない
AZ_EL_SEGMENT_VALID_TIME_RANGE_INVALID	セグメントの有効な時間範囲が無効です
AZ_EL_SEGMENT_END_TIME_TOO_LATE	セグメントの終了時刻が参照エポックの最大許容期間を超えています
AZ_EL_TOTAL_DURATION_EXCEEDED	すべてのセグメントの合計時間が、許容される最大ポイント角度期間を超えています

### 無効な理由: **TRAJECTORY\_INVALID**

これらのエラーは、エフェメリスに、軌道パラメータ、角度範囲、単位の問題など、無効な軌道データが含まれている場合に発生します。

エラーコード	エラーメッセージ
MEAN_MOTION_INVALID	提供された TLE エフェメリスの平均モーションは、最新のリファレンスエフェメリスと大きく異なります。注: Ground Station は、検証中にスペーストラックエフェメリスをリファレンスとして使用します
TIME_AZ_EL_AZ_RADIAN_RANGE_INVALID	AzEl は $-\pi$ 以上、 $2\pi$ ラジアン以下 <a href="#">az</a> である必要があります

エラーコード	エラーメッセージ
TIME_AZ_EL_EL_RADIAN_RANGE_INVALID	AzEI は $-\pi/2$ 以上、 $\pi/2$ ラジアン以下 <a href="#">e1</a> である必要があります
TIME_AZ_EL_AZ_DEGREE_RANGE_INVALID	AzEI は -180 以上、360 度以下 <a href="#">az</a> である必要があります
TIME_AZ_EL_EL_DEGREE_RANGE_INVALID	AzEI は -90 度以上、90 度以下 <a href="#">e1</a> である必要があります
TIME_AZ_EL_ANGLE_UNITS_INVALID	無効な AzEI 角度単位

### 無効な理由: **KMS\_KEY\_INVALID**

これらのエラーは、エフェメリスデータの暗号化に使用される AWS Key Management Service (KMS) キーに問題がある場合に発生します。

エラーコード	エラーメッセージ
INSUFFICIENT_KMS_PERMISSIONS	Ground Station には、このエフェメリスの KMS キーにアクセスするための十分なアクセス許可がありません

### 無効な理由: **VALIDATION\_ERROR**

これらのエラーは、エフェメリスデータに他の特定のカテゴリに該当しない一般的な検証の問題がある場合に発生します。

エラーコード	エラーメッセージ
INTERNAL_ERROR	エフェメリスの検証中に内部エラーが発生しました
FILE_FORMAT_INVALID	エフェメリスファイル形式が無効または破損しています。ファイルがエフェメリスタイプの想定形式に準拠していることを確認します。

# データを受信しなかった問い合わせのトラブルシューティング

問い合わせは成功したように見えますが、データを受信しませんでした。これは、空の PCAP ファイルを受信するか、S3 データ配信を使用している場合は PCAP ファイルをまったく受信しないことを意味します。これには、いくつかの理由が考えられます。以下に、いくつかの原因とその対処方法について説明します。

## 正しくないダウンリンク設定

衛星からデータを受信する各連絡先には、[アンテナダウンリンク設定](#)または [アンテナダウンリンク復調デコード設定](#) が関連付けられます。指定された設定が衛星によって送信されるシグナルと一致しない場合、は送信されたシグナルを受信 AWS Ground Station できません。これにより、はデータを受信しません AWS Ground Station。

これを修正するには、使用している設定が衛星によって送信されるシグナルと一致していることを確認してください。例えば、正しい中心周波数、帯域幅、偏波を設定し、必要に応じて復調パラメータとデコードパラメータを設定していることを確認します。

## 衛星操作

衛星が一部の通信システムを一時的に無効にする操作を実行する場合があります。この操作により、空の衛星の位置も大きく変わる可能性があります。AWS Ground Station は、信号を送信していない衛星から信号を受信できないか、使用されているエフェメリスによって AWS Ground Station アンテナが衛星が存在しない空の場所を指すようになります。

NOAA が運営するパブリックブロードキャスト衛星と通信しようとする、NOAA [衛星アラートメッセージ](#) ページで停止または操作を説明するメッセージが表示される場合があります。メッセージには、データ転送の再開が予想される時刻のタイムラインが含まれる場合もあれば、後続のメッセージに投稿される場合もあります。

独自の衛星と通信する場合は、衛星の運用と、それが通信にどのように影響するかを理解するのはお客様の責任です AWS Ground Station。衛星軌道に影響を与える操作を実行する場合は、更新されたカスタムエフェメリスデータの提供が含まれる場合があります。カスタムエフェメリスデータの提供の詳細については、「」を参照してください [エフェメリス AWS Ground Station を使用する方法を理解する](#)。

## AWS Ground Station 停止

AWS Ground Station によって問い合わせが失敗するか、キャンセルされた場合、AWS Ground Station は問い合わせステータスを AWS\_FAILED または AWS\_CANCELLED に設定します。問い合

わせのライフサイクルの詳細については、「」を参照してください[問い合わせのライフサイクルを理解する](#)。場合によっては、AWS Ground Station に障害が発生してデータがアカウントに配信されないことがあります。問い合わせが AWS\_FAILED または AWS\_CANCELLED ステータスにならないことがあります。この場合、AWS Ground Station はアカウント固有のイベントを AWS Health ダッシュボードに投稿する必要があります。Health ダッシュボードの詳細については、AWS [AWS「Health ユーザーガイド」](#)を参照してください。

## テレメトリのトラブルシューティング

テレメトリの一般的な問題をトラブルシューティングするには、次の情報を使用します。

### 一般的なセットアップの問題

#### IAM アクセス許可エラー

##### 症状

TelemetrySinkConfig を作成CreateConfigするために を呼び出すと、エラーが表示されます。

```
Unable to write to Kinesis Data Streams stream. Ensure that Ground Station has kinesis:PutRecord permissions for the given stream
```

##### 原因

- TelemetrySinkConfig で指定された IAM ロールには、Kinesis Data Streams ストリームに書き込むために必要なアクセス許可がありません。
- IAM ロールの信頼ポリシーは、 がロールを引き受け AWS Ground Station ることを許可しません。
- TelemetrySinkConfig の Kinesis Data Streams ストリーム ARN が正しくないか、ストリームが存在しません。

##### ソリューション

1. IAM ロールが存在し、正しいアクセス許可を持っていることを確認します。すべてのステップに従っている[ステップ 2: TelemetrySinkConfig を作成する](#)ことを確認します。
2. が IAM ロールを引き受け AWS Ground Station ることができることを確認します。

```
aws iam get-role --role-name GroundStationTelemetryRole
```

信頼ポリシーに信頼されたサービスプリンシパル `groundstation.amazonaws.com` としてが含まれていることを確認します。

3. IAM ロールに必要な Kinesis アクセス許可があることを確認します。

```
aws iam list-attached-role-policies --role-name GroundStationTelemetryRole
```

ポリシーにストリームの `kinesis:DescribeStream`、`kinesis:PutRecord`、およびアクセス `kinesis:PutRecords` 許可が含まれていることを確認します。

4. Kinesis Data Streams ストリームが存在し、ARN が正しいことを確認します。

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

5. カスタマーマネージド暗号化を使用する場合は、IAM ロールに AWS KMS キーに対する `kms:GenerateDataKey` アクセス許可があることを確認します。

## PassRole アクセス許可エラー

### 症状

を呼び出すと `CreateConfig`、IAM ロールを渡すアクセス許可がないというエラーが表示されます。

### 解決策

IAM ユーザーまたはロールにテレメトリ IAM ロールの `iam:PassRole` アクセス許可があることを確認します。ユーザーまたはロールに次のポリシーを追加します。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iam:GetRole",  
        "iam:PassRole"  
      ],  
      "Resource": "arn:aws:iam::9999999999:role/your-stream-name"  
    }  
  ]  
}
```

```
}  
]  
}
```

## Kinesis Data Streams ストリーム設定の問題

### 症状

テレメトリ配信が失敗するか、断続的です。

### 原因

- Kinesis Data Streams ストリームの容量がテレメトリスループットに不足しています。
- ストリームが他のアプリケーションで使用されているため、書き込みスロットリングが発生します。

### ソリューション

1. ストリームのステータスを確認します。

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

2. CloudWatch メトリクスを使用して書き込みスロットリングをモニタリングします。

```
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. スロットリングが検出された場合は、次の点を考慮してください。

- 自動スケーリングのためにオンデマンドキャパシティモードに切り替えます。
- AWS Ground Station テレメトリに専用ストリームを使用する。
- プロビジョンドモードを使用する場合は、シャードの数を増やします。

## テレメトリ配信の問題

### テレメトリデータが表示されない

#### 症状

テレメトリが有効なミッションプロファイルを使用してコンタクトをスケジュールすると、Kinesis Data Streams ストリームにテレメトリデータは表示されません。

#### 考えられる原因と解決策

##### ミッションプロファイルでテレメトリが有効になっていない

問い合わせに使用されるミッションプロファイルに `telemetrySinkConfigArn` が含まれていることを確認します。

```
aws groundstation get-mission-profile \  
  --mission-profile-id 12345678-1234-1234-1234-123456789012 \  
  --region us-east-2
```

`telemetrySinkConfigArn` フィールドの出力を確認します。存在しない場合、ミッションプロファイルではテレメトリが有効になっていません。

#### IAM ロールのアクセス許可の問題

の IAM アクセス許可のトラブルシューティング手順を確認します [IAM アクセス許可エラー](#)。

#### Kinesis Data Streams ストリームが存在しないか、間違っただリージョンにある

ストリームが正しいリージョンに存在することを確認します。

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

#### 問い合わせはまだ開始されていません

テレメトリ配信は、問い合わせの開始時に開始されます。問い合わせのステータスを確認して、問い合わせが開始されたことを確認します。

```
aws groundstation describe-contact \  
  --contact-id your-contact-id
```

```
--contact-id 12345678-1234-1234-1234-123456789012 \  
--region us-east-2
```

## 断続的なテレメトリデータ

### 症状

テレメトリデータが、ギャップやレコードの欠落と矛盾して配信されます。

### 考えられる原因

- Kinesis Data Streams ストリームの容量の問題またはスロットリング。 [「Kinesis Data Streams ストリーム設定の問題」](#)を参照してください。
- AWS Ground Station と Kinesis Data Streams ストリーム間のネットワーク接続の問題。

### ソリューション

- CloudWatch で Kinesis Data Streams ストリームメトリクスのスロットリングまたはエラーをモニタリングします。
- ストリームがオンデマンドキャパシティモードを使用しているか、十分なプロビジョンドキャパシティがあることを確認します。
- AWS Ground Station テレメトリ専用のストリームを使用して、他のアプリケーションとの競合を回避します。

## データ形式の問題

### JSON 解析エラー

#### 症状

テレメトリレコードを JSON として解析するときに、アプリケーションでエラーが発生します。

#### ソリューション

- Base64 デコードの検証 - Kinesis Data Streams ストリームのデータが Base64-encodedされていることを確認します。JSON として解析する前に、必ずデータをデコードしてください。詳細については、 [「Kinesis Data Streams ストリームからのデータの読み取り」](#)を参照してください。

- 空のレコードをチェックする - TelemetrySinkConfig の作成時に空の検証レコードを送信 AWS Ground Station できます。アプリケーションは、空のレコードまたは不正な形式のレコードを適切に処理する必要があります。
- バージョン対応の解析を実装する - まず telemetryTypeAndVersion、telemetryType、および telemetryVersion フィールドを解析して、各レコードに適したスキーマを決定します。

## 不明なテレメトリタイプまたはバージョン

### 症状

アプリケーションが認識しないテレメトリタイプまたはバージョンを検出しました。

### 解決策

これは、新しいテレメトリタイプとスキーマバージョンが時間の経過とともに導入される可能性があるため、予想される動作です。アプリケーションは次の条件を満たす必要があります。

- モニタリング用の不明なタイプとバージョンをログに記録します。
- 既知のタイプとバージョンを引き続き処理します。
- 不明なスキーマに対して適切な処理を実装します。

スキーマのバージョンングの詳細については、「」を参照してください [スキーマのバージョンングと進化](#)。

## ヘルプの利用

トラブルシューティングの手順に従っても問題が解決しない場合は、AWS サポートにお問い合わせください。

### 提供すべき情報

サポートに連絡するときは、次の情報を入力します。

- 問題が発生し IDs
- 使用するミッションプロファイル ID
- TelemetrySinkConfig ARN
- Kinesis Data Streams ストリーム ARN
- IAM ロール ARN とアタッチされたポリシー

- CloudWatch Logs またはアプリケーションからのエラーメッセージ
- 問題が発生したときのタイムスタンプ
- 既に実行されているトラブルシューティング手順

一般的な AWS Ground Station サポートについては、[AWS Ground Station ユーザーガイド](#)を参照してください。

## クォータと制限

サポートされているリージョン、関連するエンドポイント、およびクォータは、[AWS Ground Station エンドポイントとクォータ](#)で表示できます。

必要に応じて、[Service Quotas コンソール](#)、[AWS API](#) および [AWS CLI](#) を使用して、クォータ増加のリクエストを行うことができます。

## サービス条件

AWS Ground Station サービス条件については、[「AWS サービス条件」](#)を参照してください。

# AWS Ground Station ユーザーガイドのドキュメント履歴

次の表に、AWS Ground Station ユーザーガイドの各リリースにおける重要な変更点を示します。

変更	説明	日付
<a href="#">ドキュメントの更新</a>	CancelContact API に機能を追加し、その機能と計測への影響に関する情報を追加しました。詳細については、「 <a href="#">問い合わせ計測を理解する</a> 」を参照してください。	2025 年 12 月 10 日
<a href="#">ドキュメントの更新</a>	CloudWatch メトリクスが問い合わせの地上局に関連付けられたリージョンで出力されることを明確にしました。リンク切れを修正しました。	2025 年 12 月 2 日
<a href="#">AWS 管理ポリシーの更新</a>	AWS Ground Station は、タスクレスポンス URL を取得するための追加のアクセス許可を含めるAWSGroundStationAgentInstancePolicy ように マネージドポリシーを更新しました。URLs 詳細については、「 <a href="#">AWS Ground StationAWS 管理ポリシーの更新</a> 」を参照してください。	2025 年 11 月 13 日
<a href="#">新機能</a>	アジマス標高エフェメリスを含めるようにユーザーガイドを更新しました。詳細については、「 <a href="#">方位高度エフェメリスデータの提供</a> 」を参照してください。	2025 年 10 月 22 日

<a href="#">ドキュメントの更新</a>	クロスリージョンデータ配信では、特別な設定や承認が不要になりました。詳細については、 <a href="#">「クロスリージョンデータ配信を使用する」</a> を参照してください。	2025 年 9 月 11 日
<a href="#">ドキュメントの更新</a>	設定済みリソースの問い合わせ使用率に関する説明を追加しました。	2025 年 4 月 4 日
<a href="#">新機能</a>	AWS Ground Station デジタルツインを含めるようにユーザーガイドを更新しました。	2024 年 8 月 6 日
<a href="#">ドキュメントの更新</a>	新しい図、例など、ユーザーガイドの多くのセクションを更新しました。	2024 年 7 月 18 日
<a href="#">ドキュメントの更新</a>	ユーザーガイドに RSS フィードを追加しました。	2024 年 7 月 18 日
<a href="#">ドキュメントの更新</a>	AWS Ground Station エージェントユーザーガイドを別のユーザーガイドに分割します。	2024 年 7 月 18 日
<a href="#">新機能</a>	可視性時間範囲外で最大 30 秒までコンタクトをスケジュールできるようになりました。可視性時間は DescribeContact レスポンスに含まれます。	2024 年 3 月 26 日
<a href="#">ドキュメントの更新</a>	組織を改善し、「EC2 インスタンスの選択と CPU 計画」セクションを追加しました。	2024 年 3 月 6 日

<a href="#">ドキュメントの更新</a>	AWS Ground Station エージェントと一緒にサービスとプロセスを実行するための新しいベストプラクティスを AWS Ground Station エージェントユーザーガイドに追加しました。	2024 年 2 月 23 日
<a href="#">ドキュメントの更新</a>	エージェントリリースノートページを追加しました。	2024 年 2 月 21 日
<a href="#">テンプレートの更新</a>	DirectBroadcastSatelliteWbDigIfEc2DataDelivery テンプレートに個別のパブリックサブネットのサポートが追加されました。	2024 年 2 月 14 日
<a href="#">ドキュメントの更新</a>	モニタリングドキュメント User Notifications に AWS への紹介を追加しました。	2023 年 8 月 6 日
<a href="#">ドキュメントの更新</a>	AWS Ground Station コンソールに表示される名前で衛星にタグを付ける手順を追加しました。	2023 年 7 月 26 日
<a href="#">新機能</a>	広帯域 DigIF データ配信のリリースに関する AWS Ground Station エージェントユーザーガイドを追加しました。	2023 年 4 月 12 日
<a href="#">新しい AWS マネージドポリシー</a>	AWS Ground Station は、AWSGroundStationAgentInstancePolicy という名前の新しいポリシーを追加しました。	2023 年 4 月 12 日

<a href="#">新機能</a>	CPE プレビューのリリースに関するユーザーガイドを更新しました。	2022 年 11 月 9 日
<a href="#">新しい AWS 管理ポリシー</a>	AWS Ground Station は、AWSServiceRoleForGroundStationDataflowEndpointGroup service-linked-role (SLR) を追加しました。AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy	2022 年 11 月 2 日
<a href="#">新機能</a>	ユーザーガイドを更新して、との統合を含めました AWS CLI。	2020 年 4 月 17 日
<a href="#">新機能</a>	ユーザーガイドを更新して、CloudWatch のメトリクスとの統合を追加しました。	2020 年 2 月 24 日
<a href="#">新しいテンプレート</a>	AWS Ground Station ユーザーガイドにパブリックブロードキャスト衛星 (AquaSnppJpss テンプレート) を追加しました。	2020 年 2 月 19 日
<a href="#">新機能</a>	ユーザーガイドを更新してクロスリージョンのデータ配信を含めました	2020 年 2 月 5 日
<a href="#">ドキュメントの更新</a>	CloudWatch Events AWS Ground Station によるモニタリングの例と説明を更新しました。	2020 年 2 月 4 日

<a href="#">ドキュメントの更新</a>	テンプレートの場所が更新され、「開始方法」セクションと「トラブルシューティング」セクションが改訂されました。	2019 年 12 月 19 日
<a href="#">新しいトラブルシューティングセクション</a>	トラブルシューティングセクションが AWS Ground Station ユーザーガイドに追加されました。	2019 年 11 月 7 日
<a href="#">新しい入門トピック</a>	最新の CloudFormation テンプレートを含む「開始方法」トピックを更新しました。	2019 年 7 月 1 日
<a href="#">Kindle バージョン</a>	AWS Ground Station ユーザーガイドの Kindle バージョンを公開しました。	2019 年 6 月 20 日
<a href="#">新しいサービスとガイド</a>	これは、AWS Ground Station および AWS Ground Station ユーザーガイドの初期リリースです。	2019 年 5 月 23 日

# AWS 用語集

最新の AWS 用語については、AWS の用語集 リファレンスの[AWS 用語集](#)を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。