



用户指南

AWS Ground Station



AWS Ground Station: 用户指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 AWS Ground Station?	1
常见使用案例	1
后续步骤	2
操作方法 AWS Ground Station 方式	3
卫星载入	3
任务概况构成	3
联系人日程安排	5
联系执行	6
数字双胞胎	8
明白 AWS Ground Station 核心组件	8
任务配置文件	10
配置	12
数据流终端节点组	20
AWS Ground Station 代理人	27
开始使用	28
注册参加 AWS 账户	28
添加 AWS Ground Station 您的权限 AWS account	28
机载卫星	30
客户入职流程概述	30
(可选) 命名卫星	30
公共广播卫星	33
规划您的数据流通信路径	33
异步数据传输	34
同步数据传输	34
规划您的遥测	35
创建配置	36
数据传输配置	36
遥测配置 (可选)	36
卫星配置	37
创建任务档案	37
了解后续步骤	38
AWS Ground Station 位置	39
在 AWS 区域查找地面站位置	39
AWS Ground Station 支持的 AWS 区域	40

数字双胞胎的可用性	40
专用天线	41
在地面站查看天线	41
示例：列出地面站的天线	41
查看地面站预订	42
房源预订	42
预订类型	42
代码示例	43
AWS Ground Station 网站掩码	45
客户专用口罩	45
现场口罩对可用联系时间的影响	45
AWS Ground Station 网站能力	46
明白怎么做 AWS Ground Station 使用星历	49
默认星历数据	49
提供自定义星历数据	50
概述	50
示例：使用客户提供的星历表和 AWS Ground Station	50
提供 TLE 星历数据	50
提供 OEM 星历数据	57
提供方位角高程星历数据	65
使用自定义星历表保留联系人	75
概述	75
联系人预订工作流程	75
工作流程 1：列出可用联系人然后预订	76
工作流程 2：直接联系预订	80
监控联系人状态的变化	83
最佳实践和注意事项	85
了解使用的是哪个星历	86
TLE 和 OEM 星历	86
方位角仰角星历	87
新的星历表对先前安排的联系人的影响	87
获取卫星的当前星历	88
示例 GetSatellite 使用默认星历表返回卫星	88
示例 GetSatellite 对于使用自定义星历表的卫星	89
列出方位角仰角星历表	89
恢复为默认星历数据	90

恢复 TLE 和 OEM 星历	91
管理方位角仰角星历	91
使用数据流	92
AWS Ground Station 数据平面接口	92
使用跨区域数据传输	93
设置和配置 Amazon S3	93
设置和配置 Amazon VPC	93
VPC 配置为 AWS Ground Station 座席	94
使用数据流终端节点的 VPC 配置	96
设置和配置 Amazon EC2	98
提供的通用软件	98
AWS Ground Station Amazon 机器映像 (AMIs)	99
使用遥测	100
遥测的工作原理	100
可用的遥测类型	100
区域可用性	101
设置遥测	101
步骤 1：创建先决条件 AWS 资源	101
第 2 步：创建一个 TelemetrySinkConfig	103
第 3 步：将遥测添加到任务配置文件中	103
第 4 步：安排联系人	103
后续步骤	104
了解遥测数据	104
数据格式概述	104
指向遥测	105
追踪遥测	107
从 Kinesis 数据流中读取数据	108
架构版本控制和演变	109
与联系人合作	111
了解联系生命周期	111
AWS Ground Station 联系人状态	113
联系人数据保留	114
了解联系人账单	114
带宽定义	114
调度模式	115
CancelContact	115

场景 1：单个联系人	115
场景 2：单次停止接触	116
场景 3：单一副本	117
场景 4：短复制	118
场景 5：多个重复项	118
场景 6：多次停靠	120
场景 7：没有重复 Multi-antenna 的地面站	121
场景 8：有重复触点 Multi-antenna 的地面站	121
更新联系人和联系人版本控制	123
联系人版本控制的工作原理	123
更新联系人	123
联系人版本状态	125
代码示例	125
注意事项	129
AWS Ground Station 数字双胞胎	130
AWS Ground Station 专用天线	131
什么是专用天线	131
增强预订可见性	131
相关资源	132
监控	133
利用活动实现自动化	134
AWS Ground Station 事件类型	134
联系活动时间表	135
星历活动	137
使用记录 API 调用 CloudTrail	138
AWS Ground Station 中的信息 CloudTrail	138
了解 AWS Ground Station 日志文件条目	139
使用亚马逊查看指标 CloudWatch	140
AWS Ground Station 指标与维度	141
查看 指标	144
安全性	150
身份和访问管理	150
受众	151
使用身份进行身份验证	151
使用策略管理访问	152
如何 AWS Ground Station 与 IAM 配合使用	153

基于身份的策略示例	158
问题排查	160
AWS 托管策略	162
AWSGroundStationAgentInstancePolicy	162
AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy	163
策略更新	164
使用服务相关角色	164
Ground Station 的服务相关角色权限	165
为 Ground Station 创建服务相关角色	165
为 Ground Station 编辑服务相关角色	166
为 Ground Station 删除服务相关角色	166
Ground Station 服务相关角色的受支持区域	166
问题排查	167
静态数据加密 AWS Ground Station	167
创建客户托管密钥	168
为指定客户管理的密钥 AWS Ground Station	169
AWS Ground Station 加密上下文	169
TLE 和 OEM 星历数据的静态加密	170
方位角高程星历的静态加密	178
传输期间的数据加密 AWS Ground Station	186
AWS Ground Station 代理直播	186
数据流端点流	187
任务配置文件配置示例	188
JPSS-1-公共广播卫星 (PBS)-评估	188
使用 Amazon S3 数据传输的公共广播卫星	189
通信路径	189
AWS Ground Station 配置	191
AWS Ground Station 任务简介	192
把它放在一起	193
使用数据流端点 (窄带) 的公共广播卫星	194
沟通路径	194
AWS Ground Station 配置	201
AWS Ground Station 任务概况	202
把它放在一起	203
利用数据流端点 (解调和解码) 的公共广播卫星	204
沟通路径	205

AWS Ground Station 配置	212
AWS Ground Station 任务概况	215
把它放在一起	216
使用 AWS Ground Station 代理 (宽带) 的公共广播卫星	217
沟通路径	218
AWS Ground Station 配置	229
AWS Ground Station 任务简介	230
把它放在一起	231
问题排查	233
对向 Amazon EC2 传送数据的联系人进行故障排除	233
步骤 1 : 验证您的 EC2 实例是否正在运行	233
步骤 2 : 确定使用的数据流应用程序的类型	234
步骤 3 : 验证数据流应用程序是否正在运行	234
步骤 4 : 验证您的数据流应用程序流是否已配置	236
步骤 5 : 确保您的接收器实例子网中有足够的可用的 IP 地址	237
解决失败的联系人问题	238
数据流端点失败的用例	238
AWS Ground Station 代理失败的用例	239
解决联系人更新失败的问题	239
同步验证错误	239
异步故障代码	241
检查更新状态	243
排查计划失败的联系人故障	244
不支持 Antenna Downlink Demod Decode Config 中指定的设置	244
一般故障排除步骤	244
排除 DataflowEndpointGroups 未处于正常状态的故障	245
对无效的星历进行故障排除	245
了解星历验证错误	245
TLE 星历表的常见验证错误	246
OEM 星历表的常见验证错误	246
方位角高程星历表的常见验证错误	247
故障排除步骤	248
完整的错误代码参考	248
对未收到任何数据的联系人进行故障排除	252
下行链路配置不正确	252
卫星操纵	253

AWS Ground Station 中断	253
排除遥测故障	253
常见的设置问题	253
遥测传输问题	256
数据格式问题	257
获取帮助	258
限额和限制	260
服务条款	261
文档历史记录	262
AWS 词汇表	267
.....	cclxviii

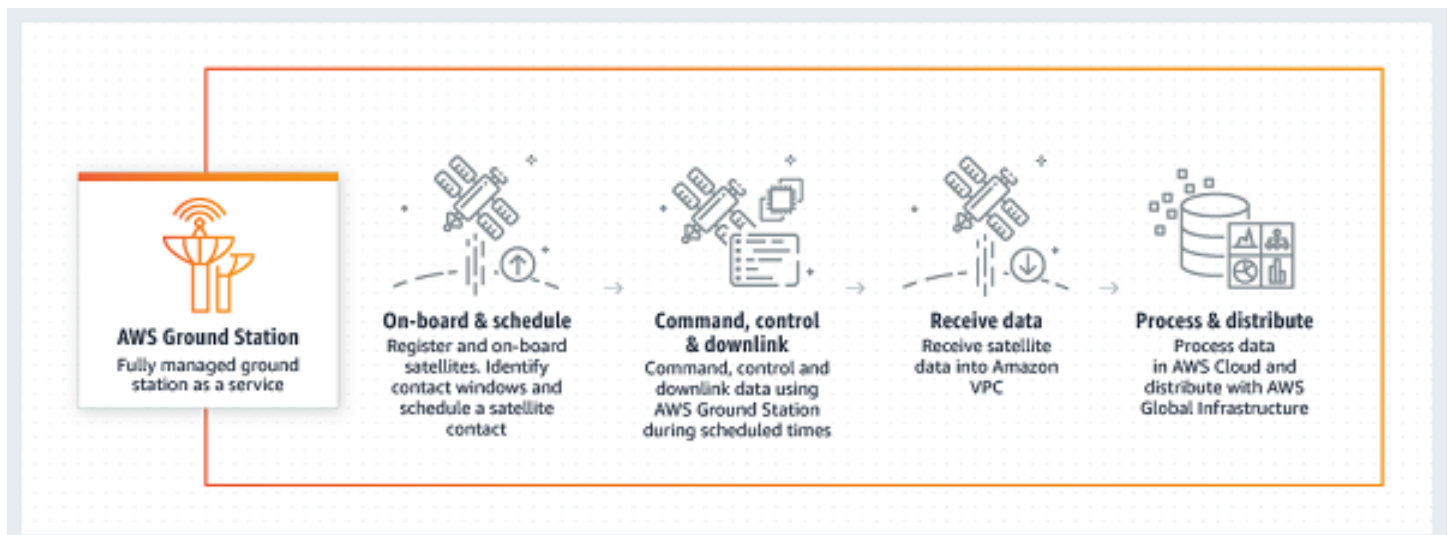
什么是 AWS Ground Station?

AWS Ground Station 是一项完全托管的服务，可在全球基础设施上提供安全、快速和可预测的卫星通信。有了 AWS Ground Station，您不再需要构建、管理或扩展自己的地面站基础设施。AWS Ground Station 使您能够专注于创新和快速尝试采集卫星数据的新应用程序，而不是将资源花费在建造、运营和扩建自己的地面站上。

使用 AWS 的低延迟、高带宽全球光纤网络，您可以在天线系统接收到卫星数据的几秒钟内开始处理您的卫星数据。这使您能够在几秒钟内将原始数据转化为经过处理的信息或经过分析的知识。

对于有特殊要求的组织，AWS Ground Station 还提供 [AWS Ground Station 专用天线](#)——可代表您进行 AWS 管理的定制天线系统，提供对根据您的规格制造的天线的专用访问权限。

常见使用案例



AWS Ground Station 允许您与卫星进行双向通信，并支持以下用例：

- 下行链路数据 — 接收来自卫星的数据、传输 X-band 和 S-band 频率，以实时（VITA-49 格式）传输到 Amazon EC2 实例，或直接传输到您账户中的 Amazon S3 存储桶（[PCAP 格式](#)）。此外，对于使用支持的调制和编码方案的卫星，您可以选择接收解调和解码的数据或原始数字中频 (DigiF) 样本（格式）。VITA-49
- 上行链路数据 — 通过发送要传输的 DigiF 数据（VITA-49 格式），向接收 S-band 频率的卫星发送数据和命令。AWS Ground Station
- Uplink echo — 通过在物理共处的天线上接收传输的信号，验证发送到航天器的命令并执行其他高级任务。

- 软件定义无线电 (SDR) /前端处理器 (FEP) — 使用能够在 Amazon EC2 实例上运行的现有 SDR and/or FEP，根据现有波形实时处理数据，并生成数据产品。 send/receive
- 遥测、跟踪和指挥 (TT&C) — 结合使用前面列出的用例执行 TT&C 来管理您的卫星舰队。
- 跨区域数据传输 — 使用来自单个 AWS 区域 AWS Ground Station 的全球天线网络同时操作多个联系人。
- 数字双胞胎 — 在不使用生产天线容量的情况下，以更低的成本进行测试调度、配置验证和正确的错误处理。

后续步骤

我们建议您首先阅读以下部分：

- 要学习基本 AWS Ground Station 概念，请参阅[操作方法 AWS Ground Station 方式](#)。
- 要了解如何设置您的账户和资源以供使用 AWS Ground Station，请参阅[开始使用](#)。
- 要以编程方式使用 AWS Ground Station，请参阅 [AWS Ground Station API 参考](#)。API 参考详细描述了所有 API 操作。AWS Ground Station 它还支持的 Web 服务协议提供请求、响应和错误示例。您可以使用所选语言的[AWS CLI](#) 或 [AWS SDK](#) 来编写与 AWS Ground Station 交互的代码。

操作方法 AWS Ground Station 方式

AWS Ground Station 操作地面天线以促进与卫星的通信。天线所能做的事情的物理特征被抽象出来，被称为能力。本[AWS Ground Station 位置](#)节中可以参考天线的物理位置及其当前能力。[AWS Support Center Console](#)如果您的用例需要额外的功能、额外的定位产品或更精确的天线位置，请通过以下方式联系我们。

要使用其中一 AWS Ground Station 根天线，必须在特定地点预留时间。此预订被称为联系人。要成功安排联系，AWS Ground Station 需要额外的数据来确保联系成功。

- 您的卫星必须载入一个或多个地点，这可确保您获得批准才能在请求的地点运行各种功能。
- 您的卫星必须具有有效的星历表 ——这可确保天线具有视线，并且在接触期间可以准确地指向您的卫星。
- 你必须拥有有效的任务配置文件 ——这允许你自定义该联系人的行为，包括你将如何接收数据和向卫星发送数据。你可以为同一辆车使用多个任务配置文件来创建不同的联系方式，以适应你遇到的不同操作姿势或场景。

卫星载入

卫星载入 AWS Ground Station 是一个多步骤的过程，涉及数据收集、技术验证、频谱许可以及集成和测试。本指南的[卫星入门](#)部分将引导您完成此过程。

任务概况构成

卫星频率信息、[数据平面](#)信息和其他详细信息封装在任务配置文件中。任务配置文件是配置组件的集合。这使您可以根据自己的用例在不同的任务配置文件中重复使用配置组件。由于任务概况不直接引用单个卫星，而是仅包含有关其技术能力的信息，因此任务配置文件也可以被具有相同配置的多颗卫星重复使用。

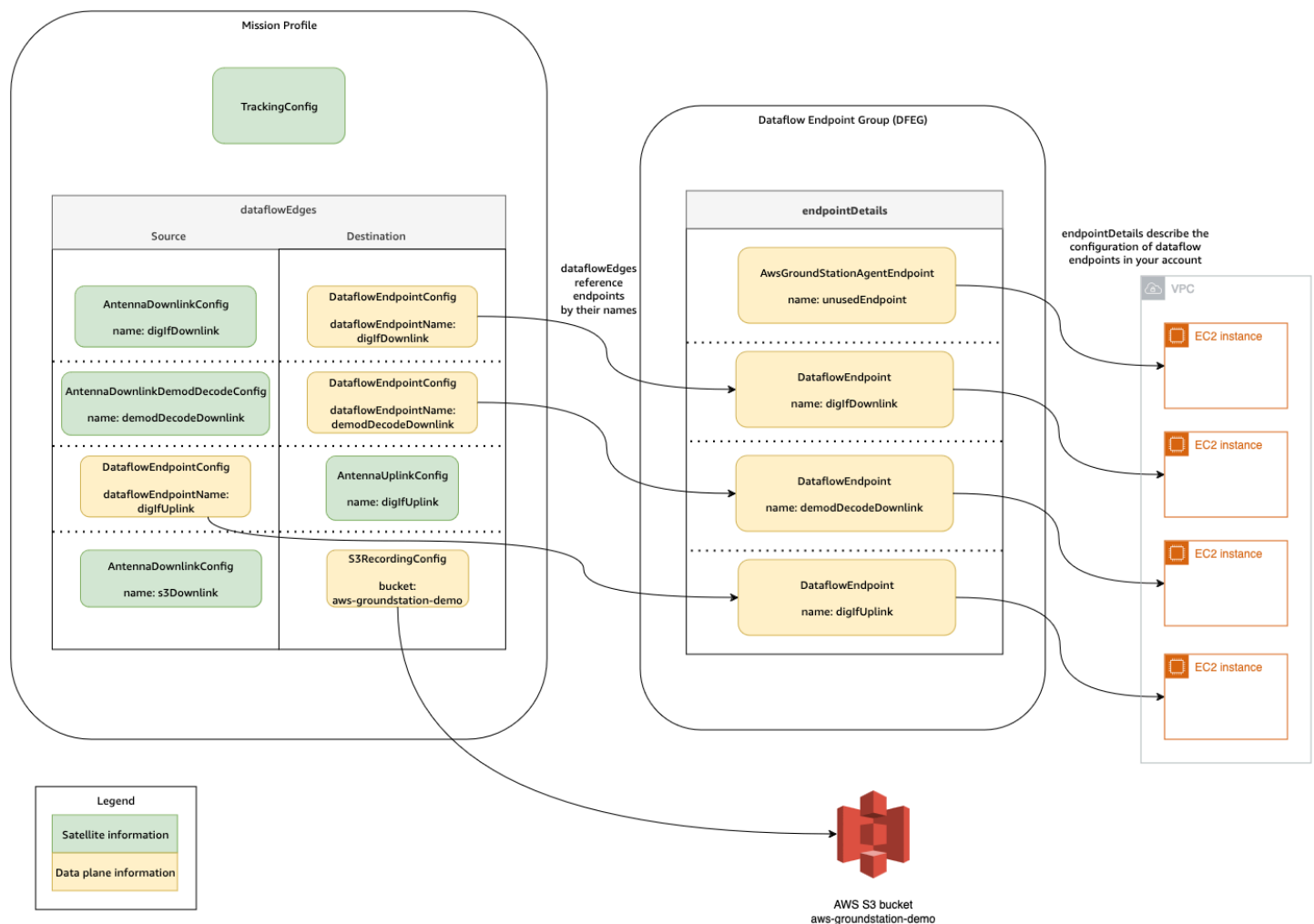
有效的任务配置文件将具有跟踪配置和一个或多个数据流。跟踪配置将指定您在联系期间的跟踪偏好。数据流中的每个配置对都会建立源和目标。根据您的卫星及其运行模式，数据流的确切数量将在任务配置文件中有所不同，以代表您的上行链路和下行链路通信路径以及任何数据处理方面。

- 有关配置将在联系期间使用的亚马逊 VPC、Amazon S3 和 Amazon EC2 资源的更多信息，请参阅[使用数据流](#)。
- 有关每种配置行为方式的详细信息，请参阅[使用 AWS Ground Station 配置](#)。

- 有关所有预期参数的具体详细信息，请参阅[使用 AWS Ground Station 任务配置文件](#)。
- 有关如何创建各种任务配置文件以支持您的用例的示例，请参阅[任务配置文件配置示例](#)。

下图显示了任务概况示例和所需的资源。请注意，该示例显示了一个名为 `unuseDendPoint` 的任务配置文件不需要的数据流端点，以证明其灵活性。该示例支持以下数据流：

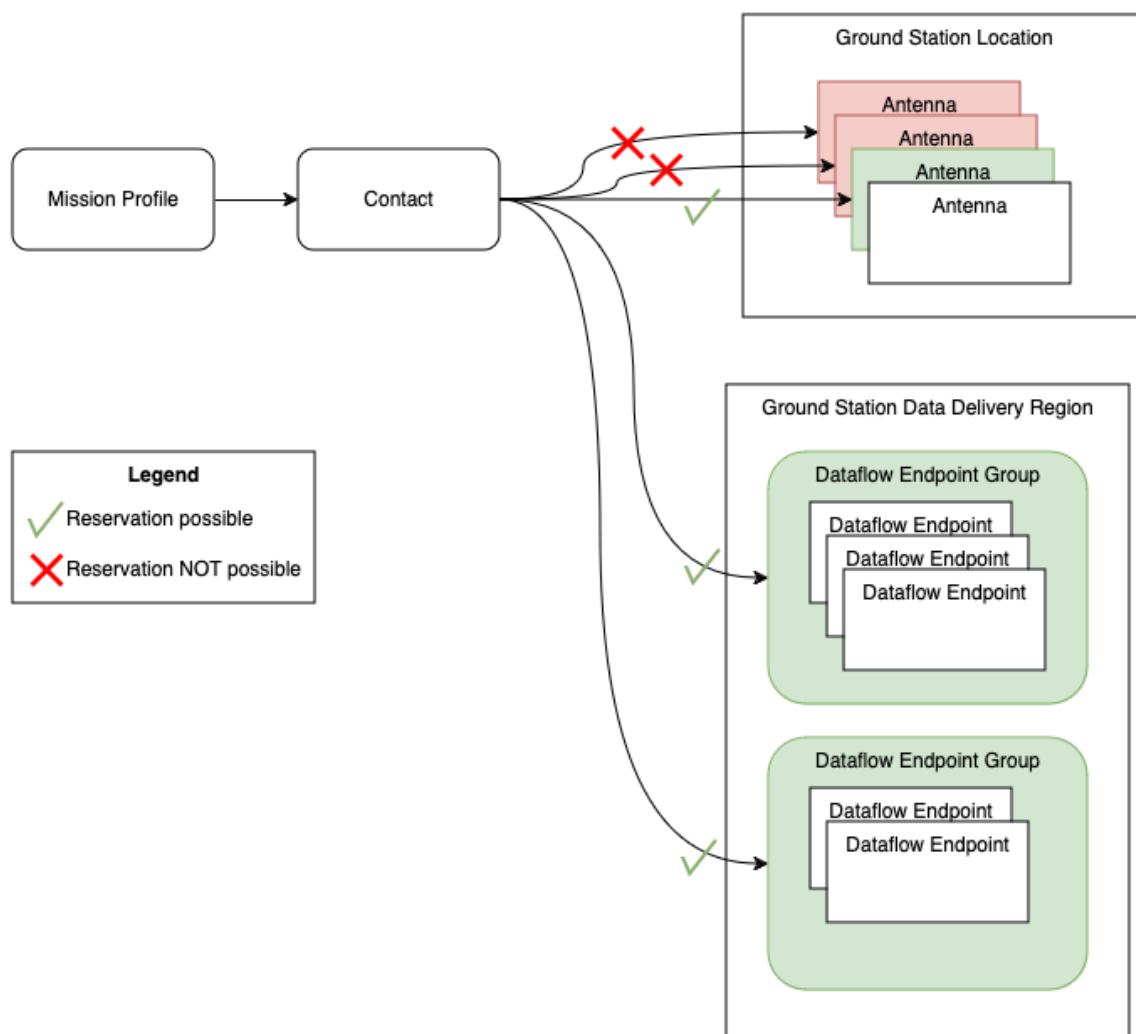
- 将数字中频数据同步下行链接到您管理的 Amazon EC2 实例。用名字 `digIfDownlink` 表示。
- 将数字中频数据异步下行链路到 Amazon S3 存储桶。由存储桶名称 `aws-groundstation-demo` 表示。
- 将解调和解码后的数据同步下行链接到您管理的 Amazon EC2 实例。用名字 `demoDecodeDownlink` 表示。
- 将数据从您管理的 Amazon EC2 实例同步上行链路到 AWS Ground Station 托管天线。用名字 `digIfUplink` 表示。



联系人日程安排

有了有效的任务资料，你就可以申请与你的机载卫星取得联系。联系人预订请求是异步的，以便全球天线服务有时间在所有相关 AWS 区域实现一致的时间表。在此过程中，将对请求的地面站位置的各种天线进行评估，以确定它们是否可用并能够处理接点。在此过程中，还会评估您配置的数据流端点以确定其可用性。进行此评估时，联系人状态将处于 SCHEDULING。

此异步调度过程将在请求发出后的五分钟内完成，但通常在一分钟内完成。请在计划[自动化 AWS Ground Station 含活动](#)期间查看基于事件的监控。



可以执行且具有可用性的联系会生成已安排的联系人。对于预定联系人，您的联系所需的资源已预留在您的任务资料所定义的所需的 AWS 区域中。无法执行或部件不可用的联系将导致 `FAILED_TO_SCHEDULE` 联系失败。有关调试[排查计划失败的联系人故障](#)的详细信息，请参见。

联系执行

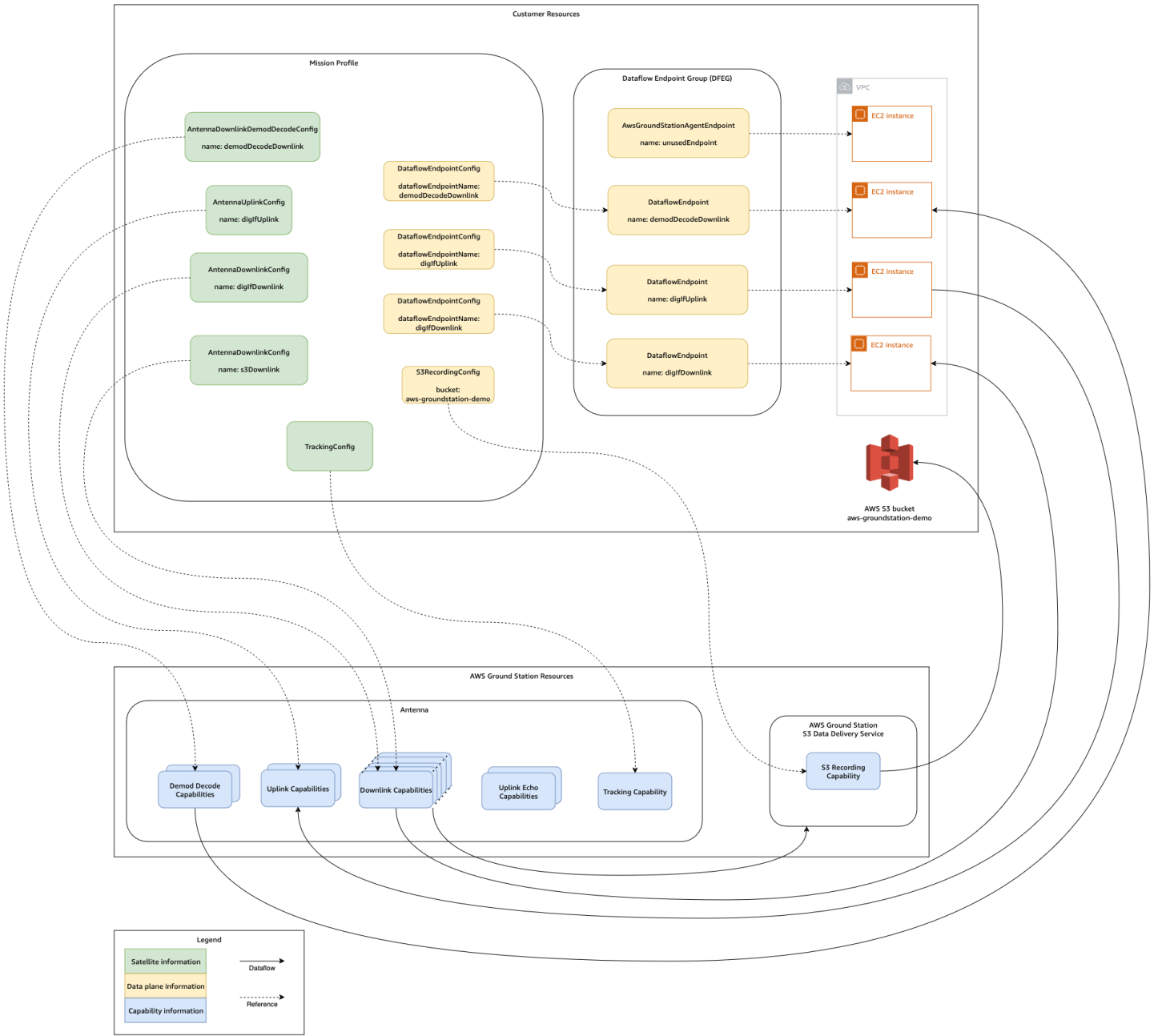
AWS Ground Station 将在联系人预约期间自动编排您的 AWS 托管资源。如果适用，您负责编排任务配置文件中定义为数据流终端节点的 EC2 资源。AWS Ground Station 提供 [AWS EventBridge 事件](#)，用于自动编排您的资源以降低成本。有关更多信息，请参阅[自动化 AWS Ground Station 含活动](#)。

在联系期间，有关您的联系表现的遥测数据将传送到 AWS CloudWatch。有关如何在执行期间监控您的联系人的信息，请参阅[通过以下方式了解监控 AWS Ground Station](#)。

下图延续了前面的示例，显示了联系期间编排的相同资源。

Note

本示例中并未使用所有天线功能。例如，每根天线都有十几种支持多种频率和极化的天线下行链路功能。有关 AWS Ground Station 天线提供的每种能力类型的数量及其支持的频率和极化的更多详细信息，请参阅[AWS Ground Station 网站能力](#)



在联系结束时，AWS Ground Station 将评估您的联系人绩效，并确定最终联系状态。未检测到错误的联系人将导致联系人状态为“已完成”。联系人如果在联系期间出现服务错误导致数据传输问题，则会导致 `AWS_FAILED` 状态。如果联系人在联系期间出现客户或用户错误导致数据传输问题，则联系人将导致“失败”状态。在裁决期间，不考虑联系时间以外的错误，即通过前或通过后的错误。

请参阅[了解联系生命周期](#)了解更多信息。

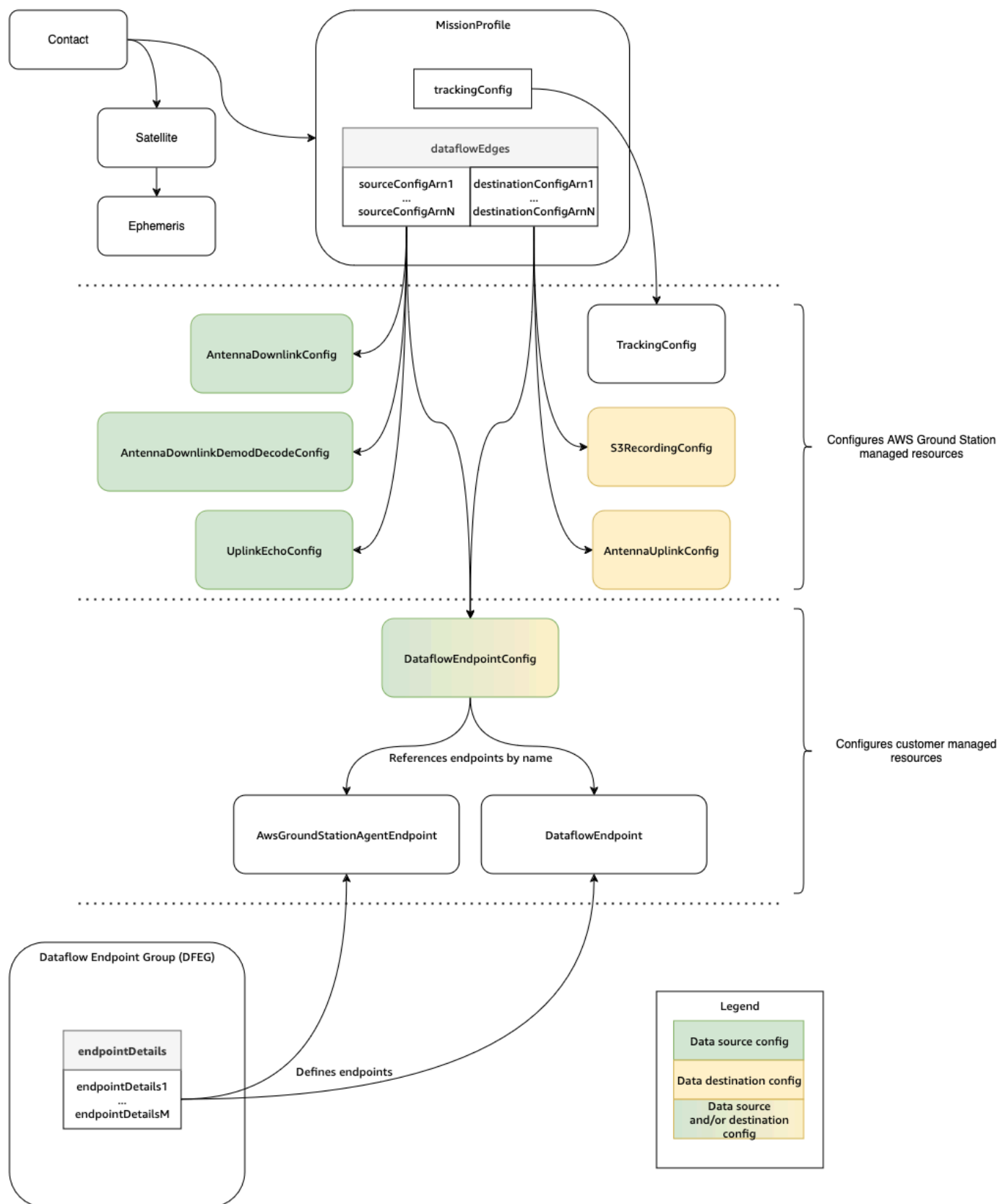
数字双胞胎

的数字双胞胎功能 AWS Ground Station 允许您根据虚拟地面站的位置安排联系。这些虚拟地面站是生产地面站的精确复制品，包括天线功能、站点掩码和实际的GPS坐标。与生产地面站相比，数字双胞胎功能使您能够测试联系人协调工作流程，而成本只是生产地面站的一小部分。参阅 [使用AWS Ground Station数字双胞胎功能](#) 了解更多信息。

明白 AWS Ground Station 核心组件

本节提供了 AWS 地面站核心组件的详细定义。

下图显示了核心组件 AWS Ground Station 及其相互关系。箭头表示组件之间依赖关系的方向，其中每个组件指向其依赖关系。



以下主题详细描述了 AWS Ground Station 核心组件。

主题

- [使用 AWS Ground Station 任务配置文件](#)
- [使用 AWS Ground Station 配置](#)
- [使用 AWS Ground Station 数据流终端节点组](#)
- [使用 AWS Ground Station 座席](#)

使用 AWS Ground Station 任务配置文件

任务配置文件包含的配置和参数说明了如何执行联络。预留联络或搜索可用联络时，您提供打算使用的任务配置文件。任务配置文件将您的所有配置集合在一起，并定义了联络期间数据将去往何处。

任务概况可以在具有相同无线电特性的卫星之间共享。您可以创建其他数据流终端节点组，以限制您希望为星座执行的最大同步联系人数量。

跟踪配置在任务配置文件中被指定为唯一字段。跟踪配置用于指定您在联系期间对使用程序跟踪和自动跟踪的偏好。有关更多信息，请参阅 [跟踪配置](#)。

所有其他配置都包含在任务配置文件dataflowEdges字段中。可以将这些配置视为数据流节点，每个节点都代表可以发送或接收数据及其关联配置的 AWS Ground Station 托管资源。

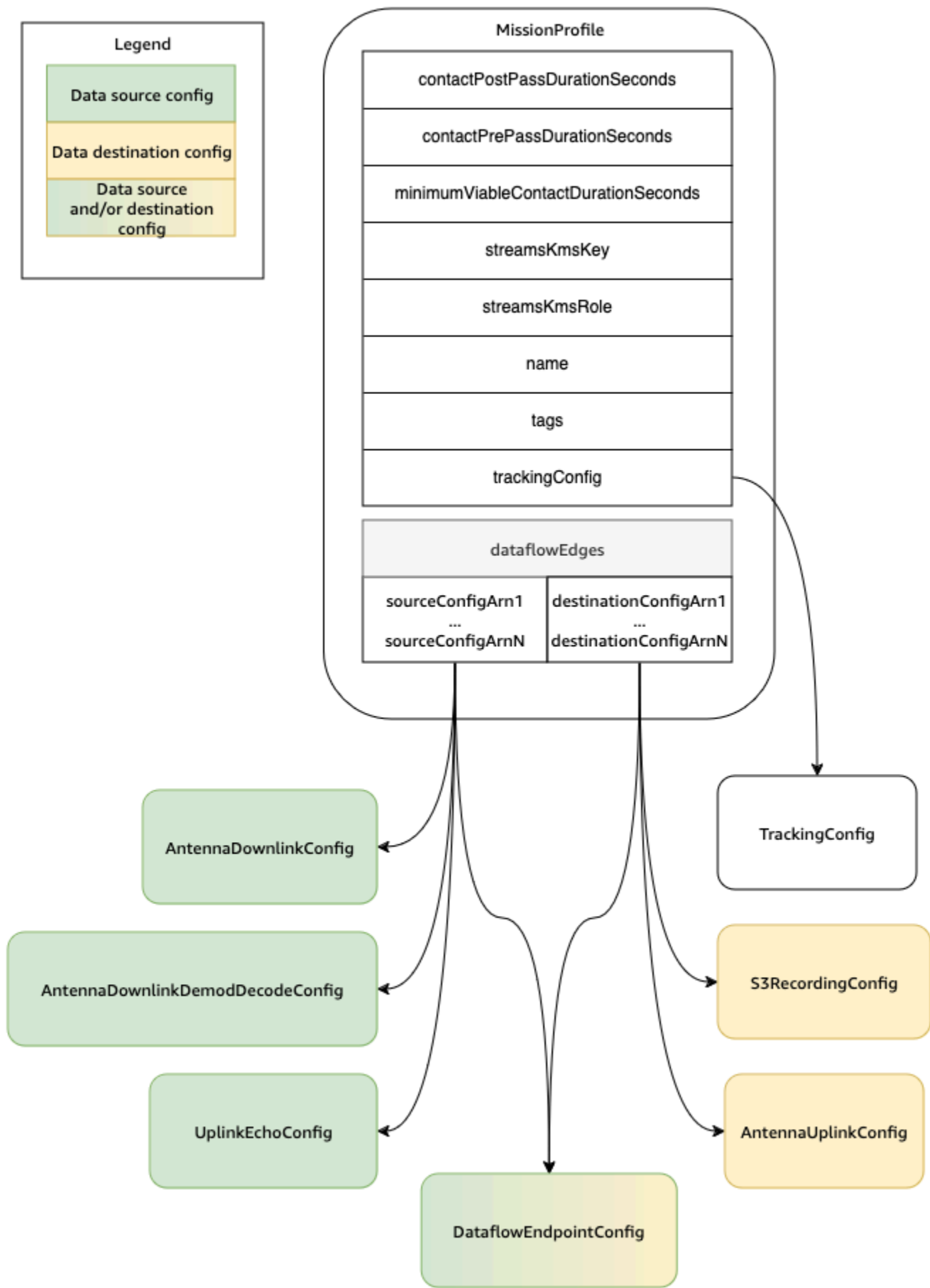
该dataflowEdges字段定义需要哪些源和目标数据流节点（配置）。单个数据流边缘是两个配置 [Amazon 资源名称 \(ARN\) 的列表](#) ——第一个是源配置，第二个是目标配置。通过在两个配置之间指定数据流边缘，您可以判断在接触期间数据应 AWS Ground Station 从何处流向何处。有关更多信息，请参阅 [使用 AWS Ground Station 配置](#)。

contactPrePassDurationSeconds和contactPostPassDurationSeconds允许您指定与您收到 CloudWatch 活动通知的联系人相关的时间。有关与您的联系人相关的事件的时间表，请阅读[了解联系生命周期](#)。

任务配置文件的 name 字段有助于区分您创建的任务配置文件。

streamsKmsRole和streamsKmsKey用于定义 AWS Ground Station 用于通过 AWS Ground Station Agent 传输数据时使用的加密。请参阅 [传输期间的数据加密 AWS Ground Station](#)。

该telemetrySinkConfigArn字段是可选的，允许您在联系期间启用 AWS Ground Station 遥测。指定后，在执行联系人期间，将近乎实时的遥测数据 AWS Ground Station 流式传输到您的账户。有关配置和使用遥测的更多信息，请参阅[使用遥测](#)。



以下文档中包含参数和示例的完整列表。

- [AWS::GroundStation:: MissionProfile CloudFormation 资源类型](#)

使用 AWS Ground Station 配置

配置是用于定义联系人各个方面的参数的资源。AWS Ground Station 将您想要的配置添加到任务配置文件，然后在执行联络时使用该任务配置文件。您可以定义几种不同类型的配置。这些配置可以分为三类：

- 跟踪配置
- 数据流配置
- 遥测配置

A TrackingConfig是唯一的跟踪配置类型。它用于在接触期间配置天线的自动跟踪设置，并且在任务配置文件中是必需的。

可以在任务配置文件数据流中使用的配置可以看作是数据流节点，每个节点都代表可以发送或接收数据的 AWS Ground Station 托管资源。任务配置文件至少需要一对这样的配置，其中一个代表数据源，一个代表目的地。下表汇总了这些配置。

Config 名称	数据流源/目的地
AntennaDownlinkConfig	来源
AntennaDownlinkDemodDecodeConfig	来源
UplinkEchoConfig	来源
S3 RecordingConfig	目标位置
AntennaUplinkConfig	目标位置
DataflowEndpointConfig	源 and/or 目的地

A TelemetrySinkConfig是遥测配置的唯一类型。它用于配置在接触期间将遥测数据传送到何处，并且在任务配置文件中是可选的。如果包括在内，则会在执行联系人期间将近乎实时的遥测 AWS Ground Station 数据流式传输到您的帐户。

有关如何使用 CloudFormation、或 AWS Ground Station API 对配置执行操作的更多信息，请参阅以下文档。AWS Command Line Interface 下面还提供了针对特定配置类型文档的链接。

- [AWS::GroundStation::Config CloudFormation 资源类型](#)
- [Config AWS CLI 参考](#)
- [配置 API 参考](#)

跟踪配置

您可以使用任务配置文件中的跟踪配置来确定是否应在您的联络期间启用自动跟踪。此配置只有一个参数：autotrack。autotrack 参数可能具有以下值：

- REQUIRED：您的联络需要自动跟踪。
- PREFERRED：最好对联络启用自动跟踪，但如果不使用，仍然可以执行联络。
- REMOVED：不应对您的联络启用自动跟踪。

AWS Ground Station 将使用编程跟踪，当不使用自动跟踪时，它会根据你的星历进行指向。有关星历构造方法[明白怎么做 AWS Ground Station 使用星历](#)的详细信息，请参考。

在找到预期信号之前，Autotrack 将使用节目跟踪。一旦发生这种情况，它将继续根据信号的强度进行跟踪。

有关如何使用 CloudFormation、或 AWS Ground Station API 对跟踪配置执行操作的更多信息，请参阅以下文档。AWS Command Line Interface

- [AWS::GroundStation::Config TrackingConfig CloudFormation 财产](#)
- [Config AWS CLI 参考 \(参见trackingConfig -> \(structure\)部分 \)](#)
- [TrackingConfig API 参考](#)

天线下行传输配置

您可在联络期间使用天线下行链路配置，以配置需要下行链路传输的天线。它们包括一个光谱配置，其中指定了下行联络期间应使用的频率、带宽和极化。

此配置表示数据流中的源节点。它负责对射频数据进行数字化。从该节点流出的数据将遵循信号 Data/IP 格式。有关如何使用此配置构造数据流的更多详细信息，请参阅 [使用数据流](#)

如果您的下行传输用例需要解调或解码，请参阅 [天线下行传输解调解码配置](#)。

有关如何使用 CloudFormation、或 API 对天线下行链路配置执行操作的更多信息，请参阅以下文档。
AWS Command Line Interface AWS Ground Station

- [AWS::GroundStation::Config AntennaDownlinkConfig CloudFormation 财产](#)
- [Config AWS CLI 参考 \(参见antennaDownlinkConfig -> \(structure\)部分\)](#)
- [AntennaDownlinkConfig API 参考](#)

天线下行传输解调解码配置

天线下行链路演示解码配置是一种更复杂且可自定义的配置类型，可用于通过解调解码执行下行链路接触。and/or 如果您有兴趣执行这些类型的联系人，AWS 支持 请通过 [AWS Support Center Console](#)。我们将帮助您确定适合您的用例的正确配置和任务配置文件。

此配置表示数据流中的源节点。它负责对射频数据进行数字化处理，并按照规定执行解调和解码。从该节点流出的数据将遵循 Demodulated/Decoded 数据/IP 格式。有关如何使用此配置构造数据流的更多详细信息，请参阅 [使用数据流](#)

有关如何使用、或 API 对天线下行链路演示解码配置执行操作的更多信息 CloudFormation，请参阅以下文档。AWS Command Line Interface AWS Ground Station

- [AWS::GroundStation::Config AntennaDownlinkDemodDecodeConfig CloudFormation 财产](#)
- [Config AWS CLI 参考 \(参见antennaDownlinkDemodDecodeConfig -> \(structure\)部分\)](#)
- [AntennaDownlinkDemodDecodeConfig API 参考](#)

天线上行传输配置

您可以在联络期间使用天线上行链路配置，以配置需要上行链路传输的天线。它们由包含频率、极化和目标有效各向同性辐射功率 (EIRP) 的频谱配置组成。有关如何配置上行回波的信息，请参阅 [天线上行传输回波配置](#)。

此配置表示数据流中的目标节点。它会将提供的数字化射频数据信号转换为模拟信号，然后将其发射给您的卫星接收。流向该节点的数据预计将符合信号 Data/IP 格式。有关如何使用此配置构造数据流的更多详细信息，请参阅 [使用数据流](#)

有关如何使用 CloudFormation、或 API 对天线上行链路配置执行操作的更多信息，请参阅以下文档。
AWS Command Line Interface AWS Ground Station

- [AWS::GroundStation::Config AntennaUplinkConfig CloudFormation 财产](#)
- [Config AWS CLI 参考 \(参见antennaUplinkConfig -> \(structure\)部分 \)](#)
- [AntennaUplinkConfig API 参考](#)

天线上行传输回波配置

上行传输回波配置告诉天线如何执行上行传输回波。上行链路回声可用于验证发送到航天器的命令，并执行其他高级任务。这是通过记录 AWS Ground Station 天线（即上行链路）传输的实际信号来实现的。这会回声天线发送回您的数据流端点的信号，并且应该与传输的信号相匹配。上行传输回波配置包含上行传输配置的 ARN。天线使用在执行上行传输回波时 ARN 指向的上行传输配置中的参数。

此配置表示数据流中的源节点。从该节点流出的数据将符合信号 Data/IP 格式。有关如何使用此配置构造数据流的更多详细信息，请参阅 [使用数据流](#)

有关如何使用 CloudFormation、或 API 对上行链路 echo 配置执行操作的更多信息，请参阅以下文档。AWS Command Line Interface AWS Ground Station

- [AWS::GroundStation::Config UplinkEchoConfig CloudFormation 财产](#)
- [Config AWS CLI 参考 \(参见uplinkEchoConfig -> \(structure\)部分 \)](#)
- [UplinkEchoConfig API 参考](#)

数据流端点配置

Note

数据流端点配置仅用于向 Amazon EC2 传输数据，不用于向 Amazon S3 传输数据。

您可以使用数据流端点配置来指定联络期间从[数据流端点组](#)中的哪个数据流端点流入或流向哪个数据流端点。数据流端点配置的两个参数指定数据流端点的名称和区域。预订联系人时，AWS Ground Station 会分析您指定的[任务配置文件](#)并尝试在 AWS 区域内找到一个数据流端点组，该组包含任务配置文件中包含的数据流端点配置所指定的所有数据流端点。如果找到了合适的数据流端点组，则联系状态将变为“已计划”，否则将变为 FAILED_TO_SCHEDULE。有关联系人可能的状态的更多信息，请参阅[AWS Ground Station 联系人状态](#)。

数据流端点配置的 dataflowEndpointName 属性指定联络期间从数据流端点组中的哪个数据流端点流入或流向哪个数据流端点。

`dataflowEndpointRegion` 属性指定数据流端点所在的区域。如果在您的数据流终端节点配置中指定了区域，则会在指定区域中 AWS Ground Station 查找数据流终端节点。如果未指定区域，AWS Ground Station 则默认为联系人的地面站区域。如果您的数据流端点的区域与联络的地面站区域不同，则该联络被视为跨区域数据传输联络。[使用数据流](#)有关跨区域数据流的更多信息，请参阅。

有关数据流[使用 AWS Ground Station 数据流终端节点组](#)的不同命名方案如何使您的用例受益的提示，请参阅。

有关如何使用此配置构造数据流的更多详细信息，请参阅 [使用数据流](#)

有关如何使用、或 API 对数据流端点配置执行操作的更多信息 CloudFormation，请参阅以下文档。
AWS Command Line Interface AWS Ground Station

- [AWS::GroundStation::Config DataflowEndpointConfig CloudFormation 财产](#)
- [Config AWS CLI 参考 \(参见dataflowEndpointConfig -> \(structure\)部分\)](#)
- [DataflowEndpointConfig API 参考](#)

亚马逊 S3 录音 Config

Note

Amazon S3 记录配置仅用于向亚马逊 S3 传输数据，不用于向亚马逊 EC2 传输数据。

此配置表示数据流中的目标节点。该节点会将来自数据流源节点的传入数据封装到 pcap 数据中。有关如何使用此配置构造数据流的更多详细信息，请参阅 [使用数据流](#)

您可以使用 S3 记录配置来指定要将下行链接数据以及使用的命名约定传送到的 Amazon S3 存储桶。以下内容指定了有关这些参数的限制和详细信息：

- Amazon S3 存储桶的名称必须以 `aws-groundstation` 开始。
- IAM 角色必须拥有信任策略，以允许 `groundstation.amazonaws.com` 服务主体承担该角色。有关示例，请参阅下面的[示例信任策略](#)部分。在配置创建过程中，配置资源 ID 不存在，信任策略必须使用星号 (*) 代替，*your-config-id*并且可以在创建后使用配置资源 ID 进行更新。

示例信任策略

有关如何更新角色的信任策略的更多信息，请参阅 IAM 用户指南中的[管理 IAM 角色](#)。

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 角色必须拥有 IAM policy，以允许该角色对存储桶执行 s3:GetBucketLocation 操作以及对存储桶对象执行 s3:PutObject 操作。如果 Amazon S3 存储桶拥有存储桶策略，则该存储桶策略还必须允许 IAM 角色执行这些操作。有关示例，请参阅下面的[示例角色策略](#)部分。

示例角色策略

有关如何更新或附加角色策略的更多信息，请参阅 IAM 用户指南中的[管理 IAM 策略](#)。

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/*"
    ]
}
]
}

```

- 命名 S3 数据对象时将使用前缀。您可以指定可选的密钥进行替换，这些值将替换为联系人详细信息中的相应信息。例如，前缀{satellite_id}/{year}/{month}/{day}将被替换，其结果将是类似的输出 fake_satellite_id/2021/01/10

替换的可选密钥：{satellite_id}{config-name}| {config-id} || {year} | {month} | {day} |

有关如何使用 CloudFormation、或 AWS Ground Station API 对 S3 录制配置执行操作的更多信息，请参阅以下文档。AWS Command Line Interface

- [AWS::GroundStation::Config S3 RecordingConfig CloudFormation 属性](#)
- [Config AWS CLI 参考 \(参见s3RecordingConfig -> \(structure\)部分\)](#)
- [S3 RecordingConfig API 参考](#)

遥测接收器 Config

您可以使用遥测接收器配置来指定在卫星接触期间要将遥测数据传送到何处。遥测接收器配置是可选的，已添加到您的任务配置文件中，用于安排支持遥测的联系人。以下内容指定了有关这些参数的限制和详细信息：

- IAM 角色必须拥有信任策略，以允许 `groundstation.amazonaws.com` 服务主体承担该角色。有关示例，请参阅下面的[示例信任策略](#)部分。

示例信任策略

有关如何更新角色的信任策略的更多信息，请参阅 IAM 用户指南中的[管理 IAM 角色](#)。

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

- IAM 角色必须有一个 IAM 策略，允许该角色在直播中执行 `kinesis:DescribeStream`、`kinesis:PutRecord` 和 `kinesis:PutRecords` 操作。有关示例，请参阅下面的[示例角色策略](#)部分。

示例角色策略

有关如何更新或附加角色策略的更多信息，请参阅 IAM 用户指南中的[管理 IAM 策略](#)。

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

当您在任务配置文件中包含遥测接收器配置时，AWS Ground Station 将在联系期间将遥测数据流式传输到您的帐户。有关遥测类型、数据格式和设置必要 AWS 资源的更多信息，请参阅[使用遥测](#)。

有关如何使用 CloudFormation、或 API 对遥测接收器配置执行操作的更多信息，请参阅以下文档。
AWS Command Line Interface AWS Ground Station

- [AWS::GroundStation::Config TelemetrySinkConfig CloudFormation 财产](#)
- [Config AWS CLI 参考 \(参见telemetrySinkConfig -> \(structure\)部分\)](#)
- [TelemetrySinkConfig API 参考](#)

使用 AWS Ground Station 数据流终端节点组

Dataflow 端点定义了您希望在联系期间同步流入或流出数据的位置。数据流终端节点总是作为数据流终端节点组的一部分创建的。在组中包括多个数据流终端节点，即表示您主张那些指定的终端节点都可以在单次联络中一起使用。例如，如果联络需要向三个独立的数据流端点发送数据，则单个数据流端点组中必须有三个端点与您的任务配置文件中的数据流端点配置相匹配。

数据流终端节点组版本

AWS Ground Station 支持两个版本的数据流端点组：

- DataflowEndpointGroup-[原始实现支持使用数据流端点的上行链路和下行链路，并且仅支持代理端点的下行链路AWS Ground Station](#)
- DataflowEndpointGroupV2-更新版本支持 AWS Ground Station 代理端点的上行链路和下行链路数据流，提高了清晰度和功能性

数据流终端节点组比较

功能	DataflowEndpointGroup	DataflowEndpointGroupV2
支持的端点类型	DataflowEndpoint, AwsGroundStationAgentEndpoint	DownlinkAwsGroundStationAgentEndpoint, UplinkAwsGroundStationAgentEndpoint
支持上行链路的终端	DataflowEndpoint	UplinkAwsGroundStationAgentEndpoint
支持下行链路的端点	DataflowEndpoint, AwsGroundStationAgentEndpoint	DownlinkAwsGroundStationAgentEndpoint

DataflowEndpointGroupV2 的创建是为了支持上行链路数据流，并使数据流端点组周围的语言更清晰。对于所有新的用例 [UplinkAwsGroundStationAgentEndpoint](#)，我们建议使用带有 [DataflowEndpointGroupV2](#) 的和 [DownlinkAwsGroundStationAgentEndpoint](#) 终端节点。

DataflowEndpointGroup 仍然支持向后兼容，但是 DataflowEndpointGroup V2 提供了增强的功能和更清晰的配置选项。

 Tip

执行联系人时，数据流端点由您选择的名称标识。这些名称在整个账户中不必是唯一的。这允许使用相同的任务配置文件在不同的卫星和天线上同时执行多个接触。如果您拥有一组具有相同运行特性的卫星，这可能会很有用。您可以向上扩展 Dataflow 端点组的数量，以适应您的卫星星座所需的最大同步联系数量。

当数据流终端节点组中的一个或多个资源正用于某个联络时，在该次联络期间整个组都予以保留。您可以同时执行多个联络，但这些联络必须在不同的数据流端点组中执行。

 Important

数据流端点组必须处于 HEALTHY 状态，才能用于安排联络。有关如何对未处于 HEALTHY 状态的数据流终端节点组进行故障排除的信息，请参阅 [排除 DataflowEndpointGroups 未处于正常状态的故障](#)

有关如何使用 CloudFormation、或 API 对数据流终端节点组执行操作的更多信息，请参阅以下文档。
AWS Command Line Interface AWS Ground Station

- [AWS::GroundStation::DataflowEndpointGroup CloudFormation 资源类型](#)
- [数据流终端节点组参考 AWS CLI](#)
- [数据流端点组 API 参考](#)

数据流终端节点

数据流终端节点组的成员是数据流端点。支持的终端节点类型取决于您使用的数据流终端节点组版本。

DataflowEndpointGroup 端点

DataflowEndpointGroup [支持使用数据流端点的上行链路和下行链路，并且仅支持代理端点的下行链路](#)。[AWS Ground Station](#) 对于这两种类型的端点，您将在创建 dataflow 端点组之前创建支持结构（例如 IP 地址）。[使用数据流](#) 有关使用哪种 dataflow 端点类型以及如何设置支持结构的建议，请参阅。

以下各节描述了两类支持的端点类型。

Important

单个数据流终端节点组中的所有数据流端点必须属于同一类型。您不能在同一个组中将 [AWS Ground Station 代理端点](#) 与 [Dataflow 端点](#) 混合使用。如果您的用例需要两种类型的终端节点，则必须为每种类型创建单独的 dataflow 端点组。

对于 DataflowEndpointGroup V2，您可以将 [UplinkAwsGroundStationAgentEndpoint](#) 和混合到同一个组 [DownlinkAwsGroundStationAgentEndpoint](#) 中。

AWS Ground Station 代理端点

AWS Ground Station 代理端点使用 AWS Ground Station 代理作为软件组件来终止连接。要构造 AWS Ground Station 代理终端节点，您只需要填充的 `AwsGroundStationAgentEndpointEndpointDetails` 字段。有关 AWS Ground Station 代理的更多信息，请参阅完整的 [《AWS Ground Station 代理用户指南》](#)。

`AwsGroundStationAgentEndpoint` 由以下内容组成：

- Name-数据流端点名称。要使联系人使用此数据流端点，此名称必须与您的数据流端点配置中使用的名称相匹配。

- EgressAddress-用于从代理输出数据的 IP 和端口地址。
- IngressAddress-用于将数据传入代理的 IP 和端口地址。

数据流端点

Dataflow Endpoint 使用网络应用程序作为软件组件来终止连接。如果要上行传输数字信号数据、下行链路少于 50% MHz 的数字信号数据或下行链路信号数据，请使用 Dataflow Endpoint。Demodulated/Decoded 要构建 Dataflow Endpoint，您需要填充Endpoint和Security Details字段。EndpointDetails

Endpoint 由以下内容组成：

- Name-数据流端点名称。要使联系人使用此数据流端点，此名称必须与您的数据流端点配置中使用的名称相匹配。
- Address-使用的 IP 和端口地址。

SecurityDetails 由以下内容组成：

- roleArn- AWS Ground Station 将担任在您的 VPC 中创建弹性网络接口 () 的角色的亚马逊资源名称 (ARNENIs)。它们 ENIs 充当联系期间流式传输的数据的入口和输出点。
- securityGroupIds：附加到弹性网络接口的安全组。
- subnetIds- AWS Ground Station 可能放置弹性网络接口以向您的实例发送流的子网列表。如果指定了多个子网，则它们必须可以相互路由。如果子网位于不同的可用区 (AZs)，则可能会产生跨可用区数据传输费用。

传递到 roleArn 的 IAM 角色必须拥有信任策略，以允许 groundstation.amazonaws.com 服务主体代入该角色。有关示例，请参阅下面的[示例信任策略](#)部分。在创建端点期间，端点资源 ID 不存在，因此信任策略必须使用星号 (*) 代替。*your-endpoint-id*这可以在创建后进行更新以使用端点资源 ID，从而将信任策略的范围限定为该特定的数据流端点组。

IAM 角色必须具有 AWS Ground Station 允许设置的 IAM 策略 ENIs。有关示例，请参阅下面的[示例角色策略](#)部分。

示例信任策略

有关如何更新角色的信任策略的更多信息，请参阅 IAM 用户指南中的[管理 IAM 角色](#)。

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"
        }
      }
    }
  ]
}
```

示例角色策略

有关如何更新或附加角色策略的更多信息，请参阅 IAM 用户指南中的[管理 IAM 策略](#)。

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 终端节点

DataflowEndpointGroupV2 引入了专门的端点类型，可提供更清晰的配置和增强的功能：

- [UplinkAwsGroundStationAgentEndpoint](#)-针对上行链路数据流进行了优化
- [DownlinkAwsGroundStationAgentEndpoint](#)-针对下行链路数据流进行了优化

这些专用端点用特定方向的配置取代了通用[AwsGroundStationAgentEndpoint](#)端点，从而使设置和管理数据流变得更加容易。

上行链路 AWS Ground Station 代理端点

专[UplinkAwsGroundStationAgentEndpoint](#)为上行链路数据流而设计，可提供更清晰的配置选项。当您需要提供要上传到卫星的数据时，AWS Ground Station 请使用此端点类型。

UplinkAwsGroundStationAgentEndpoint 由以下内容组成：

- Name-数据流端点名称。要使联系人使用此数据流端点，此名称必须与您的数据流端点配置中使用的名称相匹配。
- IngressAddressAndPort-用于向代理输入数据的单个 IP 和端口地址
- AgentIpAndPortAddress-代理通信的端口范围

下行链路 AWS Ground Station 代理端点

针对下行链路数据流进行了优化，包括窄带下行链路、宽带解调/解码和上行链路回声场景。[DownlinkAwsGroundStationAgentEndpoint](#)

DownlinkAwsGroundStationAgentEndpoint 由以下内容组成：

- Name-数据流端点名称。要使联系人使用此数据流端点，此名称必须与您的数据流端点配置中使用的名称相匹配。
- EgressAddressAndPort-代理输出数据的单个 IP 和端口地址
- AgentIpAndPortAddress-代理通信的端口范围

创建数据流终端节点组

您可以使用以下任一版本创建数据流终端节点组：

CreateDataflowEndpointGroup

[CreateDataflowEndpointGroup](#)用于向后兼容，或者在需要使用泛型[AwsGroundStationAgentEndpoint](#)或[DataflowEndpoint](#)类型时使用。

CreateDataflowEndpointGroupV2

将 [CreateDataflowEndpointGroupV2](#) 用于新实现，以利用同时支持上行链路和下行链路数据流的专用端点类型。此 API 仅支持[UplinkAwsGroundStationAgentEndpoint](#)和[DownlinkAwsGroundStationAgentEndpoint](#)。

迁移注意事项

如果您当前正在使用 DataflowEndpointGroup，则无需更改即可继续使用现有配置。AWS Ground Station 保持完全的向后兼容性。

如果您想迁移以使用新的 DataflowEndpointGroup V2，并且目前正在使用[DataflowEndpoint](#)带有 Dataflow Endpoint 应用程序的应用程序来接收数据，则需要迁移以改用代理。AWS Ground Station 如果您已经在使用 AWS Ground Station 代理进行下行链路，则也可以将相同的代理实例用于上行链路，无需其他代理实例。

要迁移到 DataflowEndpointGroup V2，请执行以下操作：

1. 如果从中迁移 DataflowEndpoint，请按照《AWS Ground Station [AWS Ground Station 代理用户指南](#)》设置代理
2. 确定您的数据流方向并创建相应的端点类型
(或) [UplinkAwsGroundStationAgentEndpointDownlinkAwsGroundStationAgentEndpoint](#)
3. 创建引用这些端点的 [DataflowEndpointGroupV2](#)
4. 创建一个按名称引用新 DataflowEndpointGroup V2 的新[数据流端点配置](#)
5. 创建新的任务配置文件，将数据流端点配置引用为数据流边缘

6. 使用新的任务配置文件安排联系人
7. 在部署到生产环境之前测试您的配置

有关完整工作流程的更多信息，请参阅[明白 AWS Ground Station 核心组件](#)和[创建配置](#)。

使用 AWS Ground Station 座席

AWS Ground Station 代理使您能够在 AWS 地面站联系期间接收（下行链路）同步宽带数字中频 (DigiF) 数据流。

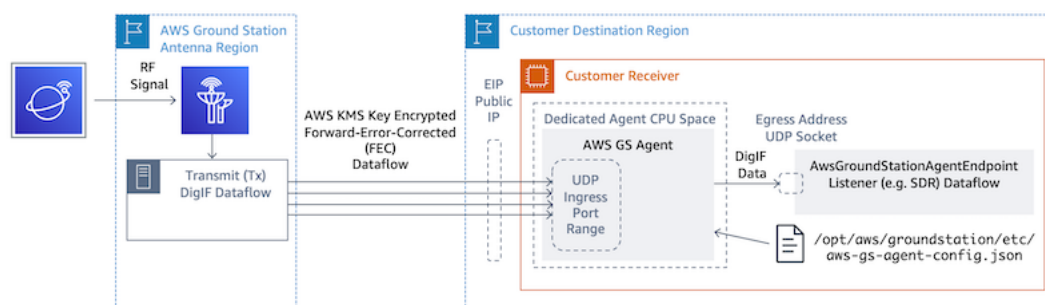
工作原理

您可以选择两个选项进行数据交付：

1. 向 EC2 实例传输数据 - 将数据传输到您拥有的 EC2 实例。您管理代 AWS Ground Station 理。如果您需要近乎实时的数据处理，则选择此选项。有关 EC2 数据传输的信息，请参阅该[使用数据流](#)部分。
2. 向 S3 存储桶传输数据 - 向 AWS S3 存储桶传输的数据完全由管理 AWS Ground Station。有关 S3 数据传输的信息，请参阅 [开始使用](#) 指南。

两种数据传输模式都需要您创建一组 AWS 资源。强烈建议使用 CloudFormation 来创建 AWS 资源，以确保可靠性、准确性和可支持性。每个联络只能向 EC2 或 S3 传输数据，但不能同时向两者传输数据。

下图显示了使用您的 Software-Defined 无线电 (SDR) 或类似监听器从 AWS Ground Station 天线区域流向您的 EC2 实例的 DigiF 数据流。



附加信息

有关更多详细信息，请参阅完整的《[AWS Ground Station 代理用户指南](#)》。

开始使用

在开始之前，您应该熟悉其中的基本概念。AWS Ground Station 有关更多信息，请参阅 [操作方法 AWS Ground Station 方式](#)。

注册参加 AWS 账户

要开始使用 AWS，你需要一个 AWS 账户。有关创建的信息 AWS 账户，请参阅《AWS 账户管理 参考指南》AWS 账户 [中的“入门”](#)。

添加 AWS Ground Station 您的权限 AWS account

要在不需要管理员用户 AWS Ground Station 的情况下使用，您需要创建新策略并将其附加到您的 AWS 账户。

1. 登录 AWS 管理控制台 并打开 [IAM 控制台](#)。
2. 创建新策略。使用以下步骤：
 - a. 在导航窗格中选择策略，然后选择创建策略。
 - b. 在 JSON 选项卡中，使用以下值之一编辑 JSON。使用最适合您的应用程序的 JSON。
 - 对于 Ground Station 管理权限，将操作设置为 groundstation:*，如下所示：

JSON

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

- 要获得 Read-only 权限，请将“操作”设置为 GroundStation: Get*、GroundStation: List* 和 GroundStation: Describe*，如下所示：

JSON

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

- 为了通过多因素身份验证提高安全性，请将 Action 设置为 groundstation:*，并 Condition/Bool 设置为 aws:MultiFactorAuthPresent:true 如下所示：

JSON

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

```
}
```

3. 在 IAM 控制台中，将您创建的策略附加到所需用户。

有关 IAM 用户并附加策略的更多信息，请参阅 [IAM 用户指南](#)。

机载卫星

将卫星载入 AWS Ground Station 是一个多步骤的过程，包括数据收集、技术验证、频谱许可，以及集成和测试。还需要保密协议 (NDAs)。

客户入职流程概述

卫星载入是一个手动流程，可以在 AWS Ground Station 控制台页面的“[卫星和资源](#)”部分找到。以下描述了整个过程。

1. 查看本[AWS Ground Station 位置](#)节以确定您的卫星是否符合地理和无线电频率特征。
2. 要开始将卫星载入 AWS Ground Station，请在 AWS Ground Station 主机页面的“[卫星和资源](#)”部分[提交卫星载](#)入问卷。请简要概述您的任务和卫星需求，包括您的组织名称、所需的频率、卫星的发射时间或发射时间、卫星的轨道类型以及您是否计划使用[使用AWS Ground Station数字双胞胎功能](#)。
3. 您的申请经过审核和批准后，AWS Ground Station 将在您计划使用的特定地点申请监管许可。此步骤的持续时间将因地点和任何现行法规而异。
4. 获得批准后，您的卫星将可见供您使用。AWS Ground Station 将向您发送更新成功通知。

(可选) 命名卫星

入职后，您可能需要在卫星记录中添加一个名称，以便更轻松地识别它。使用“联系人”页面时，AWS Ground Station 控制台能够显示用户定义的卫星名称以及 Norad ID。显示卫星名称可以在计划时更容易选择正确的卫星。为此，可以使用[标签](#)。

标记 AWS Ground Station 卫星可以通过[标签资源](#) API 和 AWS CLI 或其中一个 AWS 来完成。SDKs 本指南将介绍如何使用 AWS Ground Station CLI 标记公共广播卫星 Aqua (Norad ID 27424)。us-west-2

AWS Ground Station CLI

AWS CLI 可以用来与之交互 AWS Ground Station。在 AWS CLI 使用标记卫星之前，必须满足以下 AWS CLI 先决条件：

- 确保 AWS CLI 已安装。有关安装的信息 AWS CLI，请参阅[安装 AWS CLI 版本 2](#)。
- 确保 AWS CLI 已配置。有关配置的信息 AWS CLI，请参阅[配置 AWS CLI 版本 2](#)。
- 将您常用的配置设置和凭证保存在由 AWS CLI 维护的文件中。您需要这些设置和凭据来保留和管理您的 AWS Ground Station 联系人 AWS CLI。有关保存配置和凭据设置的更多信息，请参阅[配置和凭据文件](#)设置。

配置完成并可供使用后 AWS CLI，请查看 [AWS Ground Station CLI 命令参考](#) 页面，熟悉可用的命令。使用此服务时，请遵循 AWS CLI 命令结构，并在命令前加上前缀 `groundstation` AWS Ground Station 以指定要使用的服务。有关 AWS CLI 命令结构的更多信息，请参阅 [AWS CLI 页面中的命令结构](#)。下面提供了一个示例命令结构。

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

命名卫星

首先，您需要获取要标记的卫星的 ARN。这可以通过 AWS CLI 中的 [列表卫星](#) API 来完成：

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

运行上述 CLI 命令将返回类似于以下内容的输出：

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

找到您要标记的卫星，并记下 `satelliteArn`。[标记的一个重要注意事项是，标签资源 API 需要区域性 ARN，而列表卫星返回的 ARN 是全球性的。](#)在下一步中，您应该使用您希望看到标签的区域（可能是您安排的区域）来扩展 ARN。对于该示例，我们将使用 `us-west-2`。此次更改后，ARN 将从：

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

到：

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

为了在控制台中显示卫星名称，卫星必须有一个 “Name” 标签作为密钥。此外，由于我们使用的是 AWS CLI，因此必须使用反斜杠对引号进行转义。标签将类似于：

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

接下来，您将调用[标签资源](#) API 来标记卫星。这可以用 AWS CLI 这样的方法来完成：

```
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\"}'
```

完成此操作后，您将能够在 AWS Ground Station 控制台中看到为卫星设置的名称。

更改卫星的名称

如果要更改卫星的名称，则只需使用相同的 “Name” 密钥再次使用卫星 ARN 调用 [tag-resource](#)，但在标签中使用不同的值即可。这将更新现有标记并在控制台中显示新名称。调用示例如下：

```
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\"}'
```

删除卫星的名称

可以使用 [untag-resource](#) API 删除为卫星设置的名称。此 API 需要带有标记所在区域的卫星 ARN 以及标记密钥列表。对于名称，标记密钥为 “Name”。使用 AWS CLI 调用该 API 的示例如下所示：

```
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
```

公共广播卫星

除了登陆自己的卫星外，您还可以申请使用支持的公共广播卫星登机，这些卫星提供可公开访问的下行链路通信路径。这使您 AWS Ground Station 能够使用下行链路来自这些卫星的数据。

Note

您将无法通过上行链路连接到这些卫星。您将只能使用可公开访问的下行链路通信路径。

AWS Ground Station 支持载入以下卫星以下行链路直接广播数据：

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

这些卫星一旦登机，就可以立即使用了。AWS Ground Station 维护了许多预配置的 CloudFormation 模板，以便更轻松地开始使用该服务。[任务配置文件配置示例](#)有关 AWS Ground Station 如何使用的示例，请参阅。

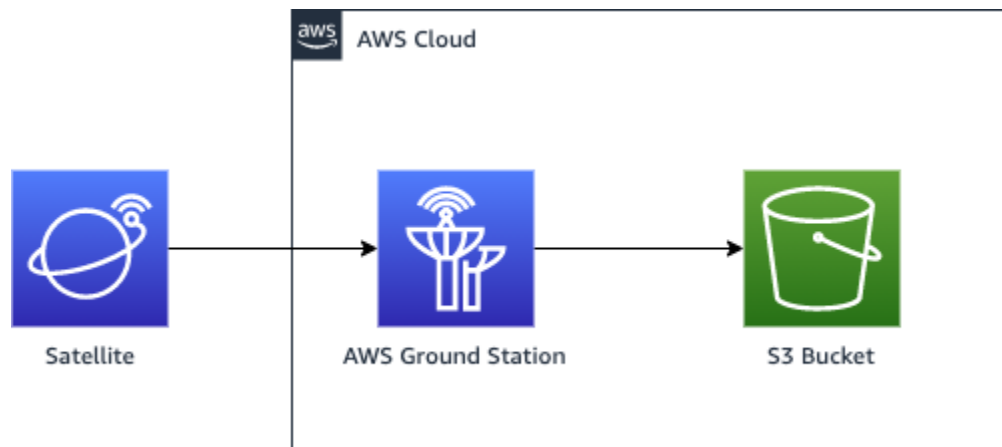
有关这些卫星及其传输的数据种类的更多信息，请参阅 [Aqua](#)、[JPSS-1/NOAA-20](#) 和 [SNPP](#) 以及 [Terra](#)。

规划您的数据流通信路径

您可以为卫星上的每条通信路径选择同步和异步通信。根据您的卫星和用例，您可能需要一种或两种类型。同步通信路径允许近乎实时的上行链路以及窄带和宽带下行链路操作。异步通信路径仅支持窄带和宽带下行链路操作。

异步数据传输

将数据传输至 Amazon S3 后，您的联络数据将异步传输到您账户中的 Amazon S3 存储桶。您的联络数据以数据包捕获 (pcap) 文件的形式传输，以允许将联络数据重放到软件定义无线电 (SDR) 中，或者从 pcap 文件中提取有效载荷数据进行处理。当天线硬件收到联络数据时，每 30 秒将 pcap 文件传输至您的 Amazon S3 存储桶，以便在联络期间需要时处理联络数据。收到数据后，您可以使用自己的后处理软件或使用其他 AWS 服务（如亚马逊 A SageMaker I 或 Amazon Rekognition）来处理数据。数据传输至 Amazon S3 仅适用于从您的卫星下载数据；无法将数据从 Amazon S3 上传到您的卫星。



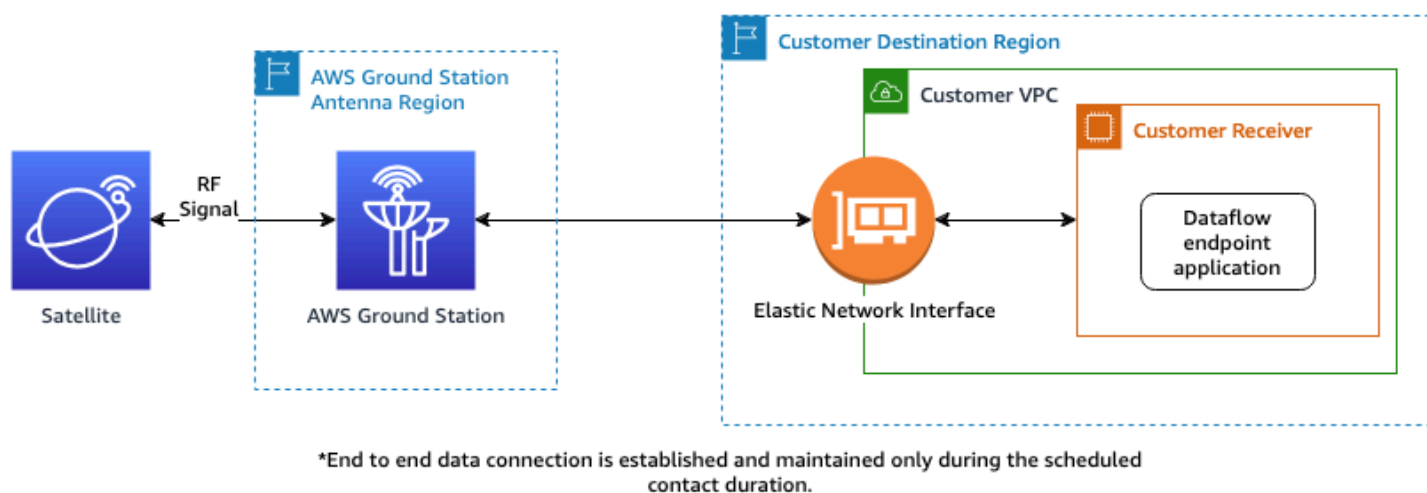
要使用此路径，您需要创建一个 Amazon S3 存储桶 AWS Ground Station 来将数据传输到。在下一步中，您还需要在下一步中创建 S3 录制配置。有关存储桶命名的限制以及如何指定文件使用的命名规范，请参阅。[亚马逊 S3 录音 Config](#)

同步数据传输

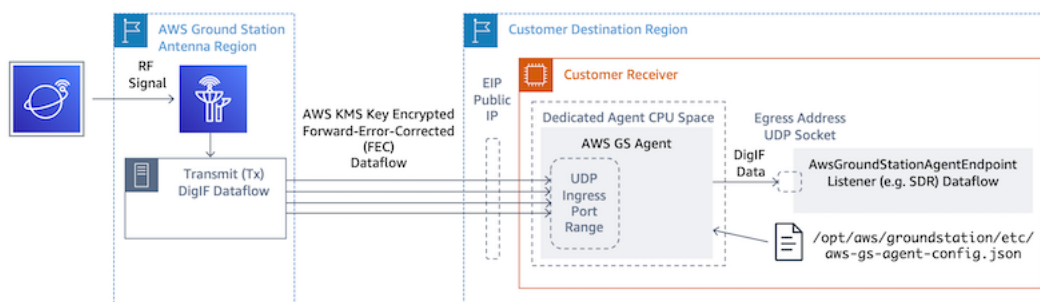
将数据传输至 Amazon EC2 后，您的联络数据将通过流式处理在 Amazon EC2 实例之间进行传输。您可以在 Amazon EC2 实例上实时处理数据，也可以转发数据进行后期处理。

要使用同步路径，您需要设置和配置您的 Amazon EC2 实例并创建一个或多个 Dataflow 终端节点组。要配置您的 Amazon EC2 实例，请参阅[设置和配置 Amazon EC2](#)。要创建您的数据流终端节点组，请参阅。[使用 AWS Ground Station 数据流终端节点组](#)

以下显示了使用数据流端点配置时的通信路径。



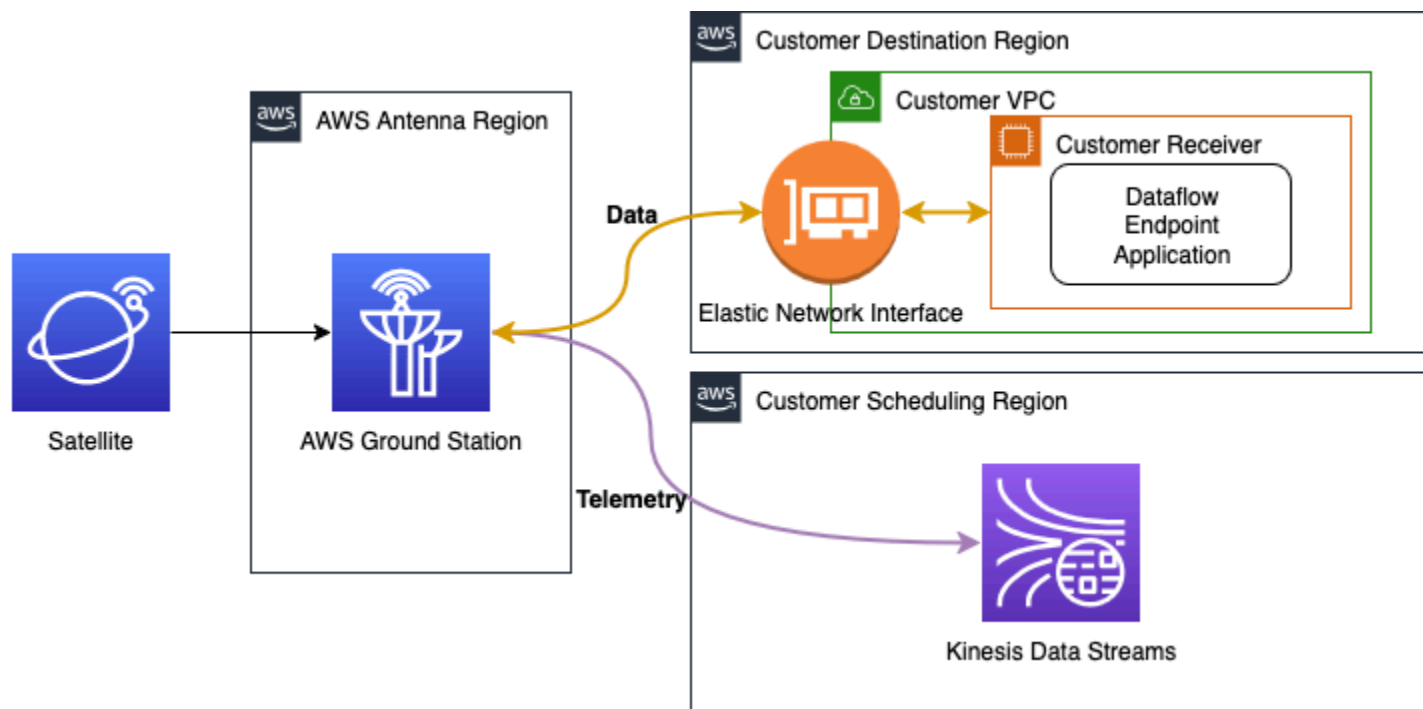
以下显示了使用 AWS Ground Station 代理配置时的通信路径。



规划您的遥测

AWS Ground Station 遥测是一项可选功能，可在卫星联系期间将指标从 AWS Ground Station 天线传输到您的 AWS 账户。这使您能够近乎实时地监控联系人表现并构建自定义监控解决方案。

使用 AWS Ground Station 遥测技术，来自 AWS Ground Station 天线的指标将直接传输到您的账户。遥测数据在接触开始时开始流式传输，并在整个接触期间持续传输。遥测数据是从天线硬件采样时近乎实时地传送到您的账户的。收到数据后，您可以使用自己的后处理软件或使用其他 AWS 服务（例如 Amazon Data Firehose 或）来处理数据。AWS Lambda



在下一步中，您将创建任务配置文件所需的配置。如果要启用遥测，除了跟踪配置和数据流配置外，还要创建遥测接收器配置。有关详细设置说明，请参阅[设置遥测](#)。

有关的更多信息 `TelemetrySinkConfig`，请参阅[遥测接收器 Config](#)。

创建配置

通过此步骤，您已经根据需要确定了卫星、通信路径以及 IAM、Amazon EC2 和 Amazon S3 资源。在此步骤中，您将创建存储各自参数的 AWS Ground Station 配置。

数据传输配置

要创建的第一个配置与您希望数据交付的地点和方式有关。使用上一步中的信息，您将构造以下许多配置类型。

- [亚马逊 S3 录音 Config](#)-将数据传输到您的 Amazon S3 存储桶。
- [数据流端点配置](#)-将数据传输到您的 Amazon EC2 实例。

遥测配置（可选）

如果您想在联系期间接收近乎实时的遥测数据，则可以创建一个 `TelemetrySinkConfig` 此配置是可选的，用于指定 AWS Ground Station 将遥测数据传送到何处。

- [遥测接收器 Config](#)-向您的账户提供遥测数据。

有关详细的设置说明，请参阅[设置遥测](#)。

卫星配置

卫星配置与 AWS Ground Station 如何与您的卫星通信有关。您将引用您在其中收集的信息[机载卫星](#)。

- [跟踪配置](#)-设置在接触期间如何对车辆进行物理跟踪的偏好。这是构建任务概况所必需的。
- [天线下行传输配置](#)-提供数字化的射频数据。
- [天线下行传输解调解码配置](#)-提供解调和解码后的射频数据。
- [天线上行传输配置](#)-将数据上行链接到您的卫星。
- [天线上行传输回波配置](#)-传送上行链路信号数据的回声。

创建任务档案

通过上一步中构建的配置，您已经确定了如何跟踪卫星、与卫星通信的可能方式以及在执行接触时如何启用近乎实时的遥测。在此步骤中，您将构建一个或多个任务配置文件。任务配置文件表示将可能的配置汇总为预期行为，然后可以对其进行计划 and 操作。

有关最新参数，请参考 [AWS::GroundStation::MissionProfile CloudFormation 资源类型](#)

1. 命名你的任务简介。这使您可以快速了解其在系统中的使用情况。例如，satellite-narrowband-emergency-operations如果有单独的窄带载波用于satellite-wideband-narrowband-nominal紧急操作，则可能有- operations 和 a。
2. 设置您的跟踪配置。
3. 设置最小可行接触时长。这使您可以筛选潜在的联系人以满足您的任务需求。
4. 设置streamsKmsKey用于streamsKmsRole在传输过程中对数据进行加密的 and。这用于所有 AWS Ground Station 代理数据流。
5. 设置您的数据流。使用您在上一步中创建的配置创建数据流以匹配您的载波信号。
6. [可选] 设置传球前和通过后的接触持续时间（秒）。这用于分别在联系人之前和之后发出每个联系人的事件。请参阅[自动化 AWS Ground Station 含活动](#)了解更多信息。
7. [可选] 将 telemetrySinkConfigArn 设置为在联系期间启用遥测。这使您可以直接在账户中接收近乎实时的遥测数据，以进行监控和分析。请参阅[使用遥测](#)了解更多信息。

8. [可选] 您可以将标签与任务配置文件相关联。这些可以用来帮助以编程方式区分您的任务配置文件。

您可以参考[任务配置文件配置示例](#)，仅查看一些潜在的配置。

了解后续步骤

既然您已经拥有了机载卫星和有效的任务配置文件，就可以安排联系人并与卫星通信了。AWS Ground Station

您可以通过以下方式之一安排联系：

- 控制[AWS Ground Station 台](#)。
- AWS CLI 的 `reserve-contact` 命令。
- AWS 软件开发工具包。 [ReserveContactAPI](#)。

有关如何 AWS Ground Station 跟踪卫星轨迹以及如何使用这些信息的信息，请参阅[明白怎么做 AWS Ground Station 使用星历](#)。

AWS Ground Station 维护了许多预配置的 CloudFormation 模板，以便更轻松地开始使用该服务。[任务配置文件配置示例](#)有关 AWS Ground Station 如何使用的示例，请参阅。

处理数字中频数据或从中提供给您解调和解码数据 AWS Ground Station 将取决于您的具体用例。以下博客文章可以帮助您了解一些可用的选项：

- [使用 AWS Ground Station Amazon S3 数据传输进行自动地球观测](#) (及其关联 GitHub 存储库 [awslabs/ aws-groundstation-eos-pipeline](#))
- [使用虚拟化卫星地面段 AWS](#)
- [地球观测使用 AWS Ground Station：如何指导](#)
- [使用 d@@ igiF AWS Ground Station WideBand 和 Amphinicy Blink SDR \(以及相关的存储库 \[aws-samples/\]\(#\) \) 构建高吞吐量的卫星数据下行链路架构 GitHub \[aws-groundstation-wbdigif-snpp\]\(#\)](#)

AWS Ground Station 位置

AWS Ground Station 提供全球地面站网络，紧邻我们的全球AWS基础设施区域网络。您可以从任何支持的 AWS 区域配置对这些位置的使用。这包括传输数据的 AWS 区域。



查找 AWS 地面站位置的区域

AWS Ground Station 全球网络包括物理上不位于其所连接的 [AWS 区域](#)的地面站位置。您可以通过 AWS 开发工具包 [ListGroundStation](#)响应检索您有权访问的地面站列表。自2026年6月16日起，AWS Ground Station 不再支持首尔的天线资源。地面站位置的完整清单如下所示。请参阅入门指南，为您的卫星添加或修改场地批准。

地面站名称	地面站位置	AWS 区域名称	AWS 区域代码	注意
阿拉斯加 1	阿拉斯加，美国	美国西部（俄勒冈州）	us-west-2	实际位置不在某个 AWS 区域
巴林 1	巴林	中东（巴林）	me-south-1	
开普敦 1	开普敦，南非	非洲（开普敦）	af-south-1	

地面站名称	地面站位置	AWS 区域名称	AWS 区域代码	注意
达博 1	澳大利亚达博	亚太地区（悉尼）	ap-southeast-2	实际位置不在某个 AWS 区域
夏威夷 1	美国夏威夷	美国西部（俄勒冈州）	us-west-2	实际位置不在某个 AWS 区域
爱尔兰 1	爱尔兰	欧洲地区（爱尔兰）	eu-west-1	
俄亥俄州 1	美国俄亥俄州	美国东部（俄亥俄州）	us-east-2	
俄勒冈 1	美国俄勒冈州	美国西部（俄勒冈州）	us-west-2	
蓬塔阿雷纳斯 1	智利蓬塔阿雷纳斯	南美洲（圣保罗）	sa-east-1	实际位置不在某个 AWS 区域
新加坡 1	新加坡	亚太地区（新加坡）	ap-southeast-1	
斯德哥尔摩 1	瑞典斯德哥尔摩	欧洲地区（斯德哥尔摩）	eu-north-1	

AWS Ground Station 支持的 AWS 区域

您可以通过 AWS 开发工具包或支持的 AWS 区域的 AWS Ground Station 控制台传送数据和配置联系人。您可以在终端节点和配额处查看支持的区域及其关联[AWS Ground Station 终端节点](#)。

数字双胞胎的可用性

[使用AWS Ground Station数字双胞胎功能](#)在所有可用的[AWS 区域](#)中可 AWS Ground Station 用。数字双胞胎地面站是生产地面站的精确副本，地面站名称的修改前缀为“数字双胞胎”。例如，“俄亥俄州1号数字双胞胎地面站”是一个数字双胞胎地面站，它是“俄亥俄1号”生产地面站的精确副本。

专用天线

除了上面列出的公开地面站位置外，还 AWS Ground Station 提供专用天线。专用天线是一种定制的天线系统，可以代您进行 AWS 管理。专用天线不限于现有的地 AWS Ground Station 面站位置，可使用公共地面站以外的功能建造，如中所述[AWS Ground Station 网站能力](#)。专用天线的位置和功能未公开披露。

有关专用天线的更多信息，请参阅[AWS Ground Station 专用天线](#)。要了解更多信息或开始使用专用天线，请 AWS 支持 通过以下[AWS Support Center Console](#)方式联系。

在地面站查看天线

每个地面站位置都有一根或多根天线。您可以使用 [ListAntennas](#) API 查看地面站的天线。此 API 返回指定地面站的天线，包括每个天线的名称。

天线信息与 [ListGroundStationReservations](#) API 结合使用时非常有用，可以了解地面站的容量和可用性。有关查看预订的更多信息，请参阅[查看地面站预订](#)。

要拨打电话ListAntennas，您必须将卫星载入地面站，或者拥有地面站的方位角仰角星历表权限。有关更多信息，请参阅 [提供方位角高程星历数据](#)。

示例：列出地面站的天线

以下示例列出了使用 Python AWS 软件开发工具包 (Boto3) 的地面站的所有天线。

```
import boto3

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

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

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

paginator = ground_station_client.get_paginator("list_antennas")
page_iterator = paginator.paginate(
    groundStationId=ground_station_id,
```

```
PaginationConfig={
    "MaxItems": 100,
    "PageSize": 20,
},
)

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

查看地面站预订

您可以使用 API 查看地面站天线上的预留情况。[ListGroundStationReservations](#) 预约代表天线上的时间段，包括您的预定联系人。[AWS Ground Station 专用天线](#) 客户还会看到维护窗口。

这些信息可以帮助您在计划联系时间表时了解天线的可用性，并提供地面站天线上正在发生的事情的可见性。

房源预订

要列出预订，请致电并[ListGroundStationReservations](#) 提供地面站标识符和时间范围。API 会返回指定时间窗口内地面站所有天线的预留量。

您看到的预订取决于您的访问级别：

- 公共 AWS Ground Station 客户-您只能看到自己的联系人预订。不包括其他账户拥有的维护窗口和联系人。
- AWS Ground Station 专属天线客户 — 您可以在专用天线上查看所有预订，包括维护时段和其他账户安排的联系方式。仅包含您拥有的联系人的联系人标识符。有关更多信息，请参阅 [AWS Ground Station 专用天线](#)。

预订类型

每个预留都有一种类型，用于指示天线时间的用途：

- 联系人-联系人预留代表为卫星通信保留的天线时间。预约开始和结束时间反映了完整的天线预留时间，包括通过前和通过后的时间，而不仅仅是卫星通过窗口。

- 维护-维护预留代表天线因维护而不可用的时间段。维护预留包括maintenanceType表明维护是计划内还是计划外的。

代码示例

以下示例列出了未来 7 天使用适用于 Python 的 AWS SDK (Boto3) 在地面站进行的预订，包括按预留类型进行筛选。

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

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

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

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

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

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

for page in page_iterator:
    for reservation in page["reservationList"]:
        reservation_type = reservation["reservationType"]
```

```

    antenna_name = reservation["antennaName"]
    res_start = reservation["startTime"]
    res_end = reservation["endTime"]

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

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

    print()

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

page_iterator = paginator.paginate(
    groundStationId=ground_station_id,
    startTime=start_time,
    endTime=end_time,
    reservationTypes=["MAINTENANCE"],
    PaginationConfig={
        "MaxItems": 100,
        "PageSize": 20,
    },
)

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

```

AWS Ground Station 网站掩码

每个 AWS Ground Station [天线位置](#) 都有相关的站点掩码。当指向某些方向（通常靠近地平线）时，这些掩码会阻止天线在该位置的发射或接收信号。掩码可以考虑：

- 天线周围的地理地形特征 — 例如，这包括山脉或建筑物之类的东西，它们会阻挡射频 (RF) 信号或阻止传输。
- 射频干扰 (RFI) — 这会影响接收（影响到 AWS Ground Station 天线的下行链路信号的外部 RFI 源）和传输（AWS Ground Station 天线传输的射频信号，对外部接收器产生不利影响）的能力。
- 法律授权 — 在每个地区运营 AWS Ground Station 的本地站点授权可能包括特定限制，例如传输的最小仰角。

这些站点掩码可能会随时间而变化。例如，可以在天线位置附近建造新建筑，RFI 来源可能会发生变化，或者可能会根据不同的限制续订法律授权。根据保密协议 (NDA)，您可以获得 AWS Ground Station 网站掩码。

客户专用口罩

除了每个站点的 AWS Ground Station 站点掩码外，您可能还会有额外的口罩，因为您自己的法律授权受到限制，无法与特定区域的卫星通信。可以在 AWS Ground Station 中配置此类掩码，以确保使用 AWS Ground Station 与这些卫星通信时的合规性。case-by-case 请联系 AWS Ground Station 团队了解详情。

现场口罩对可用联系时间的影响

有两种站点掩码：上行链路（传输）站点掩码和下行链路（接收）站点掩码。

使用该 ListContacts 操作列出可用的联系时间时，AWS Ground Station 将根据您的卫星何时升至下行链路掩码下方并设置在下行屏蔽下方返回能见时间。可用的联系时间基于此下行链路掩码可见性窗口。这样可以确保当您的卫星低于下行链路掩码时，您不会预留时间。

即使任务配置文件在数据流边缘包含[天线上行链路配置](#)，上行链接站点掩码也不会应用于可用的联络时间。这使您可以将所有可用的联系时间用于下行链路，即使由于上行链路站点掩码的原因，上行链路可能在部分时间内不可用。但是，在为卫星联络预留的部分或全部时间内，可能无法传输上行链路信号。在安排上行链路传输时，您有责任考虑所提供的上行链路掩码。

相对于天线位置的上行链路站点掩码，不可用于上行链路的部分联络会因联络期间卫星轨迹的不同而有所不同。在上行链路和下行链路站点掩码相似的区域，此持续时间通常会很短。在其他区域，上行链路

掩码可能明显高于下行链路站点掩码，这可能会导致上行链路在很大部分甚至全部联络期间无法使用。即使部分预留时间不可用于上行链路，也会向您收取全部联系时间。

AWS Ground Station 网站能力

为了简化您的体验，请为一种天线类型 AWS Ground Station 确定一组常用功能，然后将多根天线部署到地面站位置。部分入门步骤可确保您的卫星与特定位置的天线类型兼容。当您预订联系人时，您可以间接确定所使用的天线类型。这样可以确保无论使用哪种天线，您在特定地面站位置的体验都会随着时间的推移保持不变。由于各种环境问题（例如现场的天气），您的联系人的具体表现会有所不同。

目前，所有站点都支持以下功能：

 Note

除非另有说明，否则下表中的每一行都表示独立的通信路径。存在重复行是为了反映我们的多渠道功能，允许同时使用多条通信路径。

能力类型	频率范围	带宽范围	Polarization	公用名	备注
天线下行链路	7750-8500 MHz	50-400 MHz	RHCP	X 波段宽带下行链路	此功能需要使用代 AWS Ground Station 理。
天线下行链路	7750-8500 MHz	50-400 MHz	RHCP		
天线下行链路	7750-8500 MHz	50-400 MHz	RHCP		
天线下行链路	7750-8500 MHz	50-400 MHz	RHCP		阿拉斯加 1 或蓬塔阿雷纳斯 1 不支持此功能。
天线下行链路	7750-8500 MHz	50-400 MHz	RHCP		每个位置的 MHz 每次极化总带宽不得超过 400。
天线下行链路	7750-8500 MHz	50-400 MHz	LHCP		所有使用的频率范围必须不重叠。

能力类型	频率范围	带宽范围	Polarization	公用名	备注
天线下行链路	7750-8500 MHz	50-400 MHz	LHCP		
天线下行链路	7750-8500 MHz	50-400 MHz	LHCP		
天线下行链路	7750-8500 MHz	50-400 MHz	LHCP		
天线下行链路	7750-8500 MHz	50-400 MHz	LHCP		
天线下行链路	2200-2290 MHz	最多 40 MHz	RHCP	S 波段下行链路	一次只能使用一种极化
天线下行链路	2200-2290 MHz	最多 40 MHz	LHCP		
天线下行链路	7750-8500 MHz	最多 40 MHz	RHCP	X 波段窄带下行链路	一次只能使用一种极化
天线下行链路	7750-8500 MHz	最多 40 MHz	LHCP		
天线上行链路	2025-2110 MHz	最多 40 MHz	RHCP	S 波段上行链路	一次只能使用一种极化
天线上行链路	2025-2110 MHz	最多 40 MHz	LHCP		EIRP 20-50 dBW
antenna-uplink-echo	2025-2110 MHz	2 MHz	RHCP	上行链路回声	匹配天线上行链路限制
antenna-uplink-echo	2025-2110 MHz	2 MHz	LHCP		

能力类型	频率范围	带宽范围	Polarization	公用名	备注
antenna-downlink-demod-decode	7750-8500 MHz	最多 500 MHz	RHCP	X 波段解调和解码下行链路	
antenna-downlink-demod-decode	7750-8500 MHz	最多 500 MHz	LHCP		
跟踪	不适用	不适用	不适用	不适用	Support 支持自动跟踪和节目跟踪

* RHCP = 右手圆极化，LHCP = 左手圆极化。有关极化的更多信息，请参见[圆极化](#)。

明白怎么做 AWS Ground Station 使用星历

一个[星历](#)，或多个星历，是一种提供天体轨迹的文件或数据结构。从历史上看，该文件仅涉及表格数据，但逐渐演变成为指示航天器轨迹的各种数据文件。

星历表 API 允许将自定义星历上传到以用于卫星。AWS Ground Station 这些星历表会覆盖来自 (参见:) 的默认星历。[Space-Track 默认星历数据](#) AWS Ground Station 支持轨道星历信息 (OEM)、两行元素 (TLE) 和方位角仰角格式的星历数据。为获得最高的准确性和最灵活的时间跨度覆盖范围，请提供 OEM 格式的星历数据 (参见)。[提供 OEM 星历数据](#)

AWS Ground Station 使用星历数据根据提供的星历表确定联系人何时可用，并正确指挥网络中的天线。AWS Ground Station 默认情况下，如果您的卫星已分配了[NORA AWS Ground Station D ID](#)，则[无需执行任何操作即可提供星历。](#)

上传自定义星历表可以提高跟踪质量，处理没有星历[Space-Track](#)表可用的早期行动，并 AWS Ground Station 记录演习。

或者，AWS Ground Station 支持方位角仰角格式，该格式允许您在不提供卫星轨道信息的情况下直接指定天线指向方向。这对于由于卫星轨迹信息不精确或未知而需要精确天线指向的场景很有用。

主题

- [默认星历数据](#)
- [提供自定义星历数据](#)
- [使用自定义星历表保留联系人](#)
- [了解使用的是哪个星历](#)
- [获取卫星的当前星历](#)
- [恢复为默认星历数据](#)

默认星历数据

默认情况下，AWS Ground Station 使用来自的公开数据 [Space-Track](#)，无需执行任何操作即可 AWS Ground Station 提供这些默认星历。这些星历表是与卫星的 NORAD ID [相关的](#)两行元素集 (TLE)。https://en.wikipedia.org/wiki/Satellite_Catalog_Number所有默认星历表的优先级均为。0 因此，它们将始终被通过星历 API 上传的任何未过期的自定义星历表所覆盖，而星历表的优先级必须始终等于或更高。1

没有 NORAD ID 的卫星必须将自定义星历数据上传到。AWS Ground Station 例如，刚刚发射的卫星或故意从 [Space-Track](#) 目录中省略的卫星将没有 NORAD ID，需要上传自定义星历。有关提供自定义星历数据的更多信息，请参阅：[提供自定义星历数据](#)。

提供自定义星历数据

Important

星历 API 当前处于预览状态

星历 API 的访问权限仅在需要时提供。如果您需要能够上传自定义星历数据，请通过以下方式开 AWS 支持 票。[AWS Support Center Console](#) 我们的团队将与您合作，为您的特定要求启用此功能。

概述

星历表 API 允许将自定义星历上传到以用于卫星。AWS Ground Station 这些星历表会覆盖来自 (参见:) 的默认星历。[Space-Track](#) [默认星历数据](#) 我们支持接收轨道星历信息 (OEM)、两行元素 (TLE) 和方位角仰角格式的星历数据。

AWS Ground Station 将星历视为 [个性化使用数据](#)。如果您使用此可选功能，AWS 将使用您的星历数据来提供故障排除支持。

上传自定义星历表可以提高跟踪质量，处理没有星历 [Space-Track](#) 表可用的操作，并 AWS Ground Station 记录演习。

要对无效的星历表进行故障排除，请参阅：[对无效的星历进行故障排除](#)

示例：使用客户提供的星历表和 AWS Ground Station

有关使用客户提供的星历表的更多详细说明，请参阅将客户提供的星历与其关联存储库 AWS Ground Station aws--groundstation-cpe 一起 [AWS Ground Station](#) 使用。GitHub [samples/aws](#)

提供 TLE 星历数据

Important

星历 API 当前处于预览状态

星历 API 的访问权限仅在需要时提供。如果您需要能够上传自定义星历数据，请通过以下方式开 AWS 支持 票。[AWS Support Center Console](#)我们的团队将与您合作，为您的特定要求启用此功能。

概述

Two-line 元素 (TLE) 集是描述卫星轨道的标准格式。星历表API允许将TLE星历上传到以用于卫星。AWS Ground Station 这些星历表会覆盖来自 (参见:) 的默认星历。[Space-Track](#) [默认星历数据](#)

AWS Ground Station 将星历视为[个性化使用数据](#)。如果您使用此可选功能，AWS 将使用您的星历数据来提供故障排除支持。

上传自定义 TLE 星历表可以提高跟踪质量，处理没有星历[Space-Track](#)表可用的早期行动，并 AWS Ground Station记录演习。但是，TLE 精度受到传播模型的限制，并且会随着时间的推移而降低。要在更长的时间跨度内获得更高保真度的星历数据，请考虑使用 OEM 格式 (参见)。[提供 OEM 星历数据](#)

Note

在为卫星分配卫星目录号之前提供自定义星历表时，您可以使用TLE 00000 的卫星目录号字段以及 000 TLE国际标志字段的发射号部分 (例如 24000A 2024年发射的运载工具)。有关 TLE 格式的更多信息，请参阅[Two-line 元素集](#)。

Alpha-5 TLE 中的卫星目录号

Two-Line 元素集 (TLE) 使用固定宽度的格式，其中卫星目录号字段 (第 1 行和第 2 行的第 3—7 列) 限制为 5 位数，支持的最大值为 99,999。为了表示超出此限制的卫星目录号，美国太空部队开发了[Alpha-5](#)编码方案。Alpha-5 用字母字符 (A—Z，不包括 I 和 O，以避免与数字 1 和 0 混淆) 替换卫星目录编号字段的第一位数字。这使现有固定宽度的TLE格式中可表示的卫星目录号的范围从100,000 扩大到339,999。

Alpha-5 纯粹是 TLE 文本格式的显示编码。基础数字卫星目录号不变。AWS Ground Station API 中的noradSatelliteID字段仍然是整数——诸如ListSatellites和之类的 API 响应GetSatellite继续返回数字卫星目录号，而不是 Alpha-5 编码形式。Alpha-5 编码仅与 TLE 文本行本身相关。

AWS Ground Station 支持包含 Alpha-5 编码卫星编号的 TLE 星历。当您通过[CreateEphemeris](#)操作上传 TLE 时，请 AWS Ground Station 正确解析 Alpha-5 编码的卫星编号并在内部将其解析为相应的整数卫星目录号。完全支持卫星目录号在 (100,000—339,999) Alpha-5 范围内的卫星。

以下是卫星目录编号为 100,000 的卫星的 TLE 设置示例。Alpha-5 编码表示为 A0000 每行 TLE 的第 3—7 列：

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

有关 Alpha-5 编码方案的更多信息，请参阅[Space-Track Alpha-5文档](#)。

创建 TLE 星历表

可以使用 API 中的[CreateEphemeris](#)操作创建 TLE 星历。AWS Ground Station 此操作将使用请求正文或指定 S3 存储桶中的数据上传星历。

请务必注意，上传星历会将星历设置为 VALIDATING 并启动异步工作流程，该工作流程将验证您的星历并生成可能联络。只有当星历通过此工作流程并成为 ENABLED 后，才会将其用于联络。您应该轮[DescribeEphemeris](#)询星历状态或使用 CloudWatch 事件来跟踪星历表的状态变化。

要对无效的星历表进行故障排除，请参阅：[对无效的星历进行故障排除](#)

示例：通过 API 创建两行元素 (TLE) 集星历

AWS 软件开发工具包和 CLI 可用于通过调用上传两行元素 (TLE) 设置的星历表。AWS Ground Station [CreateEphemeris](#) 该星历表将用于代替卫星的默认星历数据（参见）。[默认星历数据](#)此示例说明如何使用[AWS 适用于 Python 的软件开发工具包 \(Boto3\)](#) 执行此操作。

TLE 集是 JSON 格式的对象，可将一个或多个 TLE 串在一起以构造连续轨迹。TLE 集中的 TLE 必须形成一个连续集，我们可以用它来构造轨迹（即 TLE 集中的 TLE 之间没有时间间隔）。下面显示了一个 TLE 集示例：

```
[
  {
```

```

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

```

Note

TLE 集中 TLE 的时间范围必须完全匹配才能成为有效的连续轨迹。

TLE 集可以通过 bot AWS Ground Station o3 客户端上传，如下所示：

```

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']}")

```

此调用将返回一个星历，该星历可以在将来用于引用星历。例如，我们可以使用上面电话中提供的 ephemerisID 来轮询星历表的状态：

```

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

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

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

tle_ephemeris = ground_station_client.create_ephemeris(
    name="Example TLE Ephemeris for Description",
    satelliteId="2e925701-9485-4644-b031-EXAMPLE01",
    enabled=True,
    expirationTime=datetime.now(timezone.utc) + timedelta(days=3),
    priority=2,
    ephemeris={
        "tle": {
            "tleData": [
                {
                    "tleLine1": "1 25994U 99068A   20318.54719794   .000000075   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']}")

```

下面提供了 [DescribeEphemeris](#) 操作的响应示例

```

{
  "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}}]"
    }
  }
}

```

建议轮询[DescribeEphemeris](#)路径或使用 CloudWatch 事件来跟踪上传的星历表的状态，因为该星历必须经过异步验证工作流程，然后才能将其设置为可用于安排ENABLED和执行联系人。

请注意，在上面的示例中，TLE 集合中所有 TLE 25994 中的 NORAD ID 必须与您的卫星在数据库中分配的 NORAD ID 相匹配。 [Space-Track](#)

示例：从 S3 存储桶上传 TLE 星历数据

也可以通过指向存储桶和对象密钥直接从 S3 存储桶上传 TLE 星历文件。AWS Ground Station 将代表你检索对象。有关静态数据加密的信息详 AWS Ground Station 见：[静态数据加密 AWS Ground Station](#)。

以下是从 S3 存储桶上传 TLE 星历文件的示例

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

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

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

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

```

        "tleLine2": "2 25994 98.2007 33.6589 0001234 89.2782 18.9934
14.57114995112342",
        "validTimeRange": {
            "startTime": datetime.now(timezone.utc) + timedelta(days=3),
            "endTime": datetime.now(timezone.utc) + timedelta(days=7),
        },
    },
]

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

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

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

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

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

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

```

提供 OEM 星历数据

Important

星历 API 当前处于预览状态

星历 API 的访问权限仅在需要时提供。如果您需要能够上传自定义星历数据，请通过以下方式开 AWS 支持 票。[AWS Support Center Console](#)我们的团队将与您合作，为您的特定要求启用此功能。

概述

轨道星历信息 (OEM) 是一种用于表示航天器轨迹数据的标准化格式。星历表 API 允许将 OEM 星历表上传到 AWS Ground Station 以用于卫星。这些星历表会覆盖来自 (参见:) 的默认星历。[Space-Track](#) [默认星历数据](#)

AWS Ground Station 将星历视为[个性化使用数据](#)。如果您使用此可选功能，AWS 将使用您的星历数据来提供故障排除支持。

上传自定义 OEM 星历表可以提高跟踪质量，在没有星历[Space-Track](#)表的情况下处理早期操作，并 AWS Ground Station考虑操作情况。

 Note

在为卫星分配卫星目录号之前提供自定义星历表时，您可以使用 OEM satelliteId 的OBJECT_ID部分。
有关 OEM 格式的更多信息，请参阅[OEM 星历格式](#)。

OEM 星历格式

AWS Ground Station 根据 [CCSDS 标准](#)处理 OEM 客户提供的星历，但有一些额外的限制。OEM 文件应采用 KVN 格式。下表概述了 OEM 中的不同字段以及 AWS Ground Station 与 CCSDS 标准的区别。

Section	字段	需要 CCSDS	AWS Ground Station 必填的	注意
标题	CCSDS_OEM _VERS	支持	是	所需值：2.0
	COMMENT	否	否	
	分类	否	否	
	创建日期	支持	是	

Section	字段	需要 CCSDS	AWS Ground Station 必填的	注意
元数据	创始人	支持	是	
	MESSAGE_ID	否	否	
	META_START	支持	是	
	COMMENT	否	否	
	对象名称	支持	是	
	OBJECT_ID	支持	是	
	中心名称	支持	是	所需值：地球
	REF_FRAME	支持	是	可接受的值： EME2000、ITRF 2000
	REF_FRAME_EPOCH	否	不支持*	不需要，因为接受的 ref_Frames 具有隐式纪元
	时间系统	支持	是	必填值：UTC
	START_TIME	支持	是	
	可用开始时间	否	否	
	可用停止时间	否	否	
	STOP_TIME	支持	是	
	插值	否	是	必需，这样 AWS Ground Station 才能为触点生成准确的指向角度。

Section	字段	需要 CCSDS	AWS Ground Station 必填的	注意
	插值_度	否	是	必需，这样 AWS Ground Station 才能为触点生成准确的指向角度。如果可能，将使用指定的度数，但如果分段中没有足够的数据，则将使用较低的度数。
	META_STOP	支持	是	
数据	X	支持	是	代表于 km
	Y	支持	是	代表于 km
	Z	支持	是	代表于 km
	X_DOT	支持	是	代表于 km/s
	Y_DOT	支持	是	代表于 km/s
	Z_DOT	支持	是	代表于 km/s
	X_DDOT	否	否	代表于 km/s^2
	Y_DDOT	否	否	代表于 km/s^2
	Z_DDOT	否	否	代表于 km/s^2
协方差矩阵	协方差_开始	否	否	
	EPOCH	否	否	
	COV_REF_F RAME	否	否	

Section	字段	需要 CCSDS	AWS Ground Station 必填的	注意
	协方差_停止	否	否	

* 如果提供的 OEM 中包含任何 AWS Ground Station 不支持的行，则 OEM 将无法通过验证。

与CCSDS标准的重要差异是：AWS Ground Station

- CCSDS_OEM_VERS必须是2.0。
- REF_FRAME必须为EME2000或 ITRF2000。
- REF_FRAME_EPOCH不支持 AWS Ground Station。
- CENTER_NAME必须是Earth。
- TIME_SYSTEM必须是UTC。
- INTERPOLATION并且INTERPOLATION_DEGREE都是 AWS Ground Station 客户提供的星历表所必需的。
- AWS Ground Station 与 CCSDS 5.2.4.7 不同，它允许不包含足够星历数据记录的 OEM 数据块在指定位置进行插值。INTERPOLATION_DEGREE在这种情况下，AWS Ground Station 将尽可能使用小于或等于指定值的最大插值度。INTERPOLATION_DEGREE

KVN 格式的 OEM 星历表示例

以下是公共广播卫星的 KVN 格式的 OEM 星历表的截断示例。JPSS-1

```
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

```

创建 OEM 星历表

可以使用 API 中的 [CreateEphemeris](#) 操作创建 OEM 星历。AWS Ground Station 此操作将使用请求正文或指定 S3 存储桶中的数据上传星历。

请务必注意，上传星历会将星历设置为 `VALIDATING` 并启动异步工作流程，该工作流程将验证您的星历并生成可能联络。只有当星历通过此工作流程并成为 `ENABLED` 后，才会将其用于联络。你应该轮 [DescribeEphemeris](#) 询星历状态或使用 CloudWatch 事件来跟踪星历的状态变化。

要对无效的星历表进行故障排除，请参阅：[对无效的星历进行故障排除](#)

示例：从 S3 存储桶上传 OEM 星历数据

也可以通过指向存储桶和对象密钥直接从 S3 存储桶上传 OEM 星历文件。AWS Ground Station 将代表你检索对象。有关静态数据加密的信息详 AWS Ground Station 见：[静态数据加密 AWS Ground Station](#)。

以下是从 S3 存储桶上传 OEM 星历文件的示例

```
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']}")

```

以下是为上一个示例代码块中上传的 OEM 星历调用 [DescribeEphemeris](#) 操作返回的数据示例。

```

{
  "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"
    }
  }
}
```

提供方位角高程星历数据

Important

方位角仰角星历表功能目前处于预览状态，需要明确加入。

对于数量有限的预先确定的专业用例，方位角仰角星历功能受到严格的访问控制。访问权限比客户提供的标准星历功能严格得多。有关批准的用例和访问请求流程的更多信息，请通过以下方式开 AWS 支持 票[AWS Support Center Console](#)。我们的团队将指导您完成特殊用例的批准流程。

概述

方位角仰角星历表提供了一种无需提供卫星轨道信息即可直接指定天线指向方向的方法。与其上传描述卫星轨道的星历表数据，不如提供带有时间标签的方位角和仰角来告诉天线在整个接触过程中确切指向何处。

AWS Ground Station 将星历视为[个性化使用数据](#)。如果您使用此可选功能，AWS 将使用您的星历数据来提供故障排除支持。

这种方法对以下场景特别有用：

- 早期运营支持：在发射和早期轨道阶段（LEOP）期间，当无法获得精确的轨道数据或轨道参数迅速变化时。
- 自定义指向模式：为天线测试或非标准操作实现特定的指向序列。

Note

使用方位角仰角星历表时，联系人预订请求中可能会省略卫星 ARN。如果不省略卫星 ARN，它仍将作为联系数据的一部分包括在内，但方位角仰角星历表将用于天线指向，而不是执行星历优先分辨率。方位角仰角星历表与特定的地面站相关联，用于定义该位置的天线指向方向。

方位角高程星历表数据格式

方位角高程星历表数据由按分段组织的时间标记方位角和高程值组成。每个分段都包含一系列覆盖特定时间范围的方位角和仰角。

方位角高程星历表数据的关键组成部分是：

- 地面站：将使用该方位角仰角星历表的特定地面站。
- 角度单位：角度的计量单位（DEGREE_ANGLE或RADIAN）。
- 分段：一个或多个有时间限制的方位角和仰角集合。
- Time-tagged 角度：带有相关时间戳的单个方位角和仰角值。

每个分段都需要：

- 参考时期（分段的基准时间）
- 有效的时间范围（分段的开始和结束时间）
- 至少 5 对带有时间标签 azimuth/elevation 的配对

方位角高程约束：

- 以度为单位的方位角：-180° 至 360°
- 以弧度表示的方位角：-p 到 2p
- 以度为单位的仰角：-90° 至 90°
- 以弧度为单位的海拔：- $\omega/2$ 至 $\omega/2$
- 每个分段内的时间值必须按升序排列
- 区段在时间上不得重叠

有关更多信息，请参阅 [CreateEphemeris](#) API 文档和 [TimeAzEl](#) 数据类型。

创建方位角仰角星历

方位角高程星历表使用相同的 [CreateEphemeris](#) API 操作创建，但使用星历类型。azEl与 TLE 和 OEM 星历表的主要区别在于：

- 必须指定一个groundStation参数
- 请求中必须省略该satelliteId参数
- 优先级设置不适用（每个方位角仰角星历表都特定于地面站）
- 每个分段必须包含至少 5 个 azimuth/elevation 点才能支持 4 阶拉格朗日插值
- [CreateEphemeris](#)API 文档中详细介绍了其他限制和要求

请务必注意，上传星历会将星历设置为 VALIDATING 并启动异步工作流程，该工作流程将验证您的星历并生成可能联络。星历表仅在通过此工作流程且状态变为后才用于联系人。ENABLED您应该轮[DescribeEphemeris](#)询星历状态或使用 CloudWatch 事件来跟踪星历表的状态变化。

要对无效的星历表进行故障排除，请参阅：[对无效的星历进行故障排除](#)

示例：通过 API 创建方位角仰角星历

以下示例说明如何使用适用于 Python 的 AWS 软件开发工具包 (Boto3) 创建方位角仰角星历：

```
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']}")

```

在本示例中：

- 方位角高程数据与“俄亥俄州 1”地面站相关联
- 角度以度为单位指定
- 该片段涵盖了 15 分钟的时段
- 这些dt值是相对于参考周期的原子秒偏移量
- 提供六 azimuth/elevation 对（最少 5 对）

示例：从 S3 上传方位角高程数据

对于较大的数据集，您可以从 S3 存储桶上传方位角高程数据：

```

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 对象应包含 JSON 结构，其方位角高程数据的格式与直接上传示例中所示的格式相同。

使用方位角仰角星历表保留联系人

使用方位角仰角星历表预订联系人时，该过程与 TLE 和 OEM 星历表不同：

1. 使用创建方位角高程星历 [CreateEphemeris](#)
2. 等待星历达到状态 ENABLED
3. 使用追踪[ReserveContact](#)功能预留联系人

使用方位角仰角星历表预约接触的示例：

```

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

保留方位角仰角星历表的接触时可以省略该satelliteArn参数。在接触过程中，天线将遵循指定的方位角和仰角。

列出可用联系人

使用方位角仰角星历表时，API 需要特定的参数：[ListContacts](#)

- 该satelliteArn参数可以从请求中省略
- 必须提供具有方位角仰角星历 ID 的ephemeris参数，以指定要使用哪个星历
- 当提供的方位角和仰角高于所请求地面[站的场地掩模](#)时，可用的接触窗口将显示出来
- 您仍必须提供groundStation和 missionProfileArn

创建方位角仰角星历表并列出可用联系人的示例：

```

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",
                            },
                        },
                        {
                            "referenceEpoch": "2024-04-01T12:30:00Z",
                            "validTimeRange": {
                                "startTime": "2024-04-01T12:30:00Z",
                                "endTime": "2024-04-01T13:00:00Z",
                            },
                        },
                    ],
                },
            },
        },
    },
)

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

列出联系人时必须提供具有方位角仰角 ID 的ephemeris参数，以指定应使用哪个方位角仰角星历表来确定接触窗口。如果包括satelliteArn在内，它将与联系数据相关联，但方位角仰角星历表将用于天线指向，而不是执行星历优先分辨率。

使用自定义星历表保留联系人

概述

使用自定义星历表（TLE、OEM 或方位角仰角）时，您可以使用 API 预订联系人。[ReserveContact](#) 本节介绍预约联系人的两种常见工作流程以及确保成功安排联系的重要注意事项。

AWS Ground Station 天线是多个客户之间的共享资源。这意味着，即使在您列出联系人时出现联系人窗口，其他客户也可能会在您列出联系人之前将其预订。因此，验证您的联系人是否在预订后到达该SCHEDULED州，并对联系状态的变化进行适当的监控至关重要。

Important

对于方位角仰角星历，ReserveContact请求中可以省略该satelliteArn参数，并且必须提供星历 ID。trackingOverrides对于 TLE 和 OEM 星历表，您仍需要提供 satelliteArn

联系人预订工作流程

使用自定义星历表预订联系人有两种主要工作流程：

1. List-then-reserve 工作流程：首先使用列出可用的联系人窗口 [ListContacts](#)，然后选择并保留特定窗口。当你能在做出选择之前查看所有可用机会时，这种方法很有用。
2. 直接预订工作流程：直接在特定时间段内预订联系人，无需先列出可用的联系人。当您已经知道所需的联系时间或正在按照预先确定的时间表进行工作时，这种方法很有用。

两种工作流程都是有效的，选择取决于您的操作要求。以下各节提供了每种方法的示例。

工作流程 1：列出可用联系人然后预订

此工作流程首先查询可用的联系人窗口，然后保留特定窗口。当你能在做出选择之前查看所有可用机会时，这很有用。

示例：使用方位角仰角星历表列出并保留

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

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

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

```

        }
    },
}
},
)

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

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

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

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

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

```

```

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

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

```

示例：使用 TLE 星历表列出并预订

```

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")

```

工作流程 2：直接联系预订

此工作流程直接保留联系人，无需先列出可用窗口。当您已经知道所需的联系时间或正在实施自动排程时，这种方法很有用。

示例：使用方位角仰角星历表直接预订

```

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']}")

```

示例：使用 TLE 星历表直接预订

```

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']}")

```

监控联系人状态的变化

预订联系人后，监控其状态以确保其成功过渡到联系人SCHEDULED并收到有关任何问题的通知非常重要。AWS Ground Station 向亚马逊 EventBridge 发送所有联系状态变更的事件。

联系州遵循以下生命周期：

- SCHEDULING-正在处理联系人以进行日程安排
- SCHEDULED-已成功安排联系人并将执行
- FAILED_TO_SCHEDULE-无法安排联系人（终端状态）

有关联系人状态和生命周期的更多信息，请参阅[了解联系生命周期](#)。

使用实现接触状态监控 EventBridge

要实时监控联系状态的变化，您可以设置一条亚马逊 EventBridge 规则，每当地面站联系人更改状态时，该规则就会触发 Lambda 函数。这种方法比轮询联系人状态更有效和更具可扩展性。

实现步骤

1. 创建 Lambda 函数来处理联系人状态变更事件
2. 创建与地面站联系状态变更事件相匹配的 EventBridge 规则
3. 添加 Lambda 函数作为规则的目标

示例 Lambda 函数处理器

有关处理联系人状态变更事件的 Lambda 函数的完整示例，请参阅[AquaSnppJpssTerraDigIF.yml](#) CloudFormation 模板中的GroundStationCloudWatchEventHandlerLambda资源。该模板可在 AWS Ground Station 客户加入 Amazon S3 存储桶中找到。有关访问此模板的说明，请参阅 dataflow 端点示例的[把它放在一起](#)部分。

EventBridge 规则配置

该 EventBridge 规则应使用以下事件模式来匹配所有地面站接触状态的变化：

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

要仅筛选特定状态（例如故障），可以添加详细信息过滤器：

```
{
```

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

有关使用 Lambda 目标创建 EventBridge 规则的详细说明，请参阅《亚马逊 EventBridge 用户指南》[中的](#)创建对事件做出反应的规则。

设置自动化 EventBridge 规则

您可以创建 EventBridge 规则以自动响应联系状态的变化。例如：

- 当联系人未能安排时间时发送通知
- 当联系人进入时，触发 Lambda 函数以准备资源 PREPASS
- 记录联系人完成情况以供审计

有关设置 AWS Ground Station 事件 EventBridge 规则的详细信息，请参阅[自动化 AWS Ground Station 含活动](#)。

最佳实践和注意事项

处理日程安排冲突

AWS Ground Station 由于天线是共享资源，因此在您预订之前，其他客户 ListContacts 可能会预留一个显示在中可用的联系窗口。要处理这个问题：

1. 预订后务必检查联系状态
2. 使用备用时间窗口实现重试逻辑
3. 尽可能考虑提前预约联系方式
4. 使用 EventBridge 事件监视 FAILED_TO_SCHEDULE 状态

星历验证时间

请记住，星历表必须处于ENABLED状态，然后才能使用它来保留联系人。验证过程通常需要几秒钟到几分钟，具体取决于星历的类型和大小。在尝试保留联系人之前，请务必验证星历状态。

联系时间注意事项

使用自定义星历表时：

- 确保您的星历表涵盖整个联系时间
- 对于方位角仰角星历，验证角度在整个接触过程中天线是否保持在现场掩模上方 <https://docs.aws.amazon.com/ground-station/latest/ug/locations.site-masks.html>
- 在安排未来的联系时考虑星历的到期时间

星历表类型的 API 差异

根据星历表的类型，ReserveContactAPI 的行为会有所不同：

星历表类型	SatelliteEarn 必填项	需要追踪覆盖
TLE	是	否（可选）
OEM	是	否（可选）
方位角仰角	否（可选）	是

了解使用的是哪个星历

星历具有优先级、到期时间和启用标记。它们共同决定了在接触期间使用哪些星历表进行跟踪。

TLE 和 OEM 星历

对于 OEM 和 TLE 星历表，每颗卫星只能有一个星历表处于活动状态。将要使用的星历是已启用的具有最高优先级的星历，其到期时间在未来。优先级值越大表示优先级越高。返回的可用联系时间 [ListContacts](#) 基于此星历。如果多个 ENABLED 的星历具有相同的优先级，则将使用最近创建或更新的星历。

Note

AWS Ground Station 对ENABLED客户提供的每颗卫星的星历表数量设定了服务配额（参见：[服务配额](#)）。要在达到此配额后上传星历数据，请删除（使用 [DeleteEphemeris](#)）或禁用（使用 [UpdateEphemeris](#)）客户提供的priority/earliest 创建量最低的星历表。

如果尚未创建星历表，或者没有星历表具有ENABLED状态，则 AWS Ground Station 将使用卫星（来自）的默认星历表（如果有）。[Space-Track](#) 此默认星历的优先级为 0。

方位角仰角星历

方位角仰角星历与 OEM 和 TLE 星历表的工作方式不同。每个方位角仰角星历表都与特定的地面站相关联，没有优先级。当您使用方位角仰角星历表预留接触时，您可以通过参数明确指定要使用哪个方位角高程星历表。`trackingOverrides`

方位角仰角星历表的主要区别：

- 没有优先系统-您可以为每个联系人明确选择星历
- 特定于地面站-每个星历表都与特定的地面站相关联
- 无自动回退-如果指定的星历表不可用，联系将失败

Note

方位角高度星历不与 OEM 和 TLE 星历表竞争。它们是在预订联系人时明确选择的，并且仅在指定跟踪覆盖时使用。

新的星历表对先前安排的联系人的影响

使用 [DescribeContact API 通过返回活跃可见时间](#) 来查看新星历对先前安排的联系人的影响。

对于 OEM 和 TLE 星历表，在上传新星历表之前安排的联系人将保留最初预定的联系时间，而天线跟踪将使用活动星历。如果基于活跃星历表的航天器的位置与先前的星历表有很大差异，则由于航天器在现场掩码之外运行，这可能会导致卫星与天线的接触时间缩短。`transmit/receive` 因此，我们建议您在上传与之前的星历表有很大不同的新星历表后取消并重新安排将来的联系时间。

使用 [DescribeContact API](#)，您可以通过比较您的预定联系startTime和endTime返回的联系来确定由于航天器在 transmit/receive 现场掩码之外运

行visibilityStartTime而visibilityEndTime无法使用的未来联系部分。如果您选择取消并重新安排将来的联系，则联系时间范围不得超过可见时间范围 30 秒。如果在距离联系时间过近时取消联系人，则取消的联系可能会产生费用。有关取消联络的更多信息，请参阅：[Ground Station 常见问题解答](#)。

对于方位角仰角星历，预定联系人将使用保留联系人时选择的特定星历。如果您需要更新预定联系人的方位角高程数据，则可以取消联系并使用新的星历表重新安排接触。

获取卫星的当前星历

可以通过调用或操作来检索 AWS Ground Station 用于特定卫星的当前星历。[GetSatellite](#) [ListSatellites](#) 这两种方法都将返回当前使用的星历的元数据。此星历元数据对于上传到的自定义星历表和默认星历表是不同的。AWS Ground Station

Note

方位角仰角星历表与卫星无关，因此不由或返回。[GetSatellite](#) [ListSatellites](#) 要检索有关方位角高程星历表的信息，请使用带有特定星历 ID [DescribeEphemeris](#) 的 API，或用于查看您的账户的所有可用星历。[ListEphemerides](#)

默认星历将仅包含 source 和 epoch 字段。epoch这是从[两行元素集中提取出来](#)的[时代](#) [Space-Track](#)，它目前用于计算卫星的轨迹。

自定义星历的 CUSTOMER_PROVIDED 为 source 值，且在 ephemerisId 字段中包含唯一标识符。此唯一标识符可用于通过 [DescribeEphemeris](#) 操作查询星历。如果在 AWS Ground Station 通过操作上传到星历时为星历分配了名称，则将返回一个可选name字段。[CreateEphemeris](#)

请务必注意，星历表由动态更新，AWS Ground Station 因此返回的数据只是调用 API 时使用的星历表的快照。

示例 [GetSatellite](#) 使用默认星历表返回卫星

```
{
  "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
    }
  }
}

```

示例 [GetSatellite](#) 对于使用自定义星历表的卫星

```

{
  "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"
  }
}

```

列出方位角仰角星历表

由于方位角仰角星历表与卫星无关，因此您需要使用不同的 API 来发现和检索有关它们的信息：

1. 用于列[ListEphemerides](#)出账户中的所有星历，包括方位角高程星历。您可以按状态和星历类型进行筛选。
2. [DescribeEphemeris](#)与特定的星历 ID 一起使用可获取有关方位角高程星历表的详细信息。
3. [DescribeContact](#)与特定的联系人 ID 一起使用可获取有关用于联系人的星历表的详细信息。

包含方位角仰角星历表的示例[ListEphemerides](#)响应：

```

{

```

```
"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

在[ListEphemerides](#)响应中，方位角高程星历表将使用groundStation字段而不是字段，因此易于识别satelliteId。

恢复为默认星历数据

当您上传自定义星历数据时，它将覆盖该特定卫星使用的默认星历表。AWS Ground Station AWS Ground Station 在当前没有已启用、未过期的客户提供的星历表可供使用之前，不会再次使用默认星历。AWS Ground Station 也不会列出超过当前客户提供的星历到期时间的联系人，即使在该到期时间之后有默认的星历可供使用。

Note

方位角高程星历表没有默认值，也不会覆盖卫星星历。它们是在使用trackingOverrides参数预订联系人时明确选择的。如果您不想再使用方位角仰角星历表，只需保留联系人而不指定跟踪覆盖范围，系统将改用活跃的卫星星历表。

恢复 TLE 和 OEM 星历

要恢复到卫星[Space-Track](#)的默认星历表，您需要执行以下操作之一：

- 删除（使用 [DeleteEphemeris](#)）或禁用（使用 [UpdateEphemeris](#)）所有客户提供的已启用星历。您可以使用 [ListEphemerides](#) 列出客户提供的卫星的星历。
- 等待客户提供的所有现有星历到期。

您可以通过调用 [GetSatellite](#) 并验证卫星当前星历的 `source` 是否为 `SPACE_TRACK` 来确认是否正在使用默认星历。有关默认[默认星历数据](#)星历表的更多信息，请参阅。

管理方位角仰角星历

由于方位角仰角星历是为每个接触点明确选择的，并且与卫星无关，因此没有“恢复”为默认值的概念。相反，您可以按如下方式管理方位角高程星历：

- 停止使用方位角仰角星历表：只需保留新的联系人即可，无需指定和指定。`trackingOverrides` `satelliteArn`联系人将改用指定卫星的有效星历。
- 移除未使用的方位角高程星历：用于删除不再[DeleteEphemeris](#)需要的方位角高程星历。请注意，您无法删除已安排的联系人当前正在使用的星历。

要列出账户中的所有方位角仰角星历，请使用。 [ListEphemerides](#) 方位角高程星历表可以通过字段来识别，也可以通过响应中是否存在`groundStation`字段而不是字段来识别。`ephemerisType`
`satelliteId`

使用数据流

AWS Ground Station 使用节点和边缘关系来构造数据流，以便对数据进行流处理。每个节点都由一个描述其预期处理的配置表示。为了说明这个概念，可以考虑将数据流设置antenna-downlink为a.s3-recording该antenna-downlink节点表示根据配置中定义的参数对射频频谱进行模拟到数字的转换。s3-recording表示一个计算节点，它将接收传入的数据并将其存储在您的 S3 存储桶中。生成的数据流是根据您的规格将数字化的 RF 数据异步传输到 S3 存储桶。

在任务配置文件中，您可以创建许多数据流来满足您的需求。以下各节介绍如何设置要与之配合使用的其他 AWS 资源，AWS Ground Station 并提供了构建数据流的建议。有关每个节点行为方式的详细信息，包括将其视为源节点还是目标节点，请参阅[使用 AWS Ground Station 配置](#)。

主题

- [AWS Ground Station 数据平面接口](#)
- [使用跨区域数据传输](#)
- [设置和配置 Amazon S3](#)
- [设置和配置 Amazon VPC](#)
- [设置和配置 Amazon EC2](#)

AWS Ground Station 数据平面接口

所选数据流的结果数据结构取决于数据流的来源。这些格式的详细信息将在卫星装载期间提供给您。以下汇总了每种类型的数据流所使用的格式。

- 天线下行链路
 - (带宽 less-than-or-equal至 40MHz) 数据以 [VITA-49 信号数据/IP 格式数据包](#)的形式传送。
 - (带宽大于 40MHz) 数据作为 2 AWS Ground Station 类数据包传送。
- antenna-downlink-demod-decode
 - 数据以数据 Demodulated/Decoded /IP 格式数据包的形式传送。
- 天线上行链路
 - 数据必须作为 [VITA-49 信号数据/IP 格式数据包](#)传送。
- antenna-uplink-echo
 - 数据以 [VITA-49 信号数据/IP 格式数据包](#)的形式传送。

使用跨区域数据传输

AWS Ground Station 跨区域数据传输功能使您可以灵活地将数据从天线发送到任何 AWS Ground Station 支持的 AWS 区域。这意味着，您可以在单个 AWS 区域维护您的基础设施，并安排任何[AWS Ground Station 位置](#)已上线区域的联系时间。

在 Amazon S3 存储桶中接收您的联系人数据时，AWS Ground Station 将为您管理所有配送方面。

要使用跨区域数据传输到 Amazon EC2 实例（使用 AWS Ground Station 代理或数据流终端节点），必须在您当前的 AWS 区域中创建数据流终端节点，并且必须指定相同的区域。dataflow-endpoint-config AWS Ground Station 将为您管理跨区域传输数据。

设置和配置 Amazon S3

您可以使用 Amazon S3 存储桶通过以下方式 AWS Ground Station 接收下行链路信号。要创建目标 s3 录制配置，您必须能够指定 Amazon S3 存储桶和授权 AWS Ground Station 向存储桶写入文件的 IAM 角色。

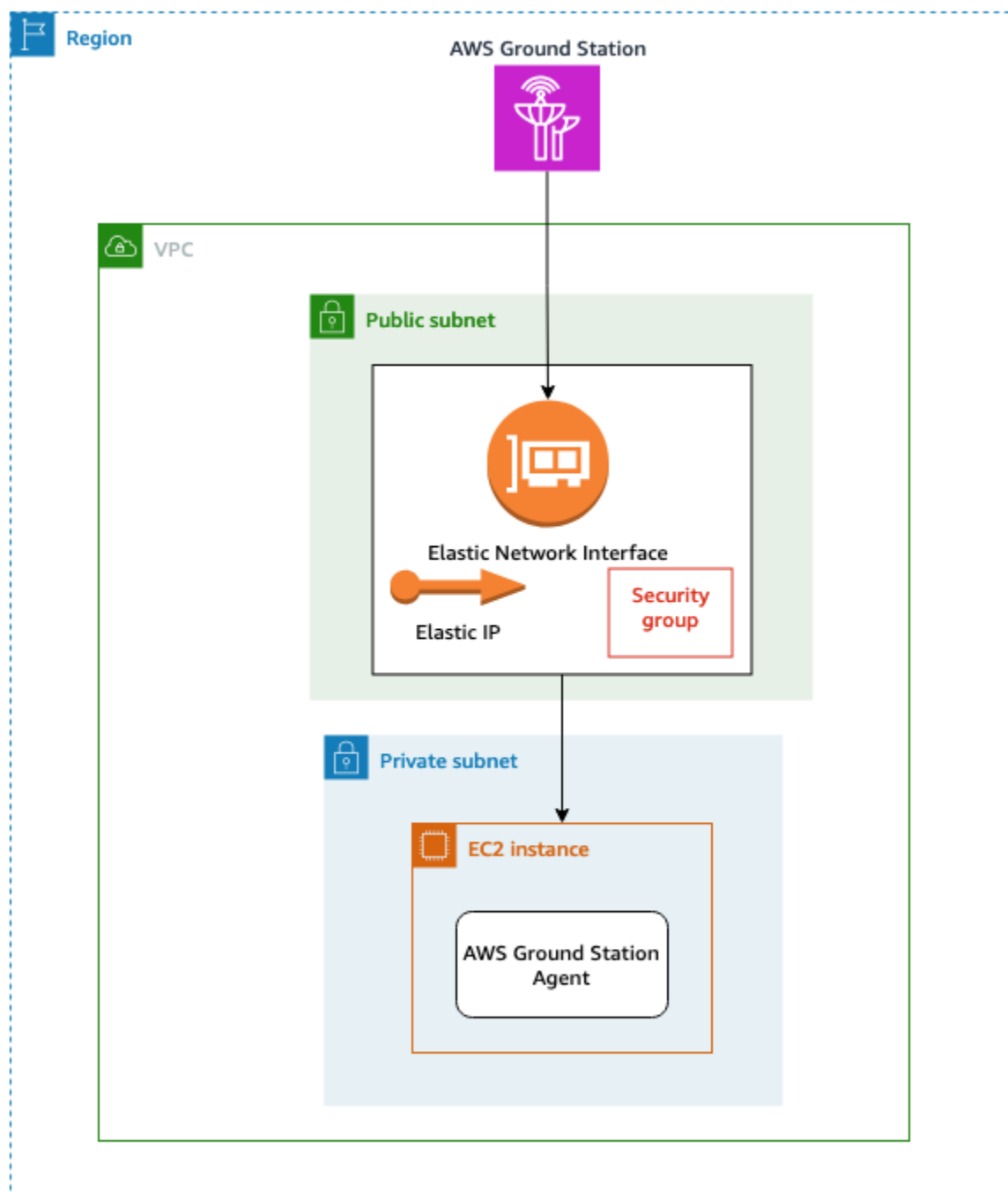
[亚马逊 S3 录音 Config](#)有关 Amazon S3 存储桶、IAM 角色或 AWS Ground Station 配置创建的限制，请参阅。

设置和配置 Amazon VPC

设置 VPC 的完整指南不在本指南的范围之内。要深入了解，请参阅 [Amazon VPC 用户指南](#)。

在本节中，描述了您的 Amazon EC2 和数据流终端节点如何存在于 VPC 中。AWS Ground Station 不支持给定数据流的多个交付点——预计每个数据流都会终止到单个 EC2 接收器。正如我们预期的那样，只有一个 EC2 接收器，该配置不是多可用区冗余的。有关将使用您的 VPC 的完整示例，请参阅[任务配置文件配置示例](#)。

VPC 配置为 AWS Ground Station 座席



您的卫星数据将提供给距离天线较近的 AWS Ground Station 代理实例。AWS Ground Station 代理将使用您提供的密 AWS KMS 钥对您的数据进行条带化处理，然后对其进行加密。每个条带都通过源天线通过 AWS 网络主干发送[到您的](#)亚马逊 EC2 弹性 IP (EIP)。数据通过[附加的](#) Amazon EC2 弹性网络接口 (ENI) 到达您的 EC2 实例。在您的 EC2 实例上，安装的 AWS Ground Station 代理将解密您的数据并执行前向错误更正 (FEC) 以恢复所有丢失的数据，然后将其转发到您在设置中指定的 IP 和端口。

以下列表列出了为 AWS Ground Station 代理交付设置 VPC 时的独特设置注意事项。

安全组 -建议您设置一个专用于 AWS Ground Station 流量的安全组。此安全组应允许在您在 Dataflow 终端节点组中指定的相同端口范围上进行 UDP 入口流量。AWS Ground Station 维护 AWS-managed 前缀列表，将您的权限限制为 AWS Ground Station IP 地址。有关如何替换部署区域的[详细信息](#)，请[参阅](#) AWS 托管前缀列表。PrefixListId

弹性网络接口 (ENI) -您需要将上述安全组与此 ENI 关联并将其放置在您的公有子网中。

Note

每个 ENI 附加的安全组数量的默认配额为 5。这是一个可调整的上限，最高可达 16 个，请[参阅亚马逊 VPC 配额](#)。

以下 CloudFormation 模板演示如何创建本节中描述的基础架构。

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:

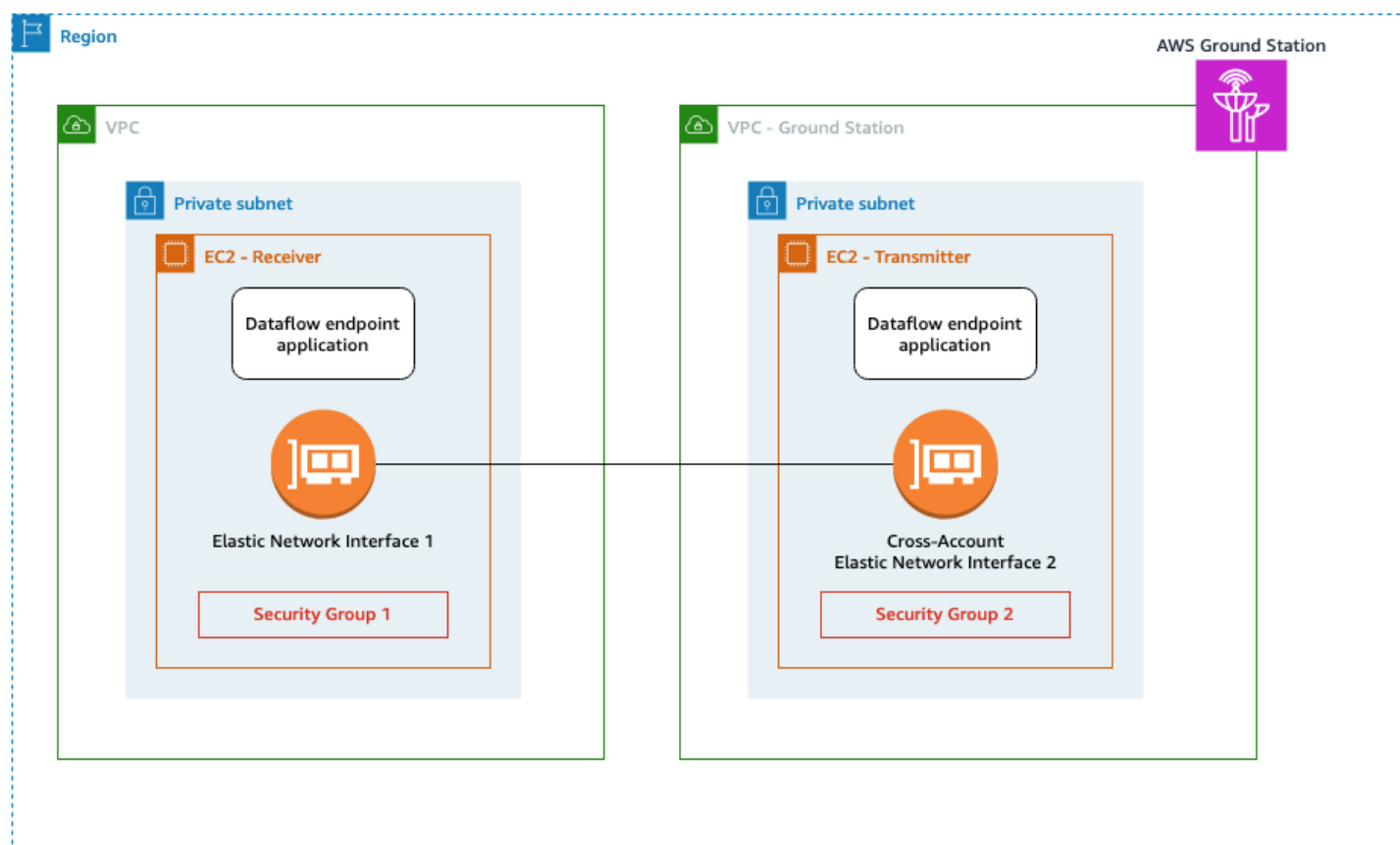
AllocationId:

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

NetworkInterfaceId:

Ref: *InstanceNetworkInterface*

使用数据流终端节点的 VPC 配置



您的卫星数据将提供给离天线最近的数据流端点应用程序实例。然后，数据通过跨账户的[亚马逊 EC2 弹性网络接口 \(ENI\)](#) 从所 AWS Ground Station 拥有的 VPC 发送。然后，数据通过附加到您的 Amazon EC2 实例的 ENI 到达您的 EC2 实例。然后，安装的 dataflow 端点应用程序会将其转发到您在设置中指定的 IP 和端口。对于上行链路连接，情况恰恰相反。

以下列表列出了为数据流终端节点交付设置 VPC 时的独特设置注意事项。

Note

每个 ENI 附加的安全组数量的默认配额为 5。这是一个可调整的上限，最高可达 16 个，请参阅[亚马逊 VPC 配额](#)。

IAM 角色 -IAM 角色是数据流终端节点的一部分，未显示在图中。用于创建跨账户 ENI 并将其连接到 AWS Ground Station Amazon EC2 实例的 IAM 角色。

安全组 1 -此安全组附加到将与您的账户中的 Amazon EC2 实例关联的 ENI。它需要允许来自安全组 2 的 UDP 流量通过您的数据流终端节点组中指定的端口。

弹性网络接口 (ENI) 1 -您需要将安全组 1 与此 ENI 关联并将其置于子网中。

子网 -您需要确保您的账户中 Amazon EC2 实例的每个数据流至少有一个可用 IP 地址。有关子网大小调整的更多详细信息，请参阅[子网 CIDR 块](#)

安全组 2 -数据流端点中引用了此安全组。该安全组将附加到用于向您的账户存入数据的 ENI。AWS Ground Station

区域 -有关跨区域连接支持的区域的更多信息，请参阅[使用跨区域数据传输](#)。

以下 CloudFormation 模板演示如何创建本节中描述的基础架构。

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*

设置和配置 Amazon EC2

要通过 AWS Ground Station 代理或数据流终端节点同步交付 VITA-49Signal/IP data or VITA-49 Extension data/IP，需要正确配置您的 Amazon EC2 实例。根据您的特定需求，您可以直接在同一个实例上执行前端 (FE) 处理器或软件定义无线电 (SDR)，或者可能需要使用其他 EC2实例。FE 或 SDR 的选择和安装超出了本用户指南的范围。有关特定数据格式的更多信息，请参阅[AWS Ground Station 数据平面接口](#)。

有关我们的服务条款的信息，请参阅[AWS 服务条款](#)。

提供的通用软件

AWS Ground Station 提供了常用软件来简化您的 Amazon EC2 实例的设置。

AWS Ground Station 代理人

AWS Ground Station 代理接收数字中频 (digiF) 下行链路数据并输出解密后的数据，从而实现以下功能：

- digiF 下行链路能力从 40 MHz 到 400 MHz 的带宽不等。
- 向网络上的 AWS 任何公共 IP AWS（弹性 IP）传输高速率、低抖动 digiF 数据。
- 使用前向纠错 (FEC) 实现可靠的数据传输。
- 使用客户托管 AWS KMS 密钥进行加密来保护数据传输。

有关更多信息，请参阅《[AWS Ground Station 代理用户指南](#)》。

数据流端点应用程序

一种网络应用程序，用于在 AWS Ground Station 天线位置和您的 AWS Ground Station Amazon EC2 实例之间发送和接收数据。它可用于数据的上行链路和下行链路。

软件定义无线电 (SDR)

一种软件定义的无线电 (SDR)，可用于调制/解调用于与卫星通信的信号。

AWS Ground Station Amazon 机器映像 (AMIs)

为了缩短这些安装的构建和配置时间，AWS Ground Station 还提供了预先配置 AMIs 的选项。AMIs 带有 dataflow 端点的网络应用程序和软件定义无线电 (SDR) 将在您的登录完成后提供给您账户。它们可以在亚马逊 EC2 控制台中通过在私有 Amazon Machine Images (AMIs) 中搜索地面站来找到。with AMIs A AWS Ground Station gent 是公开的，可以在亚马逊 EC2 控制台中通过在公共[亚马逊机器映像 \(AMIs\)](#) 中搜索地面站来找到。

使用遥测

AWS Ground Station 在卫星接触期间，遥测可以从 AWS Ground Station 天线上提供近乎实时的指标。您可以使用遥测数据来监控接触性能，检测异常情况，并就卫星通信做出明智的决策。

主题

- [遥测的工作原理](#)
- [可用的遥测类型](#)
- [区域可用性](#)
- [设置遥测](#)
- [了解遥测数据](#)

遥测的工作原理

要使用遥测，您需要配置一个来指定TelemetrySinkConfig AWS Ground Station 应在何处传送遥测数据。然后，您可以使用该telemetrySinkConfigArn字段将此配置添加到您的任务配置文件中。在使用具有遥测功能的任务配置文件进行联系时，会将遥测数据 AWS Ground Station 传输到您的帐户。

遥测传输过程的工作原理如下：

1. 您在 AWS 帐户中创建 Kinesis 数据流以接收遥测数据。该直播必须在您安排联系人的相同帐户和地区中创建。
2. 您可以创建一个 IAM 角色来授予向您的直播写入数据的 AWS Ground Station 权限。
3. 您可以创建 TelemetrySinkConfig 引用您的直播和 IAM 角色的。
4. 你 TelemetrySinkConfig 将其添加到你的任务配置文件中。
5. 您可以使用支持遥测的新任务配置文件列出和保留联系人。
6. 在使用此任务配置文件进行联系期间，将遥测数据近乎 AWS Ground Station 实时地传输到您的 Kinesis 数据流中。
7. 您可以使用 AWS 服务或自己的应用程序使用和处理来自直播的遥测数据。

可用的遥测类型

AWS Ground Station 在联系期间提供以下遥测类型：

 Note

AWS Ground Station 正在努力扩大支持的遥测类型的数量

指向遥测

提供有关卫星接触期间天线指向方向的信息。这种遥测类型总是在接触期间发送，包括实际和命令的方位角和仰角。有关更多信息，请参阅 [指向遥测](#)。

追踪遥测

提供有关天线跟踪状态和跟踪错误的信息。当您的跟踪配置中启用自动跟踪时，会发送此遥测类型。有关更多信息，请参阅 [追踪遥测](#)。

区域可用性

遥测在所有 AWS Ground Station 运营 AWS 地区都可用。在联系执行期间，遥测数据将从 AWS Ground Station 天线传送到您安排联系的区域，提供跨区域支持。

有关 AWS Ground Station 区域和地面站位置的完整列表，请参阅[AWS Ground Station 位置](#)。

设置遥测

按照以下步骤为您的 AWS Ground Station 联系人配置遥测。完成此设置后，在使用具有遥测功能的任务配置文件进行联系期间，遥测数据将传送到您的 Kinesis Data Streams 流。要深入了解 Kinesis 数据流，请参阅 [Kinesis 数据流用户指南](#)。

步骤 1：创建先决条件 AWS 资源

以下 CloudFormation 代码段演示了如何创建遥测交付的必备 AWS 资源。此代码段创建了一个 Kinesis 数据流和一个 IAM 角色，该角色授予向该流写入遥测数据的 AWS Ground Station 权限。

```
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

```

以下列表列出了为配置遥测交付时的独特设置注意事项。 AWS Ground Station

Kinesis 数据流 -该流使用按需容量模式，可根据吞吐量自动扩展。建议在大多数用例中使用此方法。该流配置为将数据保留 24 小时。默认情况下，该直播使用 AWS 托管加密。要将客户管理的加密与一起使用 AWS Key Management Service，请添加StreamEncryption属性并更新 IAM 角色策略以包括kms:GenerateDataKey权限。有关更多信息，请参阅 [Amazon Kinesis 数据流](#)中的数据保护。

IAM 角色 -IAM 角色允许groundstation.amazonaws.com服务主体代入角色并将遥测数据写入您的 Kinesis 数据流。角色策略为直播授予权限kinesis:DescribeStreamkinesis:PutRecord、和kinesis:PutRecords操作权限。有关设置信任策略和角色政策的指导，请参阅[遥测接收器 Config](#)。

其他配置 -向您用于 AWS Ground Station API 调用的 IAM 用户或角色添加iam:PassRole权限。这允许您在创建 AWS Ground Station 时将遥测角色传递给。 TelemetrySinkConfig

PassRole 策略示例

有关如何更新或附加角色策略的更多信息，请参阅 IAM 用户指南[中的](#)管理 IAM 策略。有关iam:PassRole权限的更多信息，请参阅向用户[授予向 AWS 服务传递角色的权限](#)

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

第 2 步：创建一个 TelemetrySinkConfig

创建一个 TelemetrySinkConfig，定义如何向 AWS Ground Station 您的 Kinesis 数据流传输遥测数据。使用步骤 1 中 CloudFormation 堆栈输出中的流 ARN 和角色 ARN。

Note

创建时 TelemetrySinkConfig，AWS Ground Station 将通过提供分区键为的空测试记录来验证对您的 Kinesis 数据流的 test 访问权限。

有关创建的更多信息 TelemetrySinkConfig，请参阅[遥测接收器 Config](#)。

第 3 步：将遥测添加到任务配置文件中

创建任务档案。有关创建任务配置文件的更多信息，请参阅[使用 AWS Ground Station 任务配置文件](#)。telemetrySinkConfigArn 将其添加到您的任务配置文件中，以便在联系期间启用遥测传输。使用步骤 2 中 TelemetrySinkConfig 创建的 ARN。

第 4 步：安排联系人

使用您启用遥测功能的任务配置文件安排联系人。在联系期间，AWS Ground Station 会将遥测数据传输到您的 Kinesis 数据流中。

联系期间会发生什么

- 遥测开始 - 当联系开始时，数据开始流式传输。

- 近乎实时的交付 -遥测数据几乎实时地到达您的 Kinesis 数据流中。
- 联系时长 -数据贯穿整个联系过程。
- 自动停止 -触点结束后，遥测停止直播。

监控交付

您可以使用以下方式监控遥测交付：

- Kinesis 数据流指标 -查看传入的记录。CloudWatch有关更多信息，请参阅[监控 Amazon Kinesis 数据流](#)。
- 应用程序日志 -验证从流中消耗的应用程序中的数据处理情况。
- Kinesis 数据查看器 -使用 Kinesis 数据流控制台查看流中的样本记录。

后续步骤

完成设置后，您可以：

- 了解遥测数据格式和可用的遥测类型。请参阅[了解遥测数据](#)。
- 构建应用程序来处理来自 Kinesis 数据流的遥测数据。有关更多信息，请参阅为亚马逊 Kinesis 数据流<https://docs.aws.amazon.com/streams/latest/dev/building-consumers.html>培养消费者。
- 使用和其他 AWS 服务创建仪表板 CloudWatch 和警报。
- 如果遇到问题，请查看故障排除指南。请参阅[排除遥测故障](#)。

了解遥测数据

遥测数据 Base64-encoded 以 JSON 记录的形式传送到您的 Kinesis 数据流中。每条记录都包含卫星接触期间收集的信息，包括有关接触和抽样遥测测量的元数据。

数据格式概述

每条遥测记录都包含以下组件：

遥测类型和版本

识别遥测数据的特定类型及其架构版本。这使您可以适当地解析不同的遥测类型。有关架构版本控制的更多信息，请参阅[架构版本控制和演变](#)。

范围 ID

遥测范围的唯一标识符。这使您可以将遥测数据与特定联系人关联起来。

元数据

有关遥测的上下文信息。

数据

特定于遥测类型的抽样遥测测量值。

分区键

遥测记录使用以下格式的分区键传送到您的 Kinesis 数据流中：

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

此分区键可确保将单个联系人的所有给定类型的遥测数据传送到您的 Kinesis Data Streams 流中的同一个分区，从而最大限度地对该联系人的遥测流进行排序。

指向遥测

定点遥测提供有关卫星接触期间天线指向方向的信息。这种遥测类型总是在联系期间发送的。

数据字段

示例时间戳

遥测数据采样的时间，采用 UTC ISO-8601 格式，精度为毫秒。

方位角

天线的实际方位角（以度为单位）。

仰角

天线的实际仰角（以度为单位）。

CommandedAzimuth

以度为单位的指令方位角。这是天线试图达到的目标方位角。

CommandedElevation

以度为单位的命令仰角。这是天线试图达到的目标仰角。

Note

由于物理限制或接触过程中的机械延迟，实际天线位置可能与命令位置不同。

元数据字段

地面站

地面站的名称（例如，“俄亥俄州 1”）。

卫星 ID

中卫星资源的标识符 AWS Ground Station。

contactId

联系人的标识符。

示例 JSON

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

```
}
```

追踪遥测

跟踪遥测提供有关天线跟踪状态和跟踪错误的信息。当您的跟踪配置中启用自动跟踪以及天线正在使用自动跟踪时，系统会发送此遥测类型。

Note

如果您的autotrack参数设置 TrackingConfig 为REMOVED，则不会传送任何追踪遥测数据。有关跟踪配置的更多信息，请参阅[跟踪配置](#)。

数据字段

示例时间戳

遥测数据采样的时间，采用 UTC ISO-8601 格式，精度为毫秒。

追踪状态

天线的当前跟踪状态。可能的值有：

- TRACKING— 天线已成功锁定与任务配置文件相匹配的信号，并正在积极地跟踪该信号。这是接触期间的标称运行状态。
- ACQUIRING— 天线正在定位和锁定信号。该系统目前正在使用编程跟踪，根据星历数据进行指向。
- MASKED— 卫星的预测位置位于自动跟踪掩模后面，这意味着天线无法可靠地在该特定的指向方向上使用自动跟踪。这通常发生在高射频干扰区域，例如低海拔。

追踪 ErrorAzimuth

方位角轴的跟踪误差，以度为单位测量。

追踪 ErrorElevation

仰角轴的跟踪误差，以度为单位。

Note

跟踪误差值表示对基于星历的节目轨道的调整，该调整在自动跟踪期间 AWS Ground Station 适用，以最大限度地提高信号强度。

元数据字段

跟踪遥测包括与指向遥测相同的元数据字段：groundStationsatelliteId、和。contactId

示例 JSON

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

从 Kinesis 数据流中读取数据

遥测数据将传送到您的 Kinesis 数据流中，并可使用标准流消耗模式使用。从直播中读取数据时，请记住以下注意事项。

Base64 解码

Kinesis Data Streams 流中的数据是 Base64-encoded。在将数据解析为 JSON 之前，必须对其进行解码。有关更多信息，请参阅[使用 Amazon Kinesis 数据流](#)。

使用 Kinesis 数据查看器

为了快速访问您的遥测数据，Kinesis 数据流控制台提供了数据查看器功能。使用此功能时：

- 遥测传送可能发生在直播中的任何分片上。
- 默认起始位置从分片中的最新记录中读取。
- 您可能需要调整所选分片并使用“At timestamp”的起始位置来查看收到的记录。

使用 Kinesis 客户端库

Kinesis 客户端库 (KCL) 管理与消费来自 Kinesis 数据流的数据相关的许多复杂问题，包括分片管理、检查点和负载平衡。我们建议将 KCL 用于生产遥测消费应用程序。

有关更多信息，请参阅使用 Kinesis 客户端库<https://docs.aws.amazon.com/streams/latest/dev/kcl.html>开发消费者。

最佳消费实践

- 最大限度地减少延迟 -使用增强版 Fan-Out 从 Kinesis 数据流中读取，与轮询相比，它具有专用的吞吐量和更低的延迟。有关更多信息，请参阅[开发增强型 Fan-Out 消费者](#)。
- 专用流 -使用专用 Kinesis 数据流进行 AWS Ground Station 遥测集成。与其他应用程序共享流可能会导致写入吞吐量饱和和遥测传输失败。
- On-demand 容量 -以按需配置模式部署您的 Kinesis 数据流，以允许根据吞吐量自动扩展分片。
- 监控吞吐量 -使用 CloudWatch 指标监控您的直播是否受到限制。有关更多信息，请参阅[监控 Amazon Kinesis 数据流](#)。

架构版本控制和演变

遥测架构经过版本控制，以支持随着时间的推移而演变。每条记录中的telemetryVersion字段表示架构版本。

处理架构变更

- 将来可能会引入新的遥测类型。
- 现有的遥测类型可能会收到带有重大更改的新版本。
- 您的应用程序应容忍未知的遥测类型和版本。
- 解析telemetryTypeAndVersiontelemetryType、和telemetryVersion字段以确定如何处理每条记录。

我们建议实现版本感知型负载序列化，该序列化可以正常处理多个架构版本，从而允许您的应用程序在引入新版本时继续运行。

与联系人合作

您可以使用 AWS Ground Station 控制台输入卫星数据、确定天线位置、通信和安排所选卫星的天线时间 AWS CLI，也可以使用所选语言的 AWS SDK 来安排所选卫星的天线时间。您最多可以在联系开始前 15 分钟查看、取消和重新安排联系人预订*。您还可以更新联系人以指定星历表覆盖范围（包括 azimuth/elevation OEM 或 TLE 跟踪数据）或更改目标卫星。有关更多信息，请参阅 [更新联系人和联系人版本控制](#)。此外，如果您使用的是 AWS Ground Station 预留分钟定价模式，则可以查看您的预留分钟定价计划的详细信息。

AWS Ground Station 支持跨区域数据传输。作为您选择的任务配置文件一部分的数据流终端节点配置决定了将数据传输到哪个区域。有关使用跨区域数据传输的更多信息，请参阅[使用跨区域数据传输](#)。

要安排联络，必须配置您的资源。如果您尚未配置资源，请参阅[开始使用](#)。当 [ReserveContact](#) 被调用时，AWS Ground Station 拍摄任务配置文件和配置资源的快照，以便在联系人的整个生命周期中使用。使用 [UpdateMissionProfile](#) 和 [UpdateConfig](#) API 对这些资源所做的更改不会反映在更新之前保留的联系人中。如果您需要将资源更改应用于已安排的联系人，则必须先使用取消该联系 [CancelContact](#)，然后使用重新安排该联系人。 [ReserveContact](#)

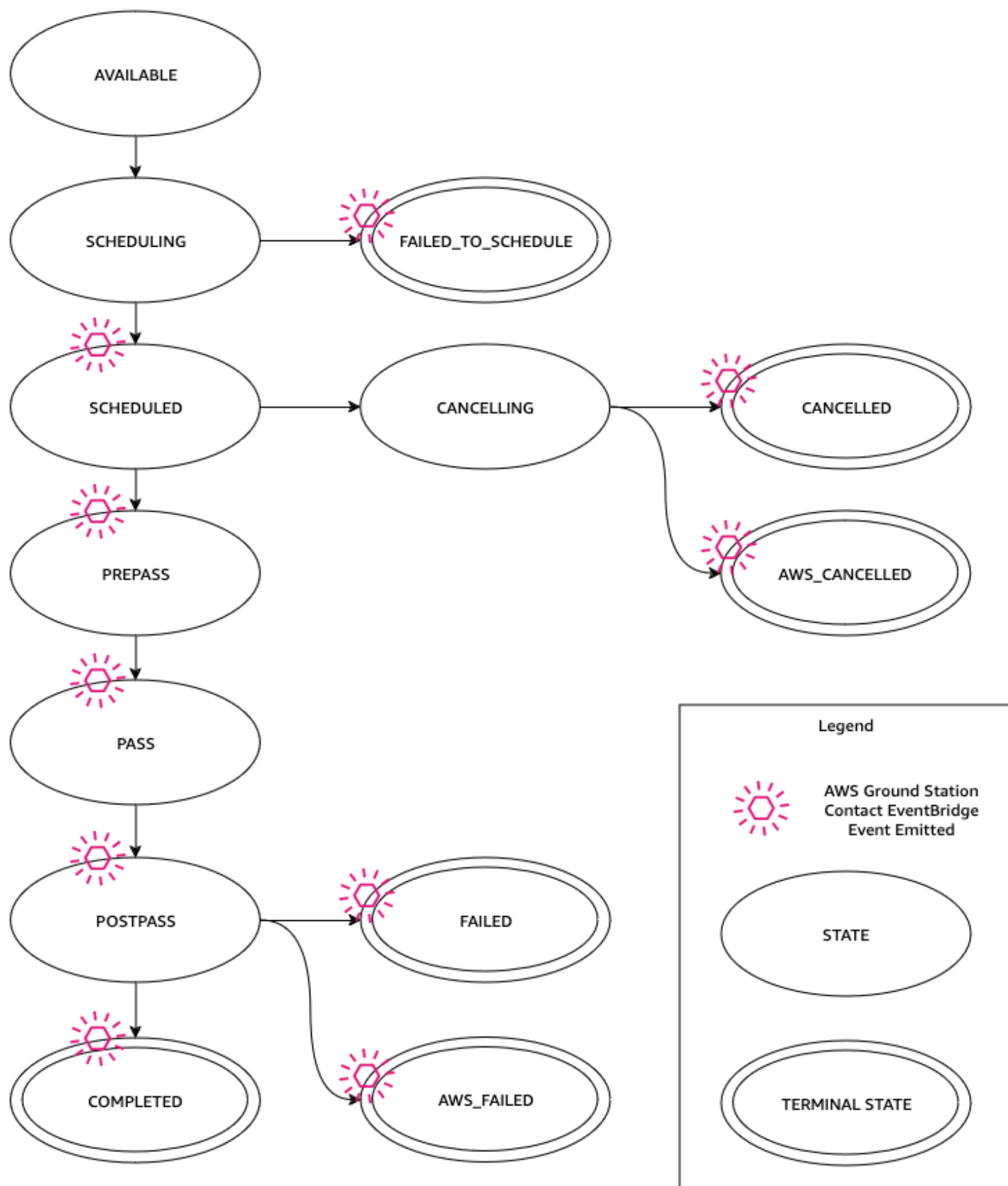
* 如果在距离联系时间过近时取消联系人，取消的联系可能会产生费用。有关取消联络的更多信息，请参阅：[Ground Station 常见问题解答](#)。

主题

- [了解联系生命周期](#)
- [了解联系人账单](#)
- [更新联系人和联系人版本控制](#)

了解联系生命周期

了解联系生命周期可以帮助您在使用时自动处理和解决各种问题 AWS Ground Station。下图显示了 AWS Ground Station 联系人生命周期以及生命周期内发出的 Event Bridge 事件。请务必注意，“已完成”、“失败”、“已失败”、“已取消”、“AWS_CANCELED”和“AWS_FAILED”是终端状态。联系人不会过渡到终端状态。有关每种状态所表示[AWS Ground Station 联系人状态](#)的内容以及使用该状态是可停止还是可取消的详细信息，请参阅。[CancelContact](#)



AWS Ground Station 联系人状态

AWS Ground Station 联系人的状态可以深入了解该联系人在给定时间发生了什么。

联系人状态

下表描述了联系人可能具有的状态：

Status	说明	Terminal	可取消	可阻挡
AVAILABLE	该联系人可供预订。	否	N/A	N/A
日程安排	联系人正在安排中。	否	是	否
SCHEDULED	联系人已成功安排。	否	是	否
未能按计划执行	联系人未能安排时间。	是	否	否
预先通过	联系即将开始，资源正在准备中。	否	是	否
PASS	该联系目前正在执行中，正在与卫星通信。	否	否	是
后期通行证	通信已完成，使用的资源正在清理中。	否	否	否
COMPLETED	联系已完成，没有错误。	是	否	否
FAILED	由于您的资源配置存在问题，联系失败。	是	否	否
AWS_失败	由于 AWS Ground Station 服务出现问题，联系失败。	是	否	否
正在取消	联系人正在取消中。	否	否	否
AWS_已取消	该 AWS Ground Station 服务取消了该联系。天线或场地维护以及星历漂移就是这种情况可能发生的例子。	是	否	否

Status	说明	Terminal	可取消	可阻挡
CANCELLED	联系已被您取消。	是	否	否

Note

有关取消或停止联系的账单影响的信息，请参阅[了解联系人账单](#)。

联系人数据保留

AWS Ground Station 在[ReserveContact](#)请求保留联系人后，将联系人数据保留 1 年。1 年期限过后，联系人数据将被删除。

如果您需要将联系人数据保留超过一年，建议在保留期到期之前导出您的数据。有关如何访问和导出联系人数据的更多信息，请参阅：

- [AWS Ground Station API 引用](#)
- [AWS Ground Station CLI 命令参考](#)

了解联系人账单

使用 AWS Ground Station，您只需为使用的天线时间付费。AWS Ground Station 以每分钟为基础的仪表接触使用量。对于每个联系人，该服务会计算从开始到结束的联系时长，并四舍五入到最接近的分钟。此计费时长决定了您为该联系人收取的费用。

您的费率取决于两个主要因素：

- 带宽 — 为联系人预留的带宽量（窄带或宽带）
- 地面站位置 — 费率因地面站位置而异

带宽定义

AWS Ground Station 根据瞬时带宽将联系人分为两个带宽层：

- 窄带 — 瞬时带宽小于或等于 40 MHz 的任何接点

- 宽带 — 瞬时带宽大于 40 MHz 的任何接点

调度模式

AWS Ground Station 提供两种调度模式：

- On-Demand— 无需长期承诺即可为天线接入付费
- 预留 — 与按月订阅相比 On-Demand，提供折扣费率和更好的日程安排。预留分钟定价适用于承诺在一段时间内按月使用的客户。

要了解您的账户的具体定价信息或要了解有关预留计划模式的更多信息，请联系您的 AWS 代表。

CancelContact

[CancelContact](#) API 的使用因调用时的联系状态而异：

- 联系开始之前 -完全取消联系
- 接触开始之后和接触结束之前 -停止正在进行的接触

当您取消联系时，计费取决于您的日程安排模式和取消时间。如需更多信息，请联系您的 AWS 代表。

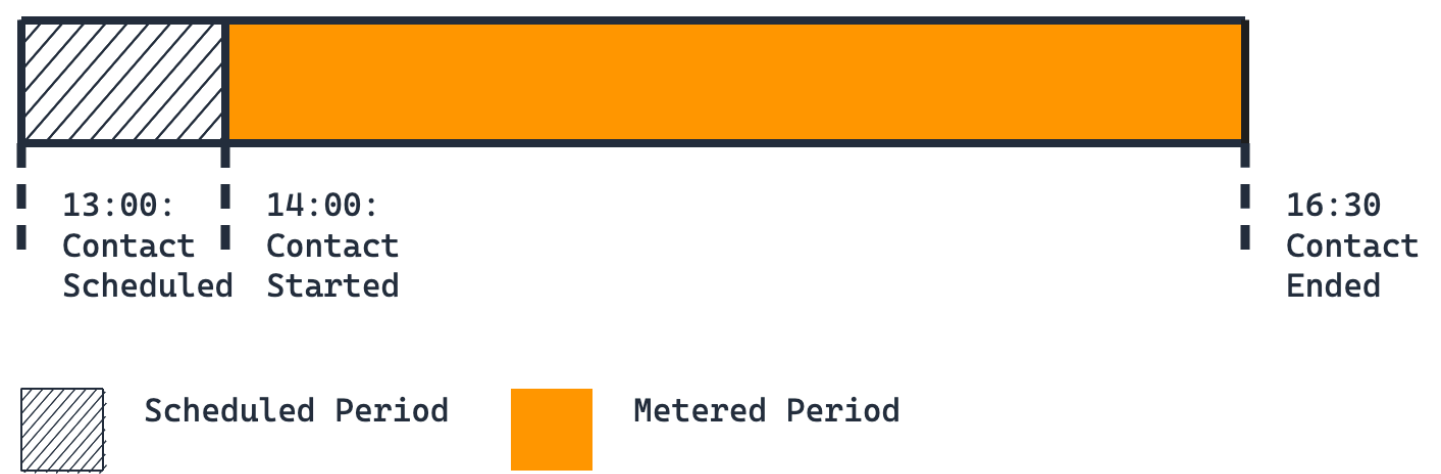
当您停止联系时，您需要按已执行的联系部分以及重复联系人未涵盖的剩余时间付费。在这种情况下，重复的联系人是：

- 与最初停止的联系人安排在同一个人地面站
- 由与最初停止的联系人相同的 AWS 账户 ID 进行安排
- 在发出停止原始联系人的命令后保留

以下场景演示了这种计量在实践中的工作原理。

场景 1：单个联系人

你在 Anytown 1 地面站安排了 150 分钟的接触，从 14:00 开始，16:30 结束。



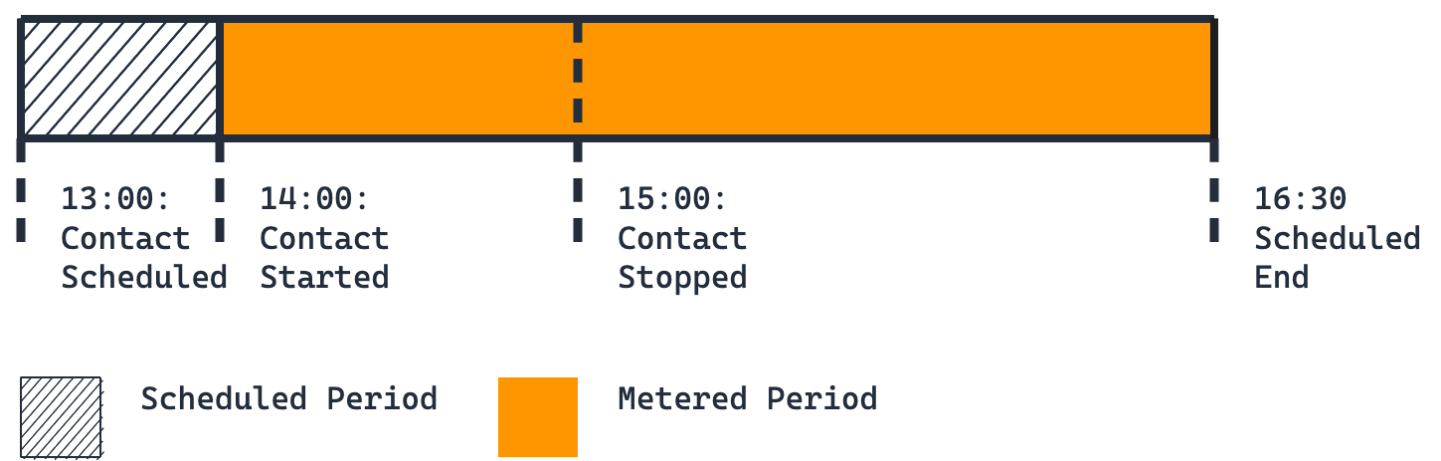
账单明细：

- 首次接触：150 分钟（完整时长）

您需要支付 150 分钟的费用。这是基准场景，即联系人一直运行到预定完成时间，没有任何中断或取消。

场景 2：单次停止接触

你在 Anytown 1 地面站安排了 150 分钟的接触，从 14:00 开始，16:30 结束。在 15:00，你调用 CancelContact API 来停止联系。



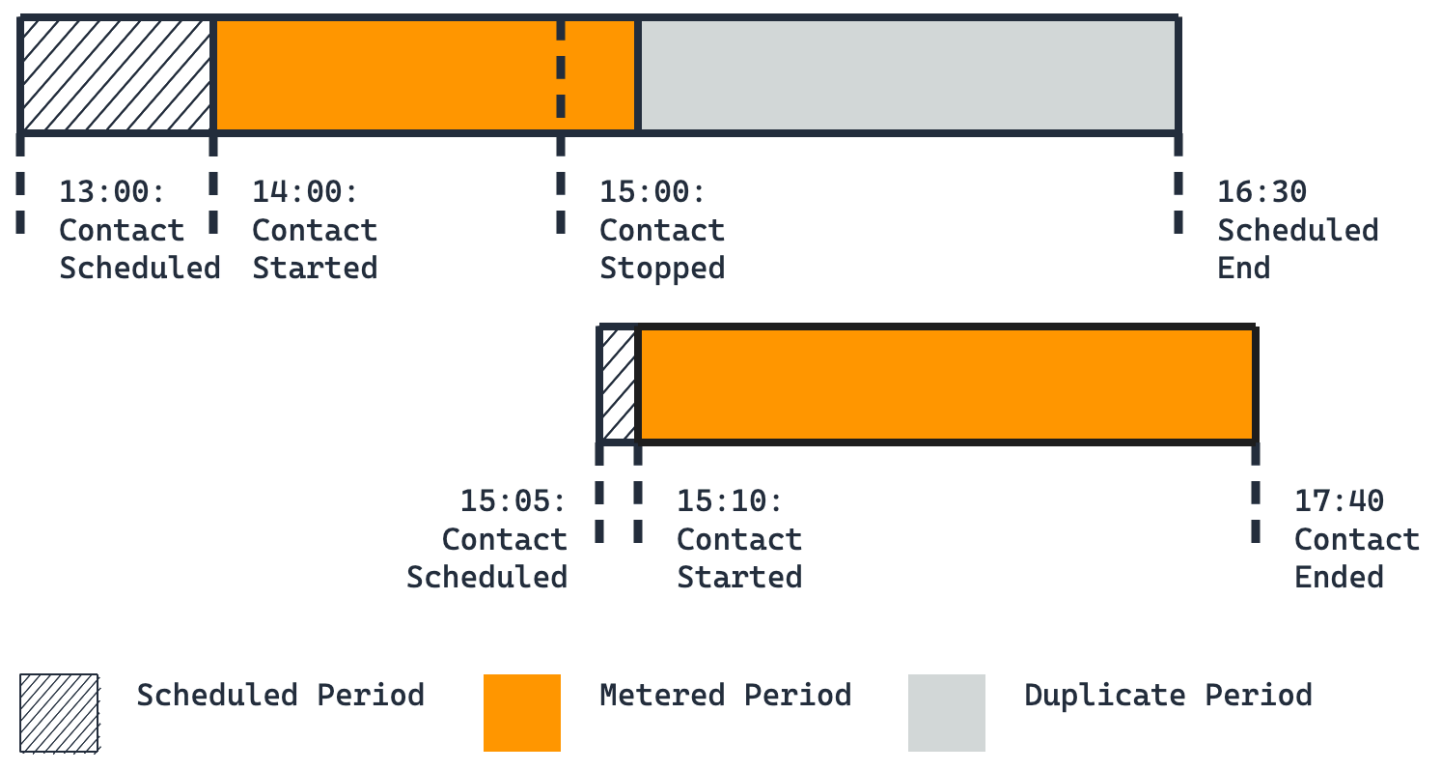
账单明细：

- 首次接触：150 分钟（完整的原始时长）

您需要为整整 150 分钟计费，因为您停止了联系但没有安排任何重复联系以覆盖剩余时间（ 15:00-16:30 ）。当您在没有安排重复联系时停止联系时，您仍需对最初安排的整个持续时间负责。

场景 3：单一副本

你在 Anytown 1 地面站安排了 150 分钟的接触，从 14:00 开始，16:30 结束。在 15:00，你调用 CancelContact API 来停止第一次接触。致电后 CancelContact，您可以在 15:10 开始在同一地面站安排另一位联系人，持续 150 分钟。



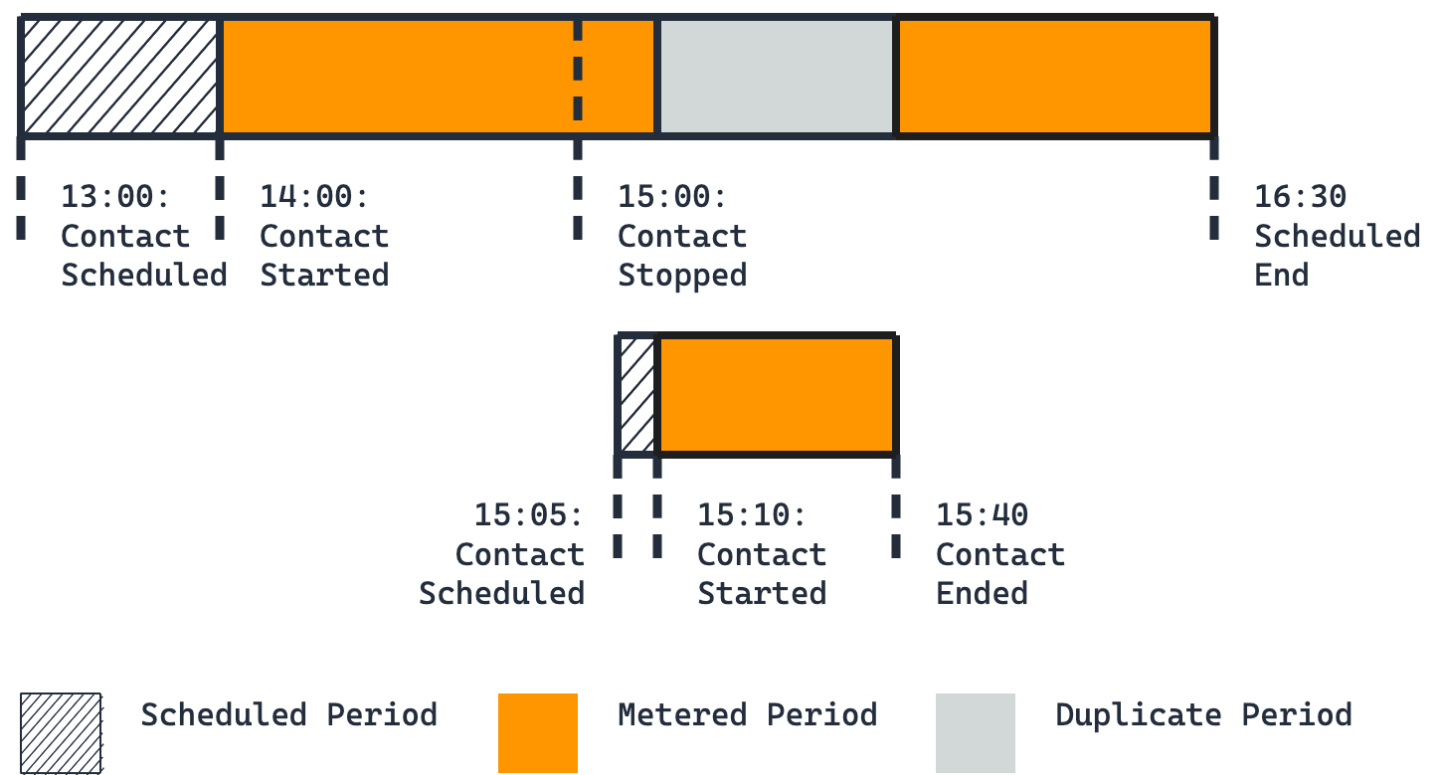
账单明细：

- 第一次接触：70 分钟（ 执行 60 分钟 + 第二次联系开始前 10 分钟的停机时间 ）
- 第二次接触：150 分钟（ 完整时长 ）

第二个联系人是重复的，因为您是在停止第一个联系之后安排的。该副本涵盖了从 15:10 到 16:30 的剩余时间，因此您只需要按第一次联系的实际运行时间加上停止和重启之间的 10 分钟间隔进行计费。

场景 4：短复制

你在 Anytown 1 地面站安排了 150 分钟的接触，从 14:00 开始，16:30 结束。在 15:00，你调用 CancelContact API 来停止第一次接触。致电后 CancelContact，您可以安排从 15:10 开始在同一个地面站进行30分钟的联系。



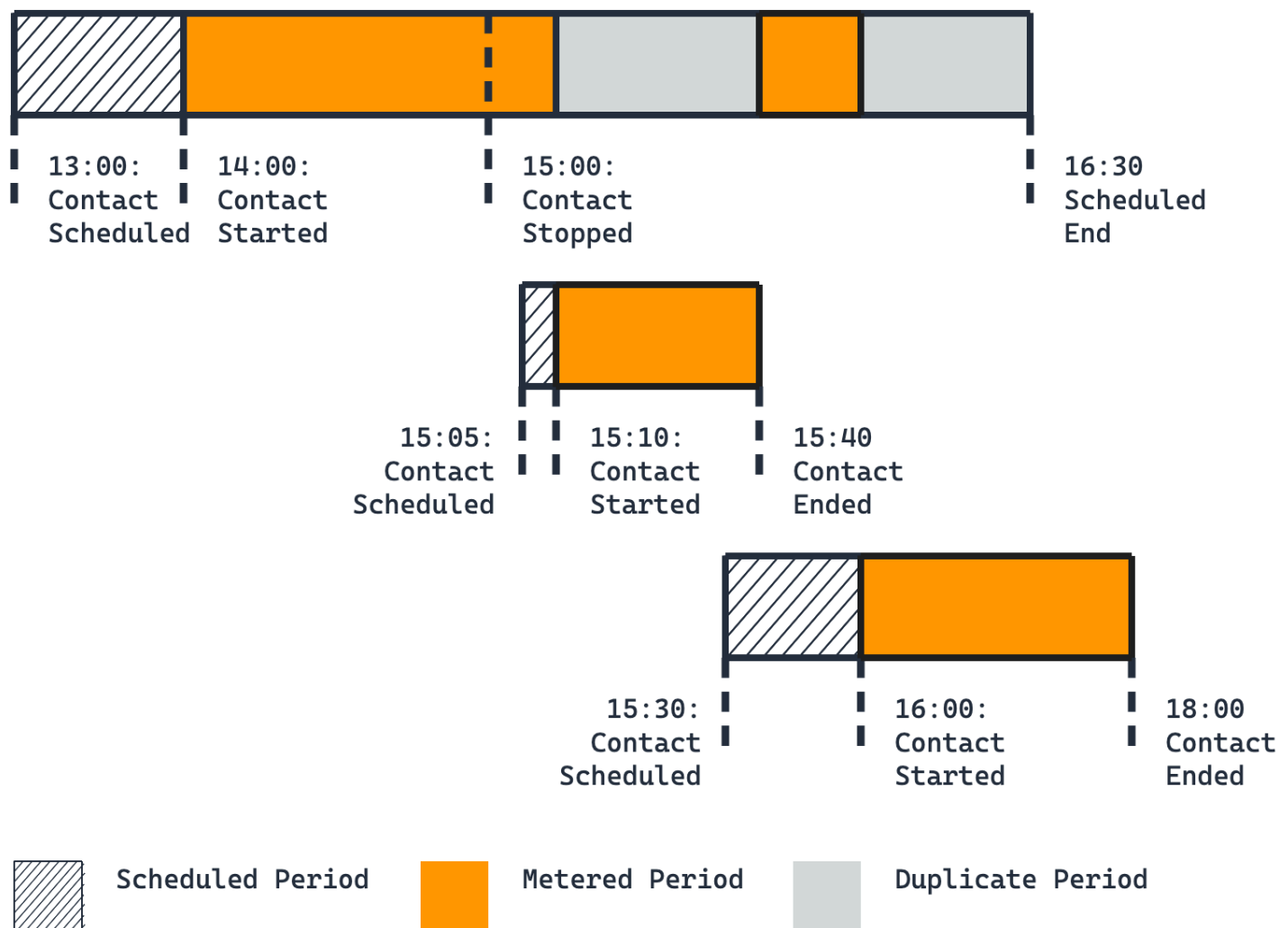
账单明细：

- 第一次接触：120 分钟（执行 60 分钟 + 第二次联系开始前 10 分钟的停机时间 + 副本未涵盖的 50 分钟剩余时间）
- 第二次接触：30 分钟（完整时长）

重复的联系仅涵盖您停止第一次联系后的剩余 90 分钟中的 30 分钟（15:10-15:40）。您需要为重复开始前的 10 分钟间隔以及重复结束后的 50 分钟（15:40-16:30）的未覆盖时间付费。

场景 5：多个重复项

你在 Anytown 1 地面站安排了 150 分钟的接触，从 14:00 开始，16:30 结束。在 15:00，你调用 CancelContact API 来停止第一次接触。致电后 CancelContact，您可以安排从 15:10 开始在同一个地面站进行30分钟的联系。稍后，在 15:30，您将从 16:00 开始安排另一位联系人，为期 120 分钟。



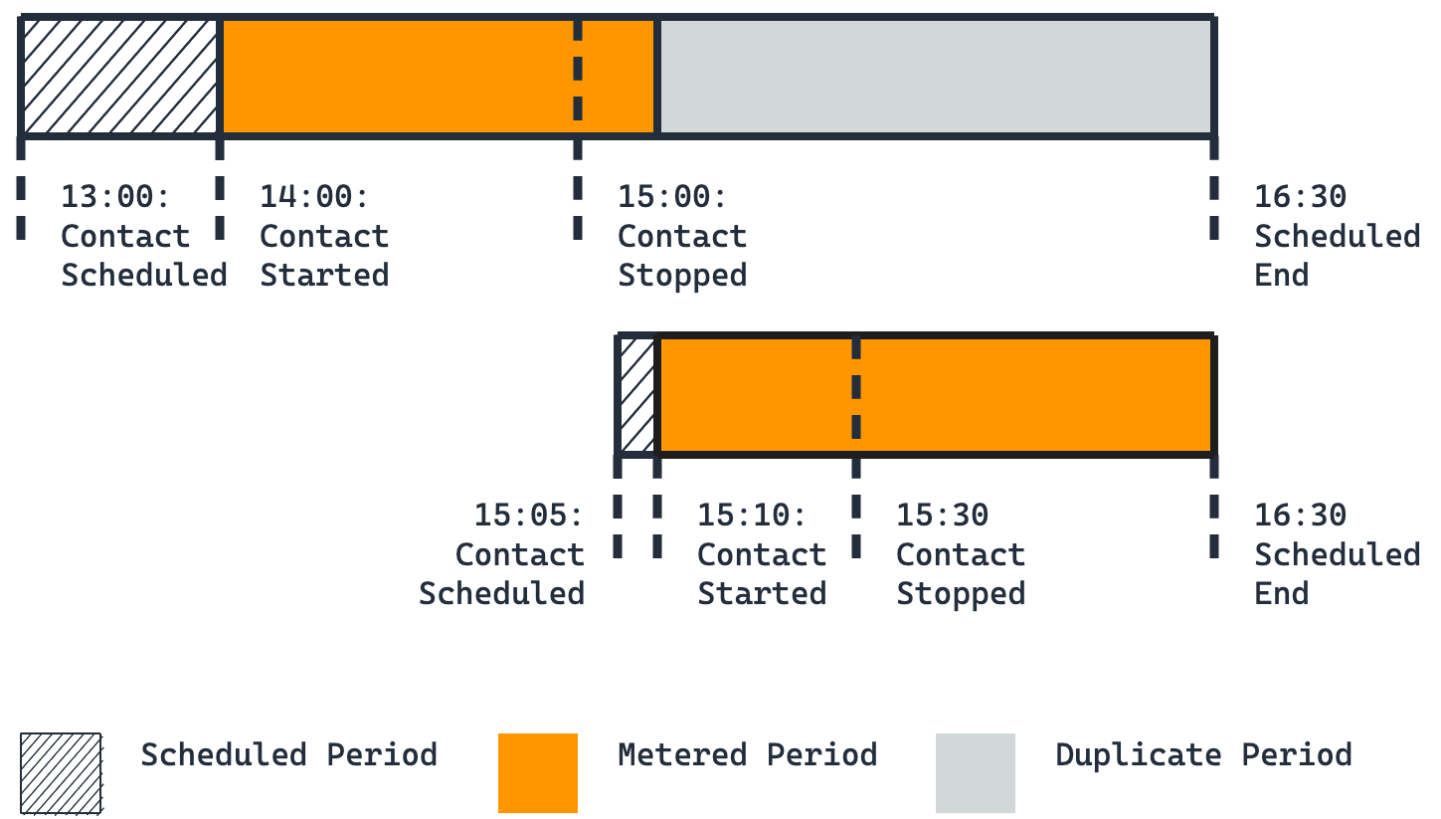
账单明细：

- 第一次接触：90 分钟（执行 60 分钟 + 第二次联系开始前 10 分钟的停机时间 + 第二和第三次联系之间的停机时间 20 分钟）
- 第二次接触：30 分钟（完整时长）
- 第三次接触：120 分钟（完整时长）

第二和第三个联系都算作重复联系人，因为您是在停止第一个联系之后安排的。但是，您仍需要为联系人之间的间隔付费：第一站（15:00）和第二次开始（15:10）之间 10 分钟，第二站（15:40）和第三次开始（16:00）之间 20 分钟。

场景 6：多次停靠

你在 Anytown 1 地面站安排了 150 分钟的接触，从 14:00 开始，16:30 结束。在 15:00，你调用 CancelContact API 来停止第一次接触。致电后 CancelContact，你在 Anytown 1 地面站预约 80 分钟的联系时间，该联系从 15:10 开始，16:30 结束。在 15:30，您再次调用 CancelContact API，停止重复联系。



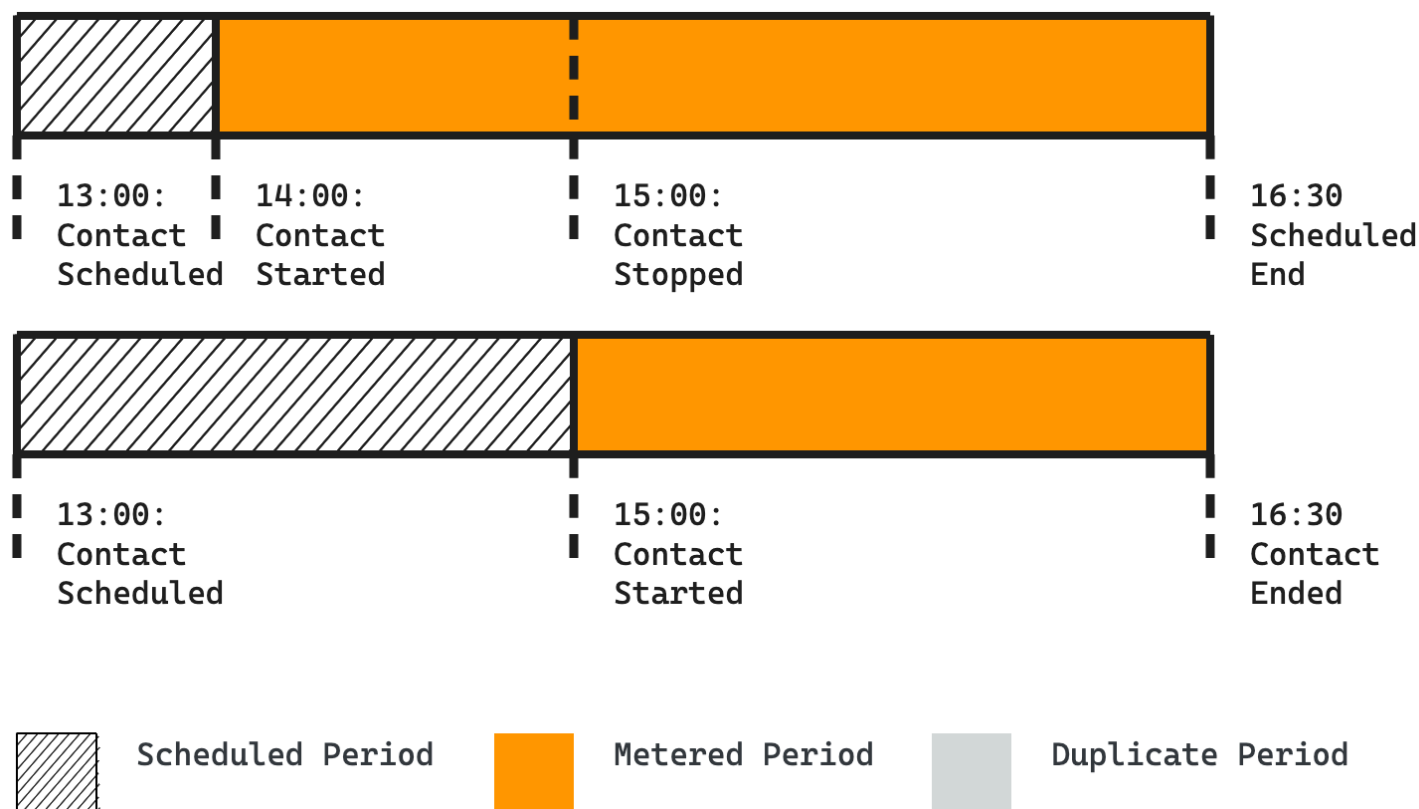
账单明细：

- 第一次接触：70 分钟（执行 60 分钟 + 第二次联系开始前 10 分钟的停机时间）
- 第二次接触：80 分钟（完整的原始时长）

第二个联系人按其 80 分钟的完整时长计费，因为您在 15:30 将其停止，剩下 60 分钟的原定时间（15:30-16:30）空缺。除非你安排另一个重复的联系以覆盖剩余的时间，否则你应对任何中断联系的整个持续时间负责。

场景 7：没有重复 Multi-antenna 的地面站

13:00，你在 Anytown 1 地面站安排两次接触。第一个是 150 分钟的接触，从 14:00 开始，到 16:30 结束。第二个是 90 分钟的接触，从 15:00 开始，到 16:30 结束。在 15:00，你调用 CancelContact API 来停止第一次接触。Anytown 1 地面站是一个多天线地面站，它允许两个触点同时运行。



账单明细：

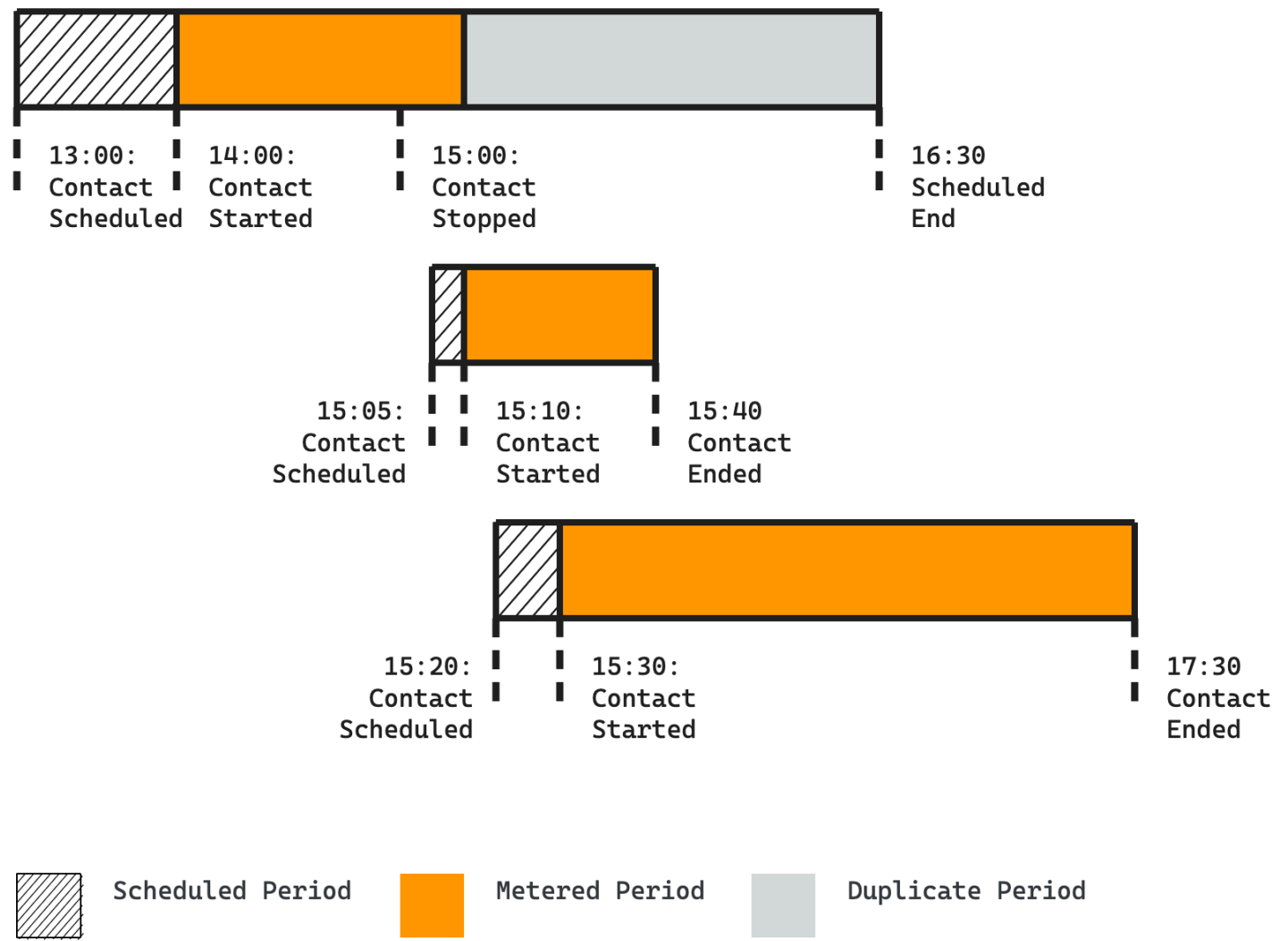
- 首次接触：150 分钟（完整的原始时长）
- 第二次接触：90 分钟（完整时长）

尽管第二个联系人与第一个联系人的停止部分重叠，但它不算作重复接触。第二个联系不符合第一个重复联系标准：它安排在 13:00，然后你在 15:00 停止了第一个联系人。因为这不是重复的，所以无论何时停止，你都将按第一次联系的原始时长进行计费。

场景 8：有重复触点 Multi-antenna 的地面站

你在 Anytown 1 地面站安排了 150 分钟的接触，从 14:00 开始，16:30 结束。在 15:00，你调用 CancelContact API 来停止第一次接触。致电后 CancelContact，您在 Anytown 1 地面站预约 30 分钟

的联系时间，从 15:10 开始，到 15:40 结束。稍后，你在 Anytown 1 地面站再安排一次 90 分钟的接触，从 15:30 开始，17:00 结束。Anytown 1 地面站是一个多天线地面站，它允许两个重复的触点在重叠时间内同时运行。



账单明细：

- 第一次接触：70 分钟（执行 60 分钟 + 第二次联系开始前 10 分钟的停机时间）
- 第二次接触：30 分钟（完整时长）
- 第三次接触：90 分钟（完整时长）

第二和第三个联系都算作重复联系人，因为您是在停止第一个联系之后安排的。停止第一个联系人（15:00）和启动第二个联系人（15:10）之间的 10 分钟间隔表示按原始联系人计费的停机时间。

更新联系人和联系人版本控制

AWS Ground Station 支持以 SCHEDULEDPREPASS、或联系人状态更新 PASS [联系人](#)。您可以使用 [UpdateContact](#) API 为联系人（包括 azimuth/elevation OEM 或 TLE 跟踪数据）指定星历表替代，而无需取消和重新安排。这对于在接触期间需要将天线重新分配给另一颗卫星的地球同步（GEO）卫星运行或需要调整指向的发射和早期操作（LEOP）非常有用。

每次预订或更新联系人时，都会 AWS Ground Station 创建一个新的联系人版本。联系人版本提供联系人更改的历史记录，并允许您跟踪每次更新的状态。

联系人版本控制的工作原理

当您呼叫时 [ReserveContact](#)，AWS Ground Station 创建联系人的第一个版本（版本 1）并在响应 versionId 中返回。每次后续调用都会 [UpdateContact](#) 创建一个版本号递增的新版本。

[DescribeContact](#) API 返回当前的 ACTIVE [联系版本](#)，包括响应 version 字段中的版本信息。该 [ListContacts](#) API 还包括每个联系人的版本信息。

要查看联系人的特定版本，请使用 [DescribeContactVersion](#) API。要列出联系人的所有版本，请使用 [ListContactVersions](#) API。

更新联系人

[UpdateContact](#) 当联系人处于“已安排”、“预通”或“通过”状态时，您可以拨打电话。API 接受以下参数：

- 联系人 ID — 要更新的联系人的标识符。
- clientToken — 一种等性令牌，可确保请求仅处理一次。如果您使用相同的客户端令牌重试请求，则在不再次执行更新的情况下 AWS Ground Station 返回原始响应。如果未提供客户端令牌，许多 AWS SDK 会自动为您生成客户端令牌。
- TrackingOverrides — 联系人的新跟踪配置。这包括节目曲目设置（azimuth/elevation TLE 或 OEM 星历）。
- SatelliteArn — 联系人卫星的 ARN。更改目标卫星和节目轨道设置时，请提供新卫星的 ARN。只有获准使用 azimuth/elevation 指向角度的客户才能将此值设置为空。所有其他客户必须包括联系人的卫星 ARN。

 Important

UpdateContactAPI 应用请求中的所有参数。任何省略或明确设置为 null 的参数都将被视为清除该值的请求，而不是使其保持不变。例如，如果您提供trackingOverrides但省略satelliteArn，则卫星 ARN 将被清除。确保在每个更新请求中包含所有所需的值。

在接触期间，您可以通过提供新的satelliteArn卫星和相应的卫星来更改目标卫星trackingOverrides。在接触期间，新卫星必须可以从地面站看见，因为使用此 API，接触的开始和结束时间不会改变。新卫星还必须登上地面站，并获得任务概况所要求的许可。接触点的任务配置文件无法更改，因此只有当两颗卫星使用相同的任务配置文件时，切换卫星才适用。

 Important

该 UpdateContact API 不支持更改联系人的开始时间、结束时间或任务资料。要更改这些值，请取消联系并保留一个新联系人。该 UpdateContact API 旨在重新分配天线指向配置，例如在卫星之间切换或更新星历数据。

 Important

UpdateContactAPI 不支持任务配置文件使用[天线下行传输解调解码配置](#)配置的联系人。要更改这些联系人的配置，请取消联系人并保留一个新联系人。

UpdateContactAPI 返回contactId和新的versionId。更新是异步处理的。[DescribeContactVersion](#)用于检查更新状态。有些 AWS 软件开发工具包会 AWS Command Line Interface 提供一个ContactUpdated服务员，负责轮询直到版本达到 ACTIVE 或 FAILED_TO_UPDATE 状态。

 Note

一次只能进行一次更新。如果最新的联系人版本处于更新状态，则 API 返回ConflictException。等待当前更新达到 ACTIVE 或 FAILED_TO_UPDATE 状态，然后再提交其他更新。

联系人版本状态

每个联系人版本都具有以下状态之一：

Status	说明
UPDATING	该版本正在应用于联系人。更新已提交并正在处理中 AWS Ground Station。
ACTIVE	该版本是联系人当前的有效配置。地面站正在使用此版本的设置。
SUPERSEDED	该版本以前处于活动状态，但已被更新的版本所取代。
更新失败	无法应用更新。联系人将恢复到先前处于活动状态的版本。有关详细信息，请查看failureCodes 和failureMessage 字段。

代码示例

以下示例演示了如何将联系人版本控制 API 与 Python AWS 软件开发工具包 (Boto3) 一起使用。

示例：更新联系人

以下示例使用新的追踪替代项更新联系人，并使用 Boto ContactUpdated 3 服务员等待更新完成。

```
import boto3
import uuid

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

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

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

```

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

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

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

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

from botocore.exceptions import WaiterError

waiter = ground_station_client.get_waiter("contact_updated")

try:
    waiter.wait(
        contactId=contact_id,
        versionId=version_id,
        WaiterConfig={
            "Delay": 5,
            "MaxAttempts": 180,
        },
    )
    print(f"Contact updated successfully. Version {version_id} is now active.")
except WaiterError as e:

```

```
# WaiterError is raised when the version reaches FAILED_TO_UPDATE
# or when MaxAttempts is exceeded. Retrieve the current version to inspect.
version_response = ground_station_client.describe_contact_version(
    contactId=contact_id,
    versionId=version_id,
)
version_status = version_response["version"]["status"]
if version_status == "FAILED_TO_UPDATE":
    failure_codes = version_response["version"].get("failureCodes", [])
    failure_message = version_response["version"].get("failureMessage", "")
    print(f"Update failed. Codes: {failure_codes}, Message: {failure_message}")
else:
    print(f"Waiter timed out. Current version status: {version_status}. Error:
{e}")
```

示例：描述联系人版本

以下示例检索特定联系人版本的详细信息，包括其状态、配置和任何故障信息。

```
import boto3

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

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

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

response = ground_station_client.describe_contact_version(
    contactId=contact_id,
    versionId=version_id,
)

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

```

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

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

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

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

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

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

```

示例：列出联系人版本

以下示例列出了联系人的所有版本，以查看完整的更改历史记录，并使用分页来处理大型结果集。

```

import boto3

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

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

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

paginator = ground_station_client.get_paginator("list_contact_versions")
page_iterator = paginator.paginate(

```

```

        contactId=contact_id,
        PaginationConfig={
            "MaxItems": 100,
            "PageSize": 20,
        },
    )

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

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

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

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

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

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

```

注意事项

- 在引入联系人版本控制功能之前创建的联系人没有版本信息。致电[DescribeContactVersion](#)或[ListContactVersions](#)为这些联系人返回 a `ResourceNotFoundException`。
- 当您更新正在进行的联系的跟踪覆盖范围时，天线会根据新的指向配置进行调整，这会有一段短暂的过渡期。在此期间，信号接收或传输可能会中断。
- 无法删除联系人版本。用于查看[ListContactVersions](#)对联系人所做更改的完整历史记录。
- 只能从联系人的调度区域调用UpdateContact该 API。

使用AWS Ground Station数字双胞胎功能

的数字双胞胎功能为您AWS Ground Station提供了一个可以测试和集成卫星任务管理和指挥与控制软件的环境。数字双胞胎功能允许您在不使用生产天线容量的情况下测试调度、验证配置和正确的错误处理。通过测试与数字双胞胎功能的AWS Ground Station集成，您可以增强对系统顺利管理卫星运行的能力的信心。它还允许您在不使用生产容量或需要频谱许可的情况下测试AWS Ground Station API。

要开始使用，请关注[机载卫星](#)，申请加入数字双胞胎功能。一旦您的卫星登上数字双胞胎功能，您就可以安排与数字双胞胎地面站的联系。您可以通过 AWS 开发工具包[ListGroundStations](#)响应检索您有权访问的地面站列表。数字双胞胎地面站是中列出的地面站的精确副本，[AWS Ground Station 位置](#)Ground Station 名称的修改前缀为“数字双胞胎”。这包括他们的天线功能和元数据，包括但不限于站点掩码和实际的 GPS 坐标。目前，数字双胞胎功能不支持数据传输，如中所述[使用数据流](#)。此外，数字双胞胎地面站[使用遥测](#)不支持中描述的遥测交付。

一旦上线，数字双胞胎功能就会发出与生产服务相同的 Amazon EventBridge 事件和 API 响应，如中所述。[自动化 AWS Ground Station 含活动](#)这些事件将允许您微调配置和数据流端点组。

AWS Ground Station 专用天线

AWS Ground Station 专用天线是代表您 AWS 管理的定制天线系统。与您与其他客户共享天线时间的公共天线不同，专用天线可让您专门访问 AWS Ground Station 按照您的规格制造的天线。您可以拥有一个或多个专用天线。专用天线连接到全球 AWS 网络，您可以使用与公共天线相同的工作流程 AWS Ground Station APIs 与之交互，此外还可以增强对天线利用率的可见性。

什么是专用天线

专用天线是一种物理天线系统，专为您的组织量身定制，完全由其 AWS 管理。您可以拥有一个或多个专用天线。专用天线与公共 AWS Ground Station 天线的不同之处在于：

- 定制 — 根据您的规格定制专用天线。专用天线的功能并不局限于公共天线的功能，如中所述。[AWS Ground Station 网站能力](#)
- 灵活的位置-专用天线不局限于现有的 AWS Ground Station 天线位置。专用天线可以连接到任何现有 AWS 区域，包括当前 AWS Ground Station 不可用的区域。您可以与一起 AWS 确定符合您要求的位置和区域。
- 专用访问权限 — 您可以专门访问天线，而不是在每次联系的基础上与其他 AWS Ground Station 客户共享天线时间。

包括专用 AWS Ground Station 天线在内的所有天线均由完全管理。AWS 这包括维护和与全球 AWS 网络的连接。您可以使用与公共天线相同的工作流程 AWS Ground Station APIs 与专用天线进行交互。您可以用同样的方式安排联系人、配置任务配置文件和提供数据。

专用天线可以由多个 AWS 账户共享，持有专用天线合同的客户可以选择加入哪些账户。每个已登录的账户都可以独立安排天线上的联系人，并且可以查看共享天线的所有账户的预订情况。

要了解有关专用天线的更多信息或开始使用，请 AWS 支持 通过以下方式联系。[AWS Support Center Console](#)

增强预订可见性

当你对专用天线使用 [ListGroundStationReservations](#) API 时，你会看到公共天线上没有的其他信息。下表汇总了专用天线 [ListGroundStationReservations](#) API 与公共 AWS Ground Station 天线相比的行为差异。

行为	专用天线	公共天线
你自己的联系人	可见完整细节，包括 contactId	可见完整细节，包括 contactId
其他账户的联系人	仅在时隙中可见，没有 contactId	不可见
维护时段	使用PLANNED或可见 UNPLANNED maintenanceType	不可见

维护时段表示天线无法用于卫星通信的时段。该maintenanceType字段表示维护是还PLANNED是UNPLANNED。计划外维护时，与维护时段重叠的联系可能会被 AWS Ground Station取消。

当您查看共享您的专用天线的其他 AWS 账户的联系人时，预留包括时隙和天线信息，但contactId不包括在内。

Important

增强的预留可见性仅适用于您的专用天线。当您使用公共 AWS Ground Station 天线时，您的可见性与其他任何客户相同。在公共天线上，您看不到维护时段或其他账户的预约。

有关列出预订的更多信息，请参阅[查看地面站预订](#)。

相关资源

- [查看地面站预订](#)
- [AWS Ground Station 位置](#)
- [AWS Ground Station 网站能力](#)
- 《AWS Ground Station API 参考》中的 [ListGroundStationReservations](#)

通过以下方式了解监控 AWS Ground Station

监控是保持 AWS Ground Station 可靠性、可用性和性能的重要环节。AWS 提供以下监控工具 AWS Ground Station，供您监视、报告问题并在适当时自动采取措施。

- Amazon EventBridge Events 提供近乎实时的系统事件流，这些事件描述了 AWS 资源的变化。EventBridge 事件支持事件驱动的自动计算，因为您可以编写规则来监视某些事件，并在这些事件发生时在其他 AWS 服务中触发自动操作。有关 EventBridge 活动的更多信息，请参阅 [Amazon EventBridge Events 用户指南](#)。
- AWS CloudTrail 捕获由您的账户或代表您的 AWS 账户进行的 API 调用和相关事件，并将日志文件传输到您指定的 Amazon S3 存储桶。您可以识别哪些用户和帐户拨打了电话 AWS、发出呼叫的源 IP 地址以及呼叫发生的时间。有关的更多信息 AWS CloudTrail，请参阅《[AWS CloudTrail 用户指南](#)》。
- 使用时，Amazon CloudWatch Metrics 会捕获您的预定联系指标 AWS Ground Station。CloudWatch Metrics 使您能够根据自己的频道、极化和卫星 ID 分析数据，以识别信号强度和联系人中的错误。有关更多信息，请参阅[使用 Amazon CloudWatch 指标](#)。
- [AWS 用户通知服务](#) 可用于设置交付渠道以获取有关 AWS Ground Station 事件的通知。当事件与指定的规则匹配时，会收到通知。您可以通过多个渠道接收事件通知，包括电子邮件、[聊天应用程序中的 Amazon Q 开发者版](#) 聊天通知或 [AWS Console Mobile Application](#) 推送通知。您还可以在 AWS 控制台[通知中心查看通知](#)。用户通知服务 支持聚合，这可以减少您在特定事件期间收到的通知数量。

使用以下主题监控 AWS Ground Station。

主题

- [自动化 AWS Ground Station 含活动](#)
- [使用记录 AWS Ground Station API 调用 AWS CloudTrail](#)
- [使用亚马逊查看指标 CloudWatch](#)

自动化 AWS Ground Station 含活动

Note

本文档始终使用“事件”一词。CloudWatch 事件和 EventBridge 是相同的底层服务和 API。可以使用任何一种服务构建用于匹配传入事件并将事件路由至目标进行处理的规则。

事件使您能够自动化 AWS 服务并自动响应系统事件，例如应用程序可用性问题和资源更改。来自 AWS 服务的事件几乎是实时交付的。您可以编写简单规则来指示您关注的事件，并指示要在事件匹配规则时执行的自动化操作。一些可以自动触发的操作包括：

- 调用函数 AWS Lambda
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams
- 激活 AWS Step Functions 状态机
- 通知 Amazon SNS 主题或 Amazon SQS 队列

将事件与一起使用的一些示例 AWS Ground Station 包括：

- 调用 Lambda 函数以根据事件状态自动启动和停止 Amazon EC2 实例。
- 每当联络变更状态时发布到 Amazon SNS 主题。这些主题可以设置为在开始或结束联络时发送电子邮件通知。

有关更多信息，请参阅[亚马逊 EventBridge 活动用户指南](#)。

AWS Ground Station 事件类型

Note

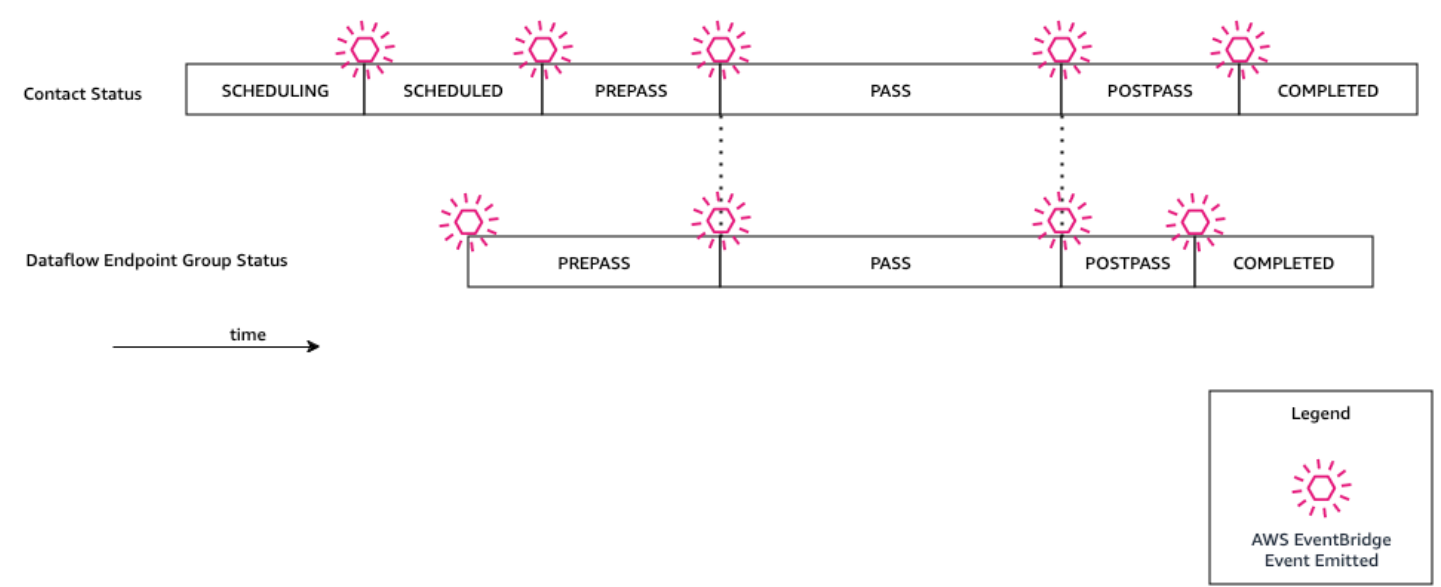
AWS Ground Station 生成的所有事件都将“aws.groundstation”作为“来源”的值。

AWS Ground Station 发出与状态更改相关的事件以支持您自定义自动化的能力。目前，AWS Ground Station 支持联系人状态更改事件、数据流端点组更改事件和星历状态更改事件。以下部分提供有关每种类型的详细信息。

联系活动时间表

AWS Ground Station 当您的联系人更改状态时发出事件。有关这些状态变化是什么以及状态本身含义的更多信息，请参阅[了解联系生命周期](#)。您的联系人中使用的任何数据流终端节点组都有一组独立的事件，这些事件也会触发。在同一时间段内，我们还会为您的数据流终端节点组发出事件。在设置任务配置文件和数据流终端节点组时，您可以配置传递前和传递后事件的确切时间。

下图显示了名义联系人及其关联的数据流端点组的状态和发出的事件。



Ground Station 联络状态变更

如果您想在即将到来的联系人更改状态时执行特定操作，则可以设置规则来自动执行此操作。当您想要接收有关联络状态变更的通知时，这很有用。如果您想更改接收这些活动的时间，可以修改任务资料的[联系方式PrePassDurationSeconds](#)和[联系方式PostPassDurationSeconds](#)。事件将发送到计划联络的区域。

下面提供了一个示例事件。

```
{
  "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 可使用的值在 [the section called “AWS Ground Station 联系人状态”](#) 中定义。

Ground Station 数据流终端节点组状态变更

如果要在数据流终端节点组正用于接收数据时执行操作，则可以设置 规则以自动执行此操作。这将允许您执行不同的操作来响应数据流终端节点组状态的不断变更的状况。如果您想更改接收这些事件的时间，请使用包含不同[联系PrePassDurationSeconds](#)人和[联系人的数据流终端节点组。PostPassDurationSeconds](#) 此事件将发送到数据流终端节点组的区域。

下面提供了一个示例。

```

{
  "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` 的可能状态包括 PREPASS、PASS、POSTPASS 和 COMPLETED。

星历活动

Ground Station Ephemeris 状态变更

如果想要在星历变更状态时执行操作，则可设置规则以自动执行此操作。这允许您根据星历的变化状态执行不同的操作。例如，您可以在星历完成验证且现在已 ENABLED 时执行操作。此事件的通知将发送到上传星历的区域。

下面提供了一个示例。

```

{
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "Ground Station Ephemeris State Change",
  "source": "aws.groundstation",
  "account": "123456789012",
  "time": "2019-12-03T21:29:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:groundstation::123456789012:satellite/10313191-c9d9-4ecb-a5f2-bc55cab050ec",
    "arn:aws:groundstation::123456789012:ephemeris/111111-cccc-bbbb-a555-bcccca005000"
  ],
  "detail": {
    "ephemerisStatus": "ENABLED",
    "ephemerisId": "111111-cccc-bbbb-a555-bcccca005000",
    "satelliteId": "10313191-c9d9-4ecb-a5f2-bc55cab050ec"
  }
}

```

```
}
```

ephemerisStatus 的可能状态包括 ENABLED、VALIDATING、INVALID、ERROR、DISABLED 和 EXPIRED

使用记录 AWS Ground Station API 调用 AWS CloudTrail

AWS Ground Station 与 AWS CloudTrail 一项服务集成，该服务提供用户、角色或 AWS 服务在中执行的操作的记录 AWS Ground Station。CloudTrail 将所有 API 调用捕获 AWS Ground Station 为事件。捕获的调用包括来自 AWS Ground Station 控制台的调用和对 AWS Ground Station API 操作的代码调用。如果您创建跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括的事件 AWS Ground Station。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的事件历史记录中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向哪个请求发出 AWS Ground Station、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅[AWS CloudTrail 用户指南](#)。

AWS Ground Station 中的信息 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上启用。当活动发生在中时 AWS Ground Station，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在自己的 AWS 账户中查看、搜索和下载最近发生的事件。有关更多信息，请参阅[使用事件历史记录查看 CloudTrail 事件](#)。

要持续记录您 AWS 账户中的事件，包括的事件 AWS Ground Station，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- [创建跟踪概述](#)
- [CloudTrail 支持的服务和集成](#)
- [配置 Amazon SNS 通知 CloudTrail](#)
- [接收来自多个区域的 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件](#)

所有 AWS Ground Station 操作均由 API 参考记录 CloudTrail 并记录在 [AWS Ground Station API 参考](#)中。例如，调用 CancelContact 和 ListConfigs 操作会在 CloudTrail 日志文件中生成条目。ReserveContact

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根证书还是 AWS Identity and Access Management (IAM) 用户凭证发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

了解 AWS Ground Station 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序出现。

以下示例显示了演示该 ReserveContact 操作的 CloudTrail 日志条目。

示例 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"
  }
}

```

使用亚马逊查看指标 CloudWatch

在联系期间，AWS Ground Station 自动捕获数据并将其发送到 CloudWatch 进行分析。可以在亚马逊 CloudWatch 控制台中查看您的数据。有关访问和 CloudWatch 指标的更多信息，请参阅[使用亚马逊 CloudWatch 指标](#)。

AWS Ground Station 遥测功能还可用于在联系期间接收近乎实时的指标。CloudWatch 指标无法近乎实时地提供，可能会延迟交付。CloudWatch 还会在一秒钟内聚合指标，这可能会降低数据粒度。遥测功能提供个人指标，并近乎实时地将其直接发送到您的 AWS 账户。有关更多信息，请参阅[使用遥测](#)。

Important

AWS Ground Station 向与联系人地面站位置相关的 AWS 区域发送 CloudWatch 指标，而不是向安排联系的 AWS 区域发送指标。要查看联系人的指标，您必须 CloudWatch 在地面站区域

进行访问。有关哪个 AWS 区域与每个地面站位置相关的信息，请参阅[查找 AWS 地面站位置的区域](#)。要在安排联系的地区接收遥测数据，可以使用 AWS Ground Station 遥测功能。有关更多信息，请参阅[使用遥测](#)。

AWS Ground Station 指标与维度

有哪些指标可用？

以下指标可从中获得 AWS Ground Station。

 Note

发出的具体指标取决于所使用的 AWS Ground Station 功能。根据您的配置，只能发出以下指标的子集。

指标	指标维度	说明
AzimuthAngle	Satelliteld	天线的方位角。 真北为 0 度，东为 90 度。 单位：度
BitErrorRate	信道，极化， Satelliteld	在给定数量的位传输中的位上误差率。位误差是由噪声、失真或干扰引起的 单位：每单位时间的位误差数量
BlockErrorRate	信道，极化， Satelliteld	给定数量的接收块中块的误差率。块误差是由干扰引起的。

指标	指标维度	说明
		单位：错误的块数/总块数
CarrierFrequencyRecovery_Cn0	类别、配置、 Satelliteld	单位带宽的载波噪声密度比。 单位：分贝赫兹 (d) B-Hz
CarrierFrequencyRecovery_Locked	类别、配置、 Satelliteld	解调器载波频率恢复环路锁定时设置为 1，解锁时设置为 0。 单位：无
CarrierFrequencyRecovery_OffsetFrequency_Hz	类别、配置、 Satelliteld	预估信号中心频率与理想中心频率之间的偏差。这是由多普勒频移和航天器与天线系统之间的局部振荡器偏移引起的。 单位：赫兹 (Hz)
ElevationAngle	Satelliteld	天线的仰角。地平线为 0 度，天顶为 90 度。 单位：度
Es/N0	信道，极化， Satelliteld	每个符号的能量与噪声功率谱密度的比率。 单位：分贝 (dB)

指标	指标维度	说明
ReceivedPower	两极分化, SatelliteId	中测得的信号强度 demodulator/decoder。 单位：分贝毫瓦 (dBm)
SymbolTimingRecovery_ErrorVectorMagnitude	类别、配置、 SatelliteId	接收符号和理想星座点之间的误差矢量幅度。 单位：百分比
SymbolTimingRecovery_Locked	类别、配置、 SatelliteId	解调器符号时序恢复环路锁定时设置为 1，解锁时设置为 0。 单位：无
SymbolTimingRecovery_OffsetSymbolRate	类别、配置、 SatelliteId	预估符号速率和理想信号符号速率之间的偏差。这是由多普勒频移和航天器与天线系统之间的局部振荡器偏移引起的。 单位：symbols/second

尺寸是用来做什么的 AWS Ground Station?

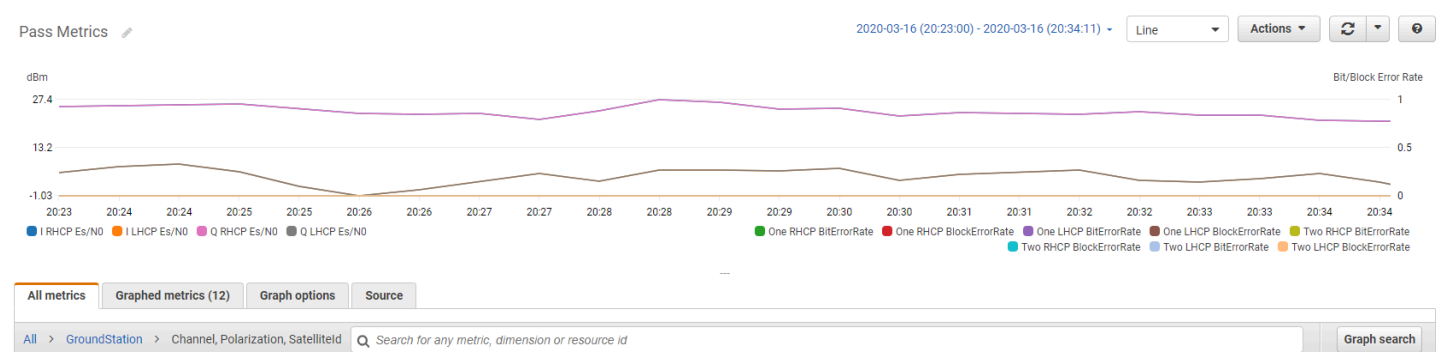
您可以使用以下维度筛选 AWS Ground Station 数据。

维度	说明
Category	解调或解码。
Channel	每次联络的通道包括 1、2、I (相中) 和 Q (正交)。
Config	天线下行链路演示解码配置 arn。
Polarization	每次联络的偏振包括 LHCP (左手圆偏振) 或 RHCP (右手圆偏振)。
SatelliteId	卫星 ID 包含用于联络的卫星的 ARN。

查看 指标

查看图形化的指标时，请务必注意，聚合窗口决定了指标的显示方式。联络中的每个指标都可以在收到数据后 3 小时内显示为每秒的数据。在 3 小时后，您的数据将按 CloudWatch 指标汇总为每分钟数据。如果您需要根据每秒数据衡量标准查看指标，建议在收到数据后的 3 小时内查看数据，或者将其保留在 M CloudWatch etrics 之外保存。有关 CloudWatch 留存率的更多信息，请参阅[亚马逊 CloudWatch 概念-指标留存率](#)。

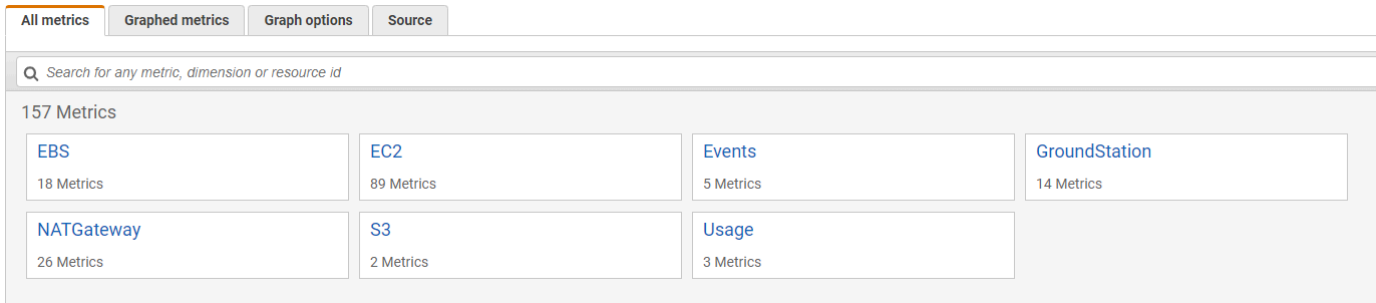
此外，在前 60 秒内捕获的任何数据都不会包含足够的信息来生成有意义的指标，并且很可能不会显示。为了查看有意义的指标，建议在 60 秒后查看您的数据。



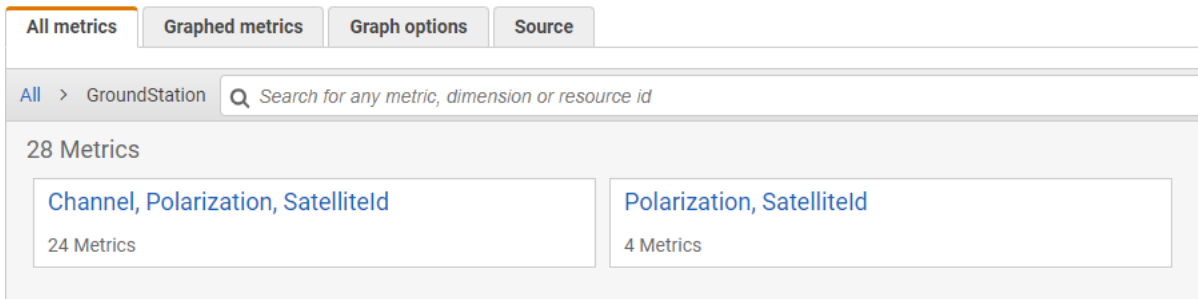
有关在中绘制 AWS Ground Station 指标的更多信息 CloudWatch，请参阅[绘制指标](#)。

使用控制台查看指标

1. 确定与您的地面站位置相关的 AWS 区域。AWS Ground Station 在与您的联系人的地面站位置相关的区域发布 CloudWatch 指标。有关地面站位置及其相关 AWS 区域的列表，请参阅[查找 AWS 地面站位置的区域](#)。
2. 打开 [CloudWatch 控制台](#)。
3. 在导航窗格中，选择指标。
4. 选择 GroundStation 命名空间。



5. 选择所需的指标维度（例如，信道、极化、Satelliteld）。



6. All metrics 选项卡显示命名空间中该维度的所有指标。您可执行以下操作：
 - a. 要对表进行排序，请使用列标题。
 - b. 要绘制指标的图表，请选中与该指标关联的复选框。要选择所有指标，请选中表格标题行中的复选框。
 - c. 要按资源进行筛选，请选择资源 ID，然后选择 Add to search。
 - d. 要按指标进行筛选，请选择指标名称，然后选择 Add to search。

要查看指标，请使用 AWS CLI

AWS Ground Station 在与您的联系人的地面站位置相关的区域发布 CloudWatch 指标。有关地面站位置及其关联 AWS 区域的列表，[查找 AWS 地面站位置的区域](#)。 *ground-station-region-code* 替换为您的地面站位置的 AWS 区域代码（us-west-2 例如，俄勒冈 1、夏威夷 1 或阿拉斯加 1）。此过程中的所有后续 AWS CLI 命令都必须使用相同的区域。

1. 确保 AWS CLI 已安装。有关安装的信息 AWS CLI，请参阅[安装 AWS CLI 版本 2](#)。
2. 确定与您的地面站位置相关的 AWS 区域。
3. 使用 CloudWatch CLI 的 [get-metric-data](#) 方法生成一个文件，该文件可以修改以指定您感兴趣的指标，然后用于查询这些指标。

为此，请运行以下命令：`aws cloudwatch get-metric-data --region ground-station-region-code --generate-cli-skeleton`。这将生成类似于以下内容的输出：

```
{
  "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": ""
    }
}

```

- 按运行列出可用 CloudWatch 指标 `aws cloudwatch list-metrics --region ground-station-region-code`。

如果你最近使用过 AWS Ground Station，该方法应返回包含以下条目的输出：

```

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

```

Note

如果自上次使用以来已超过 2 周 AWS Ground Station，则需要手动检查可用指标[表](#)，以便在指标命名空间中找到指标名称和维度。AWS/GroundStation 有关 CloudWatch 限制的更多信息，请参阅：[查看可用指标](#)

5. 修改您在步骤 2 中创建的 JSON 文件，以匹配步骤 3 和指标Polarization中的所需值。SatelliteId另外，请务必更新StartTime、和EndTime值以匹配您的联系人。例如：

```
{
  "MetricDataQueries": [
    {
      "Id": "receivedPowerExample",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/GroundStation",
          "MetricName": "ReceivedPower",
          "Dimensions": [
            {
              "Name": "SatelliteId",
              "Value":
                "arn:aws:groundstation::111111111111:satellite/aaaaaaa-bbbb-cccc-dddd-
                eeeeeeeeeee"
            }
          ],
          {
            "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 根据指标，每 1 到 60 秒发布一次指标。如果该Period字段的值小于指标的发布周期，则不会返回指标。

6. `aws cloudwatch get-metric-data`使用前面步骤中创建的配置文件运行。下面提供了一个示例。

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

将向指标提供来自联络的时间戳。下面提供 AWS Ground Station 指标的示例输出。

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

安全性 AWS Ground Station

云安全 AWS 是重中之重。作为 AWS 客户，您将受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。AWS 提供专用的安全工具和特性，以帮助实现安全目标。这些工具和功能包括网络安全性、配置管理、访问控制和数据安全性。

使用时 AWS Ground Station，我们建议您遵循行业最佳实践并实施 end-to-end 加密。AWS APIs 为您提供集成加密和数据保护的功能。有关 AWS 安全的更多信息，请参阅 [AWS 安全简介](#) 白皮书。

使用以下主题了解如何保护您的资源。

主题

- [适用于 Identity and Access 管理 AWS Ground Station](#)
- [AWS 的托管策略 AWS Ground Station](#)
- [在 Ground Station 中使用与服务相关的角色](#)
- [静态数据加密 AWS Ground Station](#)
- [传输期间的数据加密 AWS Ground Station](#)

适用于 Identity and Access 管理 AWS Ground Station

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（拥有权限）使用 AWS Ground Station 资源。您可以使用 IAM AWS 服务，无需支付额外费用。

主题

- [受众](#)
- [使用身份进行身份验证](#)
- [使用策略管理访问](#)
- [如何 AWS Ground Station 与 IAM 配合使用](#)
- [基于身份的策略示例 AWS Ground Station](#)
- [对 AWS Ground Station 身份和访问进行故障排除](#)

受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（请参阅[对 AWS Ground Station 身份和访问进行故障排除](#)）
- 服务管理员：确定用户访问权限并提交权限请求（请参阅[如何 AWS Ground Station 与 IAM 配合使用](#)）
- IAM 管理员：编写用于管理访问权限的策略（请参阅[基于身份的策略示例 AWS Ground Station](#)）

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的[如何登录您的 AWS 账户](#)。

对于编程访问，AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的[适用于 API 请求的 AWS 签名版本 4](#)。

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。AWS IAM Identity Center 有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[什么是 IAM Identity Center？](#)。

IAM 用户和群组

[IAM 用户](#)是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的[要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问](#)。

[IAM 组](#)指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的[IAM 用户使用案例](#)。

IAM 角色

[IAM 角色](#)是具有特定权限的身份，可提供临时凭证。您可以通过[从用户切换到 IAM 角色（控制台）](#)或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的[担任角色的方法](#)。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的[JSON 策略概述](#)。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管策略与内联策略之间进行选择](#)。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中[指定主体](#)。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的[服务控制策略](#)。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的[资源控制策略 \(RCPs\)](#)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的[会话策略](#)。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的[策略评估逻辑](#)。

如何 AWS Ground Station 与 IAM 配合使用

在使用 IAM 管理访问权限之前 AWS Ground Station，请先了解哪些可用的 IAM 功能 AWS Ground Station。

您可以搭配使用的 IAM 功能 AWS Ground Station

IAM 功能	AWS Ground Station 支持
基于身份的策略	是

IAM 功能	AWS Ground Station 支持
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键 (特定于服务)	是
ACLs	否
ABAC (策略中的标签)	是
临时凭证	是
主体权限	是
服务角色	否
服务关联角色	是

要全面了解 AWS Ground Station 以及其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM [配合使用的AWS 服务](#)。

基于身份的策略 AWS Ground Station

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素引用](#)。

基于身份的策略示例 AWS Ground Station

要查看 AWS Ground Station 基于身份的策略的示例，请参阅。[基于身份的策略示例 AWS Ground Station](#)

内部基于资源的政策 AWS Ground Station

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

的政策行动 AWS Ground Station

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看 AWS Ground Station 操作列表，请参阅《服务授权参考》AWS Ground Station 中[定义的操作](#)。

正在执行的策略操作在操作前 AWS Ground Station 使用以下前缀：

```
groundstation
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

要查看 AWS Ground Station 基于身份的策略的示例，请参阅。[基于身份的策略示例 AWS Ground Station](#)

的政策资源 AWS Ground Station

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

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

要查看 AWS Ground Station 资源类型及其列表 ARNs，请参阅《服务授权参考》 [AWS Ground Station 中定义的资源](#)。要了解可以在哪些操作中指定每个资源的 ARN，请参阅 [AWS Ground Station 定义的操作](#)。

要查看 AWS Ground Station 基于身份的策略的示例，请参阅。 [基于身份的策略示例 AWS Ground Station](#)

的策略条件密钥 AWS Ground Station

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用 [条件运算符](#) (例如，等于或小于) 的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的 [AWS 全局条件上下文密钥](#)。

要查看 AWS Ground Station 条件键列表，请参阅《服务授权参考》 AWS Ground Station 中的 [条件密钥](#)。要了解可以使用条件键的操作和资源，请参阅 [由定义的操作 AWS Ground Station](#)。

要查看 AWS Ground Station 基于身份的策略的示例，请参阅。 [基于身份的策略示例 AWS Ground Station](#)

ACLs in AWS Ground Station

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人 (账户成员、用户或角色) 有权访问资源。ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

ABAC with AWS Ground Station

支持 ABAC (策略中的标签) : 是

基于属性的访问权限控制 (ABAC) 是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的 [条件元素](#) 中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的 [使用 ABAC 授权定义权限](#)。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的 [使用基于属性的访问权限控制 \(ABAC \)](#)。

将临时证书与 AWS Ground Station

支持临时凭证 : 是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 中的临时安全凭证](#) 和 [使用 IAM 的 AWS 服务](#)

的跨服务主体权限 AWS Ground Station

支持转发访问会话 (FAS) : 是

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅 [转发访问会话](#)。

的服务角色 AWS Ground Station

支持服务角色 : 否

服务角色是由一项服务担任、代表您执行操作的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的 [创建向 AWS 服务委派权限的角色](#)。

 Warning

更改服务角色的权限可能会中断 AWS Ground Station 功能。只有在 AWS Ground Station 提供操作指导时才编辑服务角色。

的服务相关角色 AWS Ground Station

支持服务关联角色：是

服务相关角色是一种链接到的服务角色。AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅[能够与 IAM 搭配使用的AWS 服务](#)。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

基于身份的策略示例 AWS Ground Station

默认情况下，用户和角色没有创建或修改 AWS Ground Station 资源的权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的[创建 IAM 策略（控制台）](#)。

有关由 AWS Ground Station 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》AWS Ground Station 中的[操作、资源和条件密钥](#)。ARNs

主题

- [策略最佳实践](#)
- [使用控制 AWS Ground Station 台](#)
- [允许用户查看他们自己的权限](#)

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS Ground Station 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#) 或 [工作职能的 AWS 托管策略](#)。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的策略和权限](#)。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 [使用 IAM Access Analyzer 验证策略](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 [使用 MFA 保护 API 访问](#)。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实践](#)。

使用控制 AWS Ground Station 台

要访问 AWS Ground Station 控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您的 AWS Ground Station 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用 AWS Ground Station 控制台，还需要将 AWS Ground Station *ConsoleAccess* 或 *ReadOnly* AWS 托管策略附加到实体。有关更多信息，请参阅《IAM 用户指南》中的 [为用户添加权限](#)。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
  "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": "*"
    }
  ]
}
```

对 AWS Ground Station 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS Ground Station 和 IAM 时可能遇到的常见问题。

主题

- [我无权在以下位置执行操作 AWS Ground Station](#)
- [我无权执行 iam : PassRole](#)

- [我想允许我以外的人 AWS 账户 访问我的 AWS Ground Station 资源](#)

我无权在以下位置执行操作 AWS Ground Station

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 groundstation:*GetWidget* 权限时，会发生以下示例错误。

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

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 groundstation:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam : PassRole

如果您收到一个错误，表明您无权执行 iam:PassRole 操作，则必须更新策略以允许您将角色传递给。AWS Ground Station

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 AWS Ground Station 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人 AWS 账户 访问我的 AWS Ground Station 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解是否 AWS Ground Station 支持这些功能，请参阅[如何 AWS Ground Station 与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅[IAM 用户指南中的向您拥有 AWS 账户的另一个 IAM 用户提供访问权限](#)。
- 要了解如何向第三方提供对您的资源的访问[权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限](#)。AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的[为经过外部身份验证的用户（身份联合验证）提供访问权限](#)。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

AWS 的托管策略 AWS Ground Station

AWS 托管策略是由创建和管理的独立策略 AWS。AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住，AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于使用案例的[客户管理型策略](#)来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的[AWS 托管式策略](#)。

AWS 托管策略：AWSGroundStationAgentInstancePolicy

您可以将 AWSGroundStationAgentInstancePolicy 策略附加到 IAM 身份。

该策略向您的 Amazon EC2 实例授予 AWS Ground Station 代理权限，允许该实例在 Ground Station 联系期间发送和接收数据。本策略中的所有权限均来自 Ground Station 服务。

权限详细信息

该策略包含以下权限。

- `groundstation`— 允许数据流端点实例调用 Ground Station 代理。 APIs

要查看最新版本的 JSON 策略文档，请参阅 [AWSGroundStationAgentInstancePolicy](#) AWS 托管策略参考指南。

AWS 托管策略：

`AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy`

您无法附加 `AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy` 到您的 IAM 实体。此策略附加到允许代表您执行操作 AWS Ground Station 的服务相关角色。有关更多信息，请参阅 [使用服务相关角色](#)。

此策略向 EC2 授予 AWS Ground Station 允许查找公有 IPv4 地址的权限。

权限详细信息

该策略包含以下权限。

- `ec2:DescribeAddresses`— AWS Ground Station 允许代表您列出所有与 EIPs 之 IPs 关联的内容。
- `ec2:DescribeNetworkInterfaces`— AWS Ground Station 允许代表您获取与 EC2 实例关联的网络接口的信息。

要查看最新版本的 JSON 策略文档，请参阅

[AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy](#) AWS 托管策略参考指南。

AWS Ground Station AWS 托管策略的更新

查看 AWS Ground Station 自该服务开始跟踪这些更改以来 AWS 托管策略更新的详细信息。要获得有关此页面变更的自动提醒，请订阅“AWS Ground Station 文档历史记录”页面上的 RSS feed。

更改	描述	日期
AWSGroundStationAgentInstancePolicy ：对现有策略的更新	AWS Ground Station 添加了新的权限，允许代理检索任务响应，从而增强了 Ground Station URLs 的操作。	2025 年 11 月 13 日
AWSGroundStationAgentInstancePolicy ：新策略	AWS Ground Station 添加了一项新策略，为数据流终端节点实例提供使用 AWS Ground Station Agent 的权限。	2023 年 4 月 12 日
AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy ：新策略	AWS Ground Station 添加了一项新策略，该策略授予 EC2 权限，AWS Ground Station 允许其查找与 EIP 关联的公有 IPv4 地址以及与 EC2 实例关联的网络接口。	2022 年 11 月 2 日
AWS Ground Station 已开始跟踪更改	AWS Ground Station 已开始跟踪 AWS 托管策略的更改。	2021 年 3 月 1 日

在 Ground Station 中使用与服务相关的角色

AWS Ground Station 使用 AWS Identity and Access Management (IAM) [服务相关角色](#)。服务相关角色是一种独特类型的 IAM 角色，它与 Ground Station 直接相关。服务相关角色由 Ground Station 预定义，包括该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色可让您更轻松地了解 Ground Station，因为您无需手动添加必要权限。Ground Station 定义其服务相关角色的权限，除非另有定义，否则仅 Ground Station 可以承担该角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其他 IAM 实体。

有关支持服务相关角色的其他服务的信息，请参阅与 [IAM 配合使用的AWS 服务](#)，并在服务相关角色列中查找标有“是”的服务。选择是和链接，查看该服务的服务关联角色文档。

Ground Station 的服务相关角色权限

Ground Station 使用名为的服务相关角色

AWSServiceRoleForGroundStationDataflowEndpointGroup— AWS GroundStation 使用此服务相关角色来调用 EC2 来查找公 IPv4 有地址。

AWSServiceRoleForGroundStationDataflowEndpointGroup 服务相关角色信任以下服务来代入该角色：

- `groundstation.amazonaws.com`

名为的角色权限策略 `AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy` 允许 Ground Station 对指定资源完成以下操作：

- 操作：`all AWS resources (*)` 上的 `ec2:DescribeAddresses`

Action 允许 Ground Station 列出与之 IPs 关联的所有 EIPs 内容。

- 操作：`all AWS resources (*)` 上的 `ec2:DescribeNetworkInterfaces`

操作允许 Ground Station 获取与 EC2 实例关联的网络接口信息

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。有关更多信息，请参阅《IAM 用户指南》中的[服务关联角色权限](#)。

为 Ground Station 创建服务相关角色

您无需手动创建服务关联角色。当您在 AWS CLI 或 AWS API `DataflowEndpointGroup` 中创建时，Ground Station 会为您创建服务相关角色。

如果您删除该服务关联角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您创建时 `DataflowEndpointGroup`，Ground Station 会再次为您创建服务相关角色。

您也可以使用 IAM 控制台为数据传输至 Amazon EC2 使用案例创建服务相关角色。在 AWS CLI 或 AWS API 中，使用服务名称创建服务相关角色。`groundstation.amazonaws.com` 有关更多信息，请参阅 IAM 用户指南 中的[创建服务相关角色](#)。如果您删除了此服务相关角色，可以使用同样的过程再次创建角色。

为 Ground Station 编辑服务相关角色

Ground Station 不允许您编辑 `AWSServiceRoleForGroundStationDataflowEndpointGroup` 服务相关角色。创建服务关联角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务关联角色](#)。

为 Ground Station 删除服务相关角色

如果不再需要使用某个需要服务关联角色的功能或服务，我们建议您删除该角色。这样就没有未被主动监控或维护的未使用实体。

只有在首先 `DataflowEndpointGroups` 使用服务相关角色删除服务相关角色后，才能删除该角色。这可以保护您免于无意中撤销对您的权限。`DataflowEndpointGroups` 如果服务相关角色与多个角色一起使用 `DataflowEndpointGroups`，则必须先删除所有使用 `DataflowEndpointGroups` 该服务相关角色的角色，然后才能将其删除。

Note

如果在您尝试删除资源时 Ground Station 服务正在使用该角色，则删除可能会失败。如果发生这种情况，请等待几分钟后重试。

删除 Ground Station 使用的 Ground Station 资源 `AWSServiceRoleForGroundStationDataflowEndpointGroup`

- `DataflowEndpointGroups` 通过 AWS CLI 或 AWS API 删除。

使用 IAM 手动删除服务关联角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除

`AWSServiceRoleForGroundStationDataflowEndpointGroup` 服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[删除服务关联角色](#)。

Ground Station 服务相关角色的受支持区域

Ground Station 支持在服务可用的所有区域使用服务相关角色。有关更多信息，请参阅[区域表](#)。

问题排查

NOT_AUTHORIZED_TO_CREATE_SLR-这表示您的账户中用于调用 CreateDataflowEndpointGroup API 的角色没有 iam:CreateServiceLinkedRole 权限。具有 iam:CreateServiceLinkedRole 权限的管理员须为您的账户手动创建服务相关角色。

静态数据加密 AWS Ground Station

AWS Ground Station 默认提供加密，以使用 AWS 自有的加密密钥保护您的静态敏感数据。

- **AWS 自有密钥**-默认 AWS Ground Station 使用这些密钥自动加密可直接识别的个人数据和星历表。您无法查看、管理或使用 AWS 自有密钥，也无法审核其使用情况；但是，没有必要采取任何措施或更改程序来保护加密数据的密钥。有关更多信息，请参阅《[AWS Key Management Service 开发者指南](#)》中的[AWS 自有密钥](#)。

默认情况下，静态数据加密可帮助降低保护敏感数据时涉及的操作开销和复杂性。同时，它还支持构建符合严格加密合规性以及监管要求的安全应用程序。

AWS Ground Station 对所有敏感的静态数据强制加密，但是，对于某些 AWS Ground Station 资源（例如星历表），您可以选择使用客户托管密钥代替默认的托管密钥。AWS

- **客户托管密钥**-- AWS Ground Station 支持使用由您创建、AWS 拥有和管理的对称客户托管密钥来代替现有的自有加密。由于您可以完全控制这层加密，因此可以执行以下任务：
 - 制定和维护关键策略
 - 建立和维护 IAM 策略和授权
 - 启用和禁用密钥策略
 - 轮换加密材料
 - 添加 标签
 - 创建密钥别名
 - 计划删除密钥

有关更多信息，请参阅 [AWS Key Management Service 开发人员指南](#) 中的 [客户托管密钥](#)。

下表汇总了 AWS Ground Station 支持使用客户托管密钥的资源

数据类型	AWS 自有密钥加密	客户托管密钥加密 (可选)
用于计算卫星轨迹的星历数据	已启用	已启用
用于指挥天线的方位角仰角星历	已启用	已启用

Note

AWS Ground Station 自动启用静态加密 AWS 拥有的密钥，用于免费保护个人身份数据。但是，使用客户管理的密钥需要 AWS KMS 付费。有关定价的更多信息，请参阅 [AWS Key Management Service 定价](#)。

有关的更多信息 AWS KMS，请参阅 [《AWS Key Management Service 开发人员指南》](#)。

有关每种资源类型的特定信息，请参阅：

- [TLE 和 OEM 星历数据的静态加密](#)
- [方位角高程星历的静态加密](#)

创建客户托管密钥

您可以使用 AWS 管理控制台、或，创建对称的客户托管密钥。AWS KMS APIs

创建对称的客户托管密钥

按照 [《AWS Key Management Service 开发人员指南》](#) 中创建对称客户托管密钥的步骤进行操作。

关键政策概述

密钥政策控制对客户托管密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管式密钥时，可以指定密钥策略。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的 [管理客户托管密钥的访问权限](#)。

要将客户托管密钥与 AWS Ground Station 资源一起使用，您必须配置密钥策略以向 AWS Ground Station 服务授予适当的权限。具体的权限和策略配置取决于您要加密的资源类型：

- 有关 TLE 和 OEM 星历数据-有关具体[TLE 和 OEM 星历数据的静态加密](#)的关键政策要求和示例，请参阅。
- 有关方位角高程星历数据-有关具体的关键策略要求和示例[方位角高程星历的静态加密](#)，请参阅。

Note

密钥策略配置因星历类型而异。TLE 和 OEM 星历表数据使用密钥访问权限，而方位角提升星历使用直接密钥策略权限。确保根据要加密的特定资源类型配置密钥策略。

有关在[策略中指定权限](#)以及对[密钥访问进行故障排除](#)的更多信息，请参阅《AWS Key Management Service 开发者指南》。

为指定客户管理的密钥 AWS Ground Station

您可以指定客户托管密钥以加密以下资源：

- 星历（TLE、OEM 和方位角仰角）

创建资源时，可以通过提供一个来指定数据密钥 kmsKeyArn

- kmsKeyArn- AWS KMS 客户托管[密钥的密钥标识符](#)

AWS Ground Station 加密上下文

[加密上下文](#)是一组可选的键值对，其中包含有关数据的其他上下文信息。AWS KMS 使用加密上下文作为其他经过身份验证的数据来支持经过身份验证的加密。当您在加密数据的请求中包含加密上下文时，会将加密上下文 AWS KMS 绑定到加密数据。要解密数据，您必须在请求中包含相同的加密上下文。

AWS Ground Station 根据要加密的资源使用不同的加密上下文，并为创建的每个密钥授权指定特定的加密上下文。

有关特定于资源的加密上下文的详细信息，请参阅：

- [TLE 和 OEM 星历数据的静态加密](#)
- [方位角高程星历的静态加密](#)

TLE 和 OEM 星历数据的静态加密

TLE 和 OEM 星历表的关键政策要求

要将客户托管密钥与星历数据一起使用，您的密钥策略必须向该服务授予以下权限：AWS Ground Station

- [kms:CreateGrant](#)-为客户管理的密钥创建访问授权。[授予对客户托管密钥执行授权操作](#)的 AWS Ground Station 访问权限，以读取和存储加密数据。
- [kms:DescribeKey](#)-提供客户管理的密钥详细信息，AWS Ground Station 以便在尝试使用提供的密钥之前验证密钥。

有关[使用授权](#)的更多信息，请参阅 AWS Key Management Service 开发者指南。

使用客户托管密钥创建星历的 IAM 用户权限

在加密操作中 AWS Ground Station 使用客户托管密钥时，它代表创建星历表资源的用户行事。

要使用客户托管密钥创建星历资源，用户必须有权对客户托管密钥进行以下操作：

- [kms:CreateGrant](#)-允许用户代表对客户管理的密钥创建授权 AWS Ground Station。
- [kms:DescribeKey](#)-允许用户查看客户管理的密钥详细信息以验证密钥。

您可以在密钥策略中指定这些所需权限，或者在密钥策略允许的情况下在 IAM 策略中指定这些权限。这些权限确保用户可以授权 AWS Ground Station 他们使用客户托管密钥进行加密操作。

如何在 AWS Ground Station 星历中 AWS KMS 使用赠款

AWS Ground Station 需要[密钥授予](#)才能使用您的客户管理的密钥。

当您上传使用客户托管密钥加密的星历时，AWS Ground Station 会通过向发送请求来代表您创建密钥授予。[CreateGrant](#) AWS KMS 中的授权 AWS KMS 用于授予对您账户中 AWS KMS 密钥的 AWS Ground Station 访问权限。

这 AWS Ground Station 允许执行以下操作：

- 调用 [GenerateDataKey](#) 生成加密的数据密钥并将其存储，因为数据密钥不会立即用于加密。
- 调用 [Decrypt](#) 使用存储的加密数据密钥访问加密数据。
- 调用 [Encrypt](#) 使用数据密钥加密数据。

- 设置停用主体，以允许服务 [RetireGrant](#)。

您可以随时撤消对授权的访问权限。如果这样做，将 AWS Ground Station 无法访问由客户托管密钥加密的任何数据，这会影响依赖该数据的操作。例如，如果您从当前用于联系人的星历中删除密钥授权，则在接触期间 AWS Ground Station 将无法使用提供的星历数据来指向天线。这将导致该联系以“失败”状态结束。

星历加密上下文

加密星历资源的密钥授权绑定到特定的卫星 ARN。

```
"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

密钥授权可重复用于同一密钥卫星。

使用加密上下文进行监控

使用对称的客户托管密钥来加密您的星历表时，您还可以使用审计记录和日志中的加密上下文来识别客户托管密钥的使用情况。加密上下文还会显示在[AWS CloudTrail 或 Amazon Logs 生成的 CloudWatch 日志](#)中。

使用加密上下文控制对客户托管式密钥的访问

您可以使用密钥策略和 IAM 策略中的加密上下文作为 `conditions` 来控制对您的对称客户托管密钥的访问。您还可以在授权中使用加密上下文约束。

AWS Ground Station 在授权中使用加密上下文约束来控制对您的账户或区域中客户托管密钥的访问权限。授权约束要求授权允许的操作使用指定的加密上下文。

以下是密钥策略声明示例，用于授予对特定加密上下文的客户托管密钥的访问权限。此策略语句中的条件要求授权具有指定加密上下文的加密上下文约束。

以下示例显示了绑定到卫星的星历数据的关键策略：

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"
        }
      }
    }
  ]
}
```

监控您的加密密钥中是否有星历

当您将 AWS Key Management Service 客户托管密钥与您的星历资源一起使用时，您可以使用或 [AWS CloudTrail](#) [Amazon CloudWatch 日志](#) 来跟踪发送到的 AWS Ground Station 请求。AWS KMS 以下示例是 [CreateGrant](#)、[GenerateDataKey](#)、[Decrypt](#) 和 [DescribeKey](#) 监视 AWS KMS 操作 CloudTrail 的事件，这些操作被调用 AWS Ground Station 以访问由您的客户托管密钥加密的数据。

CreateGrant

当您使用 AWS KMS 客户管理的密钥加密您的星历表资源时，AWS Ground Station 会代表您发送访问您账户中 AWS KMS 密钥的 [CreateGrant](#) 请求。AWS Ground Station 创建的授权特定于与 AWS KMS 客户托管密钥关联的资源。此外，在您删除资源时，AWS Ground Station 使用 [RetireGrant](#) 操作来移除授权。

以下示例事件记录了 [CreateGrant](#) 星历的操作：

```
{
  "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

当您使用 AWS KMS 客户托管密钥加密您的星历表资源时，AWS Ground Station 会代表您发送 [DescribeKey](#) 请求的密钥以验证您的账户中是否存在所请求的密钥。

以下示例事件记录了 [DescribeKey](#) 操作：

```

{
  "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

当您使用 AWS KMS 客户托管密钥加密您的星历资源时，AWS Ground Station 会向发送 [GenerateDataKey](#) 请求以生成用于加密数据的数据密钥。

以下示例事件记录了 [GenerateDataKey](#) 星历的操作：

```

{
  "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

当您使用 AWS KMS 客户托管密钥加密您的星历资源时，如果提供的星历已 AWS Ground Station 使用相同的客户托管[密钥加密](#)，则使用解密操作来解密所提供的星历表。例如，如果从 S3 存储桶上传星历并使用给定密钥对该桶中星历进行加密。

以下示例事件记录了星历的[解密](#)操作：

```

{
  "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"
}
```

方位角高程星历的静态加密

方位角高程星历表的关键政策要求

要将客户托管密钥与方位角提升星历表数据一起使用，您的密钥策略必须向该服务授予以下权限。AWS Ground Station 与使用授权的 TLE 和 OEM 星历表数据不同，方位角提升星历使用直接密钥策略权限进行加密操作。这是一种管理权限和使用密钥的更简单方法。

- [kms:GenerateDataKey](#)-生成用于加密方位角高程星历数据的数据密钥。
- [kms:Decrypt](#)-在访问您的方位角高程星历表数据时解密加密的数据密钥。

授予客户托管密钥 AWS Ground Station 访问权限的密钥策略示例

Note

使用方位角提升星历表，您必须直接在密钥策略中配置这些权限。必须在您的关键政策声明中向区域 AWS Ground Station 服务委托人（例如 `groundstation.region.amazonaws.com`）授予这些权限。如果不 AWS Ground Station 将这些声明添加到密钥策略中，则无法存储或访问您的自定义方位角高程星历表。

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": "*"
    }
  ]
}
```

使用客户托管密钥创建方位角提升星历的 IAM 用户权限

在加密操作中 AWS Ground Station 使用客户托管密钥时，它代表正在创建方位角高程星历表资源的用户行事。

要使用客户托管密钥创建方位角提升星历表资源，用户必须有权对客户托管密钥进行以下操作：

- [kms:GenerateDataKey](#)-允许用户生成用于加密方位角高程星历数据的数据密钥。
- [kms:Decrypt](#)-允许用户在访问方位角高程星历表数据时解密数据密钥。
- [kms:DescribeKey](#)-允许用户查看客户管理的密钥详细信息以验证密钥。

您可以在密钥策略中指定这些所需权限，或者在密钥策略允许的情况下在 IAM 策略中指定这些权限。这些权限确保用户可以授权 AWS Ground Station 他们使用客户托管密钥进行加密操作。

如何 AWS Ground Station 使用关键策略制作方位角高程星历

当您为客户管理的密钥提供方位角高程星历表数据时，会 AWS Ground Station 使用密钥策略来访问您的加密密钥。权限是 AWS Ground Station 通过关键政策声明直接授予的，而不是像 TLE 或 OEM 星历数据那样通过授权授予的。

如果您删除 AWS Ground Station 对客户托管密钥的访问权限，将 AWS Ground Station 无法访问由该密钥加密的任何数据，这会影响依赖该数据的操作。例如，如果您删除了联系人当前使用的方位角高程星历表的关键策略权限，则在接触期间 AWS Ground Station 将无法使用提供的方位角高程数据来控制天线。这将导致该联系以“失败”状态结束。

方位角高程星历加密上下文

[当 AWS Ground Station 使用您的 AWS KMS 密钥加密方位角高程星历表数据时，该服务会指定加密上下文。](#)加密上下文是 AWS KMS 用于确保数据完整性的其他经过身份验证的数据 (AAD)。在为加密操作指定加密上下文时，该服务必须为解密操作指定同一加密上下文。否则，解密将失败。加密上下文也会写入您的 CloudTrail 日志，以帮助您了解为什么使用给定 AWS KMS 密钥。您的 CloudTrail 日志可能包含许多描述 AWS KMS 密钥使用的条目，但是每个日志条目中的加密上下文可以帮助您确定该特定使用的原因。

AWS Ground Station 当它使用您的客户托管密钥在方位角提升星历表上执行加密操作时，指定以下加密上下文：

```
{
  "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"
  }
}
```

加密上下文包含：

`aws:groundstation:ground-station-id`

与方位角高程星历表相关的地面站名称。

aws: 地面站:arn

星历资源的 ARN。

aws: s3: arn

存储在 Amazon S3 中的星历的 ARN。

使用加密上下文控制对客户托管密钥的访问

您可以使用 IAM 条件声明来控制对客户托管密钥的 AWS Ground Station 访问权限。

在kms:GenerateDataKey和kms:Decrypt操作上添加条件声明会限制 a AWS KMS 可以用于哪些地面站。

以下是密钥政策声明示例，用于授予客户在特定区域对特定地面站的托管密钥的 AWS Ground Station 访问权限。此策略声明中的条件要求所有人对指定与密钥策略中的条件匹配的加密上下文的密钥进行加密和解密访问。

授予 AWS Ground Station 访问客户管理的特定地面站密钥的密钥策略示例

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"
        }
    }
}
]
}

```

允许 AWS Ground Station 访问多个地面站的客户托管密钥的密钥策略示例

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"
    ]
    }
    }
    }
    ]
    }

```

监控加密密钥中的方位角高程星历

当您将 AWS KMS 客户托管密钥与方位角提升星历表资源一起使用时，您可以使用 [CloudTrail](#) 或 [CloudWatch 日志](#) 来跟踪发送到的请求。AWS Ground Station AWS KMS 以下示例是 [GenerateDataKey](#) 和 [Decrypt CloudTrail](#) 的事件，用于监控访问由 AWS Ground Station 您的客户托管密钥加密的数据而调用的 AWS KMS 操作。

GenerateDataKey

当您使用 AWS KMS 客户托管密钥加密您的方位角高程星历表资源时，AWS Ground Station 会向发送 [GenerateDataKey](#) 请求 AWS KMS 以生成用于加密数据的数据密钥。

以下示例事件记录了方位角高程星历的 [GenerateDataKey](#) 操作：

```

{
  "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

当您使用 AWS KMS 客户托管密钥加密您的方位角高程星历表资源时，如果提供的方位角高程星历表数据已经 AWS Ground Station 使用相同的客户管理[密钥进行加密](#)，则使用解密操作来解密这些数据。

以下示例事件记录了方位角高程星历的[解密](#)操作：

```
{
  "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"
  }
}

```

传输期间的数据加密 AWS Ground Station

AWS Ground Station 默认提供加密功能，以便在传输过程中保护您的敏感数据。根据任务配置文件配置，可以通过两种方式在 AWS Ground Station 天线位置和您的 Amazon EC2 实例之间传输数据。

- AWS Ground Station 代理人
- 数据流端点

每种流式传输数据方法对传输中数据的加密处理方式都不一样。下文将介绍每种方法。

AWS Ground Station 代理直播

AWS Ground Station 代理使用客户管理的 AWS KMS 密钥对其直播进行加密。在您的 Amazon EC2 实例上运行的 AWS Ground Station 代理将自动解密流以提供解密后的数据。

用于加密直播的 AWS KMS 密钥是在 [streamsKmsKey](#) 参数 `MissionProfile` 中创建时指定的。所有授予密钥 AWS Ground Station 访问权限的权限均通过所附的 AWS KMS 密钥策略进行处理 `streamsKmsKey`。

数据流端点流

数据流端点流使用 [数据报传输层安全性 \(DTLS\)](#) 进行加密。这是使用自签名证书完成的，不需要额外的配置。

任务配置文件配置示例

所提供的示例说明了如何使用公共广播卫星并创建支持该卫星的任务概况。生成的模板旨在帮助您与公共广播卫星进行联系，并帮助您做出有关卫星的决定。

主题

- [JPSS-1-公共广播卫星 \(PBS\)-评估](#)
- [使用 Amazon S3 数据传输的公共广播卫星](#)
- [使用数据流端点 \(窄带\) 的公共广播卫星](#)
- [利用数据流端点 \(解调和解码\) 的公共广播卫星](#)
- [使用 AWS Ground Station 代理 \(宽带\) 的公共广播卫星](#)

JPSS-1-公共广播卫星 (PBS)-评估

此示例部分与[客户入职流程概述](#)。它提供了与 AWS Ground Station 以下具体示例的简短兼容性分析，并为其奠定了基础。

如[公共广播卫星](#)本节所述，您可以使用公开可用的特定卫星或卫星的通信路径。在本节中，我们用 AWS Ground Station 术语描述 [JPSS-1](#)。作为参考，我们使用[联合极地卫星系统 1 \(JPSS-1\) 航天器高速率数据 \(HRD\) 直接广播电台 \(DBS\) 射频 \(RF\) 接口控制文档 \(ICD\)](#) 来完成示例。另外，值得注意的是，JPSS-1 与 NORAD ID 43013 有关。

JPSS-1 卫星提供一条上行链路和三条直接下行链路通信路径，如 ICD 图 1-1 所示。在这四条通信路径中，只有一条高速率数据 (HRD) 下行链路通信路径可供公众使用。基于此，你会看到这条路径还将有更具体的数据与之相关联。四条路径如下所示：

- 命令路径 (上行链路)，MHz 中心频率为 2067.27，数据速率为 2-128 kbps。此路径不可公开访问。
- MHz 中心频率为 2247.5 的遥测路径 (下行链路)，数据速率为 1-524 kbps。此路径不可公开访问。
- GHz 中心频率为 26.7034 的 SMD 路径 (下行链路)，数据速率为 150-300 Mbps。此路径不可公开访问。
- HRD 路径 (下行链路) 的 RF，MHz 中心频率为 7812，数据速率为 15 Mbps。它的 MHz 带宽为 30，而且确实如此 right-hand-circular-polarized。当您搭载 JPSS-1 时 AWS Ground Station，这是

您可以访问的通信路径。该通信路径包含仪器科学数据、仪器工程数据、仪器遥测数据和实时航天器内部管理数据。

当我们比较潜在的数据路径时，我们发现命令（上行链路）、遥测（下行链路）和 HRD（下行链路）路径符合的频率、带宽和多通道并发使用能力。AWS Ground Station SMD 路径不兼容，因为中心频率超出了现有接收器的范围。有关支持的功能的更多信息，请参阅[AWS Ground Station 网站能力](#)。

Note

由于 SMD 路径与之不兼容 AWS Ground Station，因此示例配置中不会显示。

Note

由于命令（上行链路）和遥测（下行链路）路径未在 ICD 中定义，也不可供公众使用，因此使用时提供的值是名义值。

使用 Amazon S3 数据传输的公共广播卫星

此示例建立在用户指南[JPSS-1-公共广播卫星 \(PBS\)-评估](#)部分所做的分析的基础上。

在本示例中，你需要假设一个场景，即你想将 HRD 通信路径捕获为数字中频，并将其存储起来以备将来的批处理之用。这样可以节省数字化后的原始射频 (RF) 同相正交 (I/Q) 样本。将数据放入 Amazon S3 存储桶后，您可以使用所需的任何软件对数据进行解调和解码。有关处理的详细示例，请参阅[MathWorks 教程](#)。使用此示例后，您可以考虑添加 Amazon EC2 现货定价组件来处理数据并降低总体处理成本。

通信路径

本节介绍[规划您的数据流通信路径](#)入门。

以下所有模板片段都属于 CloudFormation 模板的“资源”部分。

Resources:

```
# Resources that you would like to create should be placed within the Resources section.
```

Note

有关 CloudFormation 模板内容的更多信息，请参阅[模板部分](#)。

考虑到我们向 Amazon S3 提供单一通信路径的场景，您知道您将拥有一条异步传输路径。根据本[异步数据传输](#)节，您必须定义一个 Amazon S3 存储桶。

```
# 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]]]]]]
```

此外，您还需要创建相应的角色和策略 AWS Ground Station 才能允许使用存储桶。

```
# 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 配置

本节介绍[创建配置](#)入门。

你需要一个跟踪配置来设置你使用自动追踪的偏好。选择 PREFERRED 作为自动跟踪可以提高信号质量，但由于 JPSS-1 星历质量足够，因此不需要满足信号质量。

```

TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"

```

根据通信路径，您需要定义一个天线下行链路配置来表示卫星部分，并定义一个 s3 录音以引用您刚刚创建的 Amazon S3 存储桶。

```
# 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 任务简介

本节介绍[创建任务档案](#)入门。

现在你已经有了相关的配置，你可以用它们来构造数据流。其余参数将使用默认值。

```
# 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
```

把它放在一起

利用上述资源，您现在可以安排 JPSS-1 联系人从任何已上线人员进行异步数据传输。AWS Ground Station [AWS Ground Station 位置](#)

以下是一个完整的 CloudFormation 模板，其中包括本节中描述的所有资源，这些资源组合成一个可以直接在中使用的模板 CloudFormation。

名为的 CloudFormation 模板 `AquaSnppJpss-1TerraDigIfS3DataDelivery.yml` 包含一个 Amazon S3 存储桶以及安排联系和接收 VITA-49 信号/IP 直接广播数据所需的 AWS Ground Station 资源。

如果你的账户未登录 Aqua、SNPP、JPSS-1/NOAA-20 和 Terra，请参阅 [机载卫星](#)

Note

您可以使用有效 AWS 凭证访问客户登录 Amazon S3 存储桶，从而访问模板。以下链接使用区域性 Amazon S3 存储桶。更改 `us-west-2` 区域代码以表示要在其中创建 CloudFormation 堆栈的相应区域。

此外，以下说明使用 YAML。但是，模板有 YAML 和 JSON 这两种格式。要使用 JSON，请在下载模板 `.json` 时将 `.yaml` 文件扩展名替换为。

要使用下载模板 AWS CLI，请使用以下命令：

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/
AquaSnppJpss-1TerraDigIfS3DataDelivery.yml .
```

在浏览器中导航到以下 URL，可以在控制台中查看和下载此模板：

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml
```

您可以使用以下链接直接在中 CloudFormation 指定模板：

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml
```

使用数据流端点（窄带）的公共广播卫星

此示例建立在用户指南[JPSS-1-公共广播卫星 \(PBS\)-评估](#)部分所做的分析基础上。

要完成此示例，您需要假设一个场景——您想将 HRD 通信路径捕获为数字中频 (DigiF)，并使用 SDR 在 Amazon EC2 实例上使用数据流终端节点应用程序接收到该路径时对其进行处理。

沟通路径

本节介绍[规划您的数据流通信路径](#)入门。在本示例中，您将在 CloudFormation 模板中创建两个部分：参数和资源部分。

Note

有关 CloudFormation 模板内容的更多信息，请参阅[模板部分](#)。

在“参数”部分，您将添加以下参数。在通过 CloudFormation 控制台创建堆栈时，您需要为这些值指定值。

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

您需要创建密钥对，并提供 Amazon EC2 EC2Key 参数的名称。请参阅[为您的 Amazon EC2 实例创建密钥对](#)。

此外，在创建 CloudFormation 堆栈时，您需要提供正确的特定区域的 AMI ID。请参阅[AWS Ground Station Amazon 机器映像 \(AMIs\)](#)。

其余的模板片段属于模板的“资源”部分。CloudFormation

Resources:

Resources that you would like to create should be placed within the resource section.

鉴于我们的场景是向 EC2 实例提供单一通信路径，您将拥有一条同步传输路径。根据该[同步数据传输](#)部分，您必须使用数据流终端节点应用程序设置和配置 Amazon EC2 实例，并创建一个或多个数据流终端节点组。

```
# The EC2 instance that will send/receive data to/from your satellite using AWS
Ground Station.
```

```
ReceiverInstance:
```

```
Type: AWS::EC2::Instance
```

```
Properties:
```

```
DisableApiTermination: false
```

```
IamInstanceProfile: !Ref GeneralInstanceProfile
```

```
ImageId: !Ref ReceiverAMI
```

```
InstanceType: m5.4xlarge
```

```
KeyName: !Ref EC2Key
```

```
Monitoring: true
```

```
PlacementGroupName: !Ref ClusterPlacementGroup
```

```
SecurityGroupIds:
```

```
- Ref: InstanceSecurityGroup
```

```
SubnetId: !Ref ReceiverSubnet
```

```

BlockDeviceMappings:
  - DeviceName: /dev/xvda
    Ebs:
      VolumeType: gp2
      VolumeSize: 40
Tags:
  - Key: Name
    Value: !Join [ "-", [ "Receiver" , !Ref "AWS::StackName" ] ]
UserData:
  Fn::Base64:
    |
    #!/bin/bash
    exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console)

2>&1

    echo `date +%F %R:%S` "INFO: Logging Setup" >&2

    GROUND_STATION_DIR="/opt/aws/groundstation"
    GROUND_STATION_BIN_DIR="${GROUND_STATION_DIR}/bin"
    STREAM_CONFIG_PATH="${GROUND_STATION_DIR}/customer_stream_config.json"

    echo "Creating ${STREAM_CONFIG_PATH}"
    cat << STREAM_CONFIG > "${STREAM_CONFIG_PATH}"
    {
      "ddx_streams": [
        {
          "streamName": "Downlink",
          "maximumWanRate": 4000000000,
          "lanConfigDevice": "lo",
          "lanConfigPort": 50000,
          "wanConfigDevice": "eth1",
          "wanConfigPort": 55888,
          "isUplink": false
        }
      ]
    }
    STREAM_CONFIG

    echo "Waiting for dataflow endpoint application to start"
    while netstat -lnt | awk '$4 ~ /:80$/ {exit 1}'; do sleep 10; done

    OS_VERSION_ID=$( [ -f /etc/os-release ] && . /etc/os-release && echo
"${VERSION_ID}")
    if [ "${OS_VERSION_ID}" = "2023" ]; then PYTHON_BIN="python3"; else
PYTHON_BIN="python"; fi

```

```

    echo "Configuring dataflow endpoint application streams"
    streams_configured=false
    for attempt in $(seq 1 6); do
        if ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/configure_streams.py" --
configFileName "${STREAM_CONFIG_PATH}"; then
            streams_configured=true
            break
        fi
        echo "configure_streams attempt ${attempt} failed; retrying in 10s"
        sleep 10
    done
    if [ "${streams_configured}" = "true" ]; then
        sleep 2
        ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/save_default_config.py"
    else
        echo "ERROR: configure_streams failed after ${attempt} attempts; not saving
a default config"
    fi

    exit 0

# The AWS Ground Station Dataflow Endpoint Group that defines the endpoints that AWS
Ground
# Station will use to send/receive data to/from your satellite.
DataflowEndpointGroup:
  Type: AWS::GroundStation::DataflowEndpointGroup
  Properties:
    ContactPostPassDurationSeconds: 180
    ContactPrePassDurationSeconds: 120
    EndpointDetails:
      - Endpoint:
          Name: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ] # needs to
match DataflowEndpointConfig name
          Address:
            Name: !GetAtt ReceiverInstanceNetworkInterface.PrimaryPrivateIpAddress
            Port: 55888
    SecurityDetails:
      SecurityGroupIds:
        - Ref: "DataflowEndpointSecurityGroup"
      SubnetIds:
        - !Ref ReceiverSubnet
      RoleArn: !GetAtt DataDeliveryServiceRole.Arn

```

```
# The security group for your EC2 instance.
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: AWS Ground Station receiver instance security group.
    VpcId: !Ref ReceiverVPC
    SecurityGroupIngress:
      # To allow SSH access to the instance, add another rule allowing tcp port 22
      # from your CidrIp
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        SourceSecurityGroupId: !Ref DataflowEndpointSecurityGroup
        Description: "AWS Ground Station Downlink Stream"

# The security group that the ENI created by AWS Ground Station belongs to.
DataflowEndpointSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security Group for AWS Ground Station registration of Dataflow
    Endpoint Groups
    VpcId: !Ref ReceiverVPC
    SecurityGroupEgress:
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        CidrIp: 10.0.0.0/8
        Description: "AWS Ground Station Downlink Stream To 10/8"
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        CidrIp: 172.16.0.0/12
        Description: "AWS Ground Station Downlink Stream To 172.16/12"
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        CidrIp: 192.168.0.0/16
        Description: "AWS Ground Station Downlink Stream To 192.168/16"

# The placement group in which your EC2 instance is placed.
ClusterPlacementGroup:
  Type: AWS::EC2::PlacementGroup
  Properties:
    Strategy: cluster
```

ReceiverVPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: "10.0.0.0/16"

Tags:

- Key: "Name"
Value: "AWS Ground Station - PBS to dataflow endpoint Example VPC"
- Key: "Description"
Value: "VPC for EC2 instance receiving AWS Ground Station data"

ReceiverSubnet:

Type: AWS::EC2::Subnet

Properties:

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

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

CidrBlock: "10.0.0.0/24"

Tags:

- Key: "Name"
Value: "AWS Ground Station - PBS to dataflow endpoint Example Subnet"
- Key: "Description"
Value: "Subnet for EC2 instance receiving AWS Ground Station data"

VpcId: !Ref ReceiverVPC

An ENI providing a fixed IP address for AWS Ground Station to connect to.

ReceiverInstanceNetworkInterface:

Type: AWS::EC2::NetworkInterface

Properties:

Description: Floating network interface providing a fixed IP address for AWS Ground Station to connect to.

GroupSet:

- !Ref InstanceSecurityGroup

SubnetId: !Ref ReceiverSubnet

Attach the ENI to the EC2 instance.

ReceiverInstanceInterfaceAttachment:

Type: AWS::EC2::NetworkInterfaceAttachment

Properties:

DeleteOnTermination: false

DeviceIndex: "1"

InstanceId: !Ref ReceiverInstance

NetworkInterfaceId: !Ref ReceiverInstanceNetworkInterface

此外，您还需要创建适当的策略和角色，AWS Ground Station 以允许在您的账户中创建弹性网络接口 (ENI)。

```
# 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 配置

本节介绍[创建配置](#)入门。

你需要一个跟踪配置来设置你对使用自动跟踪的偏好。选择“首选”作为自动跟踪可以提高信号质量，但由于星 JPSS-1 历质量足够高，因此不需要满足信号质量。

```

TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"

```

根据通信路径，您需要定义天线下行链路配置以表示卫星部分，以及数据流端点配置以引用定义端点详细信息的数据流端点组。

```

# 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 任务概况

本节介绍[创建任务档案](#)入门。

现在你已经有了相关的配置，你可以使用它们来构造数据流。您将使用其余参数的默认值。

```

# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnppJpssMissionProfile:
  Type: AWS::GroundStation::MissionProfile

```

Properties:

```
Name: "37849 SNPP And 43013 JPSS"
ContactPrePassDurationSeconds: 120
ContactPostPassDurationSeconds: 60
MinimumViableContactDurationSeconds: 180
TrackingConfigArn: !Ref TrackingConfig
DataflowEdges:
  - Source: !Ref SnpJPssDownlinkDigIfAntennaConfig
    Destination: !Ref DownlinkDigIfEndpointConfig
```

把它放在一起

有了上述资源，您现在可以安排 JPSS-1 联系人，以便从任何入职人员同步传输数据。AWS Ground Station [AWS Ground Station 位置](#)

以下是一个完整的 CloudFormation 模板，其中包括本节中描述的所有资源，这些资源组合成一个可以直接在中使用的单一模板 CloudFormation。

名为 AquaSnpJPssTerraDigIF.yml 的 CloudFormation 模板旨在让您快速访问以开始接收 Aqua、SNPP 和 Terra 卫星的数字化中频 (DigiF) 数据。JPSS-1/NOAA-20 它包含一个 Amazon EC2 实例和接收原始 DigiF 直接广播数据所需的 CloudFormation 资源。

如果 Aqua、SNPP 和 Terra 未加入您的账户，请参阅。JPSS-1/NOAA-20 [机载卫星](#)

Note

您可以使用有效的 AWS 凭证访问客户加入 Amazon S3 存储桶来访问该模板。以下链接使用区域性 Amazon S3 存储桶。更改 us-west-2 区域代码以表示您要在其中创建 CloudFormation 堆栈的相应区域。

此外，以下指令使用 YAML。但是，模板有 YAML 和 JSON 这两种格式。要使用 JSON，请在下载模板.json 时将 .yaml 文件扩展名替换为。

要使用下载模板 AWS CLI，请使用以下命令：

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/
AquaSnpJPssTerraDigIF.yml .
```

在浏览器中导航到以下 URL，可以在控制台中查看和下载此模板：

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpssTerraDigIF.yml
```

您可以使用以下链接直接在中 CloudFormation 指定模板：

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpssTerraDigIF.yml
```

该模板还定义了哪些其他资源？

该AquaSnppJpssTerraDigIF模板包括以下其他资源：

- (可选) CloudWatch 事件触发器 -使用联系人 AWS Ground Station 之前和之后发送 CloudWatch 的事件触发的 AWS Lambda 函数。该 AWS Lambda 函数将启动并有选择地停止您的接收器实例。
- (可选) 联络的 EC2 验证：使用 Lambda 为带有 SNS 通知的联络设置您的 Amazon EC2 实例的验证系统的选项。需要注意的是，这可能会产生费用，具体取决于您当前的使用情况。
- Ground Station 亚马逊机器映像检索 Lambda：用于选择您的实例中安装的软件以及您选择的 AMI 的选项。软件选项包括带或不带软件定义无线电的数据流端点应用程序。随着其他软件更新和功能的发布，这些选项可能会发生变化。
- 其他任务概况 ——其他公共广播卫星 (Aqua、SNPP和Terra) 的任务概况。
- 其他天线下行链路配置 -其他公共广播卫星 (Aqua、SNPP 和 Terra) 的天线下行链路配置。

已填充此模板中卫星的值和参数。这些参数使您可以轻松地 AWS Ground Station 立即使用这些卫星。使用此模板 AWS Ground Station 时，无需配置自己的值即可使用。但是，您可以自定义这些值以使模板适用于您的使用案例。

我可以在哪里接收我的数据？

数据流终端节点组设置为使用此模板的一部分创建的接收实例网络接口。接收器实例使用数据流端点应用程序从数据流端点定义 AWS Ground Station 的端口接收数据流。接收到数据后，可通过接收实例的环回适配器上的 UDP 端口 50000 使用数据。有关设置数据流终端节点组的更多信息，请参阅 [AWS::GroundStation::DataflowEndpointGroup](#)

利用数据流端点（解调和解码）的公共广播卫星

此示例建立在用户指南[JPSS-1-公共广播卫星 \(PBS\)-评估](#)部分所做的分析基础上。

要完成此示例，您需要假设一个场景——您想使用数据流端点将 HRD 通信路径捕获为解调和解码的直接广播数据。如果你计划使用美国宇航局直接读出实验室软件（RT-STPS 和 IPOPP）处理数据，那么这个例子是一个很好的起点。

沟通路径

本节介绍[规划您的数据流通信路径](#)入门。在本示例中，您将在 CloudFormation 模板中创建两个部分：参数和资源部分。

Note

有关 CloudFormation 模板内容的更多信息，请参阅[模板部分](#)。

在“参数”部分，您将添加以下参数。在通过 CloudFormation 控制台创建堆栈时，您需要为这些值指定值。

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

您需要创建密钥对，并提供 Amazon EC2 EC2Key 参数的名称。请参阅[为您的 Amazon EC2 实例创建密钥对](#)。

此外，在创建 CloudFormation 堆栈时，您需要提供正确的特定区域的 AMI ID。请参阅[AWS Ground Station Amazon 机器映像 \(AMIs\)](#)。

其余的模板片段属于模板的“资源”部分。CloudFormation

Resources:

Resources that you would like to create should be placed within the resource section.

鉴于我们的场景是向 EC2 实例提供单一通信路径，您将拥有一条同步传输路径。根据该[同步数据传输](#)部分，您必须使用数据流终端节点应用程序设置和配置 Amazon EC2 实例，并创建一个或多个数据流终端节点组。

```
# The EC2 instance that will send/receive data to/from your satellite using AWS
Ground Station.
ReceiverInstance:
  Type: AWS::EC2::Instance
  Properties:
    DisableApiTermination: false
    IamInstanceProfile: !Ref GeneralInstanceProfile
    ImageId: !Ref ReceiverAMI
    InstanceType: m5.4xlarge
    KeyName: !Ref EC2Key
    Monitoring: true
    PlacementGroupName: !Ref ClusterPlacementGroup
    SecurityGroupIds:
      - Ref: InstanceSecurityGroup
    SubnetId: !Ref ReceiverSubnet
    BlockDeviceMappings:
      - DeviceName: /dev/xvda
        Ebs:
          VolumeType: gp2
          VolumeSize: 40
    Tags:
      - Key: Name
        Value: !Join [ "-", [ "Receiver" , !Ref "AWS::StackName" ] ]
  UserData:
    Fn::Base64:
      |
      #!/bin/bash
      exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console)
2>&1
      echo `date +%F %R:%S` "INFO: Logging Setup" >&2
```

```

GROUND_STATION_DIR="/opt/aws/groundstation"
GROUND_STATION_BIN_DIR="${GROUND_STATION_DIR}/bin"
STREAM_CONFIG_PATH="${GROUND_STATION_DIR}/customer_stream_config.json"

echo "Creating ${STREAM_CONFIG_PATH}"
cat << STREAM_CONFIG > "${STREAM_CONFIG_PATH}"
{
  "ddx_streams": [
    {
      "streamName": "Downlink",
      "maximumWanRate": 4000000000,
      "lanConfigDevice": "lo",
      "lanConfigPort": 50000,
      "wanConfigDevice": "eth1",
      "wanConfigPort": 55888,
      "isUplink": false
    }
  ]
}
STREAM_CONFIG

echo "Waiting for dataflow endpoint application to start"
while netstat -lnt | awk '$4 ~ /:80$/ {exit 1}'; do sleep 10; done

OS_VERSION_ID=$( [ -f /etc/os-release ] && . /etc/os-release && echo
"${VERSION_ID}")
if [ "${OS_VERSION_ID}" = "2023" ]; then PYTHON_BIN="python3"; else
PYTHON_BIN="python"; fi

echo "Configuring dataflow endpoint application streams"
streams_configured=false
for attempt in $(seq 1 6); do
  if ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/configure_streams.py" --
configFileName "${STREAM_CONFIG_PATH}"; then
    streams_configured=true
    break
  fi
  echo "configure_streams attempt ${attempt} failed; retrying in 10s"
  sleep 10
done
if [ "${streams_configured}" = "true" ]; then
  sleep 2
  ${PYTHON_BIN} "${GROUND_STATION_BIN_DIR}/save_default_config.py"

```

```

        else
            echo "ERROR: configure_streams failed after ${attempt} attempts; not saving
a default config"
        fi

        exit 0

```

```

# The AWS Ground Station Dataflow Endpoint Group that defines the endpoints that AWS
Ground
# Station will use to send/receive data to/from your satellite.
DataflowEndpointGroup:
  Type: AWS::GroundStation::DataflowEndpointGroup
  Properties:
    ContactPostPassDurationSeconds: 180
    ContactPrePassDurationSeconds: 120
    EndpointDetails:
      - Endpoint:
          Name: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ] # needs to
match DataflowEndpointConfig name
          Address:
            Name: !GetAtt ReceiverInstanceNetworkInterface.PrimaryPrivateIpAddress
            Port: 55888
    SecurityDetails:
      SecurityGroupIds:
        - Ref: "DataflowEndpointSecurityGroup"
      SubnetIds:
        - !Ref ReceiverSubnet
      RoleArn: !GetAtt DataDeliveryServiceRole.Arn

# The security group that the ENI created by AWS Ground Station belongs to.
DataflowEndpointSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security Group for AWS Ground Station registration of Dataflow
Endpoint Groups
    VpcId: !Ref ReceiverVPC
    SecurityGroupEgress:
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        CidrIp: 10.0.0.0/8
        Description: "AWS Ground Station Downlink Stream To 10/8"

```

```

- IpProtocol: udp
  FromPort: 55888
  ToPort: 55888
  CidrIp: 172.16.0.0/12
  Description: "AWS Ground Station Downlink Stream To 172.16/12"
- IpProtocol: udp
  FromPort: 55888
  ToPort: 55888
  CidrIp: 192.168.0.0/16
  Description: "AWS Ground Station Downlink Stream To 192.168/16"

# The placement group in which your EC2 instance is placed.
ClusterPlacementGroup:
  Type: AWS::EC2::PlacementGroup
  Properties:
    Strategy: cluster

# The security group for your EC2 instance.
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: AWS Ground Station receiver instance security group.
    VpcId: !Ref ReceiverVPC
    SecurityGroupIngress:
      # To allow SSH access to the instance, add another rule allowing tcp port 22
      # from your CidrIp
      - IpProtocol: 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

```

您还需要适当的策略、角色和配置文件，AWS Ground Station 以允许在您的账户中创建弹性网络接口 (ENI)。

```
# 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 配置

本节[创建配置](#)代表用户指南。

你需要一个跟踪配置来设置你对使用自动跟踪的偏好。选择“首选”作为自动跟踪可以提高信号质量，但由于星 JPSS-1 历质量足够高，因此不需要满足信号质量。

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"
```

根据通信路径，您需要定义天线下行解调解码配置以表示卫星部分，以及数据流端点配置以引用定义端点详细信息的数据流端点组。

Note

有关如何为和设置值的DemodulationConfig详细信息DecodeConfig，请参阅[天线下行传输解调解码配置](#)。

```
# 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 任务概况

本节[创建任务档案](#)代表用户指南。

现在你已经有了相关的配置，你可以使用它们来构造数据流。您将使用其余参数的默认值。

```

# 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
```

把它放在一起

有了上述资源，您现在可以安排 JPSS-1 联系人，以便从任何入职人员同步传输数据。AWS Ground Station [AWS Ground Station 位置](#)

以下是一个完整的 CloudFormation 模板，其中包括本节中描述的所有资源，这些资源组合成一个可以直接在中使用的单一模板 CloudFormation。

名为 AquaSnppJpss.yml 的 CloudFormation 模板旨在让您快速访问以开始接收 Aqua、SNPP 和 JPSS-1/NOAA-20 卫星的数据。它包含 Amazon EC2 实例和安排联系以及接收解调和解码后的直接广播数据所需的 AWS Ground Station 资源。

如果 Aqua、SNPP 和 Terra 未加入您的账户，请参阅。JPSS-1/NOAA-20 [机载卫星](#)

Note

您可以使用有效的 AWS 凭证访问客户加入 Amazon S3 存储桶来访问该模板。以下链接使用区域性 Amazon S3 存储桶。更改 us-west-2 区域代码以表示您要在其中创建 CloudFormation 堆栈的相应区域。

此外，以下指令使用 YAML。但是，模板有 YAML 和 JSON 这两种格式。要使用 JSON，请在下载模板.json 时将 .yaml 文件扩展名替换为。

要使用下载模板 AWS CLI，请使用以下命令：

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yml .
```

在浏览器中导航到以下 URL，可以在控制台中查看和下载此模板：

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yml
```

您可以使用以下链接直接在中 CloudFormation 指定模板：

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss.yml
```

该模板还定义了哪些其他资源？

该AquaSnppJpss模板包括以下其他资源：

- (可选) CloudWatch 事件触发器 -使用联系人 AWS Ground Station 之前和之后发送 CloudWatch 的事件触发的 AWS Lambda 函数。该 AWS Lambda 函数将启动并有选择地停止您的接收器实例。
- (可选) 联络的 EC2 验证：使用 Lambda 为带有 SNS 通知的联络设置您的 Amazon EC2 实例的验证系统的选项。需要注意的是，这可能会产生费用，具体取决于您当前的使用情况。
- Ground Station 亚马逊机器映像检索 Lambda：用于选择您的实例中安装的软件以及您选择的 AMI 的选项。软件选项包括带或不带软件定义无线电的数据流端点应用程序。要使用宽带 DigiF 数据传输和 AWS Ground Station 代理，请参阅。[使用 AWS Ground Station 代理 \(宽带 \) 的公共广播卫星](#)随着其他软件更新和功能的发布，这些选项可能会发生变化。
- 其他任务概况 ——其他公共广播卫星 (Aqua、SNPP和Terra) 的任务概况。
- 其他天线下行链路配置 -其他公共广播卫星 (Aqua、SNPP 和 Terra) 的天线下行链路配置。

已填充此模板中卫星的值和参数。这些参数使您可以轻松地 AWS Ground Station 立即使用这些卫星。使用此模板 AWS Ground Station 时，无需配置自己的值即可使用。但是，您可以自定义这些值以使模板适用于您的使用案例。

我可以在哪里接收我的数据？

数据流终端节点组设置为使用此模板的一部分创建的接收实例网络接口。接收器实例使用数据流端点应用程序从数据流端点定义 AWS Ground Station 的端口接收数据流。接收到数据后，可通过接收实例的环回适配器上的 UDP 端口 50000 使用数据。有关设置数据流终端节点组的更多信息，请参阅[AWS::GroundStation::DataflowEndpointGroup](#)

使用 AWS Ground Station 代理 (宽带) 的公共广播卫星

此示例建立在用户指南[JPSS-1-公共广播卫星 \(PBS\)-评估](#)部分所做的分析的基础上。

要完成此示例，您需要假设一个场景，即您要将 HRD 通信路径捕获为宽带数字中频 (digiF)，并在代理使用 SDR 在 Amazon EC2 实例 AWS Ground Station 上接收时对其进行处理。


```
me-south-1:
  groundstation: pl-02781422c4c792145
eu-west-1:
  groundstation: pl-03fa6b266557b0d4f
eu-north-1:
  groundstation: pl-033e44023025215c0
af-south-1:
  groundstation: pl-0382d923a9d555425
```

在“参数”部分，您将添加以下参数。在通过 CloudFormation 控制台创建堆栈时，您将为这些值指定值。

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-](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html)

[key-pairs.html](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-](https://docs.aws.amazon.com/ground-station/latest/ug/dataflows.ec2-configuration.html#dataflows.ec2-configuration.amis)

[configuration.html#dataflows.ec2-configuration.amis](https://docs.aws.amazon.com/ground-station/latest/ug/dataflows.ec2-configuration.html#dataflows.ec2-configuration.amis)

Type: AWS::EC2::Image::Id

Note

您需要创建密钥对，并提供 Amazon EC2 EC2Key 参数的名称。请参阅[为您的 Amazon EC2 实例创建密钥对](#)。

此外，在创建 CloudFormation 堆栈时，您需要提供正确的特定于区域的 AMI ID。请参阅[AWS Ground Station Amazon 机器映像 \(AMIs\)](#)。

其余的模板片段属于 CloudFormation 模板的“资源”部分。

Resources:

```
# Resources that you would like to create should be placed within the Resources section.
```

考虑到我们向 Amazon EC2 实例提供单一通信路径的场景，您知道您将拥有一条同步传输路径。根据本[同步数据传输](#)节，您必须使用 AWS Ground Station 代理设置和配置 Amazon EC2 实例，并创建一个或多个数据流终端节点组。首先，您需要为 AWS Ground Station 代理设置 Amazon VPC。

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

有关代理支持的 VPC 配置的更多信息，请参阅 AWS Ground Station [AWS Ground Station 代理要求-VPC 图表](#)。

接下来，您将设置 Receiver Amazon EC2 实例。

```
# 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

```

您还需要相应的策略、角色和配置文件，以便 AWS Ground Station 在您的账户中创建 elastic network interface (ENI)。

```

# 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 配置

本节介绍[创建配置](#)入门。

你需要一个跟踪配置来设置你使用自动追踪的偏好。选择 PREFERRED 作为自动跟踪可以提高信号质量，但由于 JPSS-1 星历质量足够，因此不需要满足信号质量。

```
TrackingConfig:  
  Type: AWS::GroundStation::Config  
  Properties:  
    Name: "JPSS Tracking Config"  
    ConfigData:  
      TrackingConfig:  
        Autotrack: "PREFERRED"
```

根据通信路径，你需要定义一个天线下行链路配置来表示卫星部分，还需要定义一个数据流端点配置来引用定义端点详细信息的数据流端点组。

```
# The AWS Ground Station Antenna Downlink Config that defines the frequency spectrum  
used to  
# downlink data from your satellite.  
SnppJpssDownlinkDigIfAntennaConfig:  
  Type: AWS::GroundStation::Config  
  Properties:  
    Name: "SNPP JPSS Downlink WBDigIF Antenna Config"  
    ConfigData:  
      AntennaDownlinkConfig:  
        SpectrumConfig:  
          Bandwidth:  
            Units: "MHz"  
            Value: 100  
          CenterFrequency:  
            Units: "MHz"  
            Value: 7812  
          Polarization: "RIGHT_HAND"
```

```
# The AWS Ground Station Dataflow Endpoint Config that defines the endpoint used to
downlink data
# from your satellite.
DownlinkDigIfEndpointConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "Aqua SNPP JPSS Terra Downlink DigIF Endpoint Config"
    ConfigData:
      DataflowEndpointConfig:
        DataflowEndpointName: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ]
        DataflowEndpointRegion: !Ref AWS::Region
```

AWS Ground Station 任务简介

本节介绍[创建任务档案](#)入门。

现在你已经有了相关的配置，你可以用它们来构造数据流。其余参数将使用默认值。

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnppJpssMissionProfile:
  Type: AWS::GroundStation::MissionProfile
  Properties:
    Name: !Sub 'JPSS WBDigIF gs-agent EC2 Delivery'
    ContactPrePassDurationSeconds: 120
    ContactPostPassDurationSeconds: 120
    MinimumViableContactDurationSeconds: 180
    TrackingConfigArn: !Ref TrackingConfig
    DataflowEdges:
      - Source: !Ref SnppJpssDownlinkDigIfAntennaConfig
        Destination: !Ref DownlinkDigIfEndpointConfig
    StreamsKmsKey:
      KmsKeyArn: !GetAtt GroundStationDataDeliveryKmsKey.Arn
    StreamsKmsRole: !GetAtt GroundStationKmsKeyRole.Arn
```

把它放在一起

利用上述资源，您现在可以安排 JPSS-1 联系人，以便从任何已上线人员同步传送数据。AWS Ground Station [AWS Ground Station 位置](#)

以下是一个完整的 CloudFormation 模板，其中包括本节中描述的所有资源，这些资源组合成一个可以直接在中使用的模板 CloudFormation。

名为 `DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml` 的 CloudFormation 模板旨在让你快速访问开始接收 Aqua、SNPP、JPSS-1/NOAA-20 和 Terra 卫星的数字化中频 (digiF) 数据。它包含一个 Amazon EC2 实例和使用 AWS Ground Station 代理接收原始的 digiF 直接广播数据所需的 CloudFormation 资源。

如果你的账户未登录 Aqua、SNPP、JPSS-1/NOAA-20 和 Terra，请参阅 [机载卫星](#)

Note

您可以使用有效 AWS 凭证访问客户登录 Amazon S3 存储桶，从而访问该模板。以下链接使用区域性 Amazon S3 存储桶。更改 `us-west-2` 区域代码以表示要在其中创建 CloudFormation 堆栈的相应区域。

此外，以下指令使用 YAML。但是，模板有 YAML 和 JSON 这两种格式。要使用 JSON，请在下载模板 `.json` 时将 `.yml` 文件扩展名替换为。

要使用下载模板 AWS CLI，请使用以下命令：

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/agent/ec2_delivery/
DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml .
```

在浏览器中导航到以下 URL，可以在控制台中查看和下载此模板：

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-
west-2/agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

您可以使用以下链接直接在中 CloudFormation 指定模板：

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/
agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

该模板还定义了哪些其他资源？

该DirectBroadcastSatelliteWbDigIfEc2DataDelivery模板包括以下其他资源：

- 接收方实例弹性网络接口-（视情况而定）在指定的子网中创建弹性网络接口（PublicSubnetId如果提供）。如果接收方实例位于私有子网中，则需要这样做。elastic network interface 将与 EIP 关联并连接到接收方实例。
- 接收器实例弹性 IP- AWS Ground Station 将连接到的弹性 IP。这会连接到接收器实例或 elastic network 接口。
- 以下弹性 IP 关联之一：
 - 接收器实例与弹性 IP 关联-弹性 IP 与您的接收器实例的关联（如果PublicSubnetId未指定）。这需要SubnetId引用公有子网。
 - 接收方实例与弹性 IP 关联的弹性网络接口-弹性 IP 与接收方实例弹性网络接口的关联（PublicSubnetId如果已指定）。
- （可选）CloudWatch 事件触发器-使用联系 AWS Ground Station 前后发送 CloudWatch 的事件触发的 AWS Lambda 函数。该 AWS Lambda 函数将启动并有选择地停止您的接收器实例。
- （可选）亚马逊联系人 EC2 验证-使用 Lambda 为带有 SNS 通知的联系人设置亚马逊 EC2 实例的验证系统的选项。需要注意的是，这可能会产生费用，具体取决于您当前的使用情况。
- 其他任务概况——其他公共广播卫星（Aqua、SNPP 和 Terra）的任务概况。
- 其他天线下行链路配置——其他公共广播卫星（Aqua、SNPP 和 Terra）的天线下行链路配置。

已填充此模板中卫星的值和参数。这些参数使您可以轻松地 AWS Ground Station 立即使用这些卫星。使用此模板 AWS Ground Station 时，您无需配置自己的值即可使用。但是，您可以自定义这些值以使模板适用于您的使用案例。

我可以在哪里接收我的数据？

数据流终端节点组设置为使用此模板的一部分创建的接收实例网络接口。接收器实例使用 AWS Ground Station 代理从数据流端 AWS Ground Station 点定义的端口接收数据流。有关设置数据流终端节点组的更多信息，请参阅[AWS::GroundStation::DataflowEndpointGroup](#)有关 AWS Ground Station 代理的更多信息，请参阅[什么是代 AWS Ground Station 理？](#)

问题排查

以下文档可以帮助您解决使用时可能出现的问题 AWS Ground Station。

主题

- [对向 Amazon EC2 传送数据的联系人进行故障排除](#)
- [解决失败的联系人问题](#)
- [解决联系人更新失败的问题](#)
- [排查计划失败的联系人故障](#)
- [排除 DataflowEndpointGroups 未处于正常状态的故障](#)
- [对无效的星历进行故障排除](#)
- [对未收到任何数据的联系人进行故障排除](#)
- [排除遥测故障](#)

对向 Amazon EC2 传送数据的联系人进行故障排除

如果您无法成功完成AWS Ground Station联系，则需要验证您的 Amazon EC2 实例是否正在运行，验证您的数据流终端节点应用程序是否正在运行，并验证您的数据流终端节点应用程序的流配置是否正确。

Note

DataDefender (DDX) 是目前支持的数据流端点应用程序的示例AWS Ground Station

先决条件

以下过程假定您已经设置了一个 EC2 instance 实例。要在中设置 Amazon EC2 实例AWS Ground Station，请参阅[入门](#)。

步骤 1：验证您的 EC2 实例是否正在运行

以下过程说明如何在控制台中找到您的 Amazon EC2 实例，并在该实例未运行时将其启动。

1. 找到用于您正在进行故障排除的联系的 Amazon EC2 实例。使用以下步骤：
 - a. 在CloudFormation控制面板中，选择包含您的 Amazon EC2 实例的堆栈。

- b. 选择资源选项卡，然后在逻辑 ID 列中找到您的 Amazon EC2 实例。在状态列中验证实例是否已创建。
 - c. 在物理 ID 列中，选择 Amazon EC2 实例的链接。这将带您到 Amazon EC2 管理控制台。
2. 在 Amazon EC2 管理控制台中，确保您的亚马逊 EC2 实例状态处于运行状态。
3. 如果您的实例正在运行，请继续执行下一步。如果您的实例没有运行，请使用以下步骤启动实例：
 - 选择 Amazon EC2 实例后，依次选择操作 > 实例状态 > 启动。

步骤 2：确定使用的数据流应用程序的类型

如果您使用 AWS Ground Station 代理进行数据传输，请重定向至 [“故障排除 AWS Ground Station 代理”](#) 部分。否则，如果您使用的是 DataDefender (DDX) 应用程序，请继续。 [the section called “步骤 3：验证数据流应用程序是否正在运行”](#)

步骤 3：验证数据流应用程序是否正在运行

验证状态 DataDefender 需要您连接到 Amazon EC2 中的实例。有关连接到您的实例的更多详细信息，请参阅 [Connect 到您的 Linux 实例](#)。

以下过程提供了在 SSH 客户端中使用命令进行故障排除的步骤。

1. 打开终端或命令提示符，然后使用 SSH 连接到您的 Amazon EC2 实例。转发远程主机的端口 80 以查看 DataDefender Web 用户界面。以下命令演示了如何使用 SSH 通过启用了端口转发的堡垒机连接到 Amazon EC2 实例。

Note

您必须将 <SSH KEY>、<BASTION HOST> 和 <HOST> 替换为特定的 SSH 密钥、堡垒主机名和 Amazon EC2 实例主机名。

对于 Windows：

```
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>
```

适用于 Mac

```
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. 通过对输出 DataDefender 中名为 ddx 的正在运行的进程执行 greping (检查) 来验证 (也称为 DDX) 是否正在运行。下面提供了针对正在运行的进程和成功的示例输出的 grep (检查) 命令。

```
[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
```

如果 DataDefender 正在运行，请跳至 [“the section called “步骤 4：验证您的数据流应用程序流是否已配置”否则”](#)，继续下一步。

3. 开始 DataDefender 使用下面的命令。

```
sudo systemctl start rtlogic-ddx
```

如果在使用命令后 DataDefender 正在运行，请跳至[the section called “步骤 4：验证您的数据流应用程序流是否已配置”](#)。否则，请继续下一步。

4. 使用以下命令检查以下文件，以查看安装和配置时是否存在任何错误 DataDefender。

```
cat /var/log/user-data.log  
cat /opt/aws/groundstation/.startup.out
```

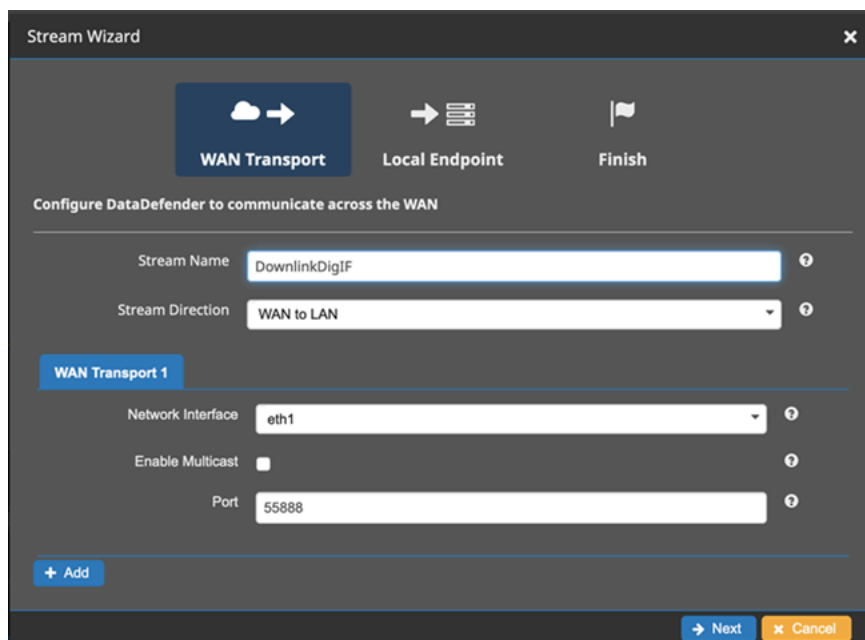
Note

检查这些文件时发现的一个常见问题是，您的 Amazon EC2 实例所在的 Amazon VPC 无法访问 Amazon S3 来下载安装文件。如果您在日志中发现这是问题所在，请检查您的 EC2 实例的 Amazon VPC 和安全组设置，确保它们不会阻止对 Amazon S3 的访问。

如果在检查您 DataDefender 的 Amazon VPC 设置后正在运行，请继续[the section called “步骤 4：验证您的数据流应用程序流是否已配置”](#)。如果问题仍然存在，请[联系 AWS Support](#)，并发送包含问题描述的日志文件。

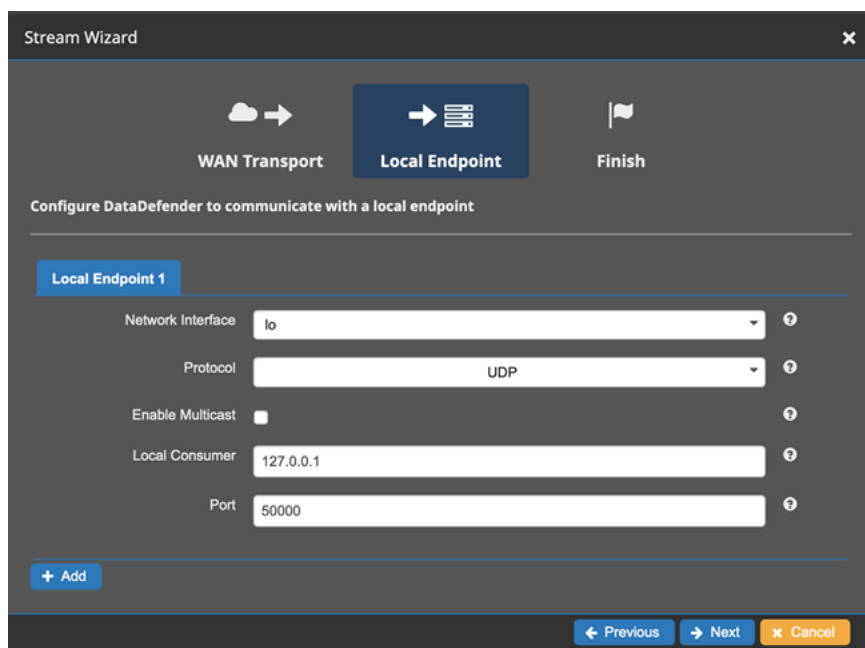
步骤 4：验证您的数据流应用程序流是否已配置

1. 在网络浏览器中，通过在地址栏中输入以下地址来访问您的 DataDefender 网络用户界面：localhost: 8080。然后按 Enter。
2. 在DataDefender仪表板上，选择“转到详细信息”。
3. 从流列表中选择您的流，然后选择编辑流。
4. 在流向导对话框中，执行以下操作：
 - a. 在广域网传输窗格中，确保将流方向选为广域网到局域网。
 - b. 在端口框中，确保您为数据流端点组选择的 WAN 端口已存在。默认情况下，此端口为 55888。然后选择下一步。



The image shows a 'Stream Wizard' dialog box with three tabs: 'WAN Transport' (selected), 'Local Endpoint', and 'Finish'. The 'WAN Transport' tab is titled 'Configure DataDefender to communicate across the WAN'. It contains the following fields: 'Stream Name' (text input with 'DownlinkDigIF'), 'Stream Direction' (dropdown menu with 'WAN to LAN'), 'WAN Transport 1' section with 'Network Interface' (dropdown menu with 'eth1'), 'Enable Multicast' (checkbox), and 'Port' (text input with '55888'). There is an '+ Add' button at the bottom left and 'Next' and 'Cancel' buttons at the bottom right.

- c. 在本地终端节点窗格中，确保端口框中存在有效端口。默认情况下，此端口为 50000。从 AWS Ground Station 服务收到数据后 DataDefender，您将在此端口上接收数据。然后选择下一步。



- d. 如果您更改了任何值，请通过其余菜单选择下一步。另外，您也可以取消流向导菜单。

现在，您已确保您的 Amazon EC2 实例 DataDefender 均处于运行状态，并且配置正确，可以从中接收数据AWS Ground Station。继续[the section called “步骤 5：确保您的接收器实例子网中有足够的可用的 IP 地址”](#)。

步骤 5：确保您的接收器实例子网中有足够的可用的 IP 地址

以下过程说明如何在控制台中查找 Amazon EC2 接收器实例中可用 IP 地址的数量。

1. 对于用于该联系的每个 Amazon EC2 接收器实例，您正在进行故障排除。使用以下步骤：
 - a. 在CloudFormation控制面板中，选择包含您的 Amazon EC2 实例的堆栈。
 - b. 选择资源选项卡，然后在逻辑 ID 列中找到您的 Amazon EC2 实例。在状态列中验证实例是否已创建。
 - c. 在物理 ID 列中，选择 Amazon EC2 实例的链接。这将带您到 Amazon EC2 管理控制台。
2. 在 Amazon EC2 管理控制台中，找到并单击您的 Amazon EC2 接收器实例的实例摘要中的子网 ID 链接。这将带您进入相应的 Amazon VPC 管理控制台。
3. 在 Amazon VPC 管理控制台中选择匹配的子网，然后查看子网的详细信息以了解可用 IPv4 地址。如果这个数字不如使用此 Amazon EC2 接收器实例的数据流终端节点那么多，请执行以下操作：

- a. 更新CloudFormation模板的相应子网CidrBlock以使其大小正确。有关子网大小的更多详细信息，请参阅[子网 CIDR 块](#)。
- b. 使用更新的CloudFormation模板重新部署堆栈。

如果您仍然遇到问题，[请联系 AWS Support](#)。

解决失败的联系人问题

当 AWS Ground Station 检测到您的资源配置存在问题时，联系人的终端联系状态将为 FAILED。下面提供了可能导致联络失败的常见用例，以及帮助排除故障的步骤。

Note

本指南专门针对失败的联系状态，不适用于其他失败状态，例如、或 FAILED_TO_SCHEDULE AWS_FAILED AWS_CANCELLED。有关联络状态的更多信息，请参阅[the section called “AWS Ground Station 联系人状态”](#)

数据流端点失败的用例

以下是可能导致基于数据流端点的数据流的联系失败状态的常见用例列表：

- Dataflow 端点从未连接 —— AWS Ground Station 天线和你的 Dataflow 端点组之间一个或多个数据流的连接从未建立。
- Dataflow 端点连接较晚- AWS Ground Station 天线和您的 Dataflow 端点组之间用于一个或多个数据流的连接是在联系开始时间之后建立的。
- Dataflow 终端节点的子网没有可用的 IP 地址- AWS Ground Station由于接收器实例的子网中没有任何可用的 IP 地址，因此的数据传输解决方案无法在您的私有网络中创建 ENI。
- Dataflow 终端节点 AWS Ground Station的子网无效-由于无法访问 Dataflow 端点组中指定的子网，数据传输解决方案无法在您的私有网络中创建 ENI。

对于任何数据流端点故障案例，建议调查以下内容：

- 确认接收方 Amazon EC2 实例在联络开始时间之前成功启动。
- 确认联系期间数据流端点软件已启动并正在运行。

- 确保每个接收方实例子网的每个数据流端点至少有一个可用的 IP 地址。
- 通过中配置的数据流，确保与您的数据流终端节点组关联的子网保持活动状态并可供[设置和配置 Amazon VPC](#)其使用。AWS Ground Station

有关更具体的故障排除步骤，请参阅有关[对向 Amazon EC2 传送数据的联系人进行故障排除](#)的部分。

AWS Ground Station 代理失败的用例

以下是可能导致基于代理的数据流出现失败联络状态的常见用例列表：

- AWS Ground Station 代理从未报告过状态-负责在您的 Dataflow Endpoint Group 上协调一个或多个数据流的数据传输的代理从未成功向其报告状态。AWS Ground Station 此状态更新应在联络结束后的几秒钟内发生。
- AWS Ground Station 代理启动较晚-负责在您的 Dataflow Endpoint Group 上为一个或多个数据流协调数据交付的代理启动得很晚，也就是联系开始时间之后。

对于任何 AWS Ground Station 代理数据流故障案例，建议调查以下内容：

- 确认接收方 Amazon EC2 实例在联络开始时间之前成功启动。
- 确认在启动时和联络期间代理应用程序已启动并正在运行。
- 确认代理应用程序和 Amazon EC2 实例在联络结束后的 15 秒内没有关闭。这为代理提供了足够的时间向 AWS Ground Station 报告状态。

有关更具体的故障排除步骤，请参阅有关[对向 Amazon EC2 传送数据的联系人进行故障排除](#)的部分。

解决联系人更新失败的问题

当您调用 [UpdateContact](#) API 时，AWS Ground Station 会对请求执行同步验证。如果验证通过，则会异步处理更新以将更改传播到天线区域。同步验证错误直接在 HTTP 响应中返回。异步故障通过 `failureCodes` 联系人版本上的 `failureMessage` 字段进行报告，您可以通过调用更新失败 [DescribeContactVersion](#) 的版本来查看这些字段。

有关联系人版本控制的更多信息，请参阅[更新联系人和联系人版本控制](#)。

同步验证错误

当 [UpdateContact](#) 请求验证失败时，将直接在 HTTP 响应中返回以下错误。

ResourceNotFoundException: 未找到联系人

常见原因

指定的contactId不存在或属于其他 AWS 账户。

解决方法

1. 验证是否contactId正确。
2. 确认您使用的是拥有该联系人的 AWS 账户的凭证。
3. 使用[ListContacts](#)来查找正确的contactId。

ConflictException: 无法更新联系人

常见原因

联系人处于不允许更新的状态。只有当联系人处于、或PASS状态时 SCHEDULEDPREPASS，才能调用[UpdateContact](#)API。如果其他更新已在进行中（最新联系人版本处于UPDATING状态），也会发生此错误。

解决方法

1. [DescribeContact](#)致电查看当前的联系人状态。
2. 如果联系人处于终止状态（例如COMPLETEDFAILED、或CANCELLED），则无法更新。只有当联系人处于、或PASS状态时 SCHEDULEDPREPASS，才能对其进行更新。有关终端状态的完整列表，请参阅[AWS Ground Station 联系人状态](#)。
3. 如果正在进行其他更新，请等待当前更新到达ACTIVE或FAILED_TO_UPDATE处于状态，然后再提交其他更新。你可以对 [DescribeContactVersion](#)API 进行民意调查，也可以使用一些人 AWS SDKs 提供的ContactUpdated服务员便利工具。AWS Command Line Interface

InvalidParameterException: 无效的请求参数

常见原因

该请求包含无效的参数。常见原因包括：

- 缺失或为空clientToken。
- 单个请求中包含多种类型ProgramTrackSettings（方位角/仰角、OEM 和 TLE）。每个请求只允许一种类型。

- `satelliteArn`在联系人的地面站未获准[使用方位角高程星历表的情况下设置为 null](#)。
- `AzElProgramTrackSettings`当`satelliteArn`为空时缺失。
- 提供`ephemerisId`与指定无关的`satelliteArn`。
- 在接触时间范围内，卫星在地面站没有有效的能见度窗口。
- 这颗卫星没有登上地面站，或者没有任务概况所要求的许可。
- 任务配置文件包含不支持联系人更新的[天线下行传输解调解码配置](#)配置。

解决方法

1. 查看响应中的错误消息，了解有关哪个参数无效的详细信息。
2. 确保`ProgramTrackSettings`每个请求只提供一种类型。
3. 如果使用不带的 `azimuth/elevation` 指向角度`satelliteArn`，请确认您的账户已获得地面站此功能的批准。有关更多信息，请参阅 [提供方位角高程星历数据](#)。
4. 验证您引用的星历是否与正确的卫星相关联并涵盖了接触时间范围。

ResourceLimitExceededException: 已达到最大版本限制

常见原因

联系人已达到最大版本数 (128)。每次调用都会[UpdateContact](#)创建一个新版本，联系人不能超过此限制。

解决方法

1. 不能提高此限制。如果您需要进一步更改，请取消联系并重新预订。

异步故障代码

以下故障代码出现在带有`FAILED_TO_UPDATE`状态的联系人版本`failureCodes`字段中。[DescribeContactVersion](#)用于检索这些详细信息。该`failureMessage`字段提供了有关失败的其他背景信息。

失败码	常见原因	解决方案
<code>INTERNAL_ERROR</code>	处理更新时出现意外内部错误。	重试更新。如果问题仍然存在，请联系 AWS 支持 。

失败码	常见原因	解决方案
INVALID_SATELLITE_ARN	更新请求中提供的卫星 ARN 无效或不存在。	验证卫星 ARN 并确认卫星已在您的账户中注册。
INVALID_UPDATE_CONTACT_REQUEST	更新请求包含同步验证期间未捕获的无效参数。	请查看failureMessage，了解详情并更正请求参数。
EPHEMERIS_NOT_FOUND	追踪覆盖中引用的星历不存在。	验证ephemerisId 并确认星历未被删除。
EPHEMERIS_TIME_RANGE_INVALID	星历不涵盖联系人的时间范围。	上传涵盖整个联系时间范围的新星历。如果无法延长星历的时间范围，请取消联系并在星历的时间段内保留新的联系人。有关更多信息，请参阅 提供自定义星历数据 。
EPHEMERIS_NOT_ENABLED	引用的星历未处于状态。ENABLED	请检查星历状态并在重试更新之前将其启用。
SATELLITE_DOES_NOT_MATCH_EPHEMERIS	星历与更新请求中指定的卫星无关。	确保ephemerisId 属于中指定的卫星satelliteArn。
NOT_ONBOARDED_TO_AZEL_EPHEMERIS	您的账户未获准在联系人的地面站 使用方位角高程星历数据 。方位角高程星历表是一项受限功能，适用于有限数量的专业用例。	如果您的用例需要方位角高程星历，请通过打开 AWS 支持 票证申请访问权限。 AWS Support Center Console 或者，如果符合您的用例，可以考虑 使用 TLE 星历表数据 或 OEM 星历数据 。
AZEL_EPHEMERIS_NOT_FOUND	请求中引用的 方位角高程星历 不存在。	验证ephemerisId 并确认方位角高程星历表未被删除。

失败码	常见原因	解决方案
AZEL_EPHE MERIS_WRO NG_GROUND _STATION	方位角高程星历表是为与接触者使用的地面站 不同的地面站创建的。	为正确的地面站上传新的方位角高程星历，或者使用与联系人地面站相匹配的现有星历表。
AZEL_EPHE MERIS_INV ALID_STATUS	方 位角高程星历表的使用状态不正确 。	检查星历状态。它必须处于一种ENABLED状态。如果星历验证失败，请上传更正后的版本。
AZEL_EPHE MERIS_TIM E_RANGE_INVALID	方位角仰角星历不 涵盖接触的时间范围。	上传涵盖整个接触时间范围的新方位角高程星历。如果无法延长星历的时间范围，请取消联系并在星历的时间段内保留新的联系人。

检查更新状态

调用后UpdateContact，新的联系人版本将在该UPDATING状态下启动。在此期间，[DescribeContact](#)将继续返回之前处于活动状态的联系人版本。新版本在传播DescribeContact到天线并达到ACTIVE状态后才会出现。要检查特定版本的状态，请使用[DescribeContactVersion](#)。

要确定更新是成功还是失败，请执行以下操作：

1. [DescribeContactVersion](#)使用调用contactId并由UpdateContact响应versionId返回。
2. 检查该version.status字段。状态为ACTIVE表示更新已成功应用。状态为FAILED_TO_UPDATE表示更新失败。
3. 如果状态为FAILED_TO_UPDATE，请查看version.failureCodes和version.failureMessage字段，详细了解出了什么问题。

Tip

有些人 AWS Command Line Interface 支持ContactUpdated服务员 AWS SDKs，该服务员会自动轮询DescribeContactVersion直到版本达到ACTIVE或FAILED_TO_UPDATE处于状

态。例如，AWS Command Line Interface 提供了 a [ws 地面站等待联系人更新命令](#)。使用服务员，而不是实现自己的投票逻辑。

排查计划失败的联系人故障

当 AWS Ground Station 检测到您的资源配置或内部系统存在问题时，联系人将以 FAILED_TO_SCHEDULE 状态结束。以 FAILED_TO_SCHEDULE 状态结束的联系人可以选择提供其他背景信息。errorMessage 有关描述联系人的信息，请参阅 [DescribeContact API](#)。

下面提供了可能导致 FAILED_TO_SCHEDULE 联系的常见用例，以及帮助进行故障排除的步骤。

Note

本指南专门针对 FAILED_TO_SCHEDULE 联系状态，不适用于其他故障状态，例如、或 FAILED。AWS_FAILED_AWS_CANCELLED 有关联络状态的更多信息，请参阅 [the section called “AWS Ground Station 联系人状态”](#)

不支持 Antenna Downlink Demod Decode Config 中指定的设置

用于安排此次联系的 [任务配置文件 antenna-downlink-demod-decode 中的配置](#) 无效。

先前存在的 AntennaDownlinkDemodDecode 配置

- 如果您的 antenna-downlink-demod-decode 配置最近发生了更改，请在尝试安排之前回滚到以前运行的版本。
- 如果这是对现有配置的故意更改，或者是以前存在的配置无法成功调度，请按照下一步操作如何载入新 AntennaDownlinkDemodDecode 配置。

新创建的 AntennaDownlinkDemodDecode 配置

请 AWS Ground Station 直接联系以加入您的新配置。使用 [AWS Support 创建案例](#) contactId，包括以 FAILED_TO_SCHEDULE 状态结束的案例

一般故障排除步骤

如果上述故障排除步骤未能解决您的问题：

- 使用相同的任务配置文件重新尝试安排联系人或安排其他联系人。有关如何预约联系人的信息，请参阅[ReserveContact](#)。
- [如果您继续收到此任务档案的 FAILED_TO_SCHEDULE 状态，请联系 AWS Support](#)

排除 DataflowEndpointGroups 未处于正常状态的故障

下面列出了您的数据流端点组可能未处于 HEALTHY 状态的原因以及需要采取的适当纠正措施。

- NO_REGISTERED_AGENT：启动您的 EC2 实例，该实例将用于注册座席。请注意，您必须拥有有效的控制器配置文件才能成功实现本次调用。有关配置该文件的详细信息，请参阅[使用 AWS Ground Station 座席](#)。
- INVALID_IP_OWNERSHIP-使用 DeleteDataflowEndpointGroup API 删除 Dataflow 端点组，然后使用 CreateDataflowEndpointGroup API 使用与 EC2 实例关联的 IP 地址和端口重新创建 Dataflow 端点组。
- UNVERIFIED_IP_OWNERSHIP：IP 地址尚未经过验证。验证会定期进行，因此这个问题应该会自行解决。
- NOT_AUTHORIZED_TO_CREATE_SLR：账户没有创建必要的服务相关角色的权限。请查看[在 Ground Station 中使用与服务相关的角色](#)中的故障排除步骤

对无效的星历进行故障排除

当您将星历数据上传到时 AWS Ground Station，它会经过异步验证工作流程。如果验证失败，则星历状态将更改为。INVALID[DescribeEphemeris](#)响应中的错误消息提供了详细信息，可帮助您识别和解决问题。

了解星历验证错误

当星历未通过验证时，[DescribeEphemeris](#)API 响应会包含两个字段来帮助诊断问题：

errorCode

标识特定验证错误的机器可读代码。这可用于编程错误处理。

errorMessage

对验证错误的简单描述，其中包含有关错误原因的具体细节以及如何修复该错误的指导。

无效[DescribeEphemeris](#)星历的响应示例：

```
{
  "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 星历表的常见验证错误

以下是上传 TLE 星历表时遇到的常见验证错误：

不匹配的卫星目录号

错误：“星历中显示的卫星目录编号与关联卫星的卫星目录号不匹配”

解决方案：验证 TLE 线路中的 NORAD ID/satellite 目录号是否与卫星的卫星目录号相匹配。00000用于没有指定目录编号的卫星。

无效的平均运动

错误：“提供的星历的平均运动与最新的参考星历表差异太大”

解决方案：验证您的 TLE 数据是否正确且代表有效的轨道。Ground Station 在验证期间使用 Space-Track 星历表作为参考。

OEM 星历表的常见验证错误

以下是上传 OEM 星历表时遇到的常见验证错误：

参考框架无效

错误：“不支持 REF_FRAME”

解决方案：更新您的 OEM 文件以使用其中一个支持的参考框架：EME2000 或 ITRF2000。

缺少必填字段

错误：“缺少元数据字段：插值”

解决方案：将 INTERPOLATION 和 INTERPOLATION_DEGREE 字段添加到您的 OEM 元数据部分。它们是生成精确的 AWS Ground Station 天线指向角度所必需的。

不支持的时间系统

错误：“不支持 TIME_SYSTEM”

解决方案：确保您的 OEM 文件使用 UTC 作为时间系统。

不支持的 OEM 版本

错误：“不支持 CCSDS_OEM_VERS”

解决方案：确保您的 OEM 文件使用 CCSDS OEM 版本 2.0。

方位角高程星历表的常见验证错误

以下是上传方位角高程星历表时遇到的常见验证错误：

缺少 azimuth/elevation 数据

错误：“至少有一个 TimeAzEl 字段中没有字段 AzElSegment”

解决方案：确保方位角高程数据中的每个分段至少包含一个带时间标签的对。 azimuth/elevation

无效的方位角范围 (度)

错误：“AzEl az 必须大于或等于 -180 且小于或等于 360 度”

解决方案：确认方位角在 [-180, 360] 度以内。

仰角范围无效 (度)

错误：“AzEl el 必须大于或等于 -90 且小于或等于 90 度”

解决方案：验证仰角是否在 [-90, 90] 度以内。

方位角范围无效 (弧度)

错误：“AzEl az 必须大于或等于 -pi 且小于或等于 2pi 弧度”

解决方案：验证方位角是否在 $[-\pi, 2\pi]$ 弧度之内。

仰角范围无效 (弧度)

错误：“AzEl el 必须大于或等于 $-\pi/2$ 且小于或等于 $\pi/2$ 弧度”

解决方案：验证仰角是否在 $[-\pi/2, \pi/2]$ 弧度之内。

非单调时间值

错误：“中的 TimeAzEl 商品 AzElSegment 必须按时间顺序排列”

解决方案：确保每个分段中的时间值严格增加。

区段出现故障

错误：“AzElSegments 必须按时间顺序排列”

解决方案：确保区段按时间顺序排列。

重叠的区段

错误：“至少一个分段的时间范围与其他分段的时间范围重叠”

解决方案：确保每个分段都有唯一的、不重叠的时间范围。一个分段endTime的值不应超过下一个分段startTime的值。

故障排除步骤

如果您的星历未能通过验证，请按照以下步骤解决问题：

1. [DescribeEphemeris](#)使用您的星历编号致电以检索和。errorCode errorMessage
2. 查看错误消息，了解有关哪些验证检查失败的具体详细信息。
3. 更正星历数据中已发现的问题。
4. 使用上传包含更正数据的新星历。[CreateEphemeris](#)
5. 监控新的星历状态，直到其达到状态。ENABLED
6. [DeleteEphemeris](#)如果不再需要无效的星历表，请使用将其删除。

完整的错误代码参考

以下各节按高级类别提供了星历验证失败时可能返回的所有errorCode值的全面映射。invalidReason

无效原因：METADATA_INVALID

当必填的元数据字段缺失、格式不正确或星历数据中包含不支持的值时，就会发生这些错误。

错误代码	错误消息
不匹配的_SATCAT_ID	TLE 星历表中的卫星目录编号与相关卫星的卫星目录号不匹配
不支持 OEM_VERSION_	不CCSDS_OEM_VERS 支持 OEM 星历中的。支持的值：[2.0]
创始人_失踪	OEM 星历中缺少ORIGINATOR 标题字段
创建日期缺失	OEM 星历中缺少CREATION_DATE 标题字段
缺少对象名称	OEM 星历中缺少OBJECT_NAME 元数据字段
缺少对象 ID	OEM 星历中缺少OBJECT_ID 元数据字段
REF_FRAME_不支持	不REF_FRAME 支持 OEM 星历中的。支持的值：[EME2000, ITRF2000]
不支持 REF_FRAME_EPOCH_EPOCH	不支持 OEM 星历中的REF_FRAME_EPOCH 元数据字段。请从星历中删除此字段
不支持时间系统	不TIME_SYSTEM 支持 OEM 星历中的。支持的值：[UTC]
CENTER_BODY_不支持	不CENTER_BODY 支持 OEM 星历中的。支持的值：[Earth]
插值_缺失	OEM 星历中缺少INTERPOLATION 元数据字段
插值_度数_无效	对于插值法，OEM 星历中的插值度必须大于 0
AZ_EL_SEGMENT_LIST_MISSING	该 azElSegmentList 字段丢失
时间不足_AZ_EL	至少有一个 TimeAzEl 字段中没有字段 azElSegmentList

无效原因：TIME_RANGE_INVALID

当星历包含无效的时间范围时，就会发生这些错误，包括时间问题、分段顺序问题、片段重叠或 start/end 时间不一致问题。

错误代码	错误消息
将来的开始时间	Ephemeris 的开始时间是未来，但必须是过去
过去 END_TIME	Ephemeris 的结束时间已经过去，但必须是将来
过期_时间_太早了_太早	提供的到期时间早于星历结束时间
START_TIME_METADATA_TOO_EARLY	START_TIME 元数据值早于 OEM 星历表数据中存在的最早时间
STOP_TIME_METADATA_TOO_LATE	STOP_TIME 元数据值晚于 OEM 星历表数据中显示的最新时间
AZ_EL_SEGMENT_END_TIME_BEFORE_START_TIME	至少有一个数据段 endTime 的位于该分段之前 startTime
AZ_EL_SEGMENT_TIMES_OVERLAP	至少一个分段的时间范围与其他分段的时间范围重叠
AZ_EL_SEGMENTS_OUT_OF_ORDER	区段不是按时间顺序排列的
TIME_AZ_EL_ITEMS_OUT_OF_ORDER	a 中的 TimeAzEl 项目 AzElSegment 必须按时间顺序排列
AZ_EL_SEGMENT_REFERENCE_EPOCH_INVALID	片段的参考周期无效或格式不正确
AZ_EL_SEGMENT_START_TIME_无效	区段有效时间范围内的开始时间不会在第一个分段之后开始
AZ_EL_SEGMENT_END_TIME_无效	区段有效时间范围内的结束时间不会在最后一个分段之后结束

错误代码	错误消息
AZ_EL_SEGMENT_VALID_TIME_RANGE_VALID_	区段的有效时间范围无效
AZ_EL_SEGMENT_END_TIME_TIME_TOO_LATE	片段的结束时间超过了从参考周期开始的最大允许持续时间
已超过 AZ_EL_TOTAL_持续时间	所有分段的总持续时间超过了允许的最大指向角度持续时间

无效原因：TRAJECTORY_INVALID

当星历中包含无效的轨迹数据（包括轨道参数、角度范围或单位问题）时，就会发生这些错误。

错误代码	错误消息
MEAN_MOTION_无效	所提供的 TLE 星历的平均动作与最新的参考星历表差异太大。注意：Ground Station 在验证期间使用 Space-Track 星历表作为参考
TIME_AZ_EL_AZ_RADIAN_RANGE_RANGE_无效	AzEl az 必须大于或等于 $-\pi$ 且小于或等于 2π 弧度
TIME_AZ_EL_EL_RADIAN_RANGE_RANGE_无效	AzEl el 必须大于或等于 $-\pi/2$ 且小于或等于 $\pi/2$ 弧度
时间_AZ_EL_AZ_DEGRE_RANGE_RANGE_无效	AzEl az 必须大于或等于 -180 且小于或等于 360 度
TIME_AZ_EL_EL_DEGREE_RANGE_RANGE_无效	AzEl el 必须大于或等于 -90 度且小于或等于 90 度
TIME_AZ_EL_ANGLE_ANGLE_UNITS 无效	AzEl 角度单位无效

无效原因：**KMS_KEY_INVALID**

当用于加密星历数据的 AWS 密钥管理服务 (KMS) 密钥出现问题时，就会发生这些错误。

错误代码	错误消息
KMS_权限不足	Ground Station 没有足够的权限访问此简历的 KMS 密钥

无效原因：**VALIDATION_ERROR**

当星历数据存在不属于其他特定类别的一般验证问题时，就会发生这些错误。

错误代码	错误消息
INTERNAL_ERROR	校验星历期间出现内部错误
文件格式无效	星历文件格式无效或已损坏。验证文件是否符合星历类型的预期格式

对未收到任何数据的联系人进行故障排除

联系人可能显示为成功，但仍未收到任何数据。这可能意味着您收到的是空的 PCAP 文件，或者如果您使用 S3 数据传输，则根本没有 PCAP 文件。这可能是由于多种原因造成。以下内容讨论了其中一些原因以及如何解决这些问题。

下行链路配置不正确

从卫星接收数据的每个联系人都将有一个关联的[天线下行传输配置](#)或[天线下行传输解调解码配置](#)。如果指定的配置与卫星传输的信号不一致，则 AWS Ground Station 将无法接收传输的信号。这将导致未收到任何数据 AWS Ground Station。

要解决此问题，请验证您使用的配置是否与卫星传输的信号一致。例如，验证您是否设置了正确的中心频率、带宽、极化以及解调和解码参数（如果需要）。

卫星操纵

有时，卫星可能会执行暂时禁用其某些通信系统的操纵。该演习还可能显著改变卫星在天空中的位置。AWS Ground Station 将无法接收来自未传输信号的卫星的信号，或者如果使用的星历使 AWS Ground Station 天线指向天空中不存在卫星的位置。

如果您正在尝试与 NOAA 运营的公共广播卫星进行通信，则可以在 NOAA [卫星警报](#) 消息页面上找到描述中断或操纵的消息。该消息可能包括预计何时恢复数据传输的时间表，也可能在随后的消息中发布。

如果你正在与自己的卫星通信，你有责任了解自己的卫星运行情况，以及这会如何影响与之通信 AWS Ground Station。如果您正在执行会影响卫星轨迹的操作，则可能包括提供更新的自定义星历数据。有关提供自定义星历数据的更多信息，请参阅 [明白怎么做 AWS Ground Station 使用星历](#)

AWS Ground Station 中断

如果 AWS Ground Station 导致联系失败或取消，则 AWS Ground Station 会将联系状态设置为 AWS_FAILED、或 AWS_CANCELLED。有关联系人生命周期的更多信息，请参阅 [了解联系生命周期](#)。在某些情况下，AWS Ground Station 可能会出现故障，导致数据无法发送到您的账户，但不会导致联系人处于 “AWS_FAILED 或 AWS_CANCELLED” 状态。发生这种情况时，AWS Ground Station 应在您的 Health AWS h 控制面板上发布特定于账户的事件。有关 Health AWS h 控制面板的更多信息，请参阅 [AWS Health 用户指南](#)。

排除遥测故障

使用以下信息对遥测的常见问题进行故障排除。

常见的设置问题

IAM 权限错误

症状

当调用 CreateConfig create a 时 TelemetrySinkConfig，你会收到一个错误：

```
Unable to write to Kinesis Data Streams stream. Ensure that Ground Station has kinesis:PutRecord permissions for the given stream
```

原因

- 中指定的 IAM 角色 TelemetrySinkConfig 没有写入 Kinesis 数据流所需的权限。
- IAM 角色的信任策略不 AWS Ground Station 允许代入该角色。
- 中的 Kinesis 数据流 ARN 不正确或该数据流不存在。 TelemetrySinkConfig

解决方案

1. 验证 IAM 角色是否存在且权限正确。审查[第 2 步：创建一个 TelemetrySinkConfig](#)并确保所有步骤都已得到遵守。
2. 检查是否 AWS Ground Station 可以代入您的 IAM 角色：

```
aws iam get-role --role-name GroundStationTelemetryRole
```

验证信任策略是否包含groundstation.amazonaws.com为可信服务主体。

3. 验证 IAM 角色是否具有所需的 Kinesis 权限：

```
aws iam list-attached-role-policies --role-name GroundStationTelemetryRole
```

确保该政策包含kinesis:DescribeStreamkinesis:PutRecord、和直播kinesis:PutRecords权限。

4. 验证 Kinesis 数据流是否存在且 ARN 是否正确：

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

5. 如果使用客户管理的加密，请验证 IAM 角色是否kms:GenerateDataKey有权使用您的 AWS KMS 密钥。

PassRole 权限错误

症状

调用时CreateConfig，您会收到一条关于无权传递 IAM 角色的错误。

解决方案

确保您的 IAM 用户或角色拥有遥测 IAM 角色的iam:PassRole权限。向您的用户或角色添加以下策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/your-stream-name"
    }
  ]
}
```

Kinesis 数据流配置问题

症状

遥测传输失败或是间歇性的。

原因

- Kinesis 数据流的容量不足以满足遥测吞吐量。
- 该流正被其他应用程序使用，导致写入限制。

解决方案

1. 检查直播状态：

```
aws kinesis describe-stream \
  --stream-name your-stream-name \
  --region us-east-2
```

2. 使用 CloudWatch 指标监控写入限制：

```
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. 如果检测到节流，请考虑：

- 切换到按需容量模式以实现自动扩展。
- 使用专用流进行 AWS Ground Station 遥测。
- 如果使用预置模式，则增加分片的数量。

遥测传输问题

没有显示遥测数据

症状

使用支持遥测的任务配置文件安排联系后，您的 Kinesis 数据流中不会显示遥测数据。

可能的原因和解决方案

任务配置文件未启用遥测

验证用于联系人的任务配置文件包括telemetrySinkConfigArn：

```
aws groundstation get-mission-profile \  
--mission-profile-id 12345678-1234-1234-1234-123456789012 \  
--region us-east-2
```

检查该telemetrySinkConfigArn字段的输出。如果不存在，则任务配置文件未启用遥测。

IAM 角色权限问题

查看中的 IAM 权限故障排除步骤[IAM 权限错误](#)。

Kinesis 数据流不存在或位于错误的区域

验证直播是否存在于正确的区域：

```
aws kinesis describe-stream \  
--stream-name your-stream-name \  
--region us-east-2
```

联系尚未开始

遥测交付从接触开始时开始。通过检查联系人状态来验证联系人是否已开始：

```
aws groundstation describe-contact \  
  --contact-id 12345678-1234-1234-1234-123456789012 \  
  --region us-east-2
```

预定在数字双胞胎地面站上进行联系

数字双胞胎地面站不支持遥测传输。在数字双天线上安排的联系不会生成遥测数据。通过检查地面站名称，确认联系人使用的地面站不是数字双胞胎。数字双胞胎地面站的名称中有“数字双胞胎”前缀。有关更多信息，请参阅 [使用AWS Ground Station数字双胞胎功能](#)。

间歇性遥测数据

症状

遥测数据的传输不一致，存在差距或缺少记录。

可能的原因

- Kinesis 数据流流容量问题或限制。请参阅[Kinesis 数据流配置问题](#)。
- 与您的 Kinesis 数据流 AWS Ground Station 之间的网络连接问题。

解决方案

- 监控 Kinesis 数据流的流式传输 CloudWatch 指标是否存在节流或错误。
- 确保您的直播使用按需容量模式或有足够的预置容量。
- 使用专用的数据流进行 AWS Ground Station 遥测，以避免与其他应用程序争用。

数据格式问题

JSON 解析错误

症状

您的应用程序在将遥测记录解析为 JSON 时遇到错误。

解决方案

- 验证 Base64 解码 -Kinesis 数据流流中的数据是。Base64-encoded确保在将数据解析为 JSON 之前对其进行解码。有关更多信息，请参阅 [从 Kinesis 数据流中读取数据](#)。
- 检查是否有空记录 -创建时 AWS Ground Station 可能会发送空的验证记录TelemetrySinkConfig。您的应用程序应正常处理空白或格式错误的记录。
- 实现版本感知解析 -首先解析telemetryTypeAndVersion、和telemetryVersion字段telemetryType，以确定每条记录的相应架构。

未知的遥测类型或版本

症状

您的应用程序会遇到它无法识别的遥测类型或版本。

解决方案

这是预期的行为，因为随着时间的推移可能会引入新的遥测类型和架构版本。您的应用程序应该：

- 记录未知类型和版本以进行监控。
- 继续处理已知类型和版本。
- 对未知架构实现优雅处理。

有关架构版本控制的更多信息，请参阅[架构版本控制和演变](#)。

获取帮助

如果您在完成故障排除步骤后仍然遇到问题，请联系 AWS 支持部门。

要提供的信息

联系支持人员时，请提供以下信息：

- 联系人 ID 遇到问题
- 使用的任务配置文件 ID
- TelemetrySinkConfig ARN
- Kinesis 数据流直播 ARN
- IAM 角色 ARN 和附加的策略

- 来自 CloudWatch 日志或应用程序的错误消息
- 问题发生的时间戳
- 已经采取了故障排除步骤

有关一般 AWS Ground Station 支持，请参阅 [《AWS Ground Station 用户指南》](#)。

限额和限制

您可以查看支持的区域、其关联的终端节点以及终端节点和配额AWS Ground Station 的[配额](#)。

如果需要，您可以使用 [Service Quotas 控制台](#)、[AWS API](#) 和 [AWS CLI](#) 请求提高配额。

服务条款

有关 AWS Ground Station 服务条款，请参阅 [AWS 服务条款](#)。

的文件历史记录 AWS Ground Station 用户指南

下表描述了每个版本的《AWS Ground Station 用户指南》中的重要更改。

变更	说明	日期
新特征	添加了有关 Alpha-5 TLE 卫星号码支持的文档。AWS Ground Station 现在接受带有 Alpha-5 编码卫星编号的 TLE 星历表，其卫星目录号介于 100,000—339,999 之间。有关更多信息，请参阅 提供 TLE 星历数据 。	2026 年 5 月 28 日
新特征	添加了 AWS Ground Station 专用天线的文档。有关更多信息，请参阅 AWS Ground Station 专用天线 。	2026 年 4 月 14 日
新特征	添加了 UpdateContact API 和联系人版本控制的文档。有关更多信息，请参阅 更新联系人和联系人版本控制 。	2026 年 4 月 14 日
新特征	添加了 ListAntennas 和 ListGroundStationReservations API 的文档。有关更多信息，请参阅 AWS Ground Station 位置 和 查看地面站预订 。	2026 年 4 月 14 日
文档更新	向 CancelContact API 添加了其他功能，包括有关上述功能和计量影响的信息。有关更多信息，请参阅 了解联系人测量 。	2025 年 12 月 10 日

文档更新	澄清说，CloudWatch 指标是在与接触者的地面站相关的区域发射的。修复了断开的链接。	2025 年 12 月 2 日
更新的 AWS 管理策略	AWS Ground Station 已更新托管策略AWSGroundStationAgentInstancePolicy，增加了检索任务响应 URL 的额外权限。有关信息，请参阅 AWS 托管策略的 AWS Ground Station 更新 。	2025 年 11 月 13 日
新特征	更新了用户指南，加入了方位角仰角星历。有关更多信息，请参阅 提供方位角高程星历数据	2025 年 10 月 22 日
文档更新	Cross-region 数据传输不再需要特殊配置或批准。有关更多信息，请参阅 使用跨区域数据交付 。	2025 年 9 月 11 日
文档更新	添加了有关已配置资源的联系人利用率的说明。	2025 年 4 月 4 日
新特征	更新了用户指南，加入了 AWS Ground Station 数字双胞胎。	2024 年 8 月 6 日
文档更新	更新了用户指南的许多部分，包括新的图表、示例等。	2024 年 7 月 18 日
文档更新	在《用户指南》中添加了 RSS 提要。	2024 年 7 月 18 日
文档更新	将《AWS Ground Station 代理用户指南》拆分为单独的《用户指南》。	2024 年 7 月 18 日

新特征	现在可以将联系人安排在可见时间范围之外最多 30 秒。可见时间包含在 DescribeContact 响应中。	2024 年 3 月 26 日
文档更新	改进了组织结构并添加了“EC2 实例选择和 CPU 规划”部分。	2024 年 3 月 6 日
文档更新	在《AWS Ground Station 代理用户指南》中添加了新的最佳实践，用于与 AWS Ground Station 代理一起运行服务和进程。	2024 年 2 月 23 日
文档更新	添加了代理发行说明页面。	2024 年 2 月 21 日
模板更新	在 DirectBroadcastSatelliteWbDigiIfEc2DataDelivery 模板中添加了对单独公有子网的支持。	2024 年 2 月 14 日
文档更新	用户通知服务 在监控文档中添加了对 AWS 的引用。	2023 年 8 月 6 日
文档更新	添加了使用要在 AWS Ground Station 控制台中显示的名称来标记卫星的说明。	2023 年 7 月 26 日
新特征	为发布宽带 DigiF 数据传输添加了 AWS Ground Station 代理用户指南。	2023 年 4 月 12 日
新的 AWS 托管策略	AWS Ground Station 添加了一个名为的新策略 AWSGroundStationAgentInstancePolicy。	2023 年 4 月 12 日
新特征	更新了 CPE 预览版发布的用户指南。	2022 年 11 月 9 日

新的 AWS 托管策略	AWS Ground Station 添加了 AWSServiceRoleForGroundStationDataflowEndpointGroup 服务关联角色 (SLR)，其中包括一个名为的新策略。AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy	2022 年 11 月 2 日
新特征	更新了用户指南，增加了与集成 AWS CLI。	2020 年 4 月 17 日
新特征	更新了用户指南，纳入了与 CloudWatch Metrics 的集成。	2020 年 2 月 24 日
新模板	公共广播卫星 (AquaSnppJpss 模板) 已添加到 AWS Ground Station 用户指南中。	2020 年 2 月 19 日
新特征	更新了用户指南以包含跨区域数据传输。	2020 年 2 月 5 日
文档更新	更新了 AWS Ground Station 使用 CloudWatch 事件进行监控的示例和描述。	2020 年 2 月 4 日
文档更新	模板位置已更新且“入门”和“故障排除”部分已修订。	2019 年 12 月 19 日
新的疑难解答部分	故障排除部分添加至 AWS Ground Station 用户指南。	2019 年 11 月 7 日
新的入门主题	更新了入门主题，其中包括最新的 CloudFormation 模板。	2019 年 7 月 1 日
Kindle 版本	发布了 AWS Ground Station 用户指南 Kindle 版本。	2019 年 6 月 20 日

[新服务和指南](#)

这是《AWS Ground Station 用户指南》AWS Ground Station 的初始版本。 2019 年 5 月 23 日

AWS 词汇表

有关最新 AWS 术语，请参阅《AWS 词汇表 参考资料》中的[AWS 词汇表](#)。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。