



Manual do usuário

AWS Ground Station



AWS Ground Station: Manual do usuário

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que AWS Ground Station é	1
Casos de uso comuns	1
Próximas etapas	2
Como AWS Ground Station funciona	3
Integração via satélite	3
Composição do perfil da missão	3
Agendamento de contatos	5
Execução de contatos	6
Gêmeo digital	9
Entenda os componentes AWS Ground Station principais	9
Perfis de missão	11
Configurações	14
Grupos de endpoints do Dataflow	24
AWS Ground Station Agente	32
Conceitos básicos	33
Inscreva-se para um Conta da AWS	33
Criar um usuário com acesso administrativo	33
Adicione AWS Ground Station permissões à sua AWS conta	35
Satélite a bordo	37
Visão geral do processo de integração de clientes	37
(Opcional) Nomeando satélites	37
Satélites de transmissão pública	40
Planeje seus caminhos de comunicação de fluxo de dados	41
Entrega assíncrona de dados	41
Entrega síncrona de dados	42
Planeje sua telemetria	43
Crie configurações	44
Configurações de entrega de dados	44
Configuração de telemetria (opcional)	45
Configurações de satélite	45
Criar perfil de missão	45
Entenda as próximas etapas	46
AWS Ground Station Localizações	48
Encontrando a região da AWS para a localização de uma estação terrestre	48

AWS Ground Station regiões da AWS suportadas	50
Disponibilidade do gêmeo digital	50
Antenas dedicadas	50
Visualizando antenas em uma estação terrestre	51
Exemplo: Listar antenas em uma estação terrestre	51
Exibir reservas de estações terrestres	52
Listando reservas	52
Tipos de reserva	52
Exemplo de código	53
AWS Ground Station máscaras do site	55
Máscaras específicas para clientes	55
Impacto das máscaras do site nos horários de contato disponíveis	56
AWS Ground Station Capacidades do site	56
Entenda como AWS Ground Station usa efemérides	60
Dados de efemérides padrão	61
Forneça dados de efemérides personalizados	61
Visão geral do	61
Exemplo: uso de efemérides fornecidas pelo cliente com AWS Ground Station	62
Forneça dados de efemérides do TLE	62
Forneça dados de efemérides do OEM	69
Forneça dados de efemérides de elevação de azimute	78
Reserve contatos com efemérides personalizadas	88
Visão geral do	88
Fluxos de trabalho de reservas de contatos	89
Fluxo de trabalho 1: liste os contatos disponíveis e reserve	89
Fluxo de trabalho 2: reserva de contato direto	94
Monitorando mudanças no estado do contato	97
Práticas recomendadas e considerações	99
Entenda quais efemérides são usadas	100
Efemérides TLE e OEM	100
Efemérides de elevação de azimute	101
Efeito de novas efemérides em contatos previamente agendados	102
Obtenha as efemérides atuais de um satélite	102
Exemplo de retorno GetSatellite para um satélite usando uma efeméride padrão	103
Exemplo de retorno GetSatellite para um satélite usando uma efeméride personalizada	104
Listando efemérides de elevação de azimute	104

Reverter para dados de efemérides padrão	105
Revertendo efemérides de TLE e OEM	106
Gerenciando efemérides de elevação de azimute	106
Trabalhe com fluxos de dados	107
AWS Ground Station interfaces de plano de dados	107
Usando a entrega de dados entre regiões	108
Instalar e configurar o Amazon S3	108
Configurar e configurar a Amazon VPC	108
Configuração de VPC com agente AWS Ground Station	109
Configuração de VPC com um endpoint de fluxo de dados	112
Configurar e configurar a Amazon EC2	114
Software comum fornecido	115
AWS Ground Station Imagens de máquinas da Amazon (AMIs)	115
Trabalhe com telemetria	116
Como funciona a telemetria	116
Tipos de telemetria disponíveis	117
Disponibilidade regional	117
Configurar a telemetria	117
Etapa 1: criar recursos de pré-requisitos AWS	118
Etapa 2: criar um TelemetrySinkConfig	120
Etapa 3: adicionar telemetria ao seu perfil de missão	120
Etapa 4: agendar um contato	120
Próximas etapas	121
Entenda os dados de telemetria	121
Visão geral do formato de dados	121
Telemetria apontadora	122
Telemetria de rastreamento	124
Lendo dados do stream do Kinesis Data Streams	126
Controle de versão e evolução do esquema	127
Trabalhe com contatos	128
Entenda o ciclo de vida do contato	128
AWS Ground Station status de contato	131
Retenção de dados de contato	132
Entenda o faturamento de contatos	133
Definições de largura de banda	133
Modos de agendamento	133

CancelContact	134
Cenário 1: contato único	134
Cenário 2: Contato único interrompido	135
Cenário 3: Duplicação única	135
Cenário 4: Duplicação curta	136
Cenário 5: Várias duplicatas	137
Cenário 6: Várias paradas	139
Cenário 7: Estação terrestre com várias antenas sem duplicata	140
Cenário 8: Estação terrestre com várias antenas com contatos duplicados	141
Atualizar contatos e controle de versão de contatos	142
Como funciona o controle de versão de contatos	142
Atualizando um contato	142
Status da versão de contato	144
Exemplos de código	145
Considerações	149
AWS Ground Station gêmeo digital	150
AWS Ground Station Antenas dedicadas	151
O que é uma antena dedicada	151
Visibilidade aprimorada da reserva	152
Recursos relacionados	153
Monitoramento	154
Automatize com eventos	155
AWS Ground Station Tipos de eventos	156
Cronograma do evento de contato	156
Eventos de efemérides	159
Registre chamadas de API com CloudTrail	160
AWS Ground Station Informações em CloudTrail	160
Entendendo as entradas do arquivo de AWS Ground Station log	161
Veja métricas com a Amazon CloudWatch	163
AWS Ground Station Métricas e dimensões	163
Visualizar métricas	169
Segurança	176
Gerenciamento de Identidade e Acesso	176
Público	177
Autenticação com identidades	177
Gerenciar o acesso usando políticas	178

Como AWS Ground Station funciona com o IAM	180
Exemplos de políticas baseadas em identidade	186
Solução de problemas	189
AWS políticas gerenciadas	191
AWSGroundStationAgentInstancePolicy	191
AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy	192
Atualizações da política	193
Usar perfis vinculados a serviços	194
Permissões de perfil vinculado a serviço para o Ground Station	194
Criar uma função vinculada a serviços para o Ground Station	195
Criar uma função vinculada a serviços para o Ground Station	195
Apagar uma função vinculada a serviços para o Ground Station	196
Regiões compatíveis com funções vinculadas ao serviço do Ground Station	196
Solução de problemas	196
Criptografia de dados em repouso para AWS Ground Station	197
Criar uma chave gerenciada pelo cliente	198
Especificando uma chave gerenciada pelo cliente para AWS Ground Station	200
AWS Ground Station contexto de criptografia	200
Criptografia em repouso para dados de efemérides TLE e OEM	200
Criptografia em repouso para efemérides de elevação de azimute	210
Criptografia de dados durante o trânsito para AWS Ground Station	219
AWS Ground Station Fluxos de agentes	219
Streams de endpoint de fluxo de dados	219
Exemplo de configurações de perfil de missão	220
JPSS-1 - Satélite de transmissão pública (PBS) - Avaliação	220
Satélite de transmissão pública utilizando a entrega de dados do Amazon S3	221
Caminhos de comunicação	222
AWS Ground Station configurações	224
AWS Ground Station perfil da missão	225
Juntando tudo	226
Satélite de transmissão pública utilizando um ponto final de fluxo de dados (banda estreita)	227
Caminhos de comunicação	227
AWS Ground Station configurações	234
AWS Ground Station perfil da missão	235
Juntando tudo	236

Satélite de transmissão pública utilizando um endpoint de fluxo de dados (demodulado e decodificado)	238
Caminhos de comunicação	238
AWS Ground Station configurações	245
AWS Ground Station perfil da missão	248
Juntando tudo	249
Satélite de transmissão pública utilizando AWS Ground Station Agent (banda larga)	251
Caminhos de comunicação	251
AWS Ground Station configurações	263
AWS Ground Station perfil da missão	264
Juntando tudo	265
Solução de problemas	268
Solucionar problemas de contatos que entregam dados para o Amazon EC2	268
Etapa 1: verificar se sua instância do EC2 está em execução	269
Etapa 2: Determinar o tipo de aplicativo de fluxo de dados usado	269
Etapa 3: verificar se o aplicativo de fluxo de dados está em execução	269
Etapa 4: verificar se o stream do aplicativo de fluxo de dados está configurado	271
Etapa 5: Certifique-se de ter endereços IP disponíveis suficientes na sub-rede da (s) instância (s) do receptor	273
Solucionar problemas de contatos com FALHA	274
Casos de uso FALHADOS do endpoint do Dataflow	274
AWS Ground Station Casos de uso com FALHA do agente	275
Solucionar problemas nas atualizações de contatos	276
Erros de validação síncrona	276
Códigos de falha assíncronos	278
Verificando o status de uma atualização	281
Solucionar problemas de contatos FAILED_TO_SCHEDULE	282
As configurações especificadas em sua Antenna Downlink Demod Decode Config não são suportadas.	282
Etapas gerais de solução de problemas	283
Solucione o problema que DataflowEndpointGroups não está em um estado SAUDÁVEL	283
Solucionar problemas de efemérides inválidas	283
Entendendo os erros de validação de efemérides	284
Erros comuns de validação para efemérides TLE	284
Erros comuns de validação para efemérides OEM	285
Erros comuns de validação para efemérides de elevação de azimute	286

Etapas de solução de problemas	287
Referência completa do código de erro	287
Solucionar problemas de contatos que não receberam dados	291
Configuração de downlink incorreta	292
Manobra de satélite	292
AWS Ground Station interrupção	292
Solucionar problemas de telemetria	293
Problemas comuns de configuração	293
Problemas de entrega de telemetria	296
Problemas de formato de dados	297
Como obter ajuda	298
Cotas e limites	300
Termos de serviço	301
Histórico do documento	302
AWS Glossário	308
.....	cccix

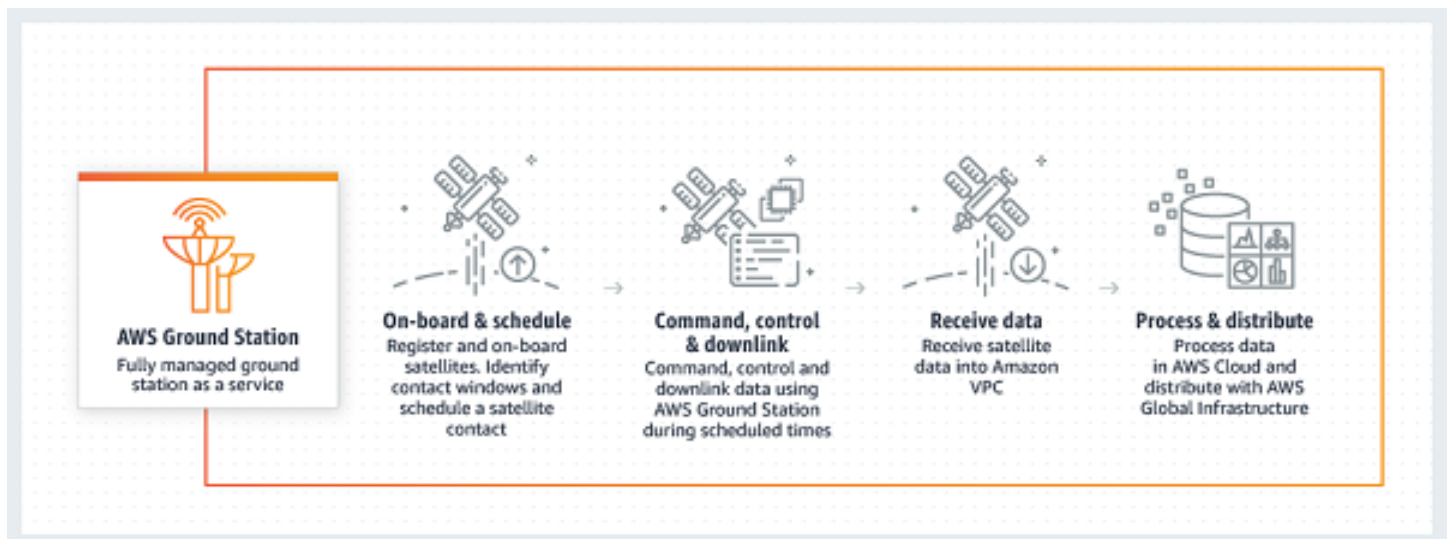
O que AWS Ground Station é

AWS Ground Station é um serviço totalmente gerenciado que fornece comunicações via satélite seguras, rápidas e previsíveis em uma infraestrutura global. Com AWS Ground Station, você não precisa mais criar, gerenciar ou escalar sua própria infraestrutura de estação terrestre. AWS Ground Station permite que você se concentre em inovar e experimentar rapidamente novos aplicativos que ingerem dados de satélite, em vez de gastar recursos na construção, operação e escalabilidade de suas próprias estações terrestres.

Usando a rede de fibra global de baixa latência e alta largura de banda da AWS, você pode começar a processar seus dados de satélite em segundos após a recepção no sistema de antena. Isso permite que você transforme dados brutos em informações processadas ou conhecimento analisado em questão de segundos.

Para organizações com requisitos especializados, AWS Ground Station também oferece [AWS Ground Station Antenas dedicadas](#) — sistemas de antenas personalizados que AWS gerencia em seu nome, fornecendo acesso dedicado às antenas criadas de acordo com suas especificações.

Casos de uso comuns



AWS Ground Station permite que você se comunique com seus satélites bidirecionalmente e oferece suporte aos seguintes casos de uso:

- **Dados de downlink** — [Receba dados de seus satélites, transmitindo frequências de banda X e banda S, entregues a uma instância do Amazon EC2 em tempo real \(formato VITA-49\) ou](#)

[diretamente para um bucket Amazon S3 em sua conta \(formato PCAP\)](#). Além disso, para satélites que usam um esquema de modulação e codificação compatível, você pode escolher entre receber dados desmodulados e decodificados ou amostras de frequência intermediária digital bruta (DigIF) (formato VITA-49).

- Dados de uplink — Envie dados e comandos para seus satélites, que recebem frequências de banda S, enviando dados DigIF (formato VITA-49) para serem transmitidos por AWS Ground Station
- Eco de uplink — Valide os comandos enviados para sua espaçonave e execute outras tarefas avançadas recebendo o sinal transmitido em uma antena fisicamente localizada.
- Rádio definido por software (SDR) /processador front-end (FEP) — Use seu SDR and/or FEP existente, capaz de ser executado em uma instância do Amazon EC2, para processar seus dados em tempo real de acordo com suas formas de onda existentes e gerar send/receive seus produtos de dados.
- Telemetria, Rastreamento e Comando (TT&C) — Execute TT&C usando uma combinação dos casos de uso listados anteriormente para gerenciar sua frota de satélites.
- Entrega de dados entre regiões — opere vários contatos simultâneos usando AWS Ground Station a rede global de antenas de uma única região da AWS.
- Digital twin — agendamento de testes, verificação de configurações e tratamento adequado de erros a um custo reduzido sem usar a capacidade da antena de produção.

Próximas etapas

Recomendamos que você comece lendo as seguintes seções:

- Para aprender AWS Ground Station conceitos essenciais, consulte [Como AWS Ground Station funciona](#).
- Para saber como configurar sua conta e seus recursos para uso AWS Ground Station, consulte [Conceitos básicos](#).
- Para usar programaticamente AWS Ground Station, consulte a Referência da [AWS Ground Station API](#). A Referência da API descreve detalhadamente todas as operações AWS Ground Station da API. Ele também fornece exemplos de solicitações, respostas e erros para os protocolos de serviços da web compatíveis. Você pode usar a [AWS CLI](#), ou um [AWS SDK](#), na linguagem de sua escolha, para escrever código que interaja com AWS Ground Station

Como AWS Ground Station funciona

AWS Ground Station opera antenas terrestres para facilitar a comunicação com seu satélite. As características físicas do que as antenas podem fazer são resumidas e chamadas de capacidades. A localização física da antena, juntamente com seus recursos atuais, podem ser referenciados na [AWS Ground Station Localizações](#) seção. Entre em contato conosco por meio do [AWS Support Center Console](#) se seu caso de uso exigir recursos adicionais, ofertas de localização adicionais ou localizações de antenas mais precisas.

Para usar uma das AWS Ground Station antenas, você deve reservar um horário em um local específico. Essa reserva é chamada de contato. Para agendar um contato com sucesso, são AWS Ground Station necessários dados adicionais para garantir seu sucesso.

- Seu satélite deve estar integrado a um ou mais locais — Isso garante que você tenha aprovação para operar os vários recursos no local solicitado.
- Seu satélite deve ter uma efeméride válida — Isso garante que as antenas tenham uma linha de visão e possam apontar com precisão para o satélite durante o contato.
- Você deve ter um perfil de missão válido — Isso permite que você personalize como esse contato se comportará, incluindo como você receberá e enviará dados para o seu satélite. Você pode utilizar vários perfis de missão para o mesmo veículo para criar contatos diferentes para se adequar a diferentes posturas operacionais ou cenários que você encontrar.

Integração via satélite

A integração de um satélite AWS Ground Station é um processo de várias etapas que envolve coleta de dados, validação técnica, licenciamento de espectro, integração e testes. A seção de [integração de satélites](#) do guia guiará você nesse processo.

Composição do perfil da missão

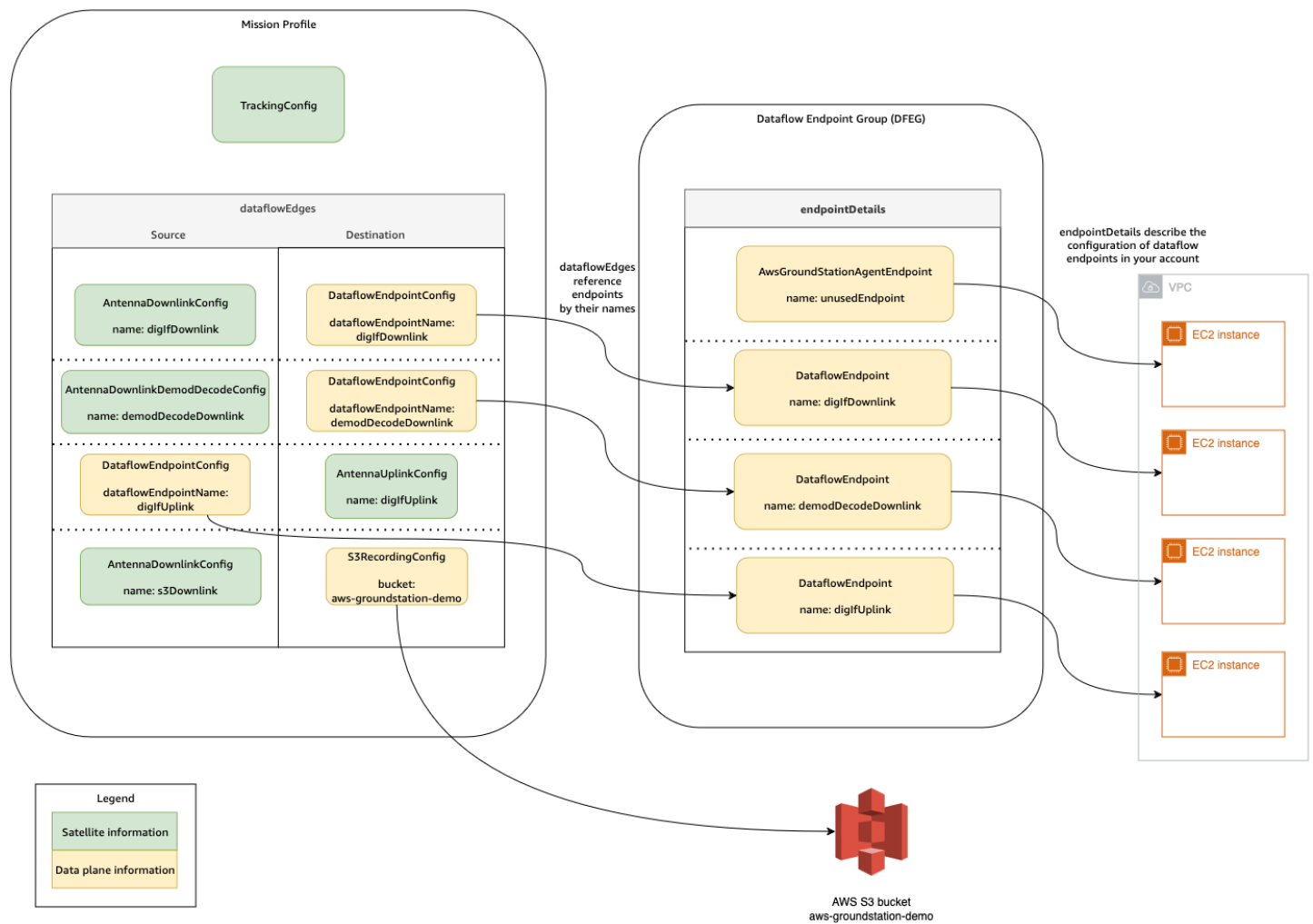
As informações de frequência do satélite, informações do [plano de dados](#) e outros detalhes são encapsulados em um perfil de missão. O perfil da missão é uma coleção de componentes de configuração. Isso permite que você reutilize componentes de configuração em diferentes perfis de missão, de acordo com seu caso de uso. Como os perfis de missão não fazem referência direta a satélites individuais, mas têm apenas informações sobre suas capacidades técnicas, os perfis de missão também podem ser reutilizados por vários satélites com a mesma configuração.

Um perfil de missão válido terá uma configuração de rastreamento e um ou mais fluxos de dados. A configuração de rastreamento especificará sua preferência de rastreamento durante um contato. Cada par de configurações em um fluxo de dados estabelece uma origem e um destino. Dependendo do seu satélite e de seus modos operacionais, o número exato de fluxos de dados variará em um perfil de missão para representar seus caminhos de comunicação de uplink e downlink, bem como quaisquer aspectos do processamento de dados.

- Para obter mais informações sobre como configurar seus recursos da Amazon VPC, Amazon S3 e EC2 Amazon que serão usados durante um contato, consulte [Trabalhe com fluxos de dados](#)
- Para obter detalhes sobre como cada configuração se comporta, consulte [Use AWS Ground Station configurações](#)
- Para obter detalhes específicos sobre todos os parâmetros esperados, consulte [Use perfis de AWS Ground Station missão](#).
- Para obter exemplos de como vários perfis de missão podem ser criados para apoiar seu caso de uso, consulte [Exemplo de configurações de perfil de missão](#).

O diagrama a seguir mostra um exemplo de perfil de missão e os recursos adicionais necessários. Observe que o exemplo mostra um endpoint de fluxo de dados que não é necessário para esse perfil de missão, chamado UnusedEndpoint, para demonstrar a flexibilidade. O exemplo é compatível com os seguintes fluxos de dados:

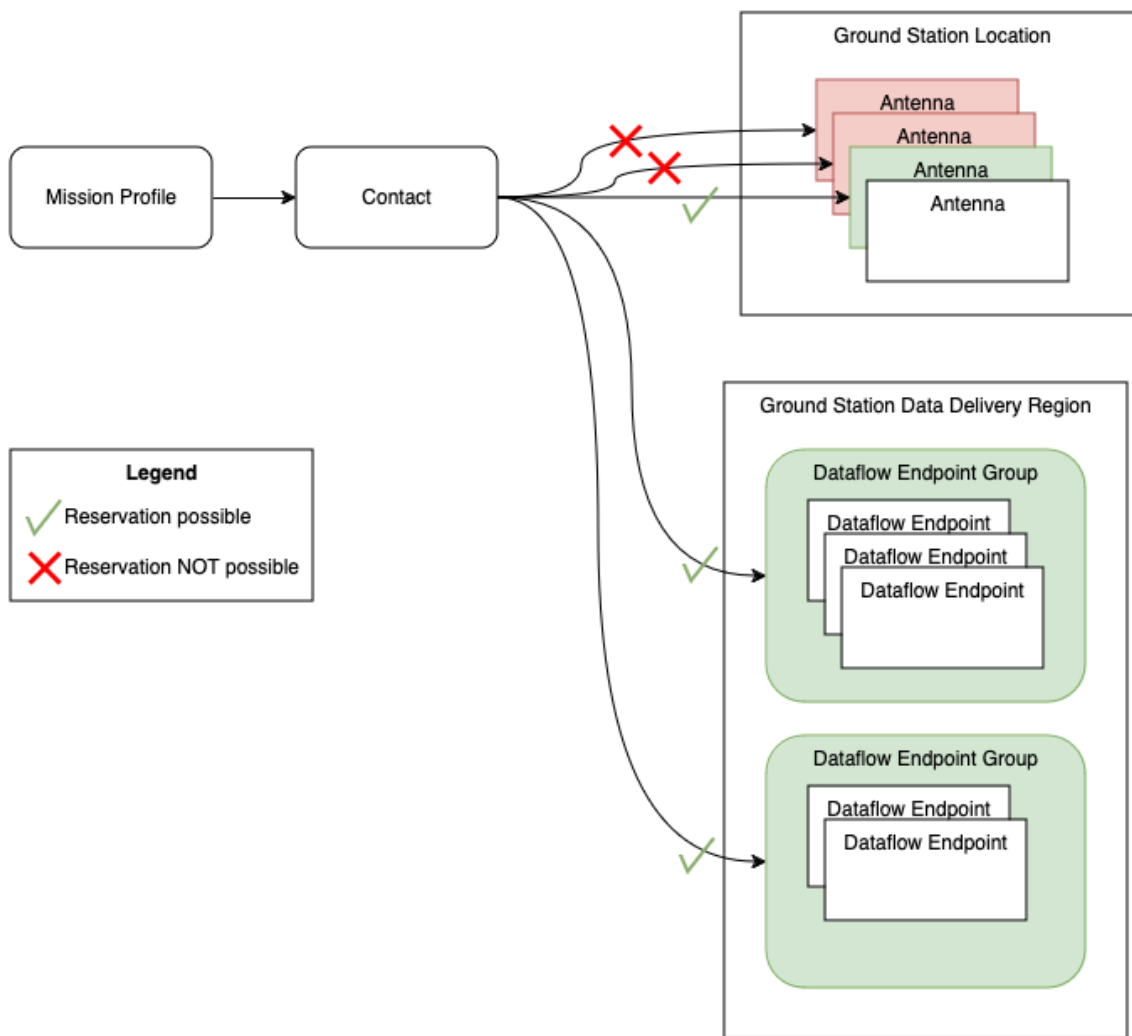
- Downlink síncrono de dados digitais de frequência intermediária para uma EC2 instância da Amazon que você gerencia. Denotado pelo nome. digIfDownlink
- Downlink assíncrono de dados digitais de frequência intermediária para um bucket Amazon S3. Indicado pelo nome do bucket. aws-groundstation-demo
- Downlink síncrono de dados desmodulados e decodificados para uma instância da Amazon EC2 que você gerencia. Denotado pelo nome. demodDecodeDownlink
- Uplink síncrono de dados de uma EC2 instância da Amazon que você gerencia para uma AWS Ground Station antena gerenciada. Denotado pelo nome. digIfUplink



Agendamento de contatos

Com um perfil de missão válido, você pode solicitar um contato com seus satélites a bordo. A solicitação de reserva de contato é assíncrona para permitir que o serviço global de antenas alcance uma programação consistente em todas as regiões envolvidas. Durante esse processo, várias antenas no local da estação terrestre solicitada são avaliadas para determinar se estão disponíveis e são capazes de processar o contato. Durante esse processo, seus endpoints de fluxo de dados configurados também são avaliados para determinar sua disponibilidade. Enquanto essa avaliação estiver ocorrendo, o status do contato estará em AGENDAMENTO.

Esse processo de agendamento assíncrono será concluído em até cinco minutos após a solicitação, mas normalmente termina em um minuto. Verifique o monitoramento baseado em eventos [Automatize AWS Ground Station com eventos](#) durante o horário de agendamento.




Contatos que podem ser realizados e têm disponibilidade resultam em contatos AGENDADOS. Com um contato agendado, os recursos necessários para realizar seu contato foram reservados nas regiões da AWS necessárias, conforme definido pelo seu perfil de missão. Contatos que não podem ser executados ou têm partes indisponíveis resultarão em contatos FAILED_TO_SCHEDULE. Consulte [Solucionar problemas de contatos FAILED_TO_SCHEDULE](#) para obter detalhes sobre a depuração.

Execução de contatos

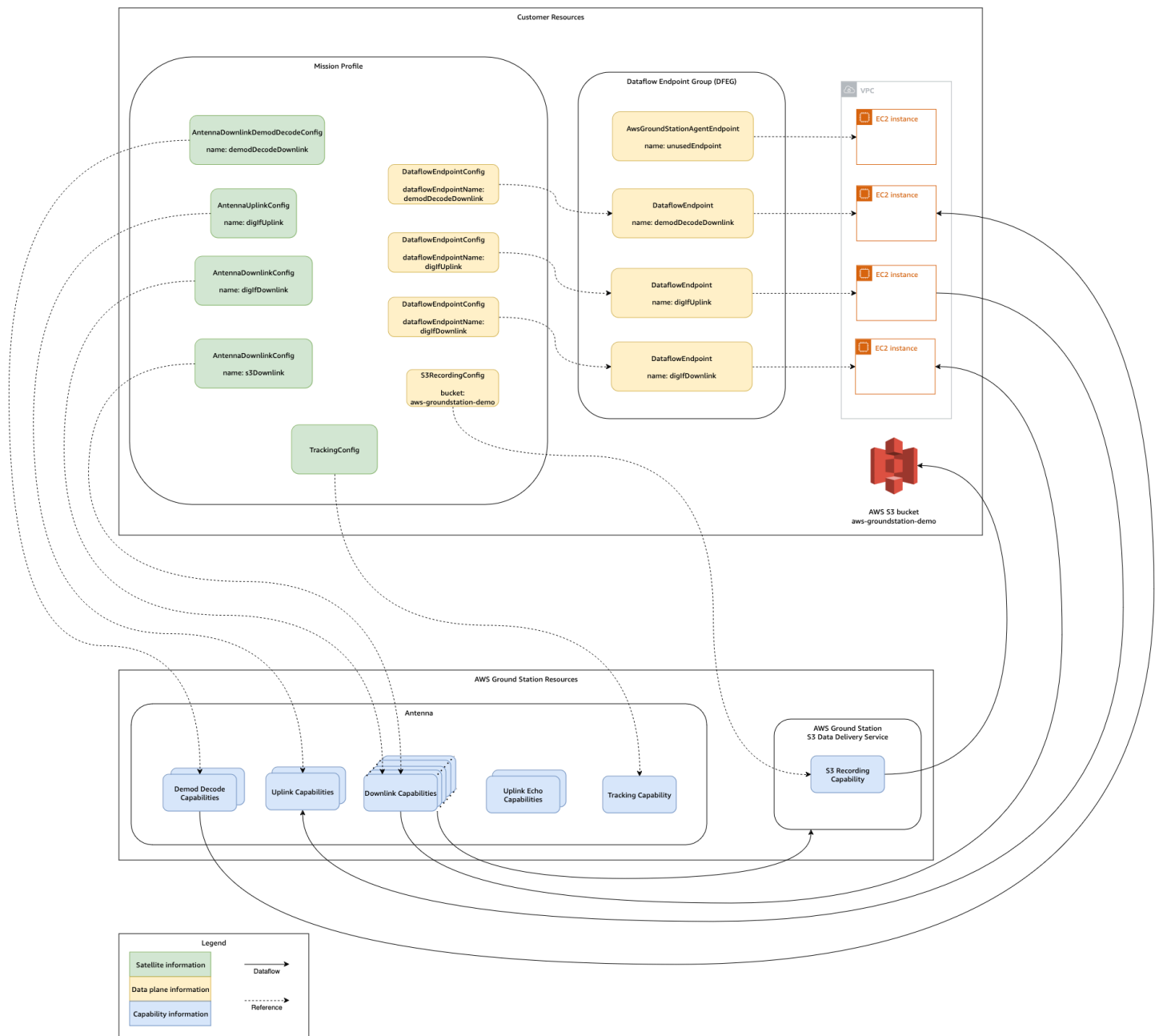
AWS Ground Station orquestrará automaticamente seus recursos gerenciados pela AWS durante sua reserva de contato. Se aplicável, você é responsável por orquestrar EC2 os recursos definidos pelo seu perfil de missão como endpoints de fluxo de dados. AWS Ground Station fornece [EventBridge eventos da AWS](#) para automatizar a orquestração de seus recursos a fim de reduzir custos. Consulte [Automatize AWS Ground Station com eventos](#) para obter mais detalhes.

Durante o contato, a telemetria sobre o desempenho do seu contato é entregue à AWS. CloudWatch Para obter informações sobre como monitorar seu contato durante a execução, consulte [Entenda o monitoramento com AWS Ground Station](#).

O diagrama a seguir dá continuidade ao exemplo anterior mostrando os mesmos recursos orquestrados durante o contato.

 Note

Nem todos os recursos da antena foram usados neste exemplo. Por exemplo, há mais de uma dúzia de recursos de downlink de antena disponíveis em cada antena que suportam várias frequências e polarizações. Para obter mais detalhes sobre o número de cada tipo de capacidade disponível nas AWS Ground Station antenas e suas frequências e polarizações suportadas, consulte. [AWS Ground Station Capacidades do site](#)



Ao final do seu contato, AWS Ground Station avaliará o desempenho do seu contato e determinará o status final do contato. Contatos em que nenhum erro for detectado resultarão em um status de contato CONCLUÍDO. Os contatos em que erros de serviço causaram problemas na entrega de dados durante o contato resultarão em um AWS_FAILED status. Contatos em que erros do cliente ou do usuário causaram problemas na entrega de dados durante o contato resultarão em um status de FALHA. Erros fora de um horário de contato, ou seja, durante a pré-aprovação ou pós-aprovação, não são levados em consideração durante a adjudicação.

Consulte [Entenda o ciclo de vida do contato](#) para obter mais informações.

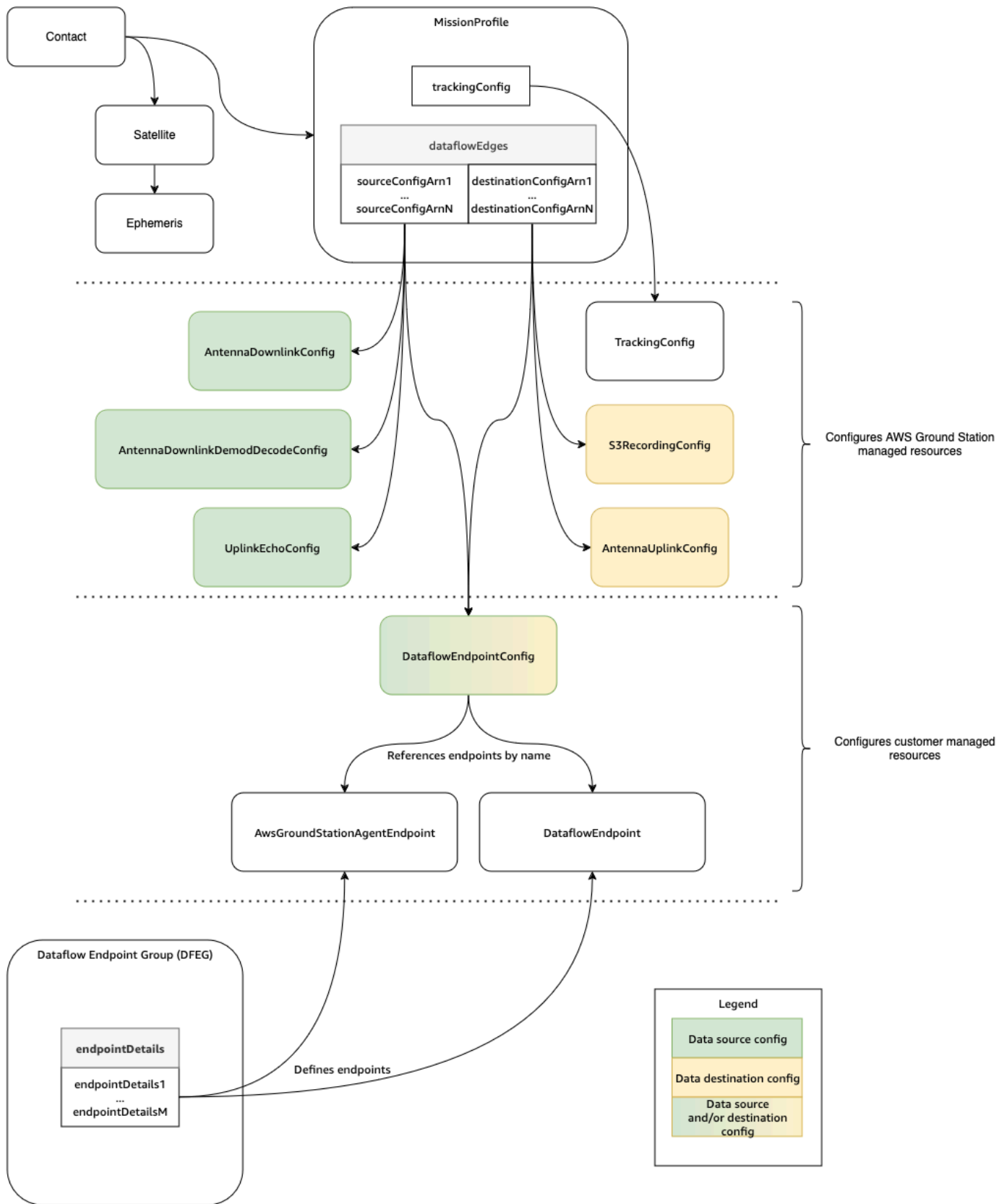
Gêmeo digital

O recurso de gêmeos digitais AWS Ground Station permite que você agende contatos em relação a localizações virtuais de estações terrestres. Essas estações terrestres virtuais são réplicas exatas das estações terrestres de produção, incluindo recursos de antena, máscaras de local e coordenadas GPS reais. O recurso digital twin permite que você teste seu fluxo de trabalho de orquestração de contatos por uma fração do custo em comparação com as estações terrestres de produção. Consulte [Use o recurso de gêmeos AWS Ground Station digitais](#) para obter mais informações.

Entenda os componentes AWS Ground Station principais

Esta seção fornece definições detalhadas dos principais componentes do AWS Ground Station.

O diagrama a seguir mostra os principais componentes AWS Ground Station e como eles se relacionam entre si. As setas indicam a direção das dependências entre os componentes, onde cada componente aponta para suas dependências.



Os tópicos a seguir descrevem detalhadamente os componentes AWS Ground Station principais.

Tópicos

- [Use perfis de AWS Ground Station missão](#)
- [Use AWS Ground Station configurações](#)
- [Use grupos AWS Ground Station de endpoints do Dataflow](#)
- [AWS Ground Station Agente de uso](#)

Use perfis de AWS Ground Station missão

Os perfis de missão contêm configurações e parâmetros de como os contatos são executados. Ao reservar um contato ou pesquisar contatos disponíveis, você fornece o perfil de missão que pretende usar. Os perfis de missão reúnem todas as suas configurações e definem para onde os dados serão direcionados durante o contato.

Os perfis de missão podem ser compartilhados entre satélites que compartilham as mesmas características de rádio. Você pode criar grupos adicionais de endpoints de fluxo de dados para limitar o máximo de contatos simultâneos que você deseja realizar para sua constelação.

As configurações de rastreamento são especificadas como um campo exclusivo no perfil da missão. As configurações de rastreamento são usadas para especificar sua preferência de usar o rastreamento de programas e o rastreamento automático durante seu contato. Para obter mais informações, consulte [Configuração de rastreamento](#).

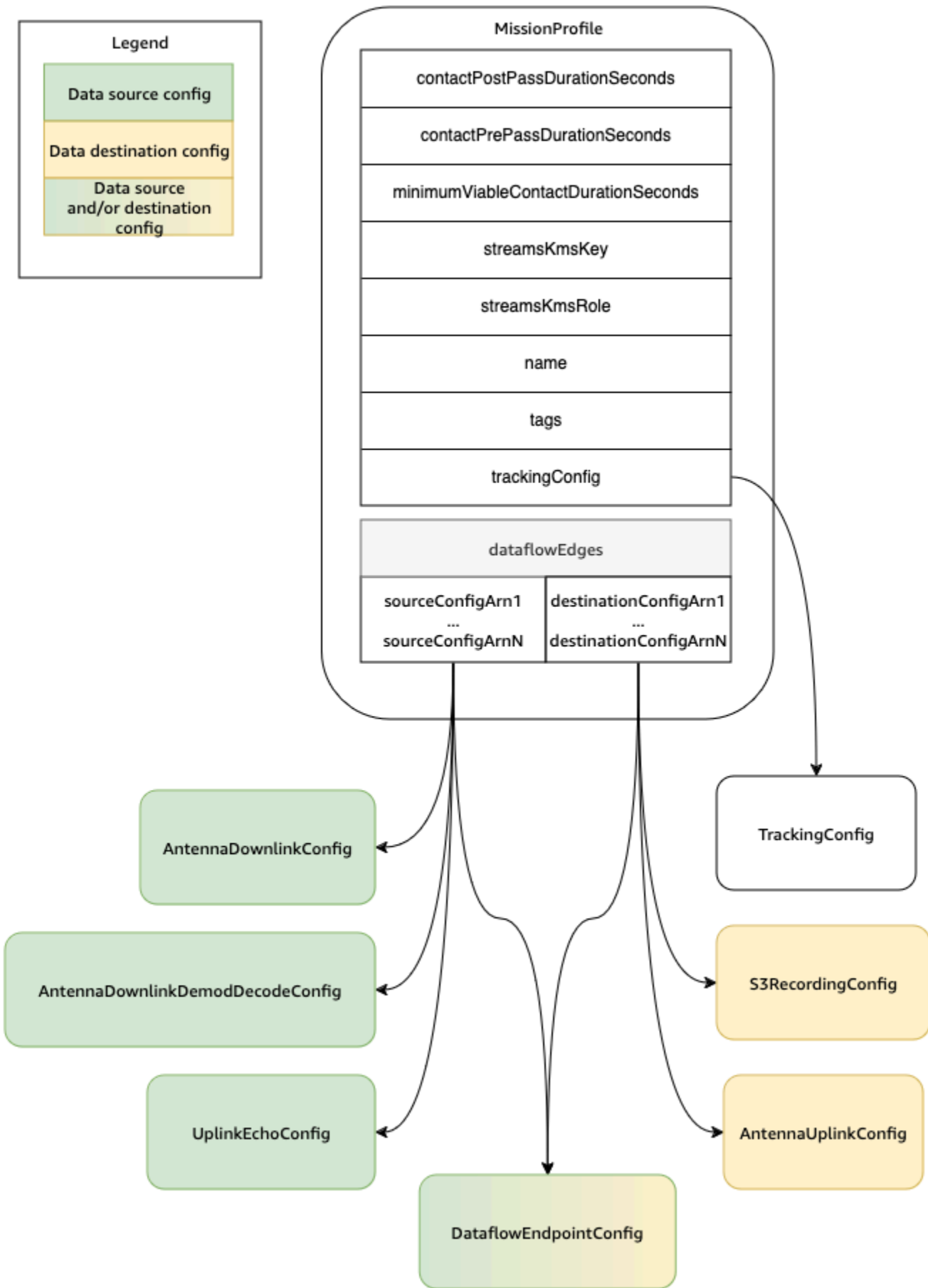
Todas as outras configurações estão contidas no `dataflowEdges` campo do perfil da missão. Essas configurações podem ser consideradas como nós de fluxo de dados, cada um representando um recurso AWS Ground Station gerenciado que pode enviar ou receber dados e sua configuração associada. O `dataflowEdges` campo define quais nós de fluxo de dados de origem e destino (configurações) são necessários. Uma única borda de fluxo de dados é uma lista de duas configurações [Amazon Resource Names ARNs \(\)](#) — a primeira é a configuração de origem e a segunda é a configuração de destino. Ao especificar uma borda de fluxo de dados entre duas configurações, você está dizendo AWS Ground Station de onde e para onde os dados devem fluir durante um contato. Para obter mais informações, consulte [Use AWS Ground Station configurações](#).

O `contactPrePassDurationSeconds` e `contactPostPassDurationSeconds` permite que você especifique horários relativos ao contato em que você receberá uma notificação de CloudWatch evento. Para obter um cronograma de eventos relacionados ao seu contato, leia [Entenda o ciclo de vida do contato](#).

O campo `name` do perfil de missão ajuda a diferenciar entre os perfis de missão criados por você.

Os `streamsKmsRole` e `streamsKmsKey` são usados para definir a criptografia usada AWS Ground Station para sua entrega de dados com o AWS Ground Station Agent. Consulte [Criptografia de dados durante o trânsito para AWS Ground Station](#).

O `telemetrySinkConfigArn` campo é opcional e permite ativar a AWS Ground Station telemetria durante os contatos. Quando especificado, AWS Ground Station transmite dados de telemetria quase em tempo real para sua conta durante a execução de seus contatos. Para obter mais informações sobre como configurar e usar a telemetria, consulte [Trabalhe com telemetria](#)



Uma lista completa de parâmetros e exemplos está incluída na documentação a seguir.

- [AWS::GroundStation::MissionProfile CloudFormation tipo de recurso](#)

Use AWS Ground Station configurações

As configurações são recursos AWS Ground Station usados para definir os parâmetros para cada aspecto do seu contato. Adicione as configurações que você deseja para um perfil de missão, e esse perfil de missão será usado ao executar o contato. Você pode definir vários tipos de configuração diferentes. As configurações podem ser agrupadas em três categorias:

- Configurações de rastreamento
- Configurações de fluxo de dados
- Configurações de telemetria

A `TrackingConfig` é o único tipo de configuração de rastreamento. Ele é usado para definir a configuração de rastreamento automático da antena durante um contato e é necessário em um perfil de missão.

As configurações que podem ser usadas em um fluxo de dados do perfil de missão podem ser consideradas como nós de fluxo de dados, cada um representando um recurso AWS Ground Station gerenciado que pode enviar ou receber dados. Um perfil de missão requer pelo menos um par dessas configurações, com uma representando uma fonte de dados e outra representando um destino. Essas configurações estão resumidas na tabela a seguir.

Nome da configuração	Origem/destino do fluxo de dados
<code>AntennaDownlinkConfig</code>	Fonte
<code>AntennaDownlinkDemodDecodeConfig</code>	Fonte
<code>UplinkEchoConfig</code>	Fonte
<code>S3 RecordingConfig</code>	Destino
<code>AntennaUplinkConfig</code>	Destino
<code>DataflowEndpointConfig</code>	and/or Destino de origem

A `TelemetrySinkConfig` é o único tipo de configuração de telemetria. Ele é usado para configurar onde os dados de telemetria serão entregues durante um contato e é opcional em um perfil de missão. Quando incluído, AWS Ground Station transmite telemetria quase em tempo real para sua conta durante a execução de seus contatos.

Consulte a documentação a seguir para obter mais informações sobre como realizar operações em configurações usando CloudFormation a AWS Command Line Interface API ou a AWS Ground Station API. Links para documentação de tipos de configuração específicos também são fornecidos abaixo.

- [AWS::GroundStation::Config CloudFormation tipo de recurso](#)
- [Referência de configuração AWS CLI](#)
- [Referência de configuração da API](#)

Configuração de rastreamento

Você pode usar configurações de rastreamento no perfil de missão para determinar se `autotrack` deve ser habilitado durante seus contatos. Essa configuração tem um único parâmetro: `autotrack`. O parâmetro `autotrack` pode ter os seguintes valores:

- `REQUIRED`: o `autotrack` é necessário para seus contatos.
- `PREFERRED`: o `autotrack` é preferido para contatos, mas os contatos ainda podem ser executados sem o `autotrack`.
- `REMOVED`: o `autotrack` deve ser usado para seus contatos.

AWS Ground Station utilizará rastreamento programático que apontará com base em suas efemérides quando o rastreamento automático não for usado. Consulte [Entenda como AWS Ground Station usa efemérides](#) para obter detalhes sobre como as efemérides são construídas.

O `Autotrack` usará o rastreamento do programa até que o sinal esperado seja encontrado. Quando isso ocorrer, ele continuará rastreando com base na intensidade do sinal.

Consulte a documentação a seguir para obter mais informações sobre como realizar operações em configurações de rastreamento usando CloudFormation a AWS Command Line Interface API ou a AWS Ground Station API.

- [AWS::GroundStation::Config TrackingConfig CloudFormation propriedade](#)

- [AWS CLI Referência de configuração](#) (consulte a trackingConfig -> (structure) seção)
- [TrackingConfig Referência de API](#)

Configuração de downlink de antena

Você pode usar as configurações de downlink da antena para configurar a antena durante o contato. Elas consistem em uma configuração espectral que especifica a largura de banda, a frequência e polarização que devem ser usadas durante o downlink de antena.

Essa configuração representa um nó de origem em um fluxo de dados. É responsável pela digitalização dos dados de radiofrequência. Os dados transmitidos desse nó seguirão o Data/IP formato do sinal. Para obter informações mais detalhadas sobre como criar fluxos de dados com essa configuração, consulte [Trabalhe com fluxos de dados](#)

Se seu caso de uso de downlink exigir demodulação ou decodificação, consulte [Configuração de decodificação de demodulação de downlink de antena](#).

Consulte a documentação a seguir para obter mais informações sobre como realizar operações nas configurações de downlink de antena usando CloudFormation AWS Command Line Interface, a ou a API. AWS Ground Station

- [AWS::GroundStation::Config AntennaDownlinkConfig CloudFormation propriedade](#)
- [AWS CLI Referência de configuração](#) (consulte a antennaDownlinkConfig -> (structure) seção)
- [AntennaDownlinkConfig Referência de API](#)

Configuração de decodificação de demodulação de downlink de antena

As configurações de decodificação de demodulação de downlink de antena são um tipo de configuração mais complexo e personalizável que você pode usar para executar contatos de downlink com decodificação de demodulação. and/or Se você estiver interessado em executar esses tipos de contatos, abra um AWS Support ticket por meio do [AWS Support Center Console](#). Nós lhe ajudaremos a definir a configuração e o perfil de missão certos para seu caso de uso.

Essa configuração representa um nó de origem em um fluxo de dados. Ele é responsável por digitalizar os dados de radiofrequência e realizar a desmodulação e decodificação conforme especificado. Os dados transmitidos desse nó seguirão o formato de Demodulated/Decoded

dados/IP. Para obter informações mais detalhadas sobre como criar fluxos de dados com essa configuração, consulte [Trabalhe com fluxos de dados](#)

Consulte a documentação a seguir para obter mais informações sobre como realizar operações nas configurações de decodificação de demod de downlink de antena usando CloudFormation, a ou a AWS Command Line Interface API. AWS Ground Station

- [AWS::GroundStation::Config AntennaDownlinkDemodDecodeConfig CloudFormation propriedade](#)
- [AWS CLI Referência de configuração](#) (consulte a `antennaDownlinkDemodDecodeConfig` -> (`structure`) seção)
- [AntennaDownlinkDemodDecodeConfig Referência da API](#)

Configuração de uplink de antena

Você pode usar configurações de uplink de antena para configurar a antena durante o contato uplink. Eles consistem em uma configuração de espectro com frequência, polarização e potência radiada isotrópica efetiva alvo (EIRP). Para obter informações sobre como configurar eco uplink, consulte [Configuração de eco de uplink de antena](#).

Essa configuração representa um nó de destino em um fluxo de dados. Ele converterá o sinal de dados de radiofrequência digitalizado fornecido em um sinal analógico e o emitirá para o seu satélite receber. Espera-se que os dados transmitidos para esse nó atendam ao Data/IP formato de sinal. Para obter informações mais detalhadas sobre como criar fluxos de dados com essa configuração, consulte [Trabalhe com fluxos de dados](#)

Consulte a documentação a seguir para obter mais informações sobre como realizar operações nas configurações de uplink de antena usando CloudFormation AWS Command Line Interface, a ou a API. AWS Ground Station

- [AWS::GroundStation::Config AntennaUplinkConfig CloudFormation propriedade](#)
- [AWS CLI Referência de configuração](#) (consulte a `antennaUplinkConfig` -> (`structure`) seção)
- [AntennaUplinkConfig Referência de API](#)

Configuração de eco de uplink de antena

As configurações de eco de uplink informam à antena como executar um eco de uplink. Um eco de uplink pode ser usado para validar comandos enviados para sua espaçonave e realizar outras

tarefas avançadas. Isso é obtido gravando o sinal real transmitido pela AWS Ground Station antena (ou seja, o uplink). Isso ecoa o sinal enviado pela antena de volta ao ponto final do fluxo de dados e deve corresponder ao sinal transmitido. A configuração de eco de uplink contém o ARN de uma configuração de uplink. A antena usa os parâmetros da configuração de uplink apontada pelo ARN ao executar um eco de uplink.

Essa configuração representa um nó de origem em um fluxo de dados. Os dados transmitidos desse nó atenderão ao Data/IP formato de sinal. Para obter informações mais detalhadas sobre como criar fluxos de dados com essa configuração, consulte [Trabalhe com fluxos de dados](#)

Consulte a documentação a seguir para obter mais informações sobre como realizar operações em configurações de eco de uplink usando CloudFormation AWS Command Line Interface, a ou a API.
AWS Ground Station

- [AWS::GroundStation::Config UplinkEchoConfig CloudFormation propriedade](#)
- [AWS CLI Referência de configuração](#) (consulte a `uplinkEchoConfig` -> (structure) seção)
- [UplinkEchoConfig Referência de API](#)

Configuração de endpoint de fluxo de dados

Note

As configurações de endpoint do Dataflow são usadas somente para entrega de dados para o Amazon EC2 e não para entrega de dados para o Amazon S3.

Você pode usar as configurações de endpoint de fluxo de dados para especificar qual endpoint de fluxo de dados em um [grupo de endpoints de fluxo de dados](#) do qual ou para o qual você deseja que os dados fluam durante um contato. Os dois parâmetros de uma configuração de endpoint do fluxo de dados especificam o nome e a região do endpoint do fluxo de dados. Ao reservar um contato, AWS Ground Station analisa o [perfil de missão](#) que você especificou e tenta encontrar um grupo de endpoints de fluxo de dados AWS na região que contenha todos os endpoints de fluxo de dados especificados pelas configurações de endpoint de fluxo de dados contidas em seu perfil de missão. Se um grupo de endpoints de fluxo de dados adequado for encontrado, o status do contato será SCHEDULADO, caso contrário, se tornará FAILED_TO_SCHEDULE. Para obter mais informações sobre os possíveis status de um contato, consulte [AWS Ground Station status de contato](#).

A propriedade `dataflowEndpointName` do endpoint em uma configuração de ponto de extremidade de fluxo de dados especifica para qual endpoint de fluxo de dados em um grupo de endpoints de fluxo de dados os dados fluirão durante um contato.

A propriedade `dataflowEndpointRegion` especifica em qual região o endpoint do fluxo de dados reside. Se uma região for especificada na configuração do endpoint do fluxo de dados, AWS Ground Station procurará um endpoint do fluxo de dados na região especificada. Se nenhuma região for especificada, o padrão AWS Ground Station será a região da estação terrestre do contato. Um contato é considerado um contato de entrega de dados entre regiões se a região do seu endpoint de fluxo de dados não for a mesma da região da estação terrestre do contato. Consulte [Trabalhe com fluxos de dados](#) para obter mais informações sobre fluxos de dados entre regiões.

Veja [Use grupos AWS Ground Station de endpoints do Dataflow](#) dicas sobre como diferentes esquemas de nomenclatura para seus fluxos de dados podem beneficiar seu caso de uso.

Para obter informações mais detalhadas sobre como criar fluxos de dados com essa configuração, consulte [Trabalhe com fluxos de dados](#)

Consulte a documentação a seguir para obter mais informações sobre como realizar operações em configurações de endpoint de fluxo de dados usando CloudFormation AWS Command Line Interface, a ou a API. AWS Ground Station

- [AWS::GroundStation::Config DataflowEndpointConfig CloudFormation propriedade](#)
- [AWS CLI Referência de configuração](#) (consulte a `dataflowEndpointConfig` -> `(structure)` seção)
- [DataflowEndpointConfig Referência de API](#)

Config de gravação do Amazon S3

Note

As configurações de gravação do Amazon S3 são usadas somente para entrega de dados para o Amazon S3 e não são usadas para entrega de dados para o Amazon EC2.

Essa configuração representa um nó de destino em um fluxo de dados. Esse nó encapsulará os dados recebidos do nó de origem do fluxo de dados em dados pcap. Para obter informações mais

detalhadas sobre como criar fluxos de dados com essa configuração, consulte [Trabalhe com fluxos de dados](#)

Você pode usar as configurações de gravação do S3 para especificar um bucket do Amazon S3 para o qual você deseja que os dados baixados sejam entregues junto com a convenção de nomenclatura usada. O seguinte especifica restrições e detalhes sobre esses parâmetros:

- O nome do bucket do Amazon S3 deve começar com `aws-groundstation`.
- O perfil do IAM deve ter uma política de confiança que permita à entidade principal do serviço `groundstation.amazonaws.com` assumir o perfil. Veja a seção [Exemplo de Política de Confiança](#) abaixo para um exemplo. Durante a criação da configuração, o ID do recurso de configuração não existe, a política de confiança deve usar um asterisco (*) no lugar `your-config-id` e pode ser atualizada após a criação com o ID do recurso de configuração.

Exemplo de política de confiança

Para obter mais informações sobre como atualizar a política de confiança de um perfil, consulte [Gerenciar perfis do IAM](#) no Guia do usuário do IAM.

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"
        }
      }
    }
  ]
}
```

```
]
}
```

- O perfil do IAM deve ter uma política do IAM que permita que a função execute a ação `s3:GetBucketLocation` no bucket e a ação `s3:PutObject` nos objetos do bucket. Se o bucket do Amazon S3 tiver uma política de bucket, a política também deverá permitir que o perfil do IAM execute essas ações. Veja a seção [Exemplo de Política de Confiança](#) abaixo para um exemplo.

Exemplo de política de funções

Para obter mais informações sobre como atualizar a política de confiança de uma função, consulte [Gerenciar políticas do IAM](#) no Guia do usuário do IAM.

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/*"
      ]
    }
  ]
}
```

- O prefixo será usado ao nomear o objeto de dados do S3. Você pode especificar chaves opcionais para substituição, esses valores serão substituídos pelas informações correspondentes dos seus detalhes de contato. Por exemplo, um prefixo de `{satellite_id}/{year}/{month}/{day}` será substituído e resultaria em uma saída como `fake_satellite_id/2021/01/10`

Teclas opcionais para substituição: `{satellite_id} | {config-name} | {config-id} | {year} | {month} | {day}`

Consulte a documentação a seguir para obter mais informações sobre como realizar operações nas configurações de gravação do S3 usando a CloudFormation API ou a AWS Command Line Interface AWS Ground Station API.

- [AWS::GroundStation::Config Propriedade S3 RecordingConfig CloudFormation](#)
- [AWS CLI Referência de configuração](#) (consulte a `s3RecordingConfig` -> (structure) seção)
- [Referência da RecordingConfig API S3](#)

Configuração do coletor de telemetria

Você pode usar as configurações do coletor de telemetria para especificar onde deseja que os dados de telemetria sejam entregues durante os contatos do satélite. A configuração do coletor de telemetria é opcional e é adicionada ao seu perfil de missão para agendar contatos habilitados para telemetria. O seguinte especifica restrições e detalhes sobre esses parâmetros:

- O perfil do IAM deve ter uma política de confiança que permita à entidade principal do serviço `groundstation.amazonaws.com` assumir o perfil. Veja a seção [Exemplo de Política de Confiança](#) abaixo para um exemplo.

Exemplo de política de confiança

Para obter mais informações sobre como atualizar a política de confiança de um perfil, consulte [Gerenciar perfis do IAM](#) no Guia do usuário do IAM.

```
{
```

```

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

```

- A função do IAM deve ter uma política do IAM que permita que a função execute `kinesis:PutRecords` as ações `kinesis:PutRecord` e no stream. `kinesis:DescribeStream` Veja a seção [Exemplo de Política de Confiança](#) abaixo para um exemplo.

Exemplo de política de funções

Para obter mais informações sobre como atualizar a política de confiança de uma função, consulte [Gerenciar políticas do IAM](#) no Guia do usuário do IAM.

```

{
  "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"
    }
  ]
}

```

Quando você inclui uma configuração de coletor de telemetria em seu perfil de missão, AWS Ground Station transmitirá dados de telemetria para sua conta durante os contatos. Para obter mais informações sobre tipos de telemetria, formato de dados e configuração dos AWS recursos necessários, consulte. [Trabalhe com telemetria](#)

Consulte a documentação a seguir para obter mais informações sobre como realizar operações nas configurações do coletor de telemetria usando CloudFormation AWS Command Line Interface, a ou a API. AWS Ground Station

- [AWS::GroundStation::Config TelemetrySinkConfig CloudFormation propriedade](#)
- [AWS CLI Referência de configuração](#) (consulte a `telemetrySinkConfig` -> (structure) seção)
- [TelemetrySinkConfig Referência da API](#)

Use grupos AWS Ground Station de endpoints do Dataflow

Os endpoints do fluxo de dados definem o local em que você deseja que os dados sejam transmitidos de forma síncrona de ou para onde você deseja que os dados sejam transmitidos de forma síncrona durante os contatos. Os endpoints de fluxo de dados sempre são criados como parte de um grupo de endpoints de fluxo de dados. Com a inclusão de vários endpoints de fluxo de dados em um grupo, você está afirmando que todos os endpoints especificados podem ser usados juntos durante um único contato. Por exemplo, se um contato precisar enviar dados para três endpoints de fluxo de dados separados, você deve ter três endpoints em um único grupo de endpoints de fluxo de dados que correspondam às configurações do endpoint de fluxo de dados em seu perfil de missão.

Versões do grupo de endpoints do Dataflow

AWS Ground Station oferece suporte a duas versões de grupos de endpoints de fluxo de dados:

- `DataflowEndpointGroup` - [A implementação original que suporta uplink e downlink usando um endpoint de fluxo de dados e somente downlink para um endpoint de agente AWS Ground Station](#)
- `DataflowEndpointGroupV2` - Versão atualizada que suporta fluxos de dados de uplink e downlink para endpoints do AWS Ground Station Agent com maior clareza e funcionalidade

Comparação de grupos de endpoints do Dataflow

Recurso	<code>DataflowEndpointGroup</code>	<code>DataflowEndpointGroupV2</code>
Tipos de endpoints compatíveis	<code>DataflowEndpoint</code> , <code>AwsGroundStationAgentEndpoint</code>	<code>DownlinkAwsGroundStationAgentEndpoint</code> , <code>UplinkAwsGroundStationAgentEndpoint</code>

Recurso	DataflowEndpointGroup	DataflowEndpointGroupV2
Endpoints que suportam uplink	DataflowEndpoint	UplinkAwsGroundStationAgentEndpoint
Endpoints com suporte para downlink	DataflowEndpoint, AwsGroundStationAgentEndpoint	DownlinkAwsGroundStationAgentEndpoint

DataflowEndpointGroup V2 foi criada para oferecer suporte a fluxos de dados de uplink e tornar mais clara a linguagem em torno dos grupos de endpoints de fluxo de dados. Recomendamos o uso [UplinkAwsGroundStationAgentEndpoint](#) de [DownlinkAwsGroundStationAgentEndpoint](#) terminais com uma [DataflowEndpointGroupV2](#) para todos os novos casos de uso. DataflowEndpointGroup continua suportado para compatibilidade com versões anteriores, mas a DataflowEndpointGroup V2 fornece funcionalidade aprimorada e opções de configuração mais claras.

Tip

Os endpoints do fluxo de dados são identificados por um nome de sua escolha ao executar contatos. Esses nomes não precisam ser exclusivos em toda a conta. Isso permite que vários contatos em diferentes satélites e antenas sejam executados ao mesmo tempo usando o mesmo perfil de missão. Isso pode ser útil se você tiver uma constelação de satélites com as mesmas características operacionais. Você pode escalar o número de grupos de endpoints de fluxo de dados para caber no número máximo de contatos simultâneos que sua constelação de satélites exige.

Quando um ou mais recursos em um grupo de endpoints de fluxo de dados está em uso para um contato, todo o grupo é reservado para a duração desse contato. É possível executar vários contatos por vez, mas esses contatos devem ser executados em diferentes grupos de endpoints de fluxo de dados.

Important

Os grupos de endpoints do Dataflow devem estar em condições HEALTHY de programar contatos com eles. Para obter informações sobre como solucionar problemas em grupos de

endpoints de fluxo de dados que não estão em um HEALTHY estado, consulte. [Solucione o problema que DataflowEndpointGroups não está em um estado SAUDÁVEL](#)

Consulte a documentação a seguir para obter mais informações sobre como realizar operações em grupos de endpoints de fluxo de dados usando a CloudFormation API ou a AWS Command Line Interface API. AWS Ground Station

- [AWS::GroundStation::DataflowEndpointGroup CloudFormation tipo de recurso](#)
- [Referência do Dataflow Endpoint Group AWS CLI](#)
- [Referência da API do Dataflow Endpoint Group](#)

Endpoints de fluxo de dados

Os membros de um grupo de endpoints de fluxo de dados são endpoints de fluxo de dados. Os tipos de endpoint compatíveis dependem da versão do grupo de endpoints de fluxo de dados que você usa.

DataflowEndpointGroup endpoints

DataflowEndpointGroup [suporta uplink e downlink usando um endpoint de fluxo de dados e somente downlink para um endpoint de agente.](#) [AWS Ground Station](#) Para os dois tipos de endpoints, você criará as construções de suporte (por exemplo, endereços IP) antes de criar o grupo de endpoints do fluxo de dados. Consulte [Trabalhe com fluxos de dados](#) as recomendações sobre qual tipo de endpoint de fluxo de dados usar e como configurar as construções de suporte.

As seções a seguir descrevem os dois tipos de endpoints compatíveis.

Important

Todos os endpoints de fluxo de dados em um único grupo de endpoints de fluxo de dados devem ser do mesmo tipo. Você não pode misturar [endpoints do AWS Ground Station Agent com endpoints](#) do [Dataflow no mesmo grupo](#). Se seu caso de uso exigir os dois tipos de endpoints, você deverá criar grupos de endpoints de fluxo de dados separados para cada tipo.

Para DataflowEndpointGroup V2, você pode misturar [UplinkAwsGroundStationAgentEndpoint](#) e [DownlinkAwsGroundStationAgentEndpoint](#) no mesmo grupo.

AWS Ground Station Endpoint do agente

O AWS Ground Station Agent Endpoint utiliza o AWS Ground Station Agente como um componente de software para encerrar conexões. Para construir um AWS Ground Station Agent Endpoint, você só preencherá o `AwsGroundStationAgentEndpoint` campo do `EndpointDetails`. Para obter mais informações sobre o AWS Ground Station agente, consulte o [Guia do usuário completo do AWS Ground Station agente](#).

O `AwsGroundStationAgentEndpoint` contém o seguinte:

- `Name`- O nome do endpoint do fluxo de dados. Para que o contato use esse endpoint de fluxo de dados, esse nome deve corresponder ao nome usado na configuração do endpoint de fluxo de dados.
- `EgressAddress`- O endereço IP e da porta usados para extrair dados do Agente.
- `IngressAddress`- O endereço IP e da porta usados para inserir dados no Agente.

Endpoint de fluxo de dados

O Dataflow Endpoint utiliza um aplicativo de rede como componente de software para encerrar conexões. Use o Dataflow Endpoint quando quiser fazer uplink de dados de sinal digital, downlink de menos de 50% de dados de sinal digital ou downlink de dados MHz de sinal. Demodulated/Decoded Para criar um endpoint do Dataflow, você preencherá os campos `Endpoint` e `Security Details` do `EndpointDetails`.

O `Endpoint` contém o seguinte:

- `Name`- O nome do endpoint do fluxo de dados. Para que o contato use esse endpoint de fluxo de dados, esse nome deve corresponder ao nome usado na configuração do endpoint de fluxo de dados.
- `Address`- O endereço IP e da porta usados.

O `SecurityDetails` contém o seguinte:

- `roleArn`- O Amazon Resource Name (ARN) de uma função que AWS Ground Station assumirá a criação de interfaces de rede elástica (ENIs) em sua VPC. Eles ENIs servem como pontos de entrada e saída de dados transmitidos durante um contato.
- `securityGroupIds`: os grupos de segurança a serem anexados às interfaces de rede elástica.

- `subnetIds`- Uma lista de sub-redes nas quais você AWS Ground Station pode colocar interfaces de rede elásticas para enviar fluxos para suas instâncias. Se várias sub-redes forem especificadas, elas deverão ser roteáveis entre si. Se as sub-redes estiverem em zonas de disponibilidade diferentes (AZs), você poderá incorrer em cobranças de transferência de dados entre AZ.

O perfil do IAM transmitido no `roleArn` deve ter uma política de confiança que permita à entidade principal do serviço `groundstation.amazonaws.com` assumir o perfil. Veja a seção [Exemplo de Política de Confiança](#) abaixo para um exemplo. Durante a criação do endpoint, o ID do recurso do endpoint não existe, portanto, a política de confiança deve usar um asterisco (*) no lugar de `your-endpoint-id`. Isso pode ser atualizado após a criação para usar o ID do recurso do endpoint a fim de definir o escopo da política de confiança para esse grupo específico de endpoints do fluxo de dados.

A função do IAM deve ter uma política do IAM que AWS Ground Station permita configurar ENIs. o. Veja a seção [Exemplo de Política de Confiança](#) abaixo para um exemplo.

Exemplo de política de confiança

Para obter mais informações sobre como atualizar a política de confiança de um perfil, consulte [Gerenciar perfis do IAM](#) no Guia do usuário do IAM.

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"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

Exemplo de política de funções

Para obter mais informações sobre como atualizar a política de confiança de uma função, consulte [Gerenciar políticas do IAM](#) no Guia do usuário do IAM.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2>DeleteNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2>DeleteNetworkInterfacePermission",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    }
  ]
}

```

DataflowEndpointGroupEndpoints V2

DataflowEndpointGroup V2 apresenta tipos de endpoints especializados que fornecem uma configuração mais clara e uma funcionalidade aprimorada:

- [UplinkAwsGroundStationAgentEndpoint](#)- Otimizado para fluxos de dados de uplink
- [DownlinkAwsGroundStationAgentEndpoint](#)- Otimizado para fluxos de dados de downlink

Esses endpoints especializados substituem o genérico [AwsGroundStationAgentEndpoint](#) por configurações específicas de direção que facilitam a configuração e o gerenciamento de seus fluxos de dados.

Ponto final do Uplink Agent AWS Ground Station

O foi [UplinkAwsGroundStationAgentEndpoint](#) projetado especificamente para fluxos de dados de uplink e fornece opções de configuração mais claras. Use esse tipo de endpoint quando precisar fornecer dados para AWS Ground Station serem conectados ao seu satélite.

O [UplinkAwsGroundStationAgentEndpoint](#) contém o seguinte:

- **Name**- O nome do endpoint do fluxo de dados. Para que o contato use esse endpoint de fluxo de dados, esse nome deve corresponder ao nome usado na configuração do endpoint de fluxo de dados.
- **IngressAddressAndPort**- IP único e endereço de porta para entrada de dados para o agente
- **AgentIpAndPortAddress**- Intervalo de portas para comunicação com agentes

Endpoint do AWS Ground Station agente de downlink

O [DownlinkAwsGroundStationAgentEndpoint](#) é otimizado para fluxos de dados de downlink, incluindo downlink de banda estreita, demodulação/decodificação de banda larga e cenários de eco de uplink.

O [DownlinkAwsGroundStationAgentEndpoint](#) contém o seguinte:

- **Name**- O nome do endpoint do fluxo de dados. Para que o contato use esse endpoint de fluxo de dados, esse nome deve corresponder ao nome usado na configuração do endpoint de fluxo de dados.
- **EgressAddressAndPort**- IP único e endereço de porta para saída de dados do agente
- **AgentIpAndPortAddress**- Intervalo de portas para comunicação com agentes

Criação de grupos de endpoints de fluxo de dados

Você pode criar grupos de endpoints de fluxo de dados usando qualquer uma das versões:

CreateDataflowEndpointGroup

Use [CreateDataflowEndpointGroup](#) para compatibilidade com versões anteriores ou quando precisar usar o genérico [AwsGroundStationAgentEndpoint](#) ou os [DataflowEndpoint](#) tipos.

CreateDataflowEndpointGroupV2

Use a [CreateDataflowEndpointGroupV2](#) para novas implementações para aproveitar os tipos de endpoints especializados que oferecem suporte a fluxos de dados de uplink e downlink. Essa API suporta apenas [UplinkAwsGroundStationAgentEndpointDownlinkAwsGroundStationAgentEndpoint](#).

Considerações sobre a migração

Se você estiver usando atualmente DataflowEndpointGroup, você pode continuar usando sua configuração existente sem alterações. AWS Ground Station mantém total compatibilidade com versões anteriores.

Se você quiser migrar para usar a nova DataflowEndpointGroup V2 e atualmente estiver usando um [DataflowEndpoint](#) aplicativo Dataflow Endpoint para receber seus dados, precisará migrar para usar o Agente em vez disso. AWS Ground Station Se você já usa um AWS Ground Station agente para downlink, também pode usar a mesma instância de agente para uplink. Nenhuma instância adicional de agente é necessária.

Para migrar para a DataflowEndpointGroup V2:

1. Se estiver migrando de DataflowEndpoint, configure o AWS Ground Station Agente seguindo o Guia do [Usuário do AWS Ground Station Agente](#)
2. Identifique a direção do fluxo de dados e crie o tipo de endpoint apropriado (ou) [UplinkAwsGroundStationAgentEndpointDownlinkAwsGroundStationAgentEndpoint](#)
3. Crie a [DataflowEndpointGroupV2](#) referenciando esses endpoints
4. Crie uma nova [configuração de endpoint de fluxo de dados](#) que faça referência à nova V2 pelo nome DataflowEndpointGroup
5. Crie um novo perfil de missão que faça referência à configuração do endpoint do fluxo de dados como uma borda do fluxo de dados
6. Use o novo perfil da missão para agendar contatos
7. Teste sua configuração antes de implantá-la na produção

Para obter mais informações sobre o fluxo de trabalho completo, consulte [Entenda os componentes AWS Ground Station principais](#) [Crie configurações](#) e.

AWS Ground Station Agente de uso

O AWS Ground Station agente permite que você receba (downlink) fluxos de dados síncronos de frequência intermediária digital de banda larga (DigiF) durante os contatos do AWS Ground Station.

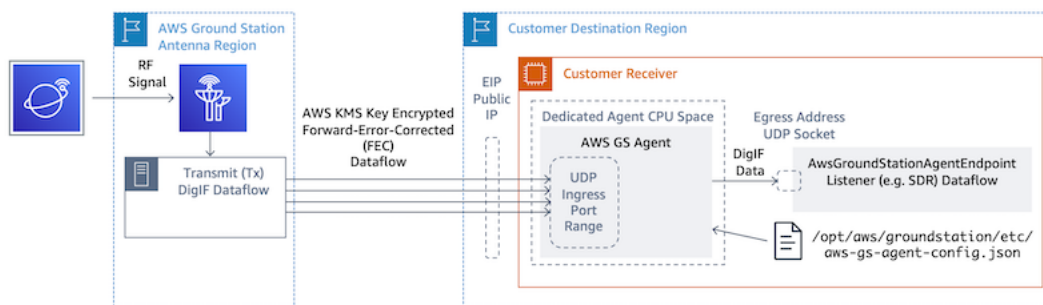
Como funciona

Você pode selecionar duas opções para entrega de dados:

1. Entrega de dados para uma EC2 instância - Entrega de dados para uma EC2 instância que você possui. Você gerencia o AWS Ground Station Agente. Essa opção pode ser mais adequada se você precisar de processamento de dados quase em tempo real. Consulte a [Trabalhe com fluxos de dados](#) seção para obter informações sobre entrega de EC2 dados.
2. Entrega de dados para um bucket S3 - A entrega de dados para seu bucket AWS S3 é totalmente gerenciada por. AWS Ground Station Consulte o guia [Conceitos básicos](#) para obter informações sobre a entrega de dados do S3.

Ambos os modos de entrega de dados exigem que você crie um conjunto de recursos da AWS. O uso de CloudFormation para criar seus recursos da AWS é altamente recomendado para garantir confiabilidade, precisão e capacidade de suporte. Cada contato só pode entregar dados para EC2 ou S3, mas não para ambos simultaneamente.

O diagrama a seguir mostra um fluxo de dados DigiF de uma região de AWS Ground Station antena para sua EC2 instância com seu rádio definido por software (SDR) ou ouvinte similar.



Mais informações

Para obter informações mais detalhadas, consulte o [Guia completo do usuário do AWS Ground Station agente](#).

Conceitos básicos

Antes de começar, você deve se familiarizar com os conceitos básicos do. AWS Ground Station Para obter mais informações, consulte [Como AWS Ground Station funciona](#).

Abaixo estão as melhores práticas para AWS Identity and Access Management (IAM) e quais permissões você precisará. Depois de configurar as funções apropriadas, você pode começar a seguir o restante das etapas.

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS Centro de Identidade do AWS IAM, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [Console de gerenciamento da AWS](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Fazer login como usuário-raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

Para obter instruções, consulte [Habilitar o Centro de Identidade do AWS IAM](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia Centro de Identidade do AWS IAM do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Criar um conjunto de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de logon único ao grupo.

Para obter instruções, consulte [Adicionar grupos](#) no Guia do usuário do Centro de Identidade do AWS IAM .

Adicione AWS Ground Station permissões à sua AWS conta

Para usar AWS Ground Station sem exigir um usuário administrativo, você precisa criar uma nova política e anexá-la à sua AWS conta.

1. Faça login no Console de gerenciamento da AWS e abra o [console do IAM](#).
2. Crie uma política. Use as seguintes etapas:
 - a. No painel de navegação, escolha Políticas e, em seguida, Criar Política.
 - b. Na guia JSON, edite o JSON com um dos seguintes valores. Use o JSON que melhor funcione para a sua aplicação.
 - Para privilégios de Admin, defina Ação como `groundstation:*`, da seguinte forma:

JSON

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

- Em privilégios Somente leitura, defina Ação como `groundstation:get*`, `groundstation:list*` e `groundstation:describe*`, da seguinte forma:

JSON

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

- Para obter segurança adicional por meio da autenticação multifatorial, defina Action como `groundstation:*` e Condition/Bool como `aws::true` da seguinte forma: `MultiFactorAuthPresent`

JSON

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

```
}
```

3. No console do IAM, anexe a política criada para o usuário desejado.

Para obter mais informações sobre criar usuários do IAM e como anexar políticas, consulte o [Guia do usuário do IAM](#).

Satélite a bordo

A integração de um satélite AWS Ground Station é um processo de várias etapas que envolve coleta de dados, validação técnica, licenciamento de espectro, integração e testes. Também são necessários acordos de não divulgação (NDAs).

Visão geral do processo de integração de clientes

A integração de satélites é um processo manual que pode ser encontrado na seção [Satélites e recursos](#) da página do AWS Ground Station console. A seguir, descrevemos o processo geral.

1. Revise a [AWS Ground Station Localizações](#) seção para determinar se seu satélite atende às características geográficas e de radiofrequência.
2. Para começar a integrar seu satélite ao AWS Ground Station, envie um questionário de integração de satélites na seção [Satélites e recursos](#) da página do console. AWS Ground Station Inclua um breve resumo de sua missão e necessidades de satélite, incluindo o nome da sua organização, as frequências necessárias, quando os satélites serão ou foram lançados, o tipo de órbita do satélite e se você planeja usá-lo [Use o recurso de gêmeos AWS Ground Station digitais](#).
3. Depois que sua solicitação for analisada e aprovada, AWS Ground Station solicitará o licenciamento regulatório nos locais específicos que você planeja usar. A duração dessa etapa variará dependendo dos locais e de quaisquer regulamentos existentes.
4. Depois que essa aprovação for obtida, seu satélite ficará visível para você usar. AWS Ground Station enviará uma notificação sobre a atualização bem-sucedida.

(Opcional) Nomeando satélites

Após a integração, talvez você queira adicionar um nome ao seu registro de satélite para reconhecê-lo mais facilmente. O AWS Ground Station console tem a capacidade de exibir um nome definido

pelo usuário para um satélite junto com o ID do Norad ao usar a página de contatos. A exibição do nome do satélite facilita muito a seleção do satélite correto durante o agendamento. Para fazer isso, as [tags](#) podem ser usadas.

A marcação de satélites do AWS Ground Station pode ser feita por meio da API de [recursos de tag](#) com a AWS CLI ou uma das AWS SDKs. Este guia abordará o uso da AWS Ground Station CLI para marcar o satélite público de transmissão Aqua (Norad ID 27424). `us-west-2`

AWS Ground Station CLI

O AWS CLI pode ser usado para interagir com AWS Ground Station. Antes de usar AWS CLI para marcar seus satélites, os seguintes AWS CLI pré-requisitos devem ser atendidos:

- Certifique-se de que AWS CLI esteja instalado. Para obter informações sobre a instalação AWS CLI, consulte [Instalação da AWS CLI versão 2](#).
- Certifique-se de que AWS CLI esteja configurado. Para obter informações sobre configuração AWS CLI, consulte [Configuração da AWS CLI versão 2](#).
- Salve as definições de configuração usadas com frequência e credenciais em arquivos que são mantidos pela AWS CLI. Você precisa dessas configurações e credenciais para reservar e gerenciar seus AWS Ground Station contatos. Para obter mais informações sobre como salvar suas configurações e configurações de credenciais, consulte [Configuração e configurações do arquivo de credenciais](#).

Quando AWS CLI estiver configurado e pronto para uso, consulte a página de [referência de comandos da CLI do AWS Ground Station](#) para se familiarizar com os comandos disponíveis. Siga a estrutura de AWS CLI comandos ao usar esse serviço e prefixe seus comandos com `groundstation` para especificar AWS Ground Station como o serviço que você deseja usar. Para obter mais informações sobre a estrutura de AWS CLI comando, consulte [Estrutura de comando na página da AWS CLI](#). Um exemplo de estrutura de comando é fornecido abaixo.

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

Nomeie um satélite

Primeiro, você precisa obter o ARN do(s) satélite(s) que deseja marcar. Isso pode ser feito por meio da API [list-satellites](#) na AWS CLI:

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

A execução do comando CLI acima retornará uma saída semelhante a esta:

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

Encontre o satélite que você deseja marcar e anote o `satelliteArn`. [Uma ressalva importante para a marcação é que a API de recursos de tags requer um ARN regional, e o ARN retornado pelos satélites de lista é global.](#) Para a próxima etapa, você deve aumentar o ARN com a região na qual gostaria de ver a tag (provavelmente a região em que você está agendando). Neste exemplo, usamos `us-west-2`. Com essa mudança, o ARN passará de:

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

para:

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

Para mostrar o nome do satélite no console, o satélite deve ter uma etiqueta com `"Name"` como a chave. Além disso, como estamos usando o AWS CLI, as aspas devem ser excluídas com uma barra invertida. A tag será semelhante a:

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

Em seguida, você chamará a API [tag-resource](#) para marcar o satélite. Isso pode ser feito da seguinte AWS CLI forma:

```
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"}'
```

Depois de fazer isso, você poderá ver o nome que definiu para o satélite no console do AWS Ground Station .

Alterar o nome de um satélite

Se você quiser alterar o nome de um satélite, basta chamar [tag-resource](#) com o ARN do satélite novamente com a mesma “Name” chave, mas com um valor diferente na tag. Isso atualizará a tag existente e mostrará o novo nome no console. Um exemplo de chamada para isso é semelhante a:

```
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"}'
```

Alterar o nome de um satélite

O nome definido para um satélite pode ser removido com a API [untag-resource](#). Essa API precisa do ARN do satélite com a região em que a tag está e de uma lista de chaves de tag. O nome da chave da tag é “Name”. Um exemplo de chamada para essa API usando a AWS CLI é este:

```
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
```

Satélites de transmissão pública

Além de integrar seus próprios satélites, você pode solicitar a integração com satélites de transmissão pública compatíveis que forneçam um caminho de comunicação de downlink acessível ao público. Isso permite que você use AWS Ground Station para baixar dados desses satélites.

Note

Você não poderá se conectar a esses satélites. Você só poderá usar os caminhos de comunicação de downlink acessíveis ao público.

AWS Ground Station suporta a integração dos seguintes satélites para baixar dados de transmissão direta:

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

Uma vez integrados, esses satélites podem ser acessados para uso imediato. AWS Ground Station mantém vários CloudFormation modelos pré-configurados para facilitar o início do serviço. Veja exemplos [Exemplo de configurações de perfil de missão](#) de como AWS Ground Station pode ser usado.

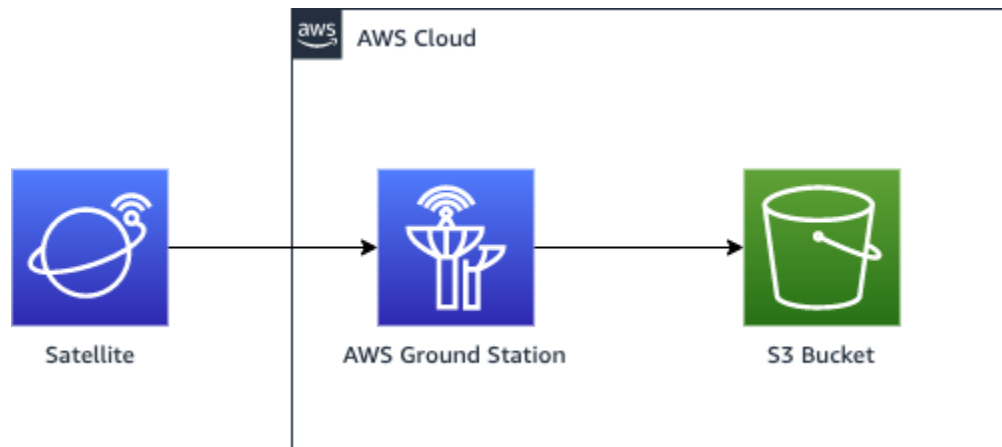
Para obter mais informações sobre esses satélites e os tipos de dados que são transmitidos, consulte [Aqua](#), [JPSS-1/NOAA-20 e SNPP](#) e [Terra](#).

Planeje seus caminhos de comunicação de fluxo de dados

Você pode escolher entre comunicação síncrona e assíncrona para cada caminho de comunicação em seu satélite. Dependendo do seu satélite e do seu caso de uso, você pode precisar de um ou dos dois tipos. Os caminhos de comunicação síncrona permitem operações de uplink quase em tempo real, bem como operações de downlink de banda estreita e banda larga. Os caminhos de comunicação assíncrona suportam somente operações de downlink de banda estreita e banda larga.

Entrega assíncrona de dados

Com a entrega de dados para o Amazon S3, seus dados de contato são entregues de forma assíncrona para um bucket do Amazon S3 em sua conta. Seus dados de contato são entregues como arquivos de captura de pacotes (pcap) para permitir a reprodução dos dados de contato em um rádio definido por software (SDR) ou para extrair os dados da carga útil dos arquivos pcap para processamento. Os arquivos pcap são entregues ao seu bucket Amazon S3 a cada 30 segundos, pois os dados de contato são recebidos pelo hardware da antena para permitir o processamento de dados de contato durante o contato, se desejado. Depois de recebidos, você pode processar os dados usando seu próprio software de pós-processamento ou usar outros serviços da AWS, como Amazon SageMaker AI ou Amazon Rekognition. A entrega de dados para o Amazon S3 só está disponível para baixar dados do seu satélite; não é possível vincular dados ao seu satélite a partir do Amazon S3.



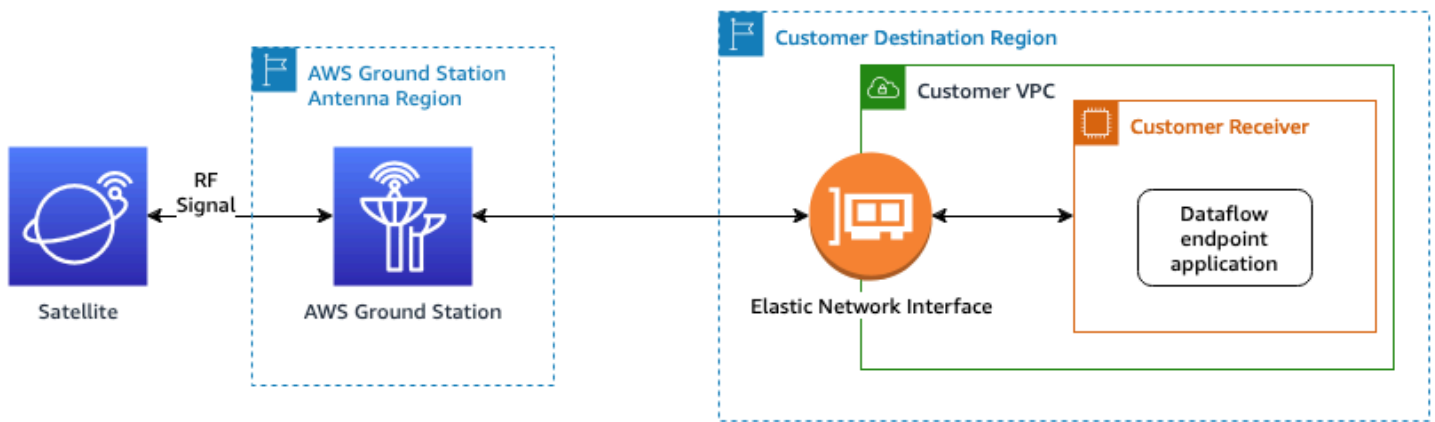
Para utilizar esse caminho, você precisará criar um bucket Amazon S3 AWS Ground Station para entregar os dados. Na próxima etapa, você também precisará criar uma Configuração de Gravação S3 na próxima etapa. Consulte as restrições sobre [Config de gravação do Amazon S3](#) a nomenclatura de buckets e como especificar a convenção de nomenclatura usada para seus arquivos.

Entrega síncrona de dados

Com a entrega de dados para o Amazon EC2, seus dados de contato são transmitidos de e para sua instância do Amazon EC2. Você pode processar seus dados em tempo real na sua instância do Amazon EC2 ou encaminhar os dados para pós-processamento.

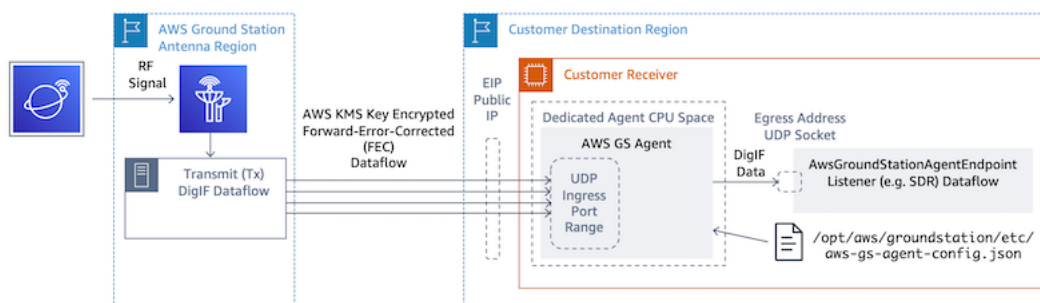
Para utilizar um caminho síncrono, você precisará instalar e configurar suas instâncias do Amazon EC2 e criar um ou mais grupos de endpoints do Dataflow. Para configurar sua instância do Amazon EC2, consulte o [Configurar e configurar a Amazon EC2](#) Para criar seu grupo de endpoints do Dataflow, consulte o [Use grupos AWS Ground Station de endpoints do Dataflow](#)

O exemplo a seguir mostra o caminho de comunicação se você estiver usando a configuração do endpoint do fluxo de dados.



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

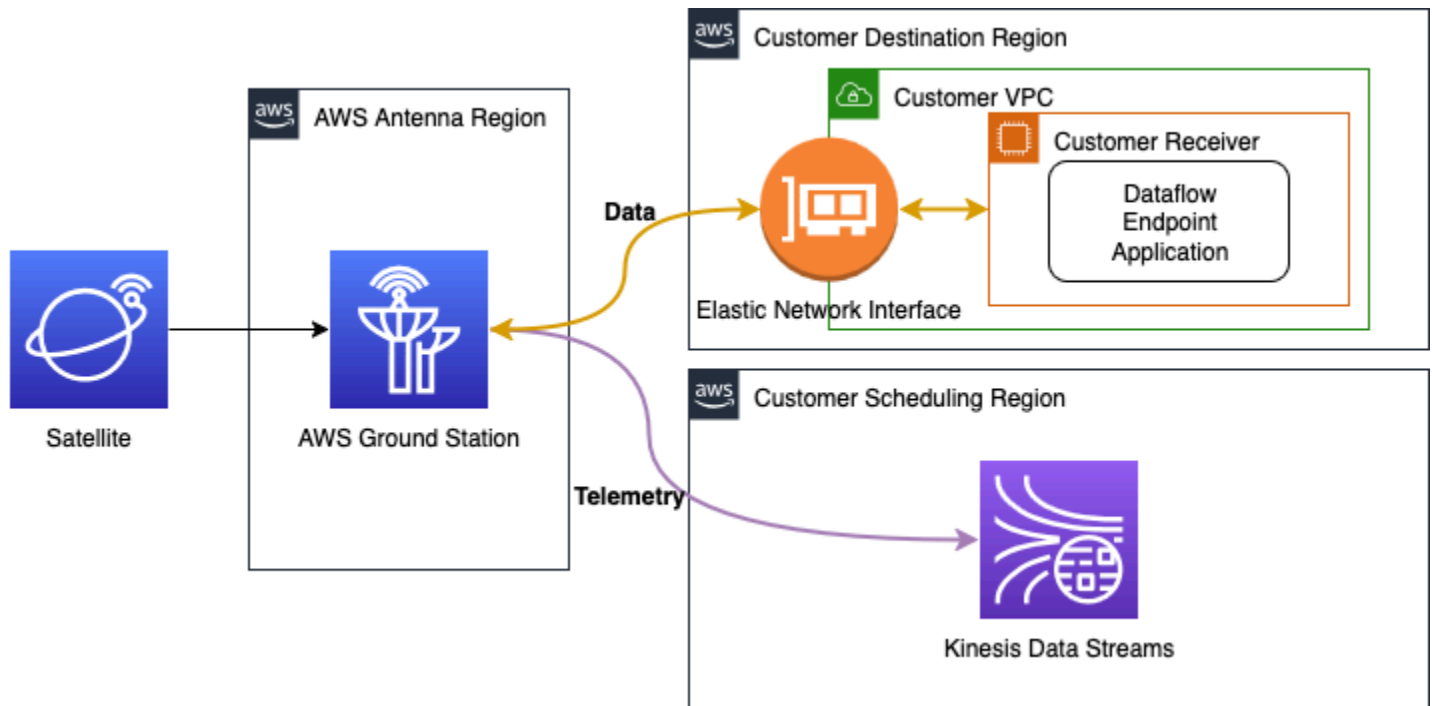
O seguinte mostra o caminho de comunicação se você estiver usando a configuração do AWS Ground Station Agente.



Planeje sua telemetria

AWS Ground Station a telemetria é um recurso opcional que transmite métricas das AWS Ground Station antenas para sua AWS conta durante os contatos via satélite. Isso permite monitorar o desempenho dos contatos quase em tempo real e criar soluções de monitoramento personalizadas.

Com a AWS Ground Station telemetria, as métricas das AWS Ground Station antenas são transmitidas diretamente para sua conta. Os dados de telemetria começam a ser transmitidos no início do contato e continuam durante toda a duração do contato. Os dados de telemetria são entregues à sua conta quase em tempo real, à medida que são amostrados do hardware da antena. Depois de recebidos, você pode processar os dados usando seu próprio software de pós-processamento ou usar outros serviços da AWS, como Amazon Data AWS Lambda Firehose ou.



Na próxima etapa, você criará as configurações necessárias para o seu perfil de missão. Se você quiser ativar a telemetria, você criará uma Configuração do coletor de telemetria, além das configurações de rastreamento e fluxo de dados. Para obter instruções detalhadas de configuração, consulte [Configurar a telemetria](#).

Para obter mais informações sobre TelemetrySinkConfig, consulte [Configuração do coletor de telemetria](#).

Crie configurações

Nessa etapa, você identificou o satélite, os caminhos de comunicação e os recursos do IAM, Amazon EC2 e Amazon S3, conforme necessário. Nesta etapa, você criará AWS Ground Station configurações que armazenam seus respectivos parâmetros.

Configurações de entrega de dados

As primeiras configurações a serem criadas estão relacionadas a onde e como você deseja que os dados sejam entregues. Usando as informações da etapa anterior, você construirá muitos dos seguintes tipos de configuração.

- [Config de gravação do Amazon S3](#)- Entregue dados para seu bucket Amazon S3.
- [Configuração de endpoint de fluxo de dados](#)- Entregue dados para sua instância do Amazon EC2.

Configuração de telemetria (opcional)

Se você quiser receber telemetria quase em tempo real durante seus contatos, você pode criar um TelemetrySinkConfig. Essa configuração é opcional e especifica onde os dados de telemetria AWS Ground Station serão entregues.

- [Configuração do coletor de telemetria](#)- Entregue dados de telemetria para sua conta.

Para obter instruções detalhadas de configuração, consulte [Configurar a telemetria](#).

Configurações de satélite

As configurações do satélite relacionam como AWS Ground Station você pode se comunicar com seu satélite. Você referenciará as informações coletadas [Satélite a bordo](#).

- [Configuração de rastreamento](#)- Define a preferência de como seu veículo é rastreado fisicamente durante um contato. Isso é necessário para a construção do perfil da missão.
- [Configuração de downlink de antena](#)- Forneça dados de radiofrequência digitalizados.
- [Configuração de decodificação de demodulação de downlink de antena](#)- Forneça dados de radiofrequência demodulados e decodificados.
- [Configuração de uplink de antena](#)- Vincule dados ao seu satélite.
- [Configuração de eco de uplink de antena](#)- Entregue um eco dos dados do seu sinal de uplink.

Criar perfil de missão

Com as configurações construídas na etapa anterior, você identificou como rastrear seu satélite, as formas possíveis de se comunicar com seu satélite e como ativar a telemetria quase em tempo real durante a execução do contato. Nesta etapa, você construirá um ou mais perfis de missão. Um perfil de missão representa a agregação das configurações possíveis em um comportamento esperado que pode ser programado e operado.

Para obter os parâmetros mais recentes, consulte o [tipo AWS::GroundStation::MissionProfile](#) [CloudFormation de recurso](#)

1. Dê um nome ao seu perfil de missão. Isso permite que você entenda rapidamente seu uso em seu sistema. Por exemplo, você pode ter satellite-wideband-narrowband-nominaloperações a e

- a satellite-narrowband-emergency-operationsse tiver uma operadora de banda estreita separada para operações de emergência.
2. Defina sua configuração de rastreamento.
 3. Defina suas durações mínimas de contato viáveis. Isso permite filtrar contatos em potencial para atender às necessidades de sua missão.
 4. Defina seus `streamsKmsKey` e `streamsKmsRole` que são usados para criptografar seus dados durante o trânsito. Isso é usado para todos os fluxos de dados AWS Ground Station do Agente.
 5. Defina seus fluxos de dados. Crie seus fluxos de dados para corresponder aos sinais da operadora usando as configurações que você criou na etapa anterior.
 6. [Opcional] Defina a duração do contato antes e depois da passagem em segundos. Isso é usado para emitir eventos por contato antes e depois do contato, respectivamente. Consulte [Automatize AWS Ground Station com eventos](#) para obter mais informações.
 7. [Opcional] Configure seu `telemetrySinkConfigArn` para ativar a telemetria durante os contatos. Isso permite que você receba telemetria quase em tempo real diretamente em sua conta para monitoramento e análise. Consulte [Trabalhe com telemetria](#) para obter mais informações.
 8. [Opcional] Você pode associar Tags ao seu perfil de missão. Eles podem ser usados para ajudar a diferenciar programaticamente seus perfis de missão.

Você pode consultar o [Exemplo de configurações de perfil de missão](#), para ver apenas algumas das configurações possíveis.

Entenda as próximas etapas

Agora que você tem um satélite a bordo e um perfil de missão válido, você está pronto para agendar contatos e se comunicar com seu satélite. AWS Ground Station

Você pode agendar um contato de uma das seguintes formas:

- O [AWS Ground Station console](#).
- O comando AWS CLI [reserve-contact](#).
- O AWS SDK. [ReserveContact](#) API.

Para obter informações sobre como AWS Ground Station rastreia a trajetória do seu satélite e como essas informações são usadas, consulte. [Entenda como AWS Ground Station usa efemérides](#)

AWS Ground Station mantém vários CloudFormation modelos pré-configurados para facilitar o início do serviço. Veja exemplos [Exemplo de configurações de perfil de missão](#) de como AWS Ground Station pode ser usado.

O processamento dos dados digitais de frequência intermediária ou dos dados desmodulados e decodificados fornecidos a você AWS Ground Station dependerá do seu caso de uso específico. As postagens de blog a seguir podem ajudar você a entender algumas das opções disponíveis:

- [Observação automatizada da Terra usando a entrega de dados do AWS Ground Station Amazon S3 \(e seu GitHub repositório associado awslabs/\) aws-groundstation-eos-pipeline](#)
- [Virtualizando o segmento terrestre de satélites com AWS](#)
- [Observação da Terra usando AWS Ground Station: Um guia prático](#)
- [Construindo arquiteturas de downlink de dados de satélite de alto rendimento com AWS Ground Station WideBand DigiF e Amphinicy Blink SDR \(e seu repositório associado aws-samples/\) GitHub aws-groundstation-wbdigif-snpp](#)

AWS Ground Station Localizações

AWS Ground Station fornece uma rede global de estações terrestres próximas à nossa rede global de regiões de infraestrutura da AWS. Você pode configurar o uso desses locais em qualquer região da AWS compatível. Isso inclui a região da AWS na qual os dados são entregues.



Encontrando a AWS região para a localização de uma estação terrestre

A rede AWS Ground Station global inclui estações terrestres que não estão fisicamente localizadas na [região da AWS](#) à qual estão conectadas. A lista de estações terrestres às quais você tem acesso pode ser recuperada por meio da resposta do SDK [ListGroundStation](#) da AWS. A lista completa das localizações das estações terrestres é apresentada abaixo, com mais informações em breve. Consulte o guia de integração para adicionar ou modificar as aprovações do site para seus satélites.

Nome da Ground Station	Localização da Ground Station	Nome da região da AWS	Código de região da AWS	Observações
Alasca 1	Alasca, EUA	Oeste dos EUA (Oregon)	us-west-2	Não está fisicamente

Nome da Ground Station	Localização da Ground Station	Nome da região da AWS	Código de região da AWS	Observações
				localizado em uma AWS região
Bahrein 1	Bahrein	Oriente Médio (Barém)	me-south-1	
Cidade do Cabo 1	Cidade do Cabo, África do Sul	África (Cidade do Cabo)	af-south-1	
Dubbo 1	Dubbo, Austrália	Ásia-Pacífico (Sydney)	ap-southeast-2	Não está fisicamente localizado em uma AWS região
Havaí 1	Havaí, EUA	Oeste dos EUA (Oregon)	us-west-2	Não está fisicamente localizado em uma AWS região
Irlanda 1	Irlanda	Europa (Irlanda)	eu-west-1	
Ohio 1	Ohio, EUA	Leste dos EUA (Ohio)	us-east-2	
Oregon 1	Oregon, EUA	Oeste dos EUA (Oregon)	us-west-2	
Punta Arenas 1	Punta Arenas, Chile	América do Sul (São Paulo)	sa-east-1	Não está fisicamente localizado em uma AWS região
Seul 1	Seul, Coreia do Sul	Ásia-Pacífico (Seul)	ap-northeast-2	

Nome da Ground Station	Localização da Ground Station	Nome da região da AWS	Código de região da AWS	Observações
Cingapura 1	Cingapura	Ásia-Pacífico (Singapura)	ap-southeast-1	
Estocolmo 1	Estocolmo, Suécia	Europa (Estocolmo)	eu-north-1	

AWS Ground Station regiões da AWS suportadas

Você pode entregar dados e configurar seus contatos por meio do SDK da AWS ou do AWS Ground Station console de regiões compatíveis da AWS. Você pode visualizar as regiões suportadas e seus endpoints associados nos [AWS Ground Station endpoints e cotas](#).

Disponibilidade do gêmeo digital

[Use o recurso de gêmeos AWS Ground Station digitais](#) está disponível em todas as [regiões da AWS](#) onde AWS Ground Station está disponível. As estações terrestres gêmeas digitais são cópias exatas das estações terrestres de produção com um prefixo modificador para o nome da estação terrestre de “Digital Twin”. Por exemplo, “Digital Twin Ohio 1” é uma estação terrestre dupla digital que é uma cópia exata da estação terrestre de produção “Ohio 1”.

Antenas dedicadas

Além das localizações de estações terrestres disponíveis ao público listadas acima, AWS Ground Station oferece antenas dedicadas. Uma antena dedicada é um sistema de antena personalizado que AWS gerencia em seu nome. Uma antena dedicada não está restrita às localizações existentes das estações AWS Ground Station terrestres e pode ser construída com recursos além dos das estações terrestres públicas, conforme descrito em [AWS Ground Station Capacidades do site](#). As localizações e capacidades das antenas dedicadas não são divulgadas publicamente.

Para obter mais informações sobre antenas dedicadas, consulte [AWS Ground Station Antenas dedicadas](#). Para saber mais ou começar a usar antenas dedicadas, entre em contato AWS Support através do [AWS Support Center Console](#).

Visualizando antenas em uma estação terrestre

Cada localização da estação terrestre tem uma ou mais antenas. Você pode ver as antenas em uma estação terrestre usando a [ListAntennas](#) API. Essa API retorna as antenas em uma estação terrestre especificada, incluindo o nome de cada antena.

As informações da antena são úteis quando combinadas com a [ListGroundStationReservations](#) API para entender a capacidade e a disponibilidade em uma estação terrestre. Para obter mais informações sobre a visualização de reservas, consulte [Exibir reservas de estações terrestres](#).

Para ligar `ListAntennas`, você deve ter um satélite a bordo da estação terrestre ou ter permissões de efemérides de elevação de azimute para a estação terrestre. Para obter mais informações, consulte [Forneça dados de efemérides de elevação de azimute](#).

Exemplo: Listar antenas em uma estação terrestre

O exemplo a seguir lista todas as antenas em uma estação terrestre usando o AWS SDK for Python (Boto3).

```
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()
```

Exibir reservas de estações terrestres

Você pode ver as reservas nas antenas em uma estação terrestre usando a [ListGroundStationReservations](#) API. As reservas representam blocos de tempo nas antenas, incluindo seus contatos agendados. [AWS Ground Station Antenas dedicadas](#) os clientes também veem janelas de manutenção.

Essas informações ajudam você a entender a disponibilidade da antena ao planejar os agendamentos de contato e fornecem visibilidade do que está acontecendo nas antenas de uma estação terrestre.

Listando reservas

Para listar reservas, ligue [ListGroundStationReservations](#) com um identificador de estação terrestre e um intervalo de tempo. A API retorna reservas em todas as antenas na estação terrestre dentro da janela de tempo especificada.

As reservas que você vê dependem do seu nível de acesso:

- AWS Ground Station Clientes públicos — Você pode ver somente suas próprias reservas de contato. Janelas de manutenção e contatos pertencentes a outras contas não estão incluídos.
- AWS Ground Station Clientes de antenas dedicadas — Você pode ver todas as reservas em suas antenas dedicadas, incluindo janelas de manutenção e contatos agendados por outras contas. Os identificadores de contato são incluídos somente para contatos que você possui. Para obter mais informações, consulte [AWS Ground Station Antenas dedicadas](#).

Tipos de reserva

Cada reserva tem um tipo que indica para que o tempo da antena está sendo usado:

- **Contato** — Uma reserva de contato representa o tempo de antena reservado para comunicação via satélite. Os horários de início e término da reserva refletem a reserva completa da antena, incluindo o tempo antes e depois da passagem, não apenas a janela do passe via satélite.
- **Manutenção** — Uma reserva de manutenção representa um período em que a antena está indisponível devido à manutenção. As reservas de manutenção incluem uma `maintenanceType` que indica se a manutenção foi planejada ou não.

Exemplo de código

O exemplo a seguir lista as reservas em uma estação terrestre para os próximos 7 dias usando o AWS SDK for Python (Boto3), incluindo a filtragem por tipo de reserva.

```
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 máscaras do site

Cada [localização AWS Ground Station da antena](#) tem máscaras de site associadas. Essas máscaras impedem que as antenas desse local transmitam ou recebam quando apontam em algumas direções, normalmente perto do horizonte. As máscaras podem levar em consideração:

- Características do terreno geográfico ao redor da antena — Por exemplo, isso inclui coisas como montanhas ou edifícios, que bloqueariam um sinal de radiofrequência (RF) ou impediriam a transmissão.
- Interferência de radiofrequência (RFI) — Isso afeta tanto a capacidade de receber (fontes externas de RFI impactando um sinal de downlink nas antenas do AWS Ground Station) quanto de transmitir (o sinal de RF transmitido pelas antenas do AWS Ground Station impactando negativamente os receptores externos).
- Autorizações legais — As autorizações locais do site para operar o AWS Ground Station em cada região podem incluir restrições específicas, como um ângulo mínimo de elevação para transmissão.

Essas máscaras do site podem ser alteradas ao longo do tempo. Por exemplo, novos edifícios podem ser construídos perto de um local de antena, as fontes de RFI podem mudar ou a autorização legal pode ser renovada com restrições diferentes. As máscaras do site AWS Ground Station estão disponíveis para você sob um acordo de confidencialidade (NDA).

Máscaras específicas para clientes

Além das máscaras de site do AWS Ground Station em cada local, você pode ter máscaras adicionais devido às restrições de sua própria autorização legal para se comunicar com seus satélites em uma determinada região. Essas máscaras podem ser configuradas no AWS Ground Station para garantir case-by-case a conformidade ao usar o AWS Ground Station para se comunicar com esses satélites. Entre em contato com a equipe do AWS Ground Station para obter detalhes.

Impacto das máscaras do site nos horários de contato disponíveis

Há dois tipos de máscaras de site: máscaras de site de uplink (transmissão) e máscaras de site de downlink (recebimento).

Ao listar os horários de contato disponíveis usando a ListContacts operação, o AWS Ground Station retornará os tempos de visibilidade com base em quando seu satélite estará acima e abaixo da máscara de downlink. Os tempos de contato disponíveis são baseados nessa janela de visibilidade da máscara de downlink. Isso garante que você não reserve tempo quando seu satélite estiver abaixo da máscara de downlink.


As máscaras do site de uplink não são aplicadas aos horários de contato disponíveis, mesmo que o perfil da missão inclua uma [configuração de uplink de antena](#) em uma borda de fluxo de dados. Isso permite que você use todo o tempo de contato disponível para downlink, mesmo que o uplink não esteja disponível por partes desse tempo devido à máscara do site de uplink. No entanto, o sinal de uplink pode não ser transmitido por parte ou por todo o tempo reservado para um contato via satélite. Você é responsável por contabilizar a máscara de uplink fornecida ao programar as transmissões de uplink.

A parte de um contato que não está disponível para uplink varia dependendo da trajetória do satélite durante o contato, em relação à máscara do local de uplink no local da antena. Em regiões em que as máscaras do site de uplink e downlink são semelhantes, essa duração normalmente será curta. Em outras regiões, em que a máscara de uplink pode ser consideravelmente maior do que a do local de downlink, isso pode fazer com que partes significativas, ou mesmo a totalidade, da duração do contato não estejam disponíveis para uplink. O tempo total de contato é cobrado de você, mesmo que partes do tempo reservado não estejam disponíveis para uplink.

AWS Ground Station Capacidades do site

Para simplificar sua experiência, AWS Ground Station determina um conjunto comum de recursos para um tipo de antena e, em seguida, implanta várias antenas em um local de estação terrestre. Parte das etapas de integração garante que seu satélite seja compatível com os tipos de antena em um local específico. Ao reservar um contato, você determina indiretamente o tipo de antena usada. Isso garante que sua experiência em um determinado local da estação terrestre permaneça a mesma ao longo do tempo, independentemente de quais antenas estejam sendo usadas. O desempenho específico do seu contato variará devido a uma grande variedade de questões ambientais, como o clima no local.

Atualmente, todos os sites oferecem suporte aos seguintes recursos:

 Note

Cada linha na tabela a seguir indica um caminho de comunicação independente, a menos que indicado de outra forma. Existem linhas duplicadas para refletir nossos recursos multicanais que permitem que vários caminhos de comunicação sejam usados simultaneamente.

Tipo de capacidade	Faixa de frequência	Faixa de largura de banda	Polarization	Nome comum	Observações
downlink de antena	750 - 8500 MHz	50 - 400 MHz	RHCP	Downlink de banda larga de banda X	Esse recurso requer o uso do AWS Ground Station Agente .
downlink de antena	750 - 8500 MHz	50 - 400 MHz	RHCP		
downlink de antena	750 - 8500 MHz	50 - 400 MHz	RHCP		Esse recurso não é suportado no Alaska 1 ou em Punta Arenas 1.
downlink de antena	750 - 8500 MHz	50 - 400 MHz	RHCP		
downlink de antena	750 - 8500 MHz	50 - 400 MHz	RHCP		
downlink de antena	750 - 8500 MHz	50 - 400 MHz	LHCP		A largura de banda agregada não deve exceder 400 MHz por polarização em cada local.
downlink de antena	750 - 8500 MHz	50 - 400 MHz	LHCP		
downlink de antena	750 - 8500 MHz	50 - 400 MHz	LHCP		
downlink de antena	750 - 8500 MHz	50 - 400 MHz	LHCP		Todas as faixas de

Tipo de capacidade	Faixa de frequência	Faixa de largura de banda	Polarization	Nome comum	Observações
downlink de antena	750 - 8500 MHz	50 - 400 MHz	LHCP		frequência utilizadas não devem ser sobrepostas.
downlink de antena	2200 - 290 MHz	Até 40 MHz	RHCP	Downlink da banda S	Somente uma polarização pode ser usada por vez
downlink de antena	2200 - 290 MHz	Até 40 MHz	LHCP		
downlink de antena	750 - 8500 MHz	Até 40 MHz	RHCP	Downlink de banda estreita de banda X	Somente uma polarização pode ser usada por vez
downlink de antena	750 - 8500 MHz	Até 40 MHz	LHCP		
uplink de antena	2025 - 2110 MHz	Até 40 MHz	RHCP	Uplink de banda S	Somente uma polarização pode ser usada por vez
uplink de antena	2025 - 2110 MHz	Até 40 MHz	LHCP		
					EIRP 20-50 dBW
antenna-uplink-echo	2025 - 2110 MHz	2 MHz	RHCP	Eco de uplink	Corresponde às restrições de uplink da antena
antenna-uplink-echo	2025 - 2110 MHz	2 MHz	LHCP		

Tipo de capacidade	Faixa de frequência	Faixa de largura de banda	Polarization	Nome comum	Observações
antenna-downlink-demod-decode	750 - 8500 MHz	Até 500 MHz	RHCP	Downlink desmodulado e decodificado em banda X	
antenna-downlink-demod-decode	750 - 8500 MHz	Até 500 MHz	LHCP		
rastreamento	N/D	N/D	N/D	N/D	Support para rastreamento automático e rastreamento de programas

* RHCP = polarização circular com a mão direita e LHCP = polarização circular com a mão esquerda. Para obter mais informações sobre polarização, consulte [Polarização circular](#).

Entenda como AWS Ground Station usa efemérides

Uma [efeméride](#), efemérides no plural, é um arquivo ou estrutura de dados que fornece a trajetória de objetos astronômicos. Historicamente, esse arquivo se referia apenas a dados tabulares, mas, gradualmente, passou a direcionar para uma ampla variedade de arquivos de dados indicando a trajetória de uma espaçonave.

A API Ephemeris permite que efemérides personalizadas sejam enviadas para AWS Ground Station uso com um satélite. [Essas efemérides substituem as efemérides padrão do Space-Track \(consulte:\). Dados de efemérides padrão](#) Oferecemos suporte ao recebimento de dados de efemérides nos formatos Orbit Ephemeris Message (OEM), elemento de duas linhas (TLE) e elevação de azimute.

AWS Ground Station usa dados de efemérides para determinar quando os contatos ficam disponíveis com base nas efemérides fornecidas e comanda corretamente as antenas na rede. AWS Ground Station [Por padrão, nenhuma ação é necessária para AWS Ground Station fornecer efemérides se seu satélite tiver um ID NORAD atribuído.](#)

O upload de efemérides personalizadas pode melhorar a qualidade do rastreamento, lidar com operações iniciais em que não há efemérides do [Space-Track disponíveis](#) e contabilizar as manobras. AWS Ground Station

Como alternativa, AWS Ground Station suporta um formato de elevação de azimute, que permite especificar diretamente as direções de apontamento da antena sem fornecer informações orbitais do satélite. Isso é útil para cenários em que o apontamento preciso da antena é necessário porque as informações da trajetória do satélite são imprecisas ou desconhecidas.

Tópicos

- [Dados de efemérides padrão](#)
- [Forneça dados de efemérides personalizados](#)
- [Reserve contatos com efemérides personalizadas](#)
- [Entenda quais efemérides são usadas](#)
- [Obtenha as efemérides atuais de um satélite](#)
- [Reverter para dados de efemérides padrão](#)

Dados de efemérides padrão

Por padrão, AWS Ground Station usa dados publicamente disponíveis do [Space-Track](#), e nenhuma ação é necessária para AWS Ground Station fornecer essas efemérides padrão. [Essas efemérides são conjuntos de elementos de duas linhas \(TLEs\) associados ao ID NORAD do seu satélite](#). Todas as efemérides padrão têm uma prioridade de 0. Como resultado, elas serão sempre substituídas por quaisquer efemérides personalizadas não expiradas enviadas por meio da API de efemérides, que sempre deve ter uma prioridade igual ou superior. 1

Os satélites sem um ID NORAD devem carregar dados de efemérides personalizados para AWS Ground Station. Por exemplo, satélites que acabaram de ser lançados ou que foram intencionalmente omitidos do catálogo do [Space-Track](#) não teriam ID NORAD e precisariam ter efemérides personalizadas carregadas. Para obter mais informações sobre como fornecer dados de efemérides personalizados, consulte: [Fornecimento de dados de efemérides personalizados](#).

Forneça dados de efemérides personalizados

Important

A API Ephemeris está atualmente em um estado de visualização

O acesso à API Ephemeris é fornecido somente conforme a necessidade. Se você precisar fazer upload de dados de efemérides personalizados, abra um AWS Support ticket por meio do [AWS Support Center Console](#). Nossa equipe trabalhará com você para habilitar esse recurso para suas necessidades específicas.

Visão geral do

A API Ephemeris permite que efemérides personalizadas sejam enviadas para AWS Ground Station uso com um satélite. [Essas efemérides substituem as efemérides padrão do Space-Track \(consulte:\). Dados de efemérides padrão](#) Oferecemos suporte ao recebimento de dados de efemérides nos formatos Orbit Ephemeris Message (OEM), elemento de duas linhas (TLE) e elevação de azimute.

AWS Ground Station trata as efemérides como dados de uso [individualizados](#). Se você usar esse recurso opcional, a AWS usará seus dados de efemérides para fornecer suporte à solução de problemas.

O upload de efemérides personalizadas pode melhorar a qualidade do rastreamento, lidar com operações nas quais não há efemérides do [Space-Track disponíveis](#) e contabilizar as manobras. AWS Ground Station

Para solucionar uma efeméride inválida, consulte: [Solucionar problemas de efemérides inválidas](#)

Exemplo: uso de efemérides fornecidas pelo cliente com AWS Ground Station

[Para obter instruções mais detalhadas sobre como usar efemérides fornecidas pelo cliente com AWS Ground Station, consulte Como usar efemérides fornecidas pelo cliente com e seu repositório associado aws-samples/. AWS Ground Station GitHub aws-groundstation-cpe](#)

Forneça dados de efemérides do TLE

Important

A API Ephemeris está atualmente em um estado de visualização

O acesso à API Ephemeris é fornecido somente conforme a necessidade. Se você precisar fazer upload de dados de efemérides personalizados, abra um AWS Support ticket por meio do. [AWS Support Center Console](#) Nossa equipe trabalhará com você para habilitar esse recurso para suas necessidades específicas.

Visão geral do

Os conjuntos de elementos de duas linhas (TLE) são um formato padronizado para descrever as órbitas dos satélites. A API Ephemeris permite que as efemérides do TLE sejam enviadas para AWS Ground Station uso com um satélite. [Essas efemérides substituem as efemérides padrão do Space-Track \(consulte:\). Dados de efemérides padrão](#)

AWS Ground Station trata as efemérides como dados de uso [individualizados](#). Se você usar esse recurso opcional, a AWS usará seus dados de efemérides para fornecer suporte à solução de problemas.

O upload de efemérides TLE personalizadas pode melhorar a qualidade do rastreamento, lidar com operações iniciais em que não há efemérides do [Space-Track disponíveis](#) e contabilizar as manobras. AWS Ground Station

Note

Ao fornecer efemérides personalizadas antes que um número de catálogo de satélite seja atribuído ao seu satélite, você pode usar 00000 para o campo de número de catálogo de satélite do TLE e 000 para a parte do número de lançamento do campo designador internacional do TLE (por exemplo, 24000A para um veículo lançado em 2024).

Para obter mais informações sobre o formato de TLEs, consulte [Conjunto de elementos de duas linhas](#).

Criando uma efeméride TLE

Uma efeméride de TLE pode ser criada usando a [CreateEphemeris](#) ação na API. AWS Ground Station Essa ação fará o upload de uma efeméride usando dados no corpo da solicitação ou de um bucket do S3 especificado.

É importante observar que o upload de uma efeméride define as efemérides como VALIDATING e inicia um fluxo de trabalho assíncrono que validará e gerará contatos potenciais a partir de suas efemérides. Somente quando uma efeméride passar por esse fluxo de trabalho e se tornar ENABLED, ela será usada para contatos. Você deve pesquisar o status das efemérides ou usar CloudWatch eventos [DescribeEphemeris](#) para rastrear as mudanças de status das efemérides.

Para solucionar uma efeméride inválida, consulte: [Solucionar problemas de efemérides inválidas](#)

Exemplo: criar efemérides de um conjunto de elementos de duas linhas (TLE) por meio da API

A AWS SDKs CLI e pode ser usada para fazer upload de efemérides de um conjunto de elementos de duas linhas (TLE) por meio da chamada. AWS Ground Station [CreateEphemeris](#) Essas efemérides serão usadas no lugar dos dados de efemérides padrão de um satélite (consulte). [Dados de efemérides padrão](#) Este exemplo mostra como fazer isso usando o [AWS SDK para Python \(Boto3\)](#).

Um conjunto TLE é um objeto formatado em JSON que TLEs une um ou mais para construir uma trajetória contínua. O TLEs conjunto TLE deve formar um conjunto contínuo que possamos usar para construir uma trajetória (ou seja, sem lacunas no tempo entre TLEs um conjunto TLE). Um conjunto de TLE de exemplo é mostrado abaixo:

```
[
```

```

    {
      "tleLine1": "1 25994U 99068A 20318.54719794 .00000075 00000-0 26688-4 0
9997",
      "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934
14.57114995111906",
      "validTimeRange": {
        "startTime": 12345,
        "endTime": 12346
      }
    },
    {
      "tleLine1": "1 25994U 99068A 20318.54719794 .00000075 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

Os intervalos de tempo do TLEs em um conjunto de TLE devem corresponder exatamente para serem uma trajetória contínua e válida.

Um conjunto TLE pode ser carregado por meio do cliente AWS Ground Station boto3 da seguinte forma:

```

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 .00000075 00000-0
26688-4 0 9997",
                "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934
14.57114995111906",
                "validTimeRange": {
                    "startTime": datetime.now(timezone.utc),
                    "endTime": datetime.now(timezone.utc) + timedelta(days=7),
                },
            }
        ]
    }
},
)

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

```

Essa chamada retornará um EphemerisID que pode ser usado para referenciar as efemérides no futuro. Por exemplo, podemos usar o EphemerisId fornecido na chamada acima para pesquisar o status da efeméride:

```

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

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

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

tle_ephemeris = ground_station_client.create_ephemeris(
    name="Example TLE Ephemeris for Description",
    satelliteId="2e925701-9485-4644-b031-EXAMPLE01",
    enabled=True,
    expirationTime=datetime.now(timezone.utc) + timedelta(days=3),
    priority=2,
    ephemeris={

```

```

        "tle": {
            "tleData": [
                {
                    "tleLine1": "1 25994U 99068A 20318.54719794 .00000075 00000-0
26688-4 0 9997",
                    "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934
14.57114995111906",
                    "validTimeRange": {
                        "startTime": datetime.now(timezone.utc),
                        "endTime": datetime.now(timezone.utc) + timedelta(days=7),
                    },
                }
            ]
        },
    ),

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

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

response = ground_station_client.describe_ephemeris(ephemerisId=ephemeris_id)

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

```

Um exemplo de resposta da ação [DescribeEphemeris](#) é fornecido abaixo

```

{
  "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}}]\"
  }
}
}

```

É recomendável pesquisar a [DescribeEphemeris](#) rota ou usar CloudWatch eventos para rastrear o status das efemérides carregadas, pois elas devem passar por um fluxo de trabalho de validação assíncrona antes de serem configuradas ENABLED e se tornarem utilizáveis para agendar e executar contatos.

Observe que o ID NORAD TLEs em todo o conjunto TLE, 25994 nos exemplos acima, deve corresponder ao ID NORAD atribuído ao seu satélite no banco de dados do [Space-Track](#).

Exemplo: carregamento de dados de efemérides do TLE de um bucket do S3

Também é possível fazer upload de um arquivo de efemérides TLE diretamente de um bucket do S3 apontando para o bucket e a chave do objeto. AWS Ground Station recuperará o objeto em seu nome. As informações sobre a criptografia de dados em repouso AWS Ground Station estão detalhadas em: [Criptografia de dados em repouso para AWS Ground Station](#).

Abaixo está um exemplo de upload de um arquivo de efemérides TLE de um bucket S3

```

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

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

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

# Create sample TLE set data
# Note: For actual satellites, use real TLE data from sources like Space-Track
tle_set_data = [
    {
        "tleLine1": "1 25994U 99068A 20318.54719794 .00000075 00000-0 26688-4 0
9997",

```

```

        "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934
14.57114995111906",
        "validTimeRange": {
            "startTime": datetime.now(timezone.utc),
            "endTime": datetime.now(timezone.utc) + timedelta(days=3),
        },
    },
    {
        "tleLine1": "1 25994U 99068A 20321.54719794 .00000075 00000-0 26688-4 0
9998",
        "tleLine2": "2 25994 98.2007 33.6589 0001234 89.2782 18.9934
14.57114995112342",
        "validTimeRange": {
            "startTime": datetime.now(timezone.utc) + timedelta(days=3),
            "endTime": datetime.now(timezone.utc) + timedelta(days=7),
        },
    },
]

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

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

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

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

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

```

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

Forneça dados de efemérides do OEM

Important

A API Ephemeris está atualmente em um estado de visualização

O acesso à API Ephemeris é fornecido somente conforme a necessidade. Se você precisar fazer upload de dados de efemérides personalizados, abra um AWS Support ticket por meio do [AWS Support Center Console](#). Nossa equipe trabalhará com você para habilitar esse recurso para suas necessidades específicas.

Visão geral do

Orbit Ephemeris Message (OEM) é um formato padronizado para representar dados de trajetória de espaçonaves. A API Ephemeris permite que efemérides OEM sejam enviadas para uso com um satélite. AWS Ground Station [Essas efemérides substituem as efemérides padrão do Space-Track \(consulte\):](#). [Dados de efemérides padrão](#)

AWS Ground Station trata as efemérides como dados de uso [individualizados](#). Se você usar esse recurso opcional, AWS usará seus dados de efemérides para fornecer suporte à solução de problemas.

O upload de efemérides OEM personalizadas pode melhorar a qualidade do rastreamento, lidar com operações iniciais em que não há efemérides do [Space-Track](#) disponíveis e contabilizar as manobras. AWS Ground Station

Note

Ao fornecer efemérides personalizadas antes que um número de catálogo de satélite seja atribuído ao seu satélite, você pode usar `satelliteId` para a `OBJECT_ID` parte do OEM. Para obter mais informações sobre o formato do OEMs, consulte [Formato de efemérides OEM](#).

Formato de efemérides OEM

AWS Ground Station processa efemérides fornecidas pelo cliente OEM de acordo com o padrão [CCSDS](#) com algumas restrições extras. Os arquivos OEM devem estar no formato KVN. A tabela a seguir descreve os diferentes campos em um OEM e como AWS Ground Station difere do padrão CCSDS.

Seção	Campo	É necessário o CCSDS	AWS Ground Station exigido	Observações
Cabeçalho	CCSDS_OEM_VERS	Sim	Sim	Valor exigido: 2,0
	COMMENT	Não	Não	
	CLASSIFICAÇÃO	Não	Não	
	DATA_DE_CRIAÇÃO	Sim	Sim	
	ORIGINADORA	Sim	Sim	
	ID DA MENSAGEM	Não	Não	
Metadados	META_START	Sim	Sim	
	COMMENT	Não	Não	
	NOME_OBJETO	Sim	Sim	
	ID_OBJETO	Sim	Sim	
	NOME_CENTRAL	Sim	Sim	Valor exigido: Terra
	QUADRO_REFERÊNCIA	Sim	Sim	Valores aceitos: EME2000, ITRF2000

Seção	Campo	É necessário o CCSDS	AWS Ground Station exigido	Observações
	REF_FRAME_EPOCH	Não	Não suportado*	Não é necessário porque os REF_ aceitos FRAMEs têm uma época implícita
	SISTEMA_TEMPO	Sim	Sim	Valor exigido: UTC
	HORÁRIO_INICIAL	Sim	Sim	
	HORÁRIO_DE_INÍCIO_UTILIZÁVEL	Não	Não	
	HORÁRIO_DE_PARADA_UTILIZÁVEL	Não	Não	
	HORÁRIO_DE_PARADA	Sim	Sim	
	INTERPOLAÇÃO	Não	Sim	Necessário para AWS Ground Station gerar ângulos de apontamento precisos para os contatos.

Seção	Campo	É necessário o CCSDS	AWS Ground Station exigido	Observações
	GRAU_DE_INTERPOLAÇÃO	Não	Sim	Necessário para AWS Ground Station gerar ângulos de apontamento precisos para contatos. O grau especificado será usado se possível, mas um grau menor será usado se não houver dados suficientes no segmento.
	META_STOP	Sim	Sim	
Dados	X	Sim	Sim	Representado em km
	S	Sim	Sim	Representado em km
	Z	Sim	Sim	Representado em km
	X_PONTO	Sim	Sim	Representado em km/s
	Y_DOT	Sim	Sim	Representado em km/s
	Z_DOT	Sim	Sim	Representado em km/s

Seção	Campo	É necessário o CCSDS	AWS Ground Station exigido	Observações
	X_DDOT	Não	Não	Representado em km/s^2
	Y_DDOT	Não	Não	Representado em km/s^2
	Z_DDOT	Não	Não	Representado em km/s^2
Matriz de covariância	INÍCIO DA COVARIÂNCIA	Não	Não	
	EPOCH	Não	Não	
	COV_REF_FRAME	Não	Não	
	COVARIANC E_STOP	Não	Não	

* Se alguma linha não suportada pelo AWS Ground Station for incluída no OEM fornecido, o OEM falhará na validação.

Os desvios importantes do padrão CCSDS para são: AWS Ground Station

- `CCSDS_OEM_VERSION` é necessário que seja `2.0`.
- `REF_FRAME` é necessário que seja um `EME2000` ou `ITRF2000`.
- `REF_FRAME_EPOCH` não é suportado pelo AWS Ground Station.
- `CENTER_NAME` é necessário que seja `Earth`.
- `TIME_SYSTEM` é necessário que seja `UTC`.
- `INTERPOLATION` e ambos `INTERPOLATION_DEGREE` são necessários para AWS Ground Station efemérides fornecidas pelo cliente.
- AWS Ground Station se desvia do CCSDS 5.2.4.7 ao permitir que blocos de dados OEM que não contêm registros de dados de efemérides suficientes realizem a interpolação no especificado.

INTERPOLATION_DEGREE Nesse caso, AWS Ground Station usará o maior grau de interpolação possível, menor ou igual ao especificado. INTERPOLATION_DEGREE

Exemplo de efemérides de OEM no formato KVN

A seguir está um exemplo truncado de uma efeméride OEM no formato KVN para o satélite de transmissão pública 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

```

Criando uma efeméride de OEM

Uma efeméride de OEM pode ser criada usando a [CreateEphemeris](#) ação na API. AWS Ground Station Essa ação fará o upload de uma efeméride usando dados no corpo da solicitação ou de um bucket do S3 especificado.

É importante observar que o upload de uma efeméride define as efemérides como `VALIDATING` e inicia um fluxo de trabalho assíncrono que validará e gerará contatos potenciais a partir de suas efemérides. Somente quando uma efeméride passar por esse fluxo de trabalho e se tornar `ENABLED`, ela será usada para contatos. Você deve pesquisar o status das efemérides ou usar CloudWatch eventos [DescribeEphemeris](#) para rastrear as mudanças de status das efemérides.

Para solucionar uma efeméride inválida, consulte: [Solucionar problemas de efemérides inválidas](#)

Exemplo: carregamento de dados de efemérides de OEM de um bucket S3

Também é possível fazer upload de um arquivo de efemérides OEM diretamente de um bucket do S3 apontando para o bucket e a chave do objeto. AWS Ground Station recuperará o objeto em seu nome. As informações sobre a criptografia de dados em repouso AWS Ground Station estão detalhadas em: [Criptografia de dados em repouso para AWS Ground Station](#).

Abaixo está um exemplo de upload de um arquivo de efemérides OEM de um bucket S3

```

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']}")
```

Abaixo está um exemplo de dados retornados da ação [DescribeEphemeris](#) que está sendo chamada para as efemérides do OEM carregadas no bloco anterior do código de exemplo.

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

Forneça dados de efemérides de elevação de azimute

Important

O recurso de efemérides de elevação de azimute está atualmente em um estado de visualização e requer integração explícita.

A funcionalidade de efemérides de elevação de azimute está sob estrito controle de acesso para um número limitado de casos de uso especializados e predeterminados.

O acesso é significativamente mais restritivo do que os recursos de efemérides padrão fornecidos pelo cliente. Para obter mais informações sobre casos de uso aprovados e o processo de solicitação de acesso, abra um [AWS Support ticket](#) por meio do [AWS Support Center Console](#). Nossa equipe o guiará pelo processo de aprovação de casos de uso especializados.

Visão geral do

As efemérides de elevação de azimute fornecem uma maneira de especificar diretamente as direções de apontamento da antena sem fornecer informações orbitais do satélite. Em vez de enviar dados de efemérides que descrevem a órbita de um satélite, você fornece azimute e ângulos de elevação marcados com o tempo que informam à antena exatamente para onde apontar durante um contato.

AWS Ground Station trata as efemérides como dados de uso [individualizados](#). Se você usar esse recurso opcional, a AWS usará seus dados de efemérides para fornecer suporte à solução de problemas.

Essa abordagem é particularmente útil para os seguintes cenários:

- Suporte às operações iniciais: durante a fase de lançamento e órbita inicial (LEOP), quando dados orbitais precisos não estão disponíveis ou os parâmetros orbitais estão mudando rapidamente.
- Padrões de apontamento personalizados: implementação de sequências de apontamento específicas para testes de antenas ou operações não padrão.

Note

Ao usar efemérides de elevação de azimute, o ARN do satélite pode ser omitido da solicitação de reserva de contato. Se o ARN do satélite não for omitido, ele ainda será

incluído como parte dos dados de contato, mas as efemérides de elevação de azimute serão usadas para apontar a antena em vez de realizar a resolução prioritária das efemérides. A efeméride de elevação do azimute está associada a uma estação terrestre específica e define as direções de apontamento da antena para esse local.

Formato de dados de efemérides de elevação de azimute

Os dados de efemérides de elevação de azimute consistem em valores de azimute e elevação com marcação temporal organizados em segmentos. Cada segmento contém uma série de ângulos de azimute e elevação que abrangem um intervalo de tempo específico.

Os principais componentes dos dados de efemérides de elevação de azimute são:

- Ground Station: A estação terrestre específica onde essa efeméride de elevação de azimute será usada.
- Unidade de ângulo: A unidade de medida para ângulos (DEGREE_ANGLE ou RADIAN).
- Segmentos: uma ou mais coleções limitadas no tempo de azimute e ângulos de elevação.
- Ângulos marcados com data e hora: valores individuais de azimute e elevação com registros de data e hora associados.

Cada segmento exige:

- Uma época de referência (a hora base para o segmento)
- Um intervalo de tempo válido (horários de início e término do segmento)
- Pelo menos 5 pares marcados com data e hora azimuth/elevation

Restrições de elevação de azimute:

- Azimute em graus: -180° a 360°
- Azimute em radianos: $-\pi$ a 2π
- Elevação em graus: -90° a 90°
- Elevação em radianos: $-\pi/2$ a $\pi/2$
- Os valores de tempo devem estar em ordem crescente dentro de cada segmento
- Os segmentos não devem se sobrepor no tempo

Para obter mais informações, consulte a documentação [CreateEphemeris](#) da API e o tipo de [TimeAzEl](#) dados.

Criando efemérides de elevação de azimute

As efemérides de elevação de azimute são criadas usando a mesma ação de [CreateEphemeris](#) API, mas com o tipo de efemérides. azE1 As principais diferenças das efemérides de TLE e OEM são:

- Você deve especificar um `groundStation` parâmetro
- O `satelliteId` parâmetro deve ser omitido da solicitação
- As configurações de prioridade não se aplicam (cada efeméride de elevação de azimute é específica para uma estação terrestre)
- Cada segmento deve conter pelo menos 5 azimuth/elevation pontos para suportar a interpolação de Lagrange de 4ª ordem
- Limites e requisitos adicionais estão detalhados na documentação [CreateEphemeris](#) da API

É importante observar que o upload de uma efeméride define as efemérides como `VALIDATING` e inicia um fluxo de trabalho assíncrono que validará e gerará contatos potenciais a partir de suas efemérides. Uma efeméride só será usada para contatos depois de passar por esse fluxo de trabalho e seu status se tornar. `ENABLED` Você deve pesquisar o status das efemérides ou usar `CloudWatch` eventos [DescribeEphemeris](#) para rastrear as mudanças de status das efemérides.

Para solucionar uma efeméride inválida, consulte: [Solucionar problemas de efemérides inválidas](#)

Exemplo: criar efemérides de elevação de azimute por meio da API

O exemplo a seguir mostra como criar uma efeméride de elevação de azimute usando o AWS SDK for Python (Boto3):

```
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={
        "azE1": {
```

```

    "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']}")

```

Neste exemplo:

- Os dados de elevação do azimute estão associados à estação terrestre “Ohio 1”
- Os ângulos são especificados em graus
- O segmento abrange um período de 15 minutos
- Os dt valores são segundos atômicos, deslocados da época de referência.
- Seis azimuth/elevation pares são fornecidos (o mínimo é 5)

Exemplo: Carregar dados de elevação de azimute do S3

Para conjuntos de dados maiores, você pode carregar dados de elevação de azimute de um bucket do S3:

```
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']}")

```

O objeto S3 deve conter uma estrutura JSON com os dados de elevação de azimute no mesmo formato mostrado no exemplo de upload direto.

Reservando contatos com efemérides de elevação de azimute

Ao usar uma efeméride de elevação de azimute para reservar um contato, o processo difere das efemérides TLE e OEM:

1. Crie as efemérides de elevação do azimute usando [CreateEphemeris](#)
2. Aguarde até que as efemérides atinjam o status ENABLED
3. Reserve o contato usando [ReserveContact](#) substituições de rastreamento

Exemplo de reserva de contato com efemérides de elevação de azimute:

```

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

O `satelliteArn` parâmetro pode ser omitido ao reservar um contato com efemérides de elevação de azimute. A antena seguirá os ângulos de azimute e elevação especificados durante o contato.

Listando os contatos disponíveis

Ao usar efemérides de elevação de azimute, a API exige parâmetros específicos: [ListContacts](#)

- O `satelliteArn` parâmetro pode ser omitido da solicitação
- Você deve fornecer um `ephemeris` parâmetro com o ID de efemérides de elevação de azimute para especificar quais efemérides usar.
- As janelas de contato disponíveis mostram quando o azimute e os ângulos de elevação fornecidos estão acima da [máscara do local](#) da estação terrestre solicitada
- Você ainda deve fornecer `groundStation` e `missionProfileArn`

Exemplo de criar uma efeméride de elevação de azimute e listar os contatos disponíveis com ela:

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

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

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

```

        },
    ],
}
),
)

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

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

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

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

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

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

contacts_response = ground_station_client.list_contacts(
    startTime=start_time,
    endTime=end_time,
    groundStation="Stockholm 1",
    statusList=["AVAILABLE"],
    ephemeris={"azEl": {"id": ephemeris_id}},

```

```
# satelliteArn is optional
satelliteArn="arn:aws:groundstation::111122223333:satellite/a88611b0-f755-404e-
b60d-57d8aEXAMPLE",
missionProfileArn="arn:aws:groundstation:eu-north-1:111122223333:mission-
profile/966b72f6-6d82-4e7e-b072-f8240EXAMPLE",
)

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

Note

O `ephemeris` parâmetro com o ID de elevação de azimute deve ser fornecido ao listar contatos para especificar quais efemérides de elevação de azimute devem ser usadas para determinar as janelas de contato. Se `satelliteArn` for incluído, ele será associado aos dados de contato, mas as efemérides de elevação do azimute serão usadas para apontar a antena, em vez de realizar a resolução prioritária das efemérides.

Reserve contatos com efemérides personalizadas

Visão geral do

Ao usar efemérides personalizadas (TLE, OEM ou elevação de azimute), você pode reservar contatos usando a API. [ReserveContact](#) Esta seção descreve dois fluxos de trabalho comuns para reservar contatos e considerações importantes para garantir o sucesso do agendamento de contatos.

AWS Ground Station antenas são recursos compartilhados entre vários clientes. Isso significa que, mesmo que uma janela de contato apareça disponível quando você lista contatos, outro cliente pode

reservá-la antes de você. Portanto, é crucial verificar se seu contato atinge o SCHEDULED estado após a reserva e implementar o monitoramento adequado das mudanças no estado do contato.

Important

Para efemérides de elevação de azimute, o `satelliteArn` parâmetro pode ser omitido da `ReserveContact` solicitação e você deve fornecer o ID da efeméride. `trackingOverrides` Para efemérides de TLE e OEM, você ainda precisa fornecer o `satelliteArn`

Fluxos de trabalho de reservas de contatos

Há dois fluxos de trabalho principais para reservar contatos com efemérides personalizadas:

1. List-then-reserve fluxo de trabalho: primeiro liste as janelas de contato disponíveis usando [ListContacts](#), depois selecione e reserve uma janela específica. Essa abordagem é útil quando você deseja ver todas as oportunidades disponíveis antes de fazer uma seleção.
2. Fluxo de trabalho de reserva direta: reserve diretamente um contato para uma janela de tempo específica sem primeiro listar os contatos disponíveis. Essa abordagem é útil quando você já sabe o horário de contato desejado ou está trabalhando com horários predeterminados.

Ambos os fluxos de trabalho são válidos e a escolha depende de seus requisitos operacionais. As seções a seguir fornecem exemplos de cada abordagem.

Fluxo de trabalho 1: liste os contatos disponíveis e reserve

Esse fluxo de trabalho primeiro consulta as janelas de contato disponíveis e, em seguida, reserva uma janela específica. Isso é útil quando você deseja ver todas as oportunidades disponíveis antes de fazer uma seleção.

Exemplo: Listar e reservar com efemérides de elevação de azimute

```
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")

```

Exemplo: Listar e reservar com efemérides de TLE

```

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 .00000075 00000-0
26688-4 0 9997",
                    "tleLine2": "2 25994 98.2007 30.6589 0001234 89.2782 18.9934
14.57114995111906",
                    "validTimeRange": {
                        "startTime": datetime.now(timezone.utc),
                        "endTime": datetime.now(timezone.utc) + timedelta(days=7),
                    },
                }
            ]
        }
    },
)

ephemeris_id = 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")
```

Fluxo de trabalho 2: reserva de contato direto

Esse fluxo de trabalho reserva diretamente um contato sem primeiro listar as janelas disponíveis. Essa abordagem é útil quando você já sabe o horário de contato desejado ou está implementando o agendamento automatizado.

Exemplo: reserva direta com efemérides de elevação de azimute

```
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']}")

```

Exemplo: reserva direta com efemérides da TLE

```
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']}")
```

Monitorando mudanças no estado do contato

Depois de reservar um contato, é importante monitorar seu estado para garantir que ele faça a transição SCHEDULED e seja notificado sobre quaisquer problemas. AWS Ground Station emite eventos para a Amazon EventBridge para todas as mudanças de estado de contato.

Os estados de contato seguem este ciclo de vida:

- SCHEDULING- O contato está sendo processado para agendamento
- SCHEDULED- O contato foi agendado com sucesso e será executado
- FAILED_TO_SCHEDULE- O contato não pôde ser agendado (estado do terminal)

Para obter mais informações sobre estados de contato e ciclo de vida, consulte. [Entenda o ciclo de vida do contato](#)

Implementando o monitoramento do estado de contato com EventBridge

Para monitorar as mudanças no estado do contato em tempo real, você pode configurar uma EventBridge regra da Amazon que aciona uma função Lambda sempre que um contato da Ground Station muda de estado. Essa abordagem é mais eficiente e escalável do que pesquisar o status do contato.

Etapas de implementação

1. Crie uma função Lambda para processar eventos de mudança de estado de contato
2. Crie uma EventBridge regra que corresponda aos eventos de mudança de estado de contato da Ground Station
3. Adicione a função Lambda como destino para a regra

Exemplo de manipulador de funções Lambda

Para ver um exemplo completo de uma função Lambda que processa eventos de mudança de estado de contato, consulte o `GroundStationCloudWatchEventHandlerLambda` recurso no `AquaSnppJpssTerraDigIF.yml` CloudFormation modelo. Esse modelo está disponível no bucket Amazon S3 de integração do AWS Ground Station cliente. Para obter instruções sobre como acessar esse modelo, consulte a [Juntando tudo](#) seção do exemplo de endpoint de fluxo de dados.

EventBridge configuração de regras

A EventBridge regra deve usar o seguinte padrão de evento para corresponder a todas as mudanças de estado de contato da Ground Station:

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

Para filtrar somente estados específicos (por exemplo, falhas), você pode adicionar um filtro de detalhes:

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

```
"detail": {
  "contactStatus": [
    "FAILED_TO_SCHEDULE",
    "FAILED",
    "AWS_FAILED",
    "AWS_CANCELLED"
  ]
}
```

Para obter instruções detalhadas sobre a criação de EventBridge regras com alvos Lambda, consulte [Criação de regras que reagem a eventos](#) no Guia EventBridge do usuário da Amazon.

Configurando EventBridge regras para automação

Você pode criar EventBridge regras para responder automaticamente às mudanças no estado do contato. Por exemplo:

- Enviar notificações quando um contato não consegue agendar
- Acione as funções do Lambda para preparar recursos quando um contato entra PREPASS
- Registre as conclusões de contatos para fins de auditoria

Para obter informações detalhadas sobre como configurar EventBridge regras para AWS Ground Station eventos, consulte [Automatize AWS Ground Station com eventos](#).

Práticas recomendadas e considerações

Lidando com conflitos de agendamento

Como as AWS Ground Station antenas são recursos compartilhados, uma janela de contato que aparece disponível em `ListContacts` pode ser reservada por outro cliente antes que você possa reservá-la. Para lidar com isso:

1. Sempre verifique o status do contato após a reserva
2. Implemente a lógica de repetição com janelas de tempo alternativas
3. Considere reservar contatos com bastante antecedência, quando possível
4. Use EventBridge eventos para monitorar `FAILED_TO_SCHEDULE` estados

Tempo de validação de efemérides

Lembre-se de que as efemérides devem estar em vigor antes ENABLED que você possa usá-las para reservar contatos. O processo de validação normalmente leva de alguns segundos a alguns minutos, dependendo do tipo e tamanho da efeméride. Sempre verifique o status das efemérides antes de tentar reservar contatos.

Considerações sobre o tempo de contato

Ao usar efemérides personalizadas:

- Certifique-se de que suas efemérides cubram toda a duração do contato
- [Para efemérides de elevação de azimute, verifique se os ângulos mantêm a antena acima da máscara local durante todo o contato](#)
- Considere os prazos de expiração das efemérides ao agendar contatos futuros

Diferenças de API por tipo de efeméride

A ReserveContact API se comporta de forma diferente dependendo do tipo de efeméride:

Tipo de efemérides	É necessário o SatellitEarn	Substituições de rastreamento necessárias
TELHA	Sim	Não (opcional)
OEM	Sim	Não (opcional)
Elevação do azimute	Não (opcional)	Sim

Entenda quais efemérides são usadas

As efemérides têm prioridade, prazo de validade e sinalizador ativado. Juntos, eles determinam quais efemérides são usadas para rastreamento durante um contato.

Efemérides TLE e OEM

Para efemérides OEM e TLE, somente uma efeméride pode estar ativa para cada satélite. A efeméride que será usada é a efeméride habilitada de maior prioridade, cujo prazo de expiração

está no futuro. Um valor de prioridade maior indica uma prioridade mais alta. Os horários de contato disponíveis retornados por [ListContacts](#) são baseados nessas efemérides. Se várias efemérides ENABLED tiverem a mesma prioridade, as efemérides criadas ou atualizadas mais recentemente serão usadas.

Note

AWS Ground Station [tem uma cota de serviço no número de efemérides ENABLED fornecidas pelo cliente por satélite \(consulte: Cotas de serviço\)](#). Para carregar dados de efemérides após atingir essa cota, exclua (usando [DeleteEphemeris](#)) ou desabilite (usando [UpdateEphemeris](#)) as efemérides de menor prioridade/mais recentes criadas pelo cliente.

[Se nenhuma efeméride tiver sido criada, ou se nenhuma efeméride tiver ENABLED status, AWS Ground Station usará uma efeméride padrão para o satélite \(do Space-Track\), se disponível.](#) Essa efeméride padrão tem prioridade zero.

Efemérides de elevação de azimute

As efemérides de elevação de azimute funcionam de forma diferente das efemérides OEM e TLE. Cada efeméride de elevação de azimute está associada a uma estação terrestre específica e não tem prioridade. Ao reservar um contato com efemérides de elevação de azimute, você especifica explicitamente quais efemérides de elevação de azimute usar por meio do parâmetro `trackingOverrides`

Principais diferenças para efemérides de elevação de azimute:

- Sem sistema prioritário - você seleciona explicitamente as efemérides para cada contato
- Estação terrestre específica - cada efeméride está associada a uma estação terrestre específica
- Sem retorno automático - se as efemérides especificadas não estiverem disponíveis, o contato falhará

Note

As efemérides de elevação de azimute não competem com as efemérides OEM e TLE. Eles são selecionados explicitamente ao reservar um contato e são usados somente quando as substituições de rastreamento são especificadas.

Efeito de novas efemérides em contatos previamente agendados

Use a [DescribeContact API](#) para visualizar os efeitos de novas efemérides em contatos previamente agendados, retornando os horários de visibilidade ativos.

Para efemérides OEM e TLE, os contatos agendados antes do upload de uma nova efeméride manterão o horário de contato originalmente programado, enquanto o rastreamento da antena usará as efemérides ativas. Se a posição da espaçonave, com base nas efemérides ativas, for muito diferente das efemérides anteriores, isso pode resultar na redução do tempo de contato do satélite com a antena devido à operação da espaçonave fora da máscara do local. transmit/receive Portanto, recomendamos que você cancele e reagende seus futuros contatos depois de fazer o upload de uma nova efeméride que seja muito diferente da efeméride anterior.

Com a [DescribeContact API](#), você pode determinar a parte do seu contato futuro que está inutilizável devido à operação da espaçonave fora da máscara do transmit/receive local, comparando seu contato agendado `startTime` e `endTime` com o retornado `e.visibilityStartTime` e `e.visibilityEndTime`. Se você optar por cancelar e reagendar seus futuros contatos, o intervalo de tempo de contato não deve estar fora do intervalo de tempo de visibilidade em mais de 30 segundos. Os contatos cancelados podem incorrer em custos quando cancelados muito perto do momento do contato. Para obter mais informações sobre contatos cancelados, consulte: [Ground Station FAQs](#).

Para efemérides de elevação de azimute, os contatos agendados usarão as efemérides específicas que foram selecionadas quando o contato foi reservado. Se precisar atualizar os dados de elevação de azimute para um contato agendado, você pode cancelar e reagendar o contato com uma nova efeméride.

Obtenha as efemérides atuais de um satélite

As efemérides atuais em uso AWS Ground Station por um satélite específico podem ser recuperadas chamando as ações ou [GetSatelliteListSatellites](#). Ambos os métodos retornarão metadados para as efemérides atualmente em uso. Esses metadados de efemérides são diferentes para efemérides personalizadas enviadas para e efemérides padrão. AWS Ground Station

Note

As efemérides de elevação de azimute não estão associadas a satélites e, portanto, não são retornadas por ou [GetSatelliteListSatellites](#). Para recuperar informações sobre efemérides

de elevação de azimute, use a [DescribeEphemeris](#) API com o ID de efemérides específico ou use [ListEphemerides](#) para ver todas as efemérides disponíveis para sua conta.

As efemérides padrão incluirão apenas campos `source` e `epoch`. Essa `epoch` é a [época](#) do [conjunto de elementos de duas linhas](#) que foi retirado do [Space-Track](#) e atualmente está sendo usado para calcular a trajetória do satélite.

Uma efeméride personalizada terá um valor `source` de `CUSTOMER_PROVIDED` e incluirá um identificador exclusivo no campo `ephemerisId`. Esse identificador exclusivo pode ser usado para consultar as efemérides por meio da ação [DescribeEphemeris](#). Um nome campo opcional será retornado se a efeméride receber um nome durante o upload AWS Ground Station por meio da ação [CreateEphemeris](#).

É importante observar que as efemérides são atualizadas dinamicamente, AWS Ground Station portanto, os dados retornados são apenas um instantâneo das efemérides que estão sendo usadas no momento da chamada para a API.

Exemplo de retorno [GetSatellite](#) para um satélite usando uma efeméride padrão

```
{
  "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
  }
}
```

Exemplo de retorno [GetSatellite](#) para um satélite usando uma efeméride personalizada

```
{
  "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"
  }
}
```

Listando efemérides de elevação de azimute

Como as efemérides de elevação de azimute não estão associadas a satélites, você precisa usar diferentes APIs para descobrir e recuperar informações sobre elas:

1. Use [ListEphemerides](#) para listar todas as efemérides em sua conta, incluindo efemérides de elevação de azimute. Você pode filtrar por status e tipo de efeméride.
2. Use [DescribeEphemeris](#) com um ID de efeméride específico para obter informações detalhadas sobre uma efeméride de elevação de azimute.
3. Use [DescribeContact](#) com uma ID de contato específica para obter informações detalhadas sobre uma efeméride usada para o contato.

Exemplo de [ListEphemerides](#) resposta incluindo uma efeméride de elevação de azimute:

```
{
  "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

Na [ListEphemerides](#) resposta, as efemérides de elevação de azimute terão um `groundStation` campo em vez de um `satelliteId` campo, facilitando sua identificação.

Reverter para dados de efemérides padrão

Quando você carrega dados de efemérides personalizados, eles substituem os usos padrão de efemérides AWS Ground Station para aquele satélite específico. AWS Ground Station não usa as efemérides padrão novamente até que não haja efemérides atualmente habilitadas e não expiradas fornecidas pelo cliente disponíveis para uso. AWS Ground Station também não lista contatos após o prazo de expiração das efemérides atuais fornecidas pelo cliente, mesmo que haja uma efeméride padrão disponível após esse prazo de expiração.

Note

As efemérides de elevação de azimute não têm valores padrão e não substituem as efemérides de satélite. Eles são selecionados explicitamente ao reservar um contato usando o `trackingOverrides` parâmetro. Se você não quiser mais usar efemérides de elevação de azimute, basta reservar contatos sem especificar substituições de rastreamento, e o sistema usará as efemérides de satélite ativas.

Revertendo efemérides de TLE e OEM

Para voltar às efemérides padrão do [Space-Track](#) de um satélite, você precisará fazer o seguinte:

- Exclua (usando [DeleteEphemeris](#)) ou desative (usando [UpdateEphemeris](#)) todas as efemérides habilitadas fornecidas pelo cliente. Você pode listar as efemérides fornecidas pelo cliente para um satélite usando [ListEphemerides](#).
- Aguarde até que todas as efemérides existentes fornecidas pelo cliente expirem.

Você pode confirmar se a efeméride padrão está sendo usada chamando [GetSatellite](#) e verificando se o `source` da efeméride atual do satélite está `SPACE_TRACK`. Consulte [Dados de efemérides padrão](#) para obter mais informações sobre efemérides padrão.

Gerenciando efemérides de elevação de azimute

Como as efemérides de elevação de azimute são selecionadas explicitamente para cada contato e não estão associadas a satélites, não existe o conceito de “reverter” para um padrão. Em vez disso, você pode gerenciar as efemérides de elevação de azimute da seguinte forma:

- Para parar de usar efemérides de elevação de azimute: basta reservar novos contatos sem especificar e especificar `a.trackingOverrides.satelliteArn`. Em vez disso, o contato usará as efemérides ativas do satélite especificado.
- Para remover efemérides de elevação de azimute não utilizadas: Use [DeleteEphemeris](#) para excluir efemérides de elevação de azimute que não são mais necessárias. Observe que você não pode excluir uma efeméride que esteja sendo usada atualmente por um contato agendado.

Para listar todas as efemérides de elevação de azimute em sua conta, use [ListEphemerides](#). As efemérides de elevação de azimute podem ser identificadas pelo `ephemerisType` campo ou pela presença de um `groundStation` campo em vez de um `satelliteId` campo na resposta.

Trabalhe com fluxos de dados

AWS Ground Station usa uma relação de nó e borda para criar fluxos de dados para permitir o processamento de fluxo de seus dados. Cada nó é representado por uma configuração que descreve o processamento esperado. Para ilustrar esse conceito, considere um fluxo de dados de antenna-downlink até a. s3-recording O antenna-downlink nó representa a transformação analógica para digital do espectro de radiofrequência de acordo com os parâmetros definidos na configuração. O s3-recording representa um nó de computação que receberá os dados recebidos e os armazenará em seu bucket do S3. O fluxo de dados resultante é uma entrega assíncrona de dados de RF digitalizados para um bucket S3 com base em suas especificações.

Em seu perfil de missão, você pode criar vários fluxos de dados para atender às suas necessidades. As seções a seguir descrevem como configurar seus outros recursos da AWS para serem usados AWS Ground Station e oferecem recomendações para criar fluxos de dados. Para obter informações detalhadas sobre como cada nó se comporta, inclusive se ele é considerado um nó de origem ou destino, consulte. [Use AWS Ground Station configurações](#)

Tópicos

- [AWS Ground Station interfaces de plano de dados](#)
- [Use a entrega de dados entre regiões](#)
- [Instalar e configurar o Amazon S3](#)
- [Configurar e configurar a Amazon VPC](#)
- [Configurar e configurar a Amazon EC2](#)

AWS Ground Station interfaces de plano de dados

A estrutura de dados resultante do fluxo de dados escolhido depende da origem do fluxo de dados. Detalhes desses formatos são fornecidos a você durante a integração de seus satélites. O seguinte resume os formatos usados para cada tipo de fluxo de dados.

- downlink de antena
 - Os dados (largura less-than-or-equal de banda até 40MHz) são entregues como pacotes de dados de [sinal/formato IP VITA-49](#).
 - (Largura de banda maior que 40MHz) os dados são entregues como AWS Ground Station pacotes de Classe 2.

- antenna-downlink-demod-decode
 - Os dados são entregues como pacotes Demodulated/Decoded de dados/formato IP.
- uplink de antena
 - Os dados devem ser entregues como pacotes de [dados de sinal/formato IP VITA-49](#).
- antenna-uplink-echo
 - Os dados são entregues como pacotes de [dados de sinal/formato IP VITA-49](#).

Use a entrega de dados entre regiões

O recurso de entrega de dados AWS Ground Station entre regiões oferece a flexibilidade de enviar seus dados de uma antena para qualquer região AWS Ground Station compatível AWS . Isso significa que você pode manter sua infraestrutura em uma única região da AWS e agendar contatos em qualquer região à [AWS Ground Station Localizações](#) qual você esteja integrado.

Ao receber seus dados de contato em um Amazon S3 Bucket, AWS Ground Station gerenciaremos todos os aspectos de entrega para você.

Para usar a entrega de dados entre regiões para uma EC2 instância da Amazon (usando o AWS Ground Station Agente ou um endpoint de fluxo de dados), o endpoint de fluxo de dados deve ser criado na sua região atual da AWS e você deve especificar a mesma região. dataflow-endpoint-config AWS Ground Station gerenciará a entrega dos dados entre regiões para você.

Instalar e configurar o Amazon S3

Você pode utilizar um bucket do Amazon S3 para receber seus sinais de downlink usando. AWS Ground Station Para criar o s3-recording-config de destino, você deve ser capaz de especificar um bucket do Amazon S3 e uma função do IAM que autorize a gravação de arquivos no bucket. AWS Ground Station

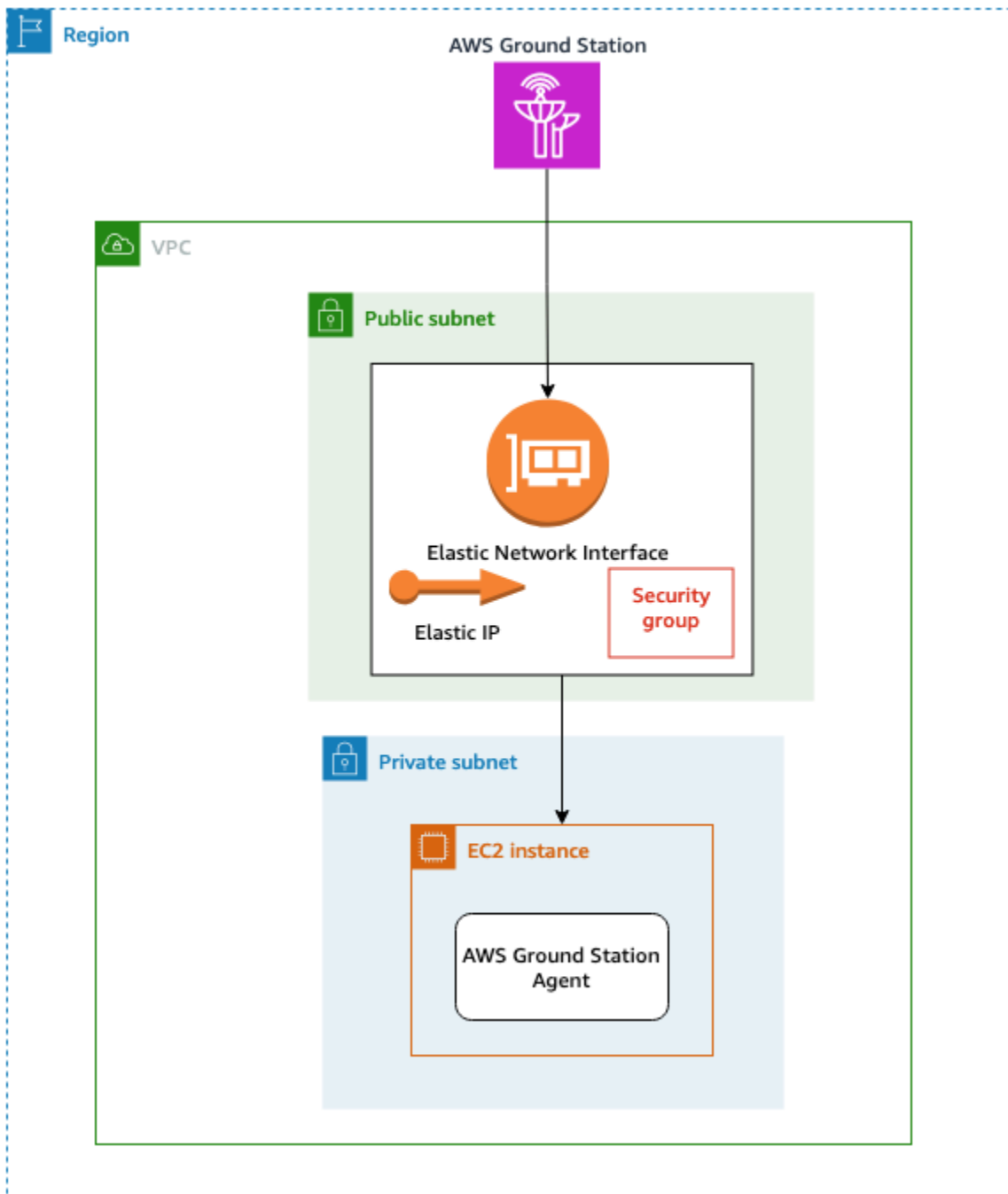
Consulte [Config de gravação do Amazon S3](#) as restrições sobre o bucket do Amazon S3, a função do IAM ou a criação de AWS Ground Station configurações.

Configurar e configurar a Amazon VPC

Um guia completo para configurar uma VPC está além do escopo deste guia. Para uma compreensão aprofundada, consulte o Guia do usuário da [Amazon VPC](#).

Nesta seção, é descrito como seu endpoint Amazon EC2 e de fluxo de dados podem existir em uma VPC. AWS Ground Station não oferece suporte a vários pontos de entrega para um determinado fluxo de dados. Espera-se que cada fluxo de dados termine em um único receptor. EC2 Como esperamos um único EC2 receptor, a configuração não é redundante Multi-AZ. Para ver exemplos completos de como usar sua VPC, consulte. [Exemplo de configurações de perfil de missão](#)

Configuração de VPC com agente AWS Ground Station



Seus dados de satélite são fornecidos para uma instância do AWS Ground Station Agente que está próxima à antena. O AWS Ground Station agente extrairá e criptografará seus dados usando a AWS KMS chave que você fornecer. Cada faixa é enviada para seu [Amazon EC2 Elastic IP \(EIP\)](#) a partir da antena de origem em todo o backbone da rede AWS. Os dados chegam à sua EC2 instância por meio da [Amazon EC2 Elastic Network Interface \(ENI\)](#) anexada. Uma vez na sua EC2 instância, o AWS Ground Station Agente instalado descriptografará seus dados e executará a correção de erros de encaminhamento (FEC) para recuperar os dados perdidos e, em seguida, os encaminhará para o IP e a porta especificados na sua configuração.

A lista abaixo destaca considerações de configuração exclusivas ao configurar sua VPC AWS Ground Station para entrega de agentes.

Grupo de segurança - É recomendável configurar um grupo de segurança dedicado somente ao AWS Ground Station tráfego. Esse grupo de segurança deve permitir o tráfego de entrada UDP no mesmo intervalo de portas especificado no seu grupo de endpoints do Dataflow. AWS Ground Station mantém uma lista de prefixos gerenciada pela AWS para restringir suas permissões somente AWS Ground Station a endereços IP. Consulte [as listas de prefixos gerenciados da AWS](#) para obter detalhes sobre como substituí-las em suas regiões de implantação. PrefixListId

Interface de rede elástica (ENI) - Você precisará associar o grupo de segurança acima a essa ENI e colocá-la em sua sub-rede pública.

Note

A cota padrão para o número de grupos de segurança anexados por ENI é 5. Esse é um limite ajustável de até 16, consulte Cotas [da Amazon VPC](#).

O CloudFormation modelo a seguir demonstra como criar a infraestrutura descrita nesta seção.

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:

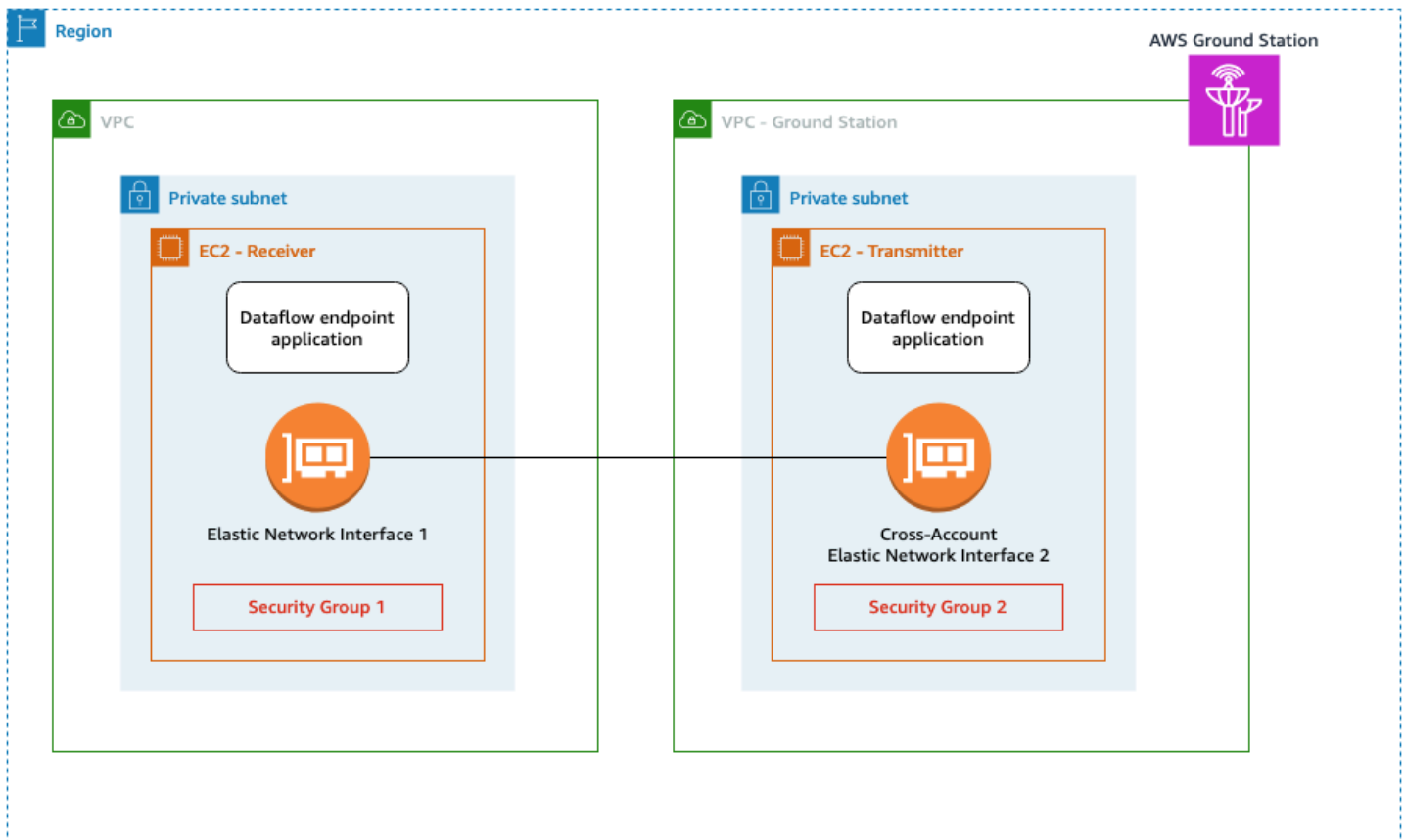
AllocationId:

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

NetworkInterfaceId:

Ref: *InstanceNetworkInterface*

Configuração de VPC com um endpoint de fluxo de dados



Seus dados de satélite são fornecidos a uma instância do aplicativo de endpoint de fluxo de dados que está próxima à antena. Em seguida, os dados são enviados por meio da [Amazon EC2 Elastic Network Interface \(ENI\)](#) entre contas de uma VPC de propriedade da AWS Ground Station. Em seguida, os dados chegam à sua EC2 instância por meio da ENI anexada à sua EC2 instância da Amazon. O aplicativo de endpoint de fluxo de dados instalado o encaminhará para o IP e a porta especificados na sua configuração. O inverso desse fluxo ocorre nas conexões de uplink.

A lista abaixo destaca considerações de configuração exclusivas ao configurar sua VPC para entrega de endpoints de fluxo de dados.

Note

A cota padrão para o número de grupos de segurança anexados por ENI é 5. Esse é um limite ajustável de até 16, consulte [Cotas da Amazon VPC](#).

Função do IAM — A função do IAM faz parte do endpoint do Dataflow e não é mostrada no diagrama. A função do IAM usada para criar e vincular a ENI entre contas à instância da AWS Ground Station Amazon EC2.

Grupo de segurança 1 - Esse grupo de segurança é anexado à ENI, que será associada à EC2 instância da Amazon em sua conta. Ele precisa permitir o tráfego UDP do Grupo de Segurança 2 nas portas especificadas em seu dataflow-endpoint-group.

Interface de rede elástica (ENI) 1 - Você precisará associar o grupo de segurança 1 a essa ENI e colocá-la em uma sub-rede.

Sub-rede - Você precisará garantir que haja pelo menos um endereço IP disponível por fluxo de dados para a EC2 instância da Amazon em sua conta. Para obter mais detalhes sobre o tamanho da sub-rede, consulte Blocos CIDR de [sub-rede](#)

Grupo de segurança 2 - Esse grupo de segurança é referenciado no Dataflow Endpoint. Esse grupo de segurança será anexado à ENI que AWS Ground Station será usada para colocar dados em sua conta.

Região - Para obter mais informações sobre as regiões suportadas para conexões entre regiões, consulte [Use a entrega de dados entre regiões](#).

O CloudFormation modelo a seguir demonstra como criar a infraestrutura descrita nesta seção.

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*

Configurar e configurar a Amazon EC2

É necessário configurar adequadamente sua EC2 instância da Amazon para que a entrega síncrona do VITA-49 Signal/IP data or VITA-49 Extension data/IP seja entregue por meio do AWS Ground Station Agente ou de um endpoint de fluxo de dados. Dependendo de suas necessidades específicas, você pode executar o processador Front End (FE) ou o Software Defined Radio (SDR) diretamente na mesma instância ou pode precisar utilizar EC2 instâncias adicionais. A seleção e instalação do seu FE ou SDR estão além do escopo deste guia do usuário. Para obter mais informações sobre os formatos de dados específicos, consulte [AWS Ground Station interfaces de plano de dados](#).

Para obter informações sobre nossos termos de serviço, consulte os [Termos AWS de Serviço](#).

Software comum fornecido

AWS Ground Station fornece software comum para facilitar a configuração da sua EC2 instância Amazon.

AWS Ground Station Agente

O AWS Ground Station agente recebe dados de downlink de frequência intermediária digital (DigiF) e emite dados descriptografados que permitem o seguinte:

- Capacidade de downlink do DigiF de 40 a 400 MHz de largura MHz de banda.
- Entrega de dados DigiF de alta taxa e baixa instabilidade para qualquer IP público (IP AWS elástico) na rede. AWS
- Entrega confiável de dados usando Forward Error Correction (FEC).
- Entrega segura de dados usando uma AWS KMS chave gerenciada pelo cliente para criptografia.

Para obter mais informações, consulte o [Guia do usuário do AWS Ground Station agente](#).

Aplicativo de endpoint Dataflow

Um aplicativo de rede usado AWS Ground Station para enviar e receber dados entre os locais da AWS Ground Station antena e suas EC2 instâncias da Amazon. Ele pode ser usado para o uplink e downlink de dados.

Rádio definido por software (SDR)

Um rádio definido por software (SDR) que pode ser usado para modular/desmodular o sinal usado para se comunicar com seu satélite.

AWS Ground Station Imagens de máquinas da Amazon (AMIs)

Para reduzir os tempos de construção e configuração dessas instalações, AWS Ground Station também oferece ofertas AMIs pré-configuradas. AMIs Com um aplicativo de rede de endpoint de fluxo de dados e um rádio definido por software (SDR), são disponibilizados para sua conta após a conclusão da integração. Eles podem ser encontrados no EC2 console da Amazon pesquisando por estação terrestre em [Amazon Machine Images \(AMIs\)](#) privadas. Os AMIs with AWS Ground Station Agent são públicos e podem ser encontrados no EC2 console da Amazon pesquisando por groundstation em [Amazon Machine Images \(AMIs\)](#) públicas.

Trabalhe com telemetria

AWS Ground Station a telemetria fornece métricas quase em tempo real das AWS Ground Station antenas durante seus contatos via satélite. Você pode usar dados de telemetria para monitorar o desempenho do contato, detectar anomalias e tomar decisões informadas sobre suas comunicações via satélite.

Tópicos

- [Como funciona a telemetria](#)
- [Tipos de telemetria disponíveis](#)
- [Disponibilidade regional](#)
- [Configurar a telemetria](#)
- [Entenda os dados de telemetria](#)

Como funciona a telemetria

Para usar a telemetria, você configura um `TelemetrySinkConfig` que especifica onde AWS Ground Station deve entregar os dados de telemetria. Em seguida, você adiciona essa configuração ao seu perfil de missão usando o `telemetrySinkConfigArn` campo. Durante os contatos que usam um perfil de missão habilitado para telemetria, AWS Ground Station transmite dados de telemetria para sua conta.

O processo de entrega por telemetria funciona da seguinte forma:

1. Você cria um stream do Kinesis Data Streams AWS em sua conta para receber dados de telemetria. O stream deve ser criado na mesma conta e região em que você agenda seus contatos.
2. Você cria uma função do IAM que concede AWS Ground Station permissão para gravar dados no seu stream.
3. Você cria um `TelemetrySinkConfig` que faz referência ao seu stream e à função do IAM.
4. Você adiciona o `TelemetrySinkConfig` ao seu perfil de missão.
5. Você lista e reserva contatos usando o novo perfil de missão habilitado para telemetria.
6. Durante os contatos usando esse perfil de missão, AWS Ground Station transmite dados de telemetria para seu stream do Kinesis Data Streams quase em tempo real.

7. Você consome e processa os dados de telemetria do seu stream usando AWS serviços ou seus próprios aplicativos.

Tipos de telemetria disponíveis

AWS Ground Station fornece os seguintes tipos de telemetria durante os contatos:

Note

AWS Ground Station está trabalhando na expansão do número de tipos de telemetria compatíveis

Telemetria apontadora

Fornecer informações sobre a direção do apontamento da antena durante os contatos do satélite. Esse tipo de telemetria é sempre enviado durante um contato e inclui ângulos de azimute e elevação reais e comandados. Para obter mais informações, consulte [Telemetria apontadora](#).

Telemetria de rastreamento

Fornecer informações sobre o status de rastreamento da antena e os erros de rastreamento. Esse tipo de telemetria é enviado quando o rastreamento automático está ativado em sua configuração de rastreamento. Para obter mais informações, consulte [Telemetria de rastreamento](#).

Disponibilidade regional

A telemetria está disponível em todas as AWS regiões onde AWS Ground Station opera. Durante a execução do contato, a telemetria será entregue da AWS Ground Station antena para a região da qual você agendou seu contato, fornecendo suporte entre regiões.

Para obter uma lista completa das AWS Ground Station regiões e localizações das estações terrestres, consulte [AWS Ground Station Localizações](#).

Configurar a telemetria

Siga estas etapas para configurar a telemetria para seus AWS Ground Station contatos. Depois de concluir essa configuração, os dados de telemetria serão entregues ao seu stream do Kinesis Data Streams durante os contatos que usam um perfil de missão habilitado para telemetria. Para uma

compreensão aprofundada do Kinesis Data Streams, consulte o Guia do usuário do Kinesis [Data Streams](#).

Etapa 1: criar recursos de pré-requisitos AWS

O CloudFormation trecho a seguir demonstra como criar os recursos necessários para a entrega de AWS telemetria. Esse trecho cria um stream do Kinesis Data Streams e uma função do IAM que AWS Ground Station concede permissão para gravar dados de telemetria no stream.

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
```

A lista abaixo destaca considerações de configuração exclusivas ao configurar a entrega de telemetria para AWS Ground Station

Stream do Kinesis Data Streams — O stream usa o modo de capacidade sob demanda, que é escalado automaticamente com base na taxa de transferência. Isso é recomendado para a maioria dos casos de uso. O stream está configurado para reter dados por 24 horas. Por padrão, o stream usa criptografia AWS gerenciada. Para usar a criptografia gerenciada pelo cliente AWS Key Management Service, adicione a `StreamEncryption` propriedade e atualize a política de função do IAM para incluir `kms:GenerateDataKey` a permissão. Para obter mais informações, consulte [Proteção de dados no Amazon Kinesis Data Streams](#).

Função do IAM - A função do IAM permite que o diretor do `groundstation.amazonaws.com` serviço assuma a função e grave dados de telemetria no seu stream do Kinesis Data Streams. A política de funções concede permissões para `kinesis:DescribeStreamkinesis:PutRecord`, e `kinesis:PutRecords` ações no stream. Consulte [Configuração do coletor de telemetria](#) para obter orientação sobre como configurar a política de confiança e a política de funções.

Configuração adicional: adicione `iam:PassRole` permissões ao usuário ou à função do IAM que você usa para chamadas de AWS Ground Station API. Isso permite que você passe a função de telemetria para AWS Ground Station ao criar um. `TelemetrySinkConfig`

Exemplo `PassRole` de política

Para obter mais informações sobre como atualizar ou anexar uma política de função, consulte [Gerenciamento de políticas do IAM](#) no Guia do usuário do IAM. Para obter mais informações sobre a `iam:PassRole` permissão, consulte [Conceder permissões a um usuário para passar uma função para um serviço da AWS](#)

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

Etapa 2: criar um TelemetrySinkConfig

Crie um TelemetrySinkConfig que defina como AWS Ground Station entregará dados de telemetria ao seu stream do Kinesis Data Streams. Use o ARN do stream e o ARN da função das saídas da CloudFormation pilha na Etapa 1.

Note

Ao criar um TelemetrySinkConfig, AWS Ground Station verificará o acesso ao seu stream do Kinesis Data Streams entregando um registro de teste vazio com uma chave de partição de `test`.

Para obter mais informações sobre como criar um TelemetrySinkConfig, consulte [Configuração do coletor de telemetria](#).

Etapa 3: adicionar telemetria ao seu perfil de missão

Crie um perfil de missão. Para obter mais informações sobre a criação de perfis de missão, consulte [Use perfis de AWS Ground Station missão](#). Adicione o `telemetrySinkConfigArn` ao seu perfil de missão para permitir a entrega de telemetria durante os contatos. Use o ARN do TelemetrySinkConfig criado na Etapa 2.

Etapa 4: agendar um contato

Agende um contato usando seu perfil de missão habilitado para telemetria. Durante o contato, AWS Ground Station transmitirá dados de telemetria para seu stream do Kinesis Data Streams.

O que esperar durante os contatos

- Início da telemetria - Os dados começam a ser transmitidos quando o contato é iniciado.
- Entrega quase em tempo real — a telemetria chega ao seu stream do Kinesis Data Streams quase em tempo real.
- Duração do contato - Os dados continuam durante todo o contato.
- Parada automática - A telemetria interrompe a transmissão quando o contato termina.

Monitorando a entrega

Você pode monitorar a entrega de telemetria usando:

- Métricas de stream do Kinesis Data Streams - Verifique os registros recebidos. CloudWatch Para obter mais informações, consulte [Monitoramento do Amazon Kinesis Data Streams](#).
- Registros do aplicativo - verifique o processamento de dados em seus aplicativos que consomem do stream.
- Kinesis Data Viewer — use o console de stream do Kinesis Data Streams para ver exemplos de registros do seu stream.

Próximas etapas

Depois de concluir a configuração, você pode:

- Saiba mais sobre o formato dos dados de telemetria e os tipos de telemetria disponíveis. Consulte [Entenda os dados de telemetria](#).
- Crie aplicativos para processar dados de telemetria do seu stream do Kinesis Data Streams. Para obter mais informações, consulte [Criação de consumidores para o Amazon Kinesis Data Streams](#).
- Crie painéis e alertas usando CloudWatch outros AWS serviços.
- Revise as orientações de solução de problemas se você encontrar problemas. Consulte [Solucionar problemas de telemetria](#).

Entenda os dados de telemetria

Os dados de telemetria são entregues como registros JSON codificados em Base64 para seu stream do Kinesis Data Streams. Cada registro contém informações coletadas durante seu contato via satélite, incluindo metadados sobre o contato e as medições de telemetria amostradas.

Visão geral do formato de dados

Cada registro de telemetria contém os seguintes componentes:

Tipo e versão da telemetria

Identifica o tipo específico de dados de telemetria e sua versão do esquema. Isso permite que você analise diferentes tipos de telemetria de forma adequada. Para obter mais informações sobre o controle de versão do esquema, consulte [Controle de versão e evolução do esquema](#)

ID do escopo

Um identificador exclusivo para o escopo da telemetria. Isso permite que você correlacione dados de telemetria com contatos específicos.

Metadados

Informações contextuais sobre a telemetria.

Dados

As medidas de telemetria amostradas específicas para o tipo de telemetria.

Chave de partição

Os registros de telemetria são entregues ao seu stream do Kinesis Data Streams com uma chave de partição no formato:

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

Essa chave de partição garante que toda a telemetria de um determinado tipo para um único contato seja entregue ao mesmo fragmento em seu stream do Kinesis Data Streams, oferecendo a melhor ordenação de esforços para o stream de telemetria desse contato.

Telemetria apontadora

A telemetria de apontamento fornece informações sobre a direção do apontamento da antena durante os contatos do satélite. Esse tipo de telemetria é sempre enviado durante um contato.

Campos de dados

Carimbo de data/hora da amostra

Hora em que os dados de telemetria foram amostrados, no formato ISO-8601 em UTC com precisão de milissegundos.

azimute

Ângulo de azimute real da antena em graus.

elevação

Ângulo de elevação real da antena em graus.

Comandou Azimuth

Ângulo de azimute comandado em graus. Esse é o ângulo de azimute alvo que a antena está tentando alcançar.

Elevação comandada

Ângulo de elevação comandado em graus. Esse é o ângulo de elevação alvo que a antena está tentando alcançar.

Note

A posição real da antena pode ser diferente da posição comandada devido a limitações físicas ou atrasos mecânicos durante o contato.

Campos de metadados

Estação Terrestre

Nome da estação terrestre (por exemplo, "Ohio 1").

ID de satélite

Identificador do recurso de satélite em AWS Ground Station.

contactId

Identificador do contato.

Exemplo de 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
}
```

Telemetria de rastreamento

A telemetria de rastreamento fornece informações sobre o status de rastreamento da antena e os erros de rastreamento. Esse tipo de telemetria é enviado quando o rastreamento automático está ativado em sua configuração de rastreamento e quando a antena está usando ativamente o rastreamento automático.

Note

Se o `autotrack` parâmetro em seu `TrackingConfig` estiver definido como `REMOVED`, nenhuma telemetria de rastreamento será fornecida. Para obter mais informações sobre configurações de rastreamento, consulte [Configuração de rastreamento](#).

Campos de dados

Carimbo de data/hora da amostra

Hora em que os dados de telemetria foram amostrados, no formato ISO-8601 em UTC com precisão de milissegundos.

Status de rastreamento

Status atual de rastreamento da antena. Os valores possíveis são:

- **TRACKING**— A antena foi bloqueada com sucesso em um sinal que corresponde ao perfil da missão e o está seguindo ativamente pelo céu. Esse é o estado operacional nominal durante um contato.
- **ACQUIRING**— A antena está em processo de localização e bloqueio do sinal. Atualmente, o sistema está usando rastreamento programático, apontando com base em dados de efemérides.

- MASKED— A posição prevista do satélite está atrás de uma máscara de rastreamento automático, o que significa que a antena não pode utilizar o rastreamento automático de forma confiável nessa direção específica de apontamento. Isso normalmente ocorre em áreas de alta interferência de RF, como baixas elevações.

trackingErrorAzimuth

Erro de rastreamento no eixo do azimute, medido em graus.

trackingErrorElevation

Erro de rastreamento no eixo de elevação, medido em graus.

Note

Os valores de erro de rastreamento representam ajustes da trilha do programa baseada em efemérides que AWS Ground Station se aplica durante o rastreamento automático para maximizar a intensidade do sinal.

Campos de metadados

A telemetria de rastreamento inclui os mesmos campos de metadados da telemetria apontadora:, e
groundStation satelliteId contactId

Exemplo de 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
  }
}
```

```
}  
}
```

Lendo dados do stream do Kinesis Data Streams

Os dados de telemetria são entregues ao seu stream do Kinesis Data Streams e podem ser consumidos usando padrões de consumo de stream padrão. Ao ler os dados do seu stream, tenha em mente as seguintes considerações.

Decodificação Base64

Os dados no stream do Kinesis Data Streams são codificados em Base64. Você deve decodificar os dados antes de analisá-los como JSON. Para obter mais informações, consulte Como [trabalhar com o Amazon Kinesis Data Streams](#).

Usando o Kinesis Data Viewer

Para acesso rápido aos seus dados de telemetria, o console de stream do Kinesis Data Streams oferece um recurso Visualizador de dados. Ao usar esse recurso:

- A entrega de telemetria pode ocorrer em qualquer fragmento em seu stream.
- A posição inicial padrão é a dos registros mais recentes no fragmento.
- Talvez seja necessário ajustar o fragmento selecionado e usar a posição inicial “No timestamp” para visualizar os registros recebidos.

Usando a biblioteca de cliente Kinesis

A Kinesis Client Library (KCL) gerencia muitas das complexidades associadas ao consumo de dados do stream do Kinesis Data Streams, incluindo gerenciamento de fragmentos, pontos de verificação e balanceamento de carga. Recomendamos usar o KCL para aplicativos de consumo de telemetria de produção.

Para obter mais informações, consulte [Desenvolvendo consumidores usando a biblioteca cliente Kinesis](#).

Melhores práticas de consumo

- Minimize a latência — Use o Enhanced Fan-Out para ler o stream do Kinesis Data Streams com taxa de transferência dedicada e menor latência em comparação com a pesquisa. Para obter mais informações, consulte [Desenvolvendo consumidores avançados de fan-out](#).

- **Stream dedicado** — Use um stream dedicado do Kinesis Data Streams AWS Ground Station para sua integração de telemetria. Compartilhar um stream com outros aplicativos pode causar saturação da taxa de transferência de gravação e falhas na entrega de telemetria.
- **Capacidade sob demanda** — implante seu stream do Kinesis Data Streams no modo de provisionamento sob demanda para permitir o escalonamento automático de fragmentos com base na taxa de transferência.
- **Monitore a taxa de transferência** - monitore a limitação do fluxo usando métricas. CloudWatch Para obter mais informações, consulte [Monitoramento do Amazon Kinesis Data Streams](#).

Controle de versão e evolução do esquema

Os esquemas de telemetria são versionados para apoiar a evolução ao longo do tempo. O `telemetryVersion` campo em cada registro indica a versão do esquema.

Lidando com alterações no esquema

- Novos tipos de telemetria podem ser introduzidos no futuro.
- Os tipos de telemetria existentes podem receber novas versões com alterações significativas.
- Seus aplicativos devem tolerar tipos e versões de telemetria desconhecidos.
- Analise os `telemetryVersion` campos `telemetryTypeAndVersion` e `telemetryType`, e para determinar como processar cada registro.

Recomendamos implementar a serialização de carga útil com reconhecimento de versão que possa lidar com várias versões do esquema normalmente, permitindo que seus aplicativos continuem funcionando quando novas versões forem introduzidas.

Trabalhe com contatos

Você pode inserir dados de satélite, identificar localizações de antenas, comunicar-se e programar o horário da antena para satélites selecionados usando o AWS Ground Station console ou o AWS SDK no idioma de sua escolha. AWS CLI Você pode revisar, cancelar e reagendar reservas de contato até 15 minutos antes do início do contato*. Você também pode atualizar um contato para especificar uma anulação de efemérides — incluindo dados de rastreamento de azimute/elevação, OEM ou TLE — ou alterar o satélite alvo. Para obter mais informações, consulte [Atualizar contatos e controle de versão de contatos](#). Além disso, você pode visualizar os detalhes dos planos de preço de minutos reservados, se estiver usando o modelo de definição de preço de minutos reservados do AWS Ground Station .

AWS Ground Station oferece suporte à entrega de dados entre regiões. As configurações de endpoint do fluxo de dados que são parte do perfil da missão selecionado determinam para quais regiões os dados são entregues. Para obter mais informações sobre como usar a entrega de dados entre regiões, consulte [Use a entrega de dados entre regiões](#).

Para agendar contatos, os recursos devem estar configurados. Se você não configurou seus recursos, consulte [Conceitos básicos](#). Quando [ReserveContact](#) é chamado, AWS Ground Station tira uma foto do perfil da missão e configura os recursos para uso em todo o ciclo de vida do contato. As alterações nesses recursos usando o [UpdateMissionProfile](#) e não [UpdateConfig](#) APIs serão refletidas nos contatos reservados antes das atualizações. Se você precisar que as alterações de recursos sejam aplicadas a um contato já agendado, você deve primeiro cancelar o contato usando o [CancelContact](#), em seguida, reagendá-lo usando [ReserveContact](#).

* Os contatos cancelados podem incorrer em custos quando cancelados muito perto do momento do contato. Para obter mais informações sobre contatos cancelados, consulte: [Ground Station FAQs](#).

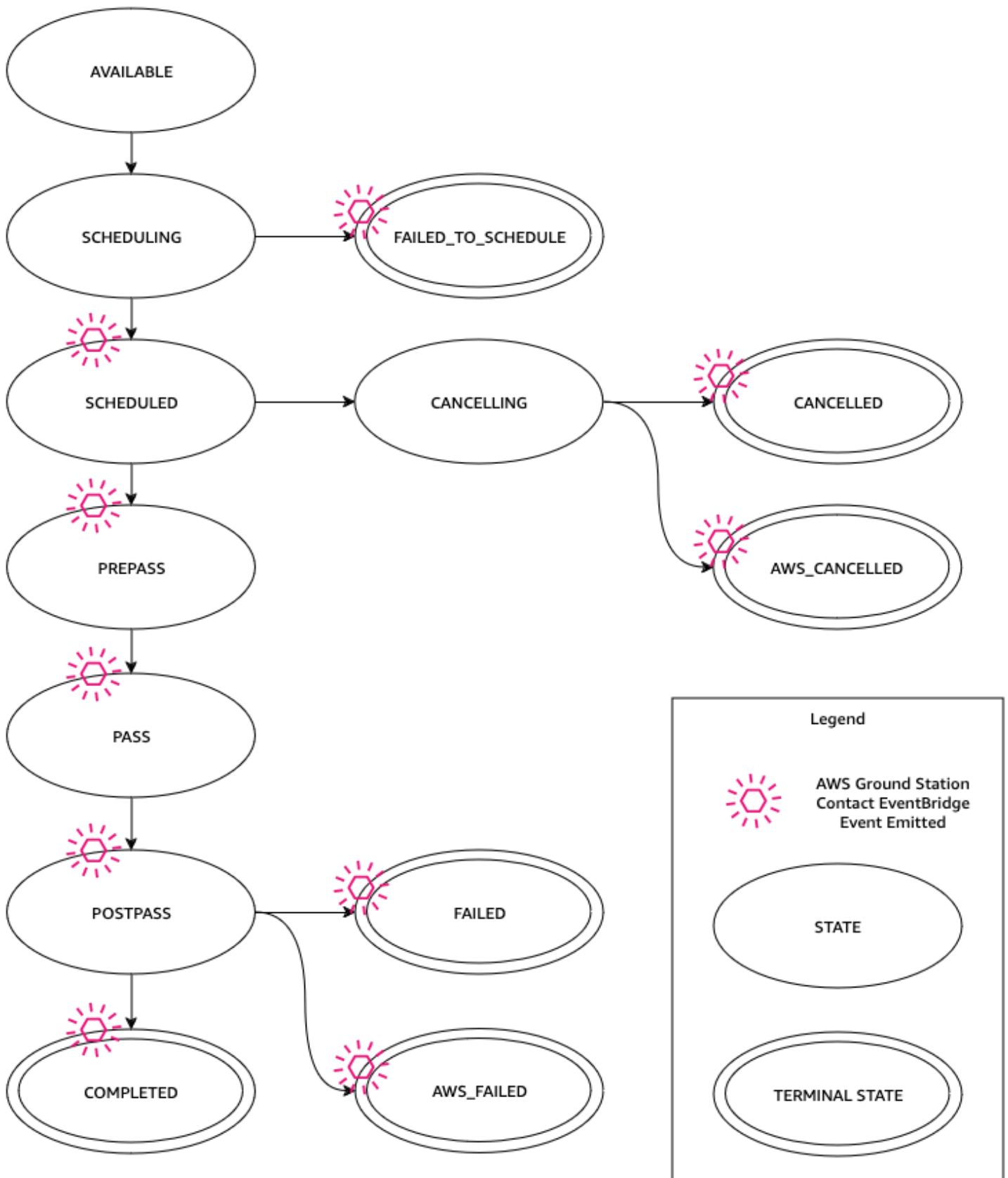
Tópicos

- [Entenda o ciclo de vida do contato](#)
- [Entenda o faturamento de contatos](#)
- [Atualizar contatos e controle de versão de contatos](#)

Entenda o ciclo de vida do contato

Compreender o ciclo de vida do contato pode ajudá-lo a automatizar e solucionar vários problemas durante o uso. AWS Ground Station O diagrama a seguir mostra o ciclo de vida do AWS Ground

Station contato, bem como os eventos do Event Bridge emitidos durante o ciclo de vida. É importante observar que COMPLETED, FAILED, FAILED_TO_SCHEDULE, CANCELLED, AWS_CANCELLED e são estados terminais. AWS_FAILED Os contatos não sairão de um estado terminal. Consulte o [AWS Ground Station status de contato](#) para obter detalhes sobre o que cada status indica e se ele pode ser interrompido ou cancelado usando. [CancelContact](#)



AWS Ground Station status de contato

O status de um AWS Ground Station contato fornece informações sobre o que está acontecendo com esse contato em um determinado momento.

Status de contato

A tabela a seguir descreve os status que um contato pode ter:

Status	Description	Terminal	Cancelável	Parável
AVAILABLE	O contato está disponível para ser reservado.	Não	N/D	N/D
PROGRAMAÇÃO	O contato está em processo de agendamento.	Não	Sim	Não
SCHEDULED	O contato foi agendado com sucesso.	Não	Sim	Não
NÃO FOI POSSÍVEL AGENDAR	O contato falhou no agendamento.	Sim	Não	Não
PRÉ-PASSAGEM	O contato começará em breve e os recursos estão sendo preparados.	Não	Sim	Não
PASS	O contato está sendo executado no momento e o satélite está sendo comunicado com.	Não	Não	Sim
POSTPASS	A comunicação foi concluída e os recursos usados estão sendo limpos.	Não	Não	Não
CONCLUÍDO	O contato foi concluído sem erros.	Sim	Não	Não

Status	Description	Terminal	Cancelável	Parável
FAILED	O contato falhou devido a um problema com a configuração do seu recurso.	Sim	Não	Não
AWS_FAILED	O contato falhou devido a um problema no AWS Ground Station serviço.	Sim	Não	Não
CANCELANDO	O contato está em processo de cancelamento.	Não	Não	Não
AWS_CANCELED	O contato foi cancelado pelo AWS Ground Station serviço. A manutenção da antena ou do local e o desvio de efemérides são exemplos de quando isso pode acontecer.	Sim	Não	Não
CANCELADO	O contato foi cancelado por você.	Sim	Não	Não

Note

Para obter informações sobre as implicações de cobrança de contatos cancelados ou interrompidos, consulte. [Entenda o faturamento de contatos](#)

Retenção de dados de contato

AWS Ground Station retém os dados de contato por 1 ano após a [ReserveContacts](#) solicitação de reserva de um contato. Após o período de 1 ano, os dados de contato são excluídos.

Se você precisar reter dados de contato por mais de um ano, é recomendável exportar seus dados antes que o período de retenção expire. Para obter mais informações sobre como acessar e exportar dados de contato, consulte:

- [AWS Ground Station API Reference](#)

- [AWS Ground Station Referência de comandos da CLI](#)

Entenda o faturamento de contatos

Com AWS Ground Station, você paga apenas pelo tempo de uso da antena. AWS Ground Station mede o uso de contatos por minuto. Para cada contato, o serviço calcula a duração do contato do início ao fim e arredonda para o minuto mais próximo. Essa duração medida determina suas cobranças por esse contato.

Sua tarifa depende de dois fatores principais:

- Largura de banda — A quantidade de largura de banda reservada para o contato (banda estreita ou banda larga)
- Localização da estação terrestre — As tarifas variam de acordo com a localização da estação terrestre

Definições de largura de banda

AWS Ground Station categoriza os contatos em dois níveis de largura de banda com base na largura de banda instantânea:

- Banda estreita — Qualquer contato em que a largura de banda instantânea seja menor ou igual a 40 MHz
- Banda larga — Qualquer contato em que a largura de banda instantânea seja maior que 40 MHz

Modos de agendamento

AWS Ground Station oferece dois modos de agendamento:

- Sob demanda — pague pelo acesso à antena sem compromissos de longo prazo
- Reservado — oferece uma tarifa com desconto e um agendamento aprimorado em comparação com o On-Demand, com um compromisso mensal. O preço por minuto reservado está disponível para clientes que se comprometem com o uso mensal por um determinado período de tempo.

Para obter informações específicas sobre preços para sua conta ou para saber mais sobre o modo de agendamento reservado, entre em contato com seu representante da AWS.

CancelContact

O uso da [CancelContact](#) API varia de acordo com o estado do contato quando você a chama:

- Antes do início do contato - cancela totalmente o contato
- Após o início do contato e antes do término do contato - interrompe o contato em andamento

Quando você cancela um contato, o faturamento depende do seu modo de agendamento e de quando você cancela. Para obter mais informações, entre em contato com seu representante da AWS.

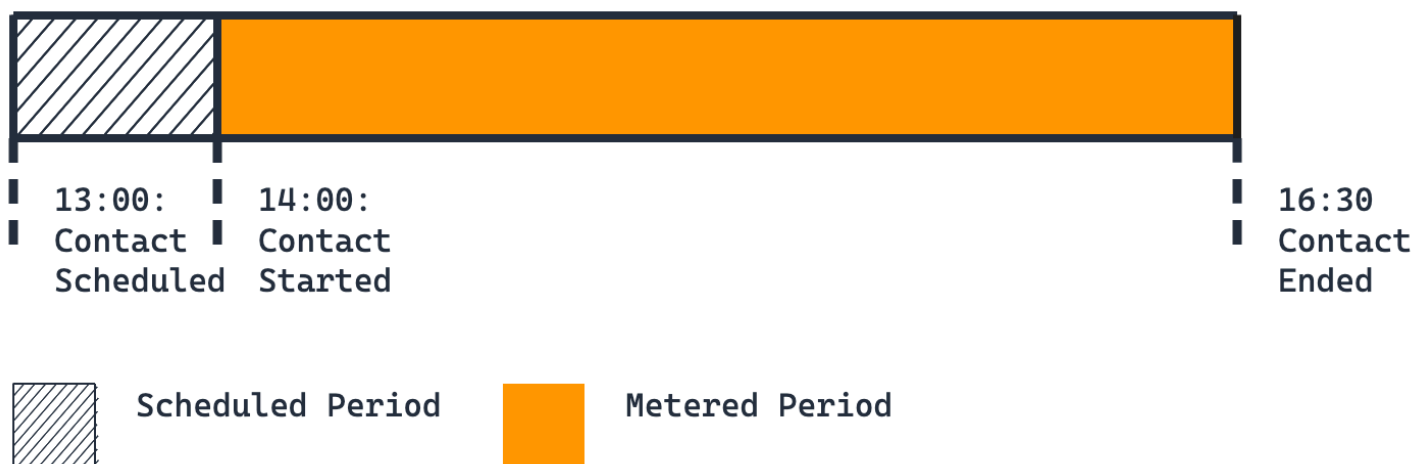
Quando você interrompe um contato, você é cobrado pela parte do contato que foi executada e pelo tempo restante que não é coberto por contatos duplicados. Um contato duplicado nesse contexto foi:

- Programado na mesma estação terrestre do contato interrompido original
- Programado com o mesmo ID de conta da AWS do contato original interrompido
- Reservado após a emissão do comando para interromper o contato original

Os cenários a seguir demonstram como essa medição funciona na prática.

Cenário 1: contato único

Você agenda um contato de 150 minutos na Ground Station Anytown 1 para começar às 14:00 e terminar às 16:30.



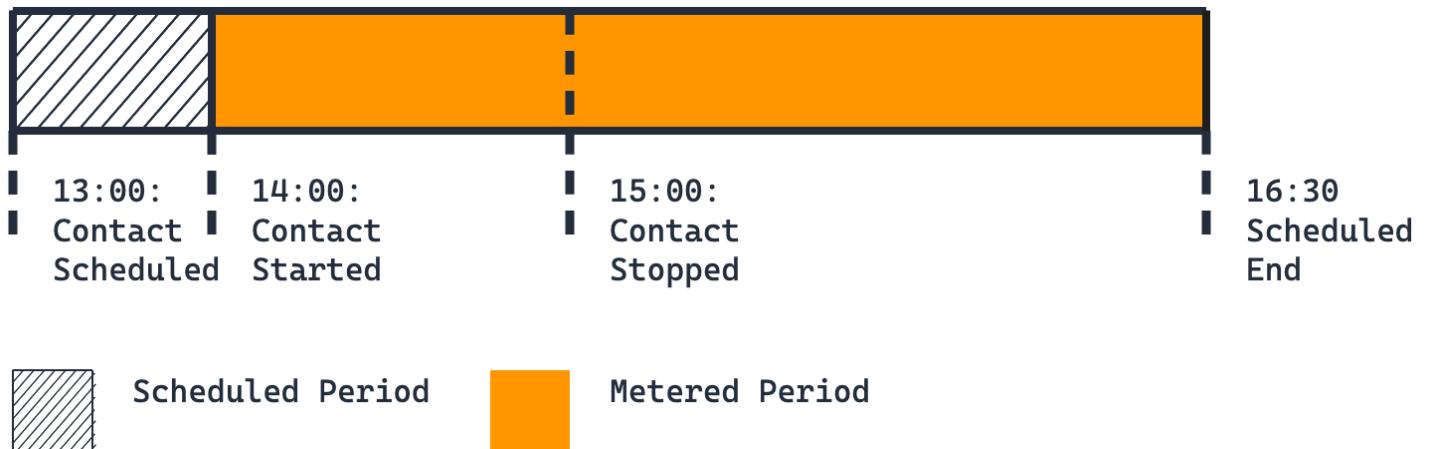
Detalhamento do faturamento:

- Primeiro contato: 150 minutos (duração total)

Você é cobrado por 150 minutos. Esse é o cenário básico em que um contato segue até a conclusão programada sem paradas ou cancelamentos.

Cenário 2: Contato único interrompido

Você agenda um contato de 150 minutos na Ground Station Anytown 1 para começar às 14:00 e terminar às 16:30. Às 15:00, você liga para a CancelContact API para interromper seu contato.



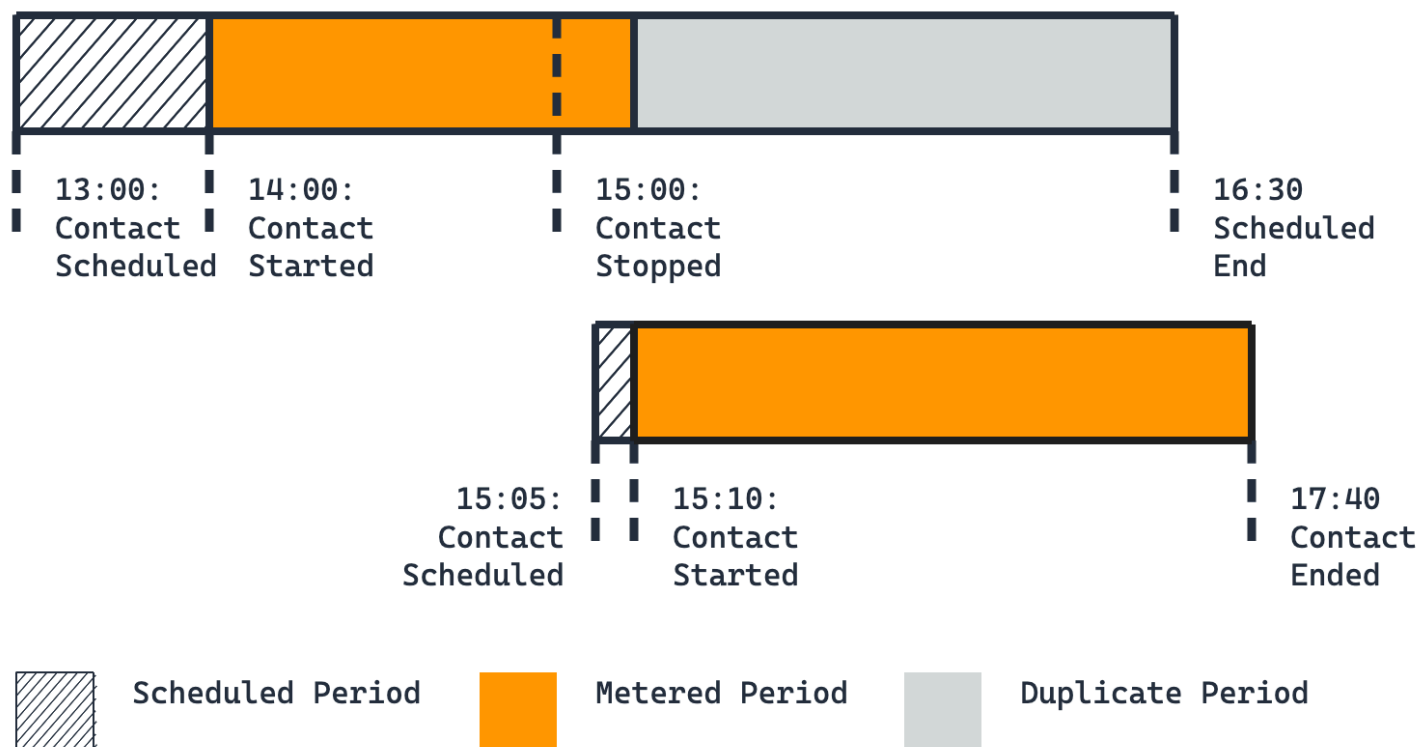
Detalhamento do faturamento:

- Primeiro contato: 150 minutos (duração original completa)

Você é cobrado pelos 150 minutos completos porque interrompeu o contato, mas não agendou nenhum contato duplicado para cobrir o tempo restante (15:00-16:30). Quando você interrompe um contato sem agendar duplicatas, você permanece responsável por toda a duração originalmente programada.

Cenário 3: Duplicação única

Você agenda um contato de 150 minutos na Ground Station Anytown 1 para começar às 14:00 e terminar às 16:30. Às 15:00, você chama a CancelContact API para interromper seu primeiro contato. Depois de ligar CancelContact, você agenda outro contato na mesma Ground Station a partir das 15:10 por 150 minutos.



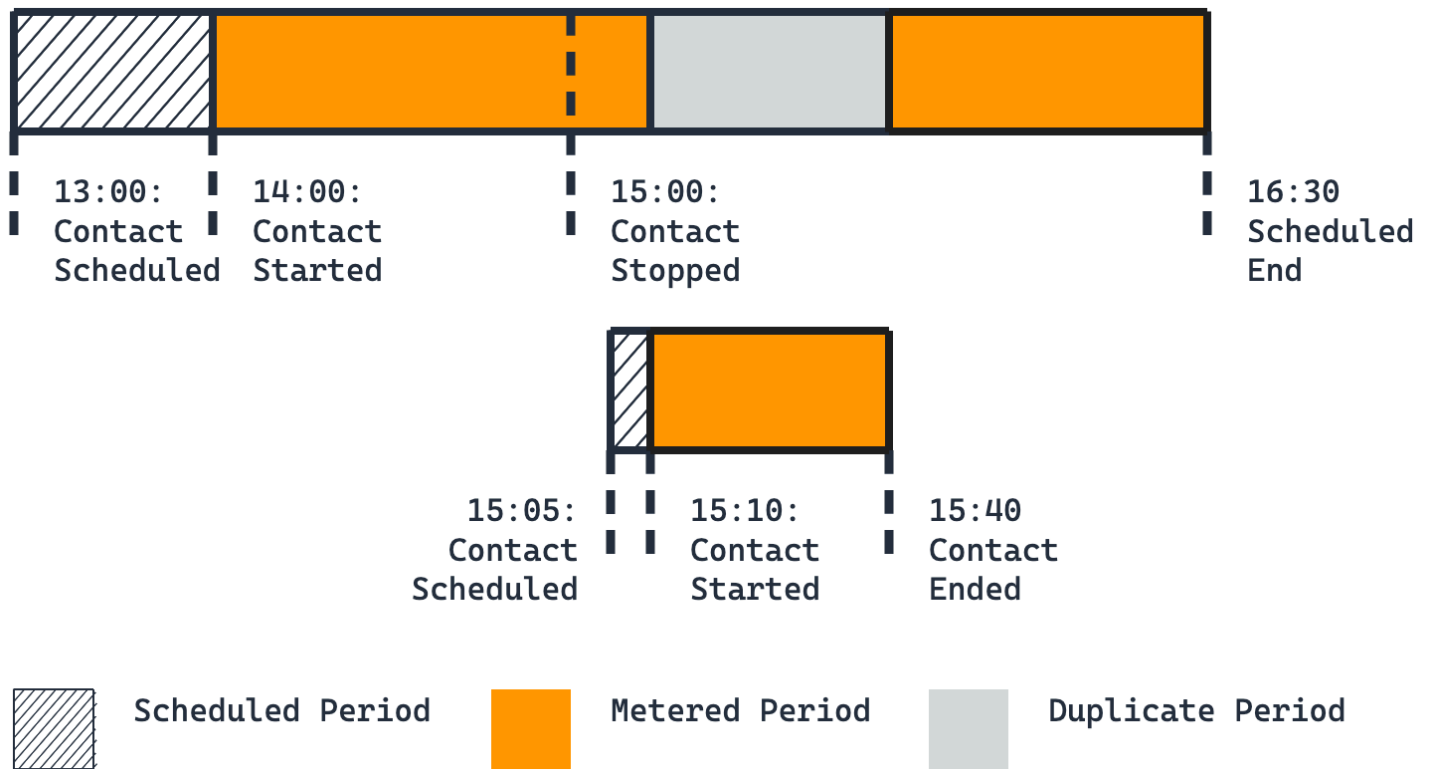
Detalhamento do faturamento:

- Primeiro contato: 70 minutos (60 minutos executados + 10 minutos de inatividade antes do início do segundo contato)
- Segundo contato: 150 minutos (duração total)

O segundo contato é duplicado porque você o agendou depois de interromper o primeiro contato. A duplicata cobre o tempo restante das 15:10 às 16:30, então você só é cobrado pelo horário em que o primeiro contato realmente foi executado, mais o intervalo de 10 minutos entre a parada e a reinicialização.

Cenário 4: Duplicação curta

Você agenda um contato de 150 minutos na Ground Station Anytown 1 para começar às 14:00 e terminar às 16:30. Às 15:00, você chama a `CancelContact` API para interromper seu primeiro contato. Depois de ligar `CancelContact`, você agenda um contato de 30 minutos na mesma Ground Station a partir das 15:10.



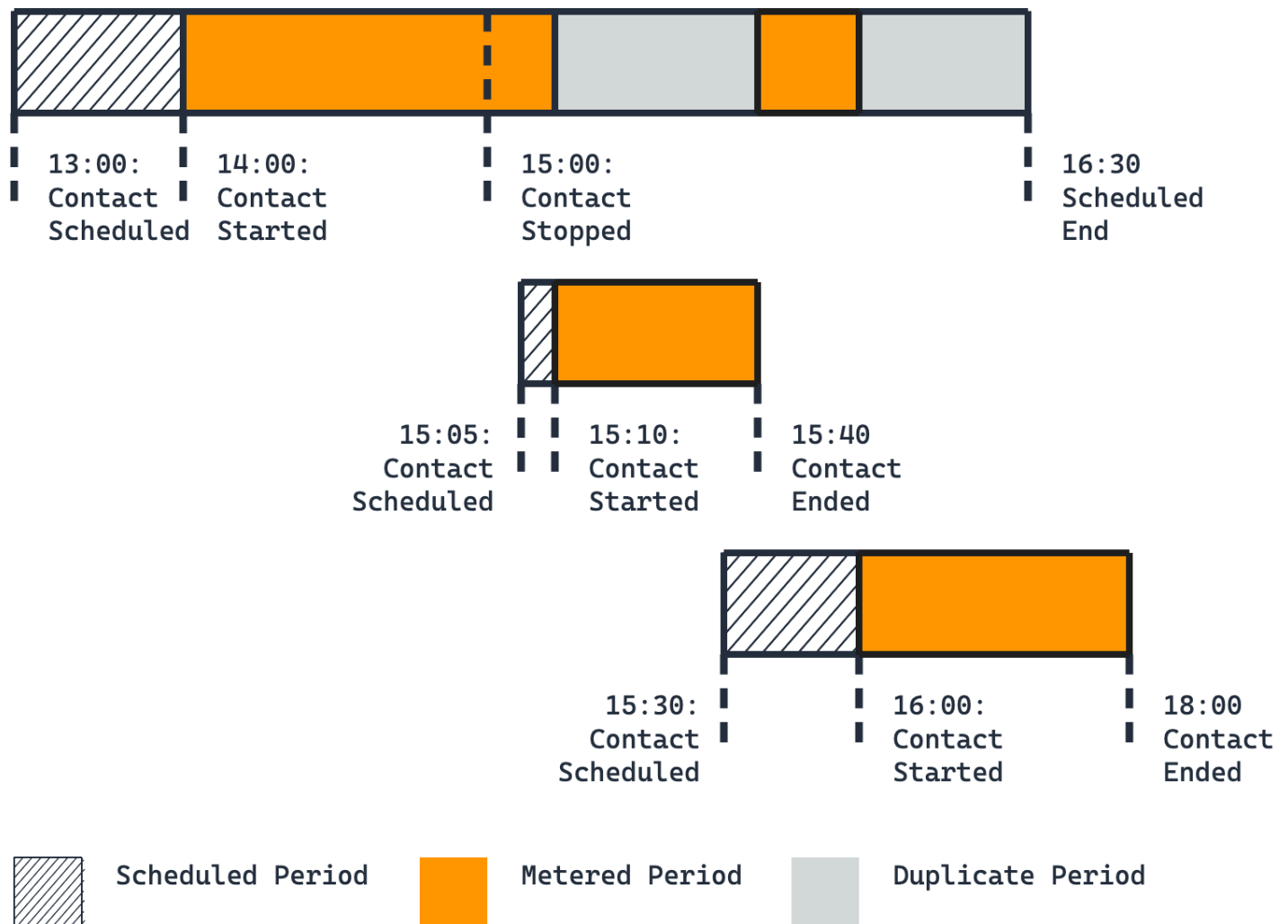
Detalhamento do faturamento:

- Primeiro contato: 120 minutos (60 minutos executados + 10 minutos de tempo de inatividade antes do início do segundo contato + 50 minutos de tempo restante que a duplicata não cobriu)
- Segundo contato: 30 minutos (duração total)

O contato duplicado cobre apenas 30 minutos (15:10-15:40) dos 90 minutos restantes depois que você interrompeu o primeiro contato. Você será cobrado pelo intervalo de 10 minutos antes do início da duplicação e pelos 50 minutos do tempo descoberto após o término da duplicação (15:40-16:30).

Cenário 5: Várias duplicatas

Você agenda um contato de 150 minutos na Ground Station Anytown 1 para começar às 14:00 e terminar às 16:30. Às 15:00, você chama a CancelContact API para interromper seu primeiro contato. Depois de ligar CancelContact, você agenda um contato de 30 minutos na mesma Ground Station a partir das 15:10. Mais tarde, às 15:30, você agenda outro contato a partir das 16:00 por 120 minutos.



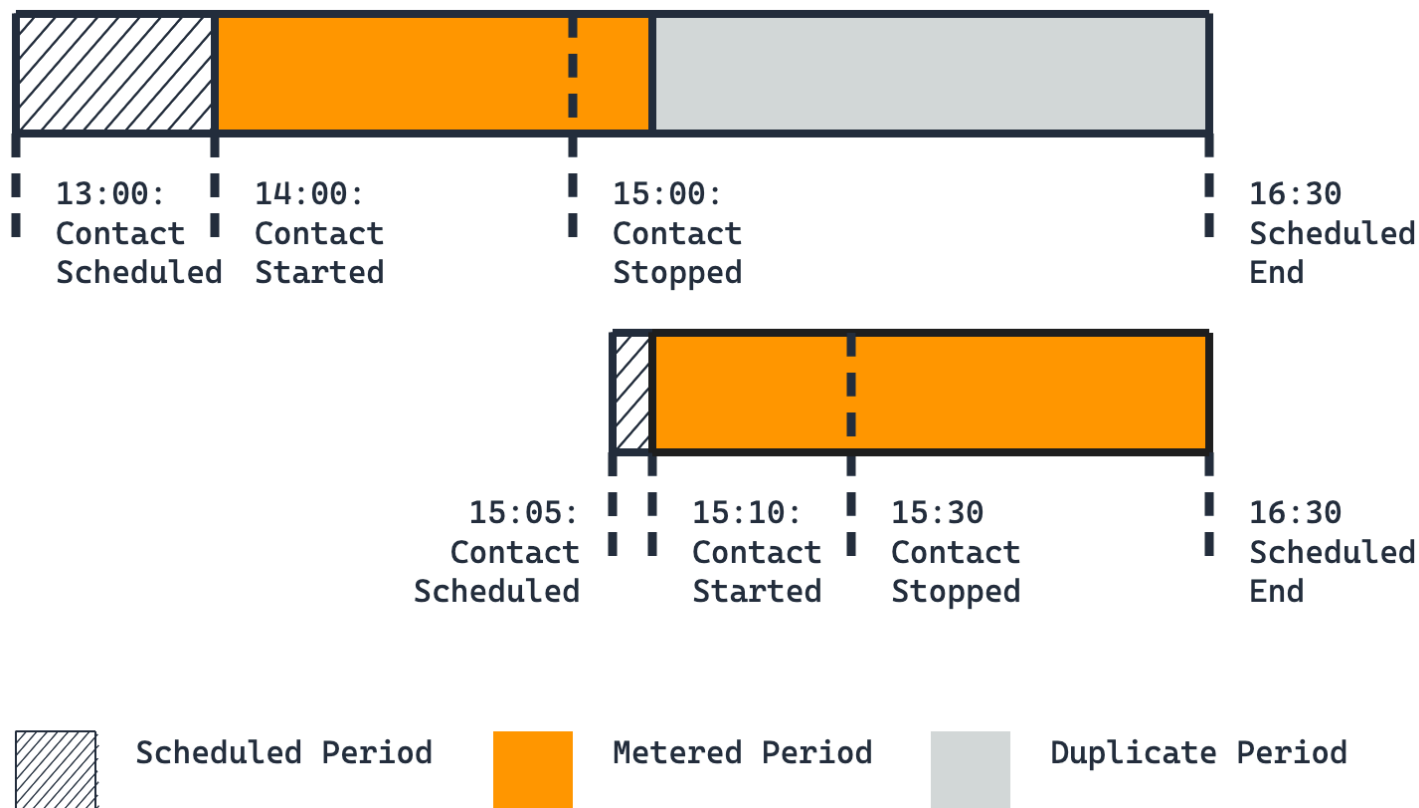
Detalhamento do faturamento:

- Primeiro contato: 90 minutos (60 minutos executados + 10 minutos de tempo de inatividade antes do início do segundo contato + 20 minutos de inatividade entre o segundo e o terceiro contato)
- Segundo contato: 30 minutos (duração total)
- Terceiro contato: 120 minutos (duração total)

Tanto o segundo quanto o terceiro contatos contam como duplicatas porque você os agendou depois de interromper o primeiro contato. No entanto, você ainda é cobrado pelas lacunas entre os contatos: 10 minutos entre a primeira parada (15:00) e a segunda partida (15:10) e 20 minutos entre a segunda extremidade (15:40) e a terceira partida (16:00).

Cenário 6: Várias paradas

Você agenda um contato de 150 minutos na Ground Station Anytown 1 para começar às 14:00 e terminar às 16:30. Às 15:00, você chama a CancelContact API para interromper seu primeiro contato. Depois de ligar CancelContact, você agenda um contato de 80 minutos na Ground Station Anytown 1, que começa às 15:10 e termina às 16:30. Às 15:30, você chama a CancelContact API novamente, interrompendo seu contato duplicado.



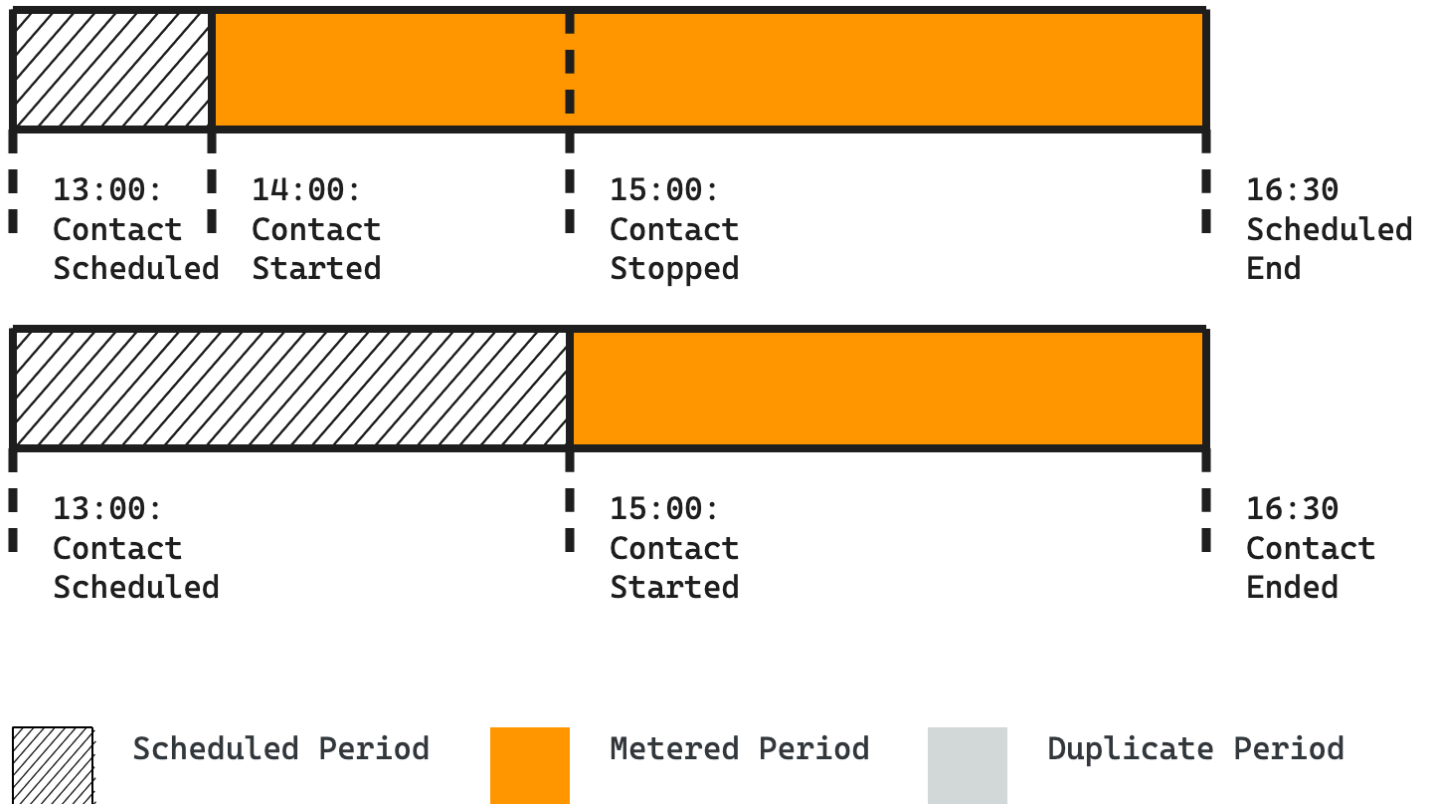
Detalhamento do faturamento:

- Primeiro contato: 70 minutos (60 minutos executados + 10 minutos de inatividade antes do início do segundo contato)
- Segundo contato: 80 minutos (duração original completa)

O segundo contato é cobrado por sua duração total de 80 minutos porque você o interrompeu às 15h30, deixando 60 minutos do horário originalmente programado (15h30 às 16h30) não preenchidos. A menos que você agende outro contato duplicado para cobrir o tempo restante, você é responsável por toda a duração de qualquer contato interrompido.

Cenário 7: Estação terrestre com várias antenas sem duplicata

Às 13:00, você agenda dois contatos na Ground Station Anytown 1. O primeiro é um contato de 150 minutos começando às 14:00 e terminando às 16:30. O segundo é um contato de 90 minutos começando às 15:00 e terminando às 16:30. Às 15:00, você chama a CancelContact API para interromper seu primeiro contato. A Ground Station Anytown 1 é uma estação terrestre com várias antenas, que permite que os dois contatos funcionem simultaneamente.



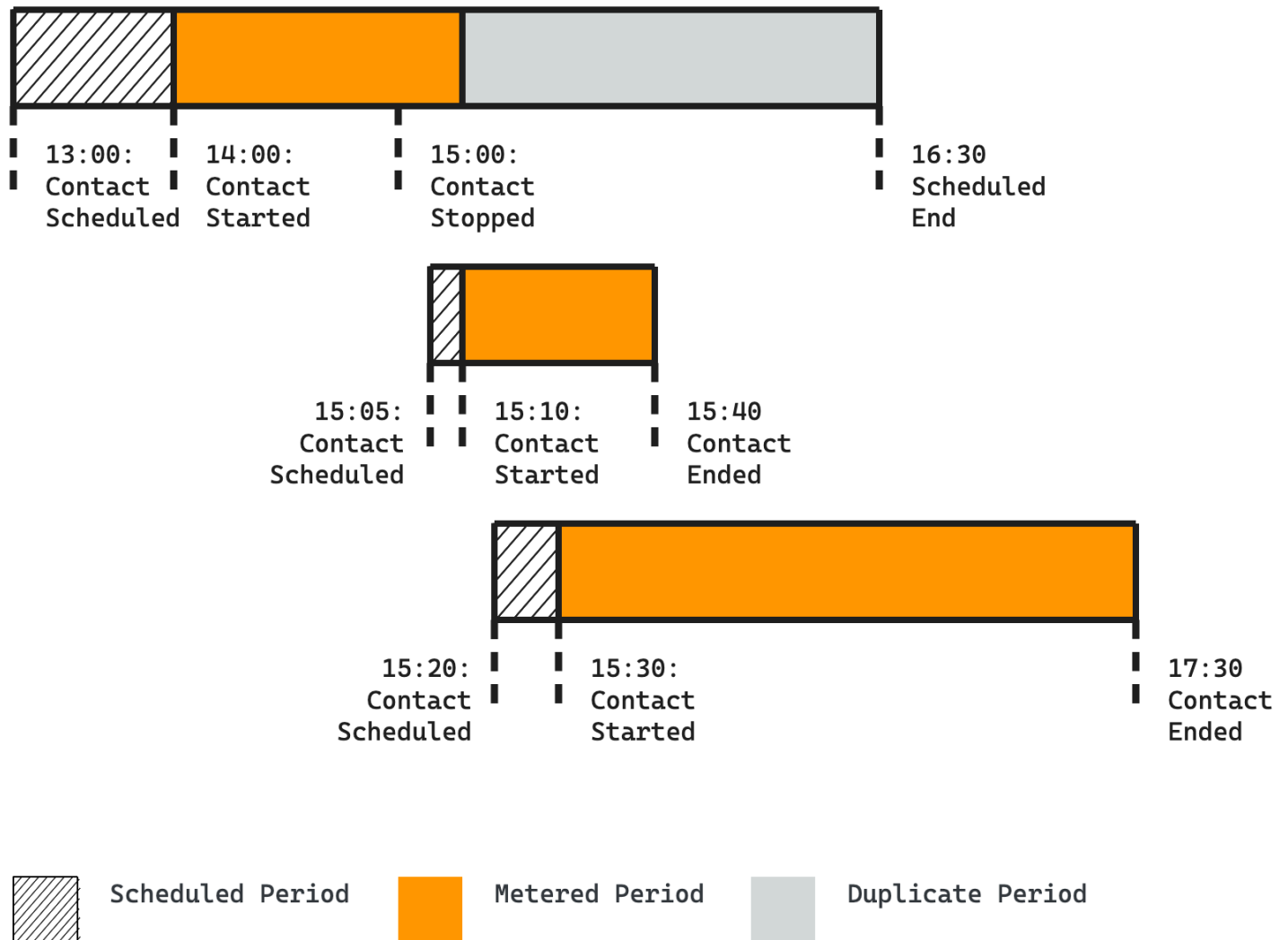
Detalhamento do faturamento:

- Primeiro contato: 150 minutos (duração original completa)
- Segundo contato: 90 minutos (duração total)

Embora o segundo contato se sobreponha à parte interrompida do primeiro contato, ele não conta como uma duplicata. O segundo contato não atende ao primeiro critério para duplicatas: foi agendado às 13:00, antes de você interromper o primeiro contato às 15:00. Como não é uma cópia, você é cobrado pela duração total original do primeiro contato, independentemente de quando você o interrompeu.

Cenário 8: Estação terrestre com várias antenas com contatos duplicados

Você agenda um contato de 150 minutos na Ground Station Anytown 1 para começar às 14:00 e terminar às 16:30. Às 15:00, você chama a CancelContact API para interromper seu primeiro contato. Depois de ligar CancelContact, você agenda um contato de 30 minutos na Ground Station Anytown 1, começando às 15:10 e terminando às 15:40. Posteriormente, você agenda outro contato de 90 minutos na Ground Station Anytown 1, começando às 15:30 e terminando às 17:00. A Ground Station Anytown 1 é uma estação terrestre com várias antenas, que permite que os dois contatos duplicados funcionem simultaneamente com tempos sobrepostos.



Detalhamento do faturamento:

- Primeiro contato: 70 minutos (60 minutos executados + 10 minutos de inatividade antes do início do segundo contato)

- Segundo contato: 30 minutos (duração total)
- Terceiro contato: 90 minutos (duração total)

Tanto o segundo quanto o terceiro contatos contam como duplicatas porque você os agendou depois de interromper o primeiro contato. O intervalo de 10 minutos entre a interrupção do primeiro contato (15:00) e o início do segundo contato (15:10) representa o tempo de inatividade cobrado do contato original.

Atualizar contatos e controle de versão de contatos

AWS Ground Station suporta a atualização de contatos com um SCHEDULED [status de PASS contato](#) ou. PREPASS Você pode usar a [UpdateContact](#) API para especificar uma substituição de efemérides para um contato, incluindo dados de rastreamento de azimute/elevação, OEM ou TLE, sem cancelá-lo e reagendá-lo. Isso é útil para operações de satélite geossíncronas (GEO), nas quais você precisa reatribuir uma antena a um satélite diferente durante um contato, ou para operações iniciais e de lançamento (LEOPs), onde ajustes de apontamento são necessários.

Sempre que você reserva ou atualiza um contato, AWS Ground Station cria uma nova versão do contato. As versões de contato fornecem um histórico das alterações feitas em um contato e permitem que você acompanhe o status de cada atualização.

Como funciona o controle de versão de contatos

Quando você liga [ReserveContact](#), AWS Ground Station cria a primeira versão do contato (versão 1) e retorna a `versionId` na resposta. Cada chamada subsequente [UpdateContact](#) cria uma nova versão com um número de versão incrementado.


A [DescribeContact](#) API retorna a [versão atual do ACTIVE contato](#), incluindo informações da versão no `version` campo da resposta. A [ListContacts](#) API também inclui informações sobre a versão de cada contato.

Para ver uma versão específica de um contato, use a [DescribeContactVersion](#) API. Para listar todas as versões de um contato, use a [ListContactVersions](#) API.

Atualizando um contato


Você pode ligar [UpdateContact](#) quando um contato estiver no estado SCHEDULED, PREPASS ou PASS. A API aceita os seguintes parâmetros:

- `contactiD` — O identificador do contato a ser atualizado.
- `ClientToken` — Um token de idempotência que garante que a solicitação seja processada somente uma vez. Se você tentar novamente uma solicitação com o mesmo token de cliente, AWS Ground Station retornará a resposta original sem realizar a atualização novamente. Muitos geram AWS SDKs automaticamente um token de cliente para você, se um não for fornecido.
- `TrackingOverrides` — A nova configuração de rastreamento para o contato. Isso inclui as configurações da trilha do programa (azimute/elevação, TLE ou efemérides de OEM).
- `SatelliteArn` — O ARN do satélite para o contato. Ao alterar o satélite alvo junto com as configurações de rastreamento do programa, forneça o ARN do novo satélite. Somente clientes aprovados para ângulos de apontamento de azimute/elevação podem definir esse valor como nulo. Todos os outros clientes devem incluir o ARN do satélite do contato.

 Important

A `UpdateContact` API aplica todos os parâmetros na solicitação. Qualquer parâmetro omitido ou definido explicitamente como nulo é tratado como uma solicitação para limpar esse valor, não para deixá-lo inalterado. Por exemplo, se você fornecer, `trackingOverrides` mas omitir `satelliteArn`, o ARN do satélite será apagado. Certifique-se de incluir todos os valores desejados em cada solicitação de atualização.

Você pode alterar o satélite alvo durante um contato fornecendo um novo `satelliteArn` junto com o `correspondentetrackingOverrides`. O novo satélite deve estar visível da estação terrestre durante o contato, porque a hora de início e término do contato não muda com essa API. O novo satélite também deve estar embarcado na estação terrestre e ter o licenciamento exigido pelo perfil da missão. O perfil de missão do contato não pode ser alterado, portanto, a troca de satélites só é aplicável quando os dois satélites usam o mesmo perfil de missão.

 Important

A `UpdateContact` API não suporta a alteração da hora de início, hora de término ou perfil da missão de um contato. Para alterar esses valores, cancele o contato e reserve um novo. A `UpdateContact` API foi projetada para refazer a configuração de apontamento da antena, como alternar entre satélites ou atualizar dados de efemérides.

⚠ Important

A UpdateContact API não oferece suporte a contatos que tenham um perfil de missão que usa [Configuração de decodificação de demodulação de downlink de antena](#) configurações. Para alterar a configuração desses contatos, cancele o contato e reserve um novo.

A UpdateContact API retorna o `contactId` e o `novoversionId`. A atualização é processada de forma assíncrona. Use [DescribeContactVersion](#) para verificar o status da atualização. Alguns AWS SDKs e outros AWS Command Line Interface fornecem um `ContactUpdated` garçom que pesquisa até que a versão atinja o status `ACTIVE` ou `FAILED_TO_UPDATE`.

ℹ Note

Somente uma atualização pode estar em andamento por vez. Se a versão mais recente do contato estiver no estado `ATUALIZANDO`, a API retornará `ConflictException`. Aguarde até que a atualização atual alcance o status `ACTIVE` ou `FAILED_TO_UPDATE` antes de enviar outra atualização.

Status da versão de contato

Cada versão de contato tem um dos seguintes status:

Status	Description
ATUALIZANDO	A versão está sendo aplicada ao contato. A atualização foi enviada e está sendo processada pelo AWS Ground Station.
ATIVO	A versão é a configuração atualmente ativa para o contato. A estação terrestre está usando as configurações desta versão.
SUPERSEDED	A versão estava ativa anteriormente, mas foi substituída por uma versão mais recente.
FALHA_NA ATUALIZAÇÃO	A atualização não pôde ser aplicada. O contato é revertido para a versão anteriormente ativa. Verifique os <code>failureMessage</code> campos <code>failureCodes</code> e para obter detalhes.

Exemplos de código

Os exemplos a seguir demonstram como usar o controle de versão de contatos APIs com o AWS SDK para Python (Boto3).

Exemplo: atualizar um contato

O exemplo a seguir atualiza um contato com novas substituições de rastreamento e aguarda a conclusão da atualização usando o garçom do Boto3. `ContactUpdated`

```
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}")
```

Exemplo: descrever uma versão de contato

O exemplo a seguir recupera os detalhes de uma versão específica do contato, incluindo seu status, configuração e qualquer informação de falha.

```
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}")

```

Exemplo: Listar versões de contatos

O exemplo a seguir lista todas as versões de um contato para ver o histórico completo das alterações, usando a paginação para lidar com grandes conjuntos de resultados.

```

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.")
```

Considerações

- Os contatos criados antes da introdução do recurso de controle de versão de contatos não têm informações de versão. Ligar [DescribeContactVersion](#) ou [ListContactVersions](#) receber esses contatos retorna um `ResourceNotFoundException`.
- Quando você atualiza as substituições de rastreamento de um contato em andamento, há um breve período de transição enquanto a antena se ajusta à nova configuração de apontamento. Durante esse período, a recepção ou transmissão do sinal pode ser interrompida.
- As versões de contato não podem ser excluídas. Use [ListContactVersions](#) para ver o histórico completo das alterações feitas em um contato.
- A `UpdateContact` API só pode ser chamada na região de agendamento do contato.

Use o recurso de gêmeos AWS Ground Station digitais

O recurso digital twin AWS Ground Station fornece um ambiente onde você pode testar e integrar seu software de gerenciamento de missões de satélite e comando e controle. O recurso digital twin permite testar o agendamento, a verificação de configurações e o tratamento adequado de erros sem usar a capacidade da antena de produção. Testar sua AWS Ground Station integração com o recurso digital twin permite que você tenha maior confiança na capacidade do seu sistema de gerenciar suas operações de satélite sem problemas. Também permite testar AWS Ground Station APIs sem usar a capacidade de produção ou exigir licenciamento de espectro.

Para começar, siga [Satélite a bordo](#), solicitando a integração com o recurso de gêmeos digitais. Depois que seu satélite estiver integrado ao recurso de gêmeos digitais, você poderá agendar contatos com estações terrestres duplas digitais. A lista de estações terrestres às quais você tem acesso pode ser recuperada por meio da resposta do SDK [ListGroundStations](#) da AWS. As estações terrestres gêmeas digitais são cópias exatas das estações terrestres listadas [AWS Ground Station Localizações](#) com um prefixo modificador para o nome da estação terrestre de “Digital Twin”. Isso inclui seus recursos de antena e metadados, incluindo, mas não se limitando a, máscara do local e coordenadas GPS reais. No momento, o recurso de gêmeos digitais não oferece suporte à entrega de dados conforme descrito em [Trabalhe com fluxos de dados](#).

Uma vez integrado, o recurso digital twin emite os mesmos EventBridge eventos da Amazon e respostas de API que o serviço de produção, conforme descrito em [Automatize AWS Ground Station com eventos](#). Esses eventos permitirão que você ajuste suas configurações e grupos de endpoints de fluxo de dados.

AWS Ground Station Antenas dedicadas

AWS Ground Station Antenas dedicadas são sistemas de antenas personalizados que AWS gerencia em seu nome. Ao contrário das AWS Ground Station antenas públicas em que você compartilha o tempo da antena com outros clientes, uma antena dedicada fornece acesso dedicado a uma antena construída de acordo com suas especificações. Você pode ter uma ou mais antenas dedicadas. Uma antena dedicada está conectada à AWS rede global e você interage com ela usando os mesmos AWS Ground Station APIs fluxos de trabalho que usa com antenas públicas, com a adição de maior visibilidade na utilização da antena.

O que é uma antena dedicada

Uma antena dedicada é um sistema de antena física desenvolvido sob medida para sua organização e totalmente gerenciado pela AWS. Você pode ter uma ou mais antenas dedicadas. Uma antena dedicada difere das AWS Ground Station antenas públicas das seguintes maneiras:

- **Personalizado** — Uma antena dedicada é construída de acordo com suas especificações. Os recursos de uma antena dedicada não se limitam aos recursos das antenas públicas, conforme descrito em [AWS Ground Station Capacidades do site](#)
- **Localização flexível** — Antenas dedicadas não estão restritas às localizações de AWS Ground Station antenas existentes. Uma antena dedicada pode ser conectada a qualquer AWS região existente, incluindo regiões onde não AWS Ground Station está disponível no momento. Você pode trabalhar com ele AWS para determinar o local e a região que atendem às suas necessidades.
- **Acesso dedicado** — Você tem acesso dedicado à antena em vez de compartilhar o tempo da antena com outros AWS Ground Station clientes por contato.

Todas as AWS Ground Station antenas, incluindo antenas dedicadas, são totalmente gerenciadas pelo AWS. Isso inclui manutenção e conectividade com a AWS rede global. Você interage com uma antena dedicada usando os mesmos fluxos AWS Ground Station APIs de trabalho que usa com antenas públicas. Você agenda contatos, configura perfis de missão e entrega dados da mesma forma.

Uma antena dedicada pode ser compartilhada por várias AWS contas, onde o cliente que detém o contrato de antena dedicada escolhe quais contas integrar. Cada conta integrada pode agendar

contatos na antena de forma independente e tem visibilidade das reservas em todas as contas que compartilham a antena.

Para saber mais sobre antenas dedicadas ou para começar, entre em contato AWS Support através do. [AWS Support Center Console](#)

Visibilidade aprimorada da reserva

Ao usar a [ListGroundStationReservations](#) API em sua antena dedicada, você vê informações adicionais que não estão disponíveis nas antenas públicas. A tabela a seguir resume as diferenças comportamentais da [ListGroundStationReservations](#) API para antenas dedicadas em comparação com antenas públicas. AWS Ground Station

Comportamento	Antena dedicada	Antena pública
Seus próprios contatos	Visível com todos os detalhes, incluindo <code>contactId</code>	Visível com todos os detalhes, incluindo <code>contactId</code>
Contatos de outras contas	Visível somente com horários, sem <code>contactId</code>	Não visível
Janelas de manutenção	Visível com <code>PLANNED</code> ou <code>UNPLANNED</code> <code>maintenanceType</code>	Não visível

As janelas de manutenção representam os períodos em que a antena não está disponível para comunicação via satélite. O `maintenanceType` campo indica se a manutenção foi `PLANNED` ou `UNPLANNED`. Quando a manutenção não planejada é programada, os contatos que se sobrepõem à janela de manutenção podem ser cancelados por. AWS Ground Station

Quando você visualiza contatos de outras AWS contas que compartilham sua antena dedicada, a reserva inclui o horário e as informações da antena, mas não `contactId` estão incluídas.

⚠ Important

A visibilidade aprimorada da reserva se aplica somente à sua antena dedicada. Ao usar AWS Ground Station antenas públicas, você tem a mesma visibilidade de qualquer outro cliente. Você não vê janelas de manutenção ou reservas de outras contas nas antenas públicas.

Para obter mais informações sobre como anunciar reservas, consulte [Exibir reservas de estações terrestres](#).

Recursos relacionados

- [Exibir reservas de estações terrestres](#)
- [AWS Ground Station Localizações](#)
- [AWS Ground Station Capacidades do site](#)
- [ListGroundStationReservations](#) na Referência de API do AWS Ground Station

Entenda o monitoramento com AWS Ground Station

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e o desempenho do AWS Ground Station. A AWS fornece as seguintes ferramentas de monitoramento para observar AWS Ground Station, relatar quando algo está errado e realizar ações automáticas quando apropriado.

- A Amazon EventBridge Events fornece um fluxo quase em tempo real de eventos do sistema que descrevem mudanças nos AWS recursos. EventBridge Os eventos permitem a computação automatizada baseada em eventos, pois você pode criar regras que observam determinados eventos e acionam ações automatizadas em outros AWS serviços quando esses eventos acontecem. Para obter mais informações sobre EventBridge eventos, consulte o [Guia do usuário do Amazon EventBridge Events](#).
- AWS CloudTrail captura chamadas de API e eventos relacionados feitos por ou em nome de sua AWS conta e entrega os arquivos de log para um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações sobre AWS CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).
- O Amazon CloudWatch Metrics captura métricas para seus contatos agendados durante o uso AWS Ground Station. CloudWatch As métricas permitem que você analise dados com base em seu canal, polarização e ID de satélite para identificar a intensidade do sinal e os erros em seus contatos. Para obter mais informações, consulte [Usando CloudWatch métricas da Amazon](#).
- A [AWS Notificações de Usuários](#) pode ser usada para configurar canais de entrega para receber notificações sobre AWS Ground Station eventos. Você recebe uma notificação quando um evento corresponde a uma regra especificada. É possível receber notificações para eventos por meio de diversos canais, incluindo o e-mail, o [Amazon Q Developer em aplicações de chat](#), notificações por chat ou notificações push do [AWS Console Mobile Application](#). Você também pode ver as notificações na [Central de Notificações](#) do AWS Console. Notificações de Usuários agregação de suporte, que pode reduzir o número de notificações que você recebe durante eventos específicos.

Use os seguintes tópicos para monitorar o AWS Ground Station.

Tópicos

- [Automatize AWS Ground Station com eventos](#)
- [Registre chamadas de AWS Ground Station API com AWS CloudTrail](#)

- [Veja métricas com a Amazon CloudWatch](#)

Automatize AWS Ground Station com eventos

Note

Este documento usa o termo “evento” por toda parte. CloudWatch Eventos e EventBridge são o mesmo serviço e API subjacentes. Regras para corresponder a eventos de entrada e roteá-los para destinos para processamento podem ser construídas usando qualquer um dos serviços.

Os eventos permitem que você automatize seus AWS serviços e responda automaticamente aos eventos do sistema, como problemas de disponibilidade de aplicativos ou alterações de recursos. Os eventos dos AWS serviços são entregues quase em tempo real. Você pode escrever regras simples para indicar quais eventos são do seu interesse, e as ações automatizadas a serem tomadas quando um evento corresponder à regra. Algumas das ações que podem ser acionadas automaticamente incluem o seguinte:

- Invocando uma função AWS Lambda
- Invocar o comando de execução do Amazon EC2
- Transmitir o evento Amazon Kinesis Data Streams
- Ativando uma máquina de AWS Step Functions estado
- Notificar um tópico do Amazon SNS ou uma fila do Amazon SQS

Alguns exemplos de uso de eventos com AWS Ground Station incluem:

- Invocar uma função do Lambda para automatizar o início e a interrupção das instâncias do Amazon EC2 com base no estado do evento.
- Publicar um tópico do Amazon SNS sempre que ocorre uma mudança de estado em um contato. Esses tópicos podem ser configurados para enviar avisos por e-mail no início ou no final dos contatos.

Para obter mais informações, consulte o [Guia do usuário do Amazon EventBridge Events](#).

AWS Ground Station Tipos de eventos

Note

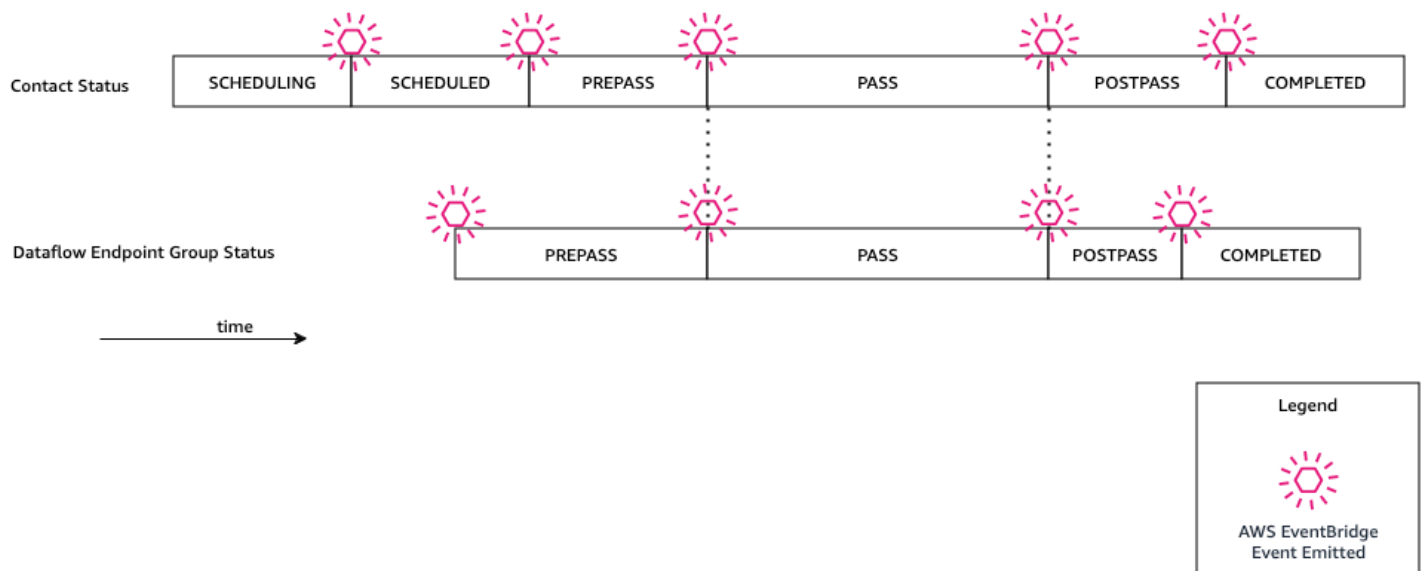
Todos os eventos gerados pelo AWS Ground Station têm “aws.groundstation” como valor para “fonte”.

AWS Ground Station emite eventos relacionados a mudanças de estado para apoiar sua capacidade de personalizar sua automação. Atualmente, AWS Ground Station oferece suporte a eventos de mudança de estado de contato, eventos de alteração de grupos de endpoints de fluxo de dados e eventos de mudança de estado de efemérides. As seções a seguir fornecem informações detalhadas sobre cada tipo.

Cronograma do evento de contato

AWS Ground Station emite eventos quando seu contato muda de estado. Para obter mais informações sobre quais são essas mudanças de estado e o que os próprios estados significam, consulte [Entenda o ciclo de vida do contato](#). Todos os grupos de endpoints de fluxo de dados usados em seu contato têm um conjunto independente de eventos que também são emitidos. Durante esse mesmo período, também emitimos eventos para seu grupo de endpoints de fluxo de dados. O horário exato dos eventos de pré-passagem e pós-passagem pode ser configurado por você à medida que configura o perfil da missão e o grupo de endpoints do fluxo de dados.

O diagrama a seguir mostra os status e os eventos emitidos por um contato nominal e seu grupo de endpoints de fluxo de dados associado.



Alteração de estado do contato do Ground Station

Se você quiser realizar uma ação específica quando um contato futuro estiver mudando de estado, você pode configurar uma regra para automatizar essa ação. Isso é útil quando quiser receber notificações sobre as alterações de estado do contato. Se você quiser mudar quando receber esses eventos, você pode modificar o perfil da sua missão [contactPrePassDurationSeconds](#) [contactPostPassDurationSeconds](#). Os eventos são enviados para a região na qual o contato foi agendado.

Um exemplo de evento é fornecido abaixo.

```
{
  "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"
  }
}

```

Os possíveis valores para `contactStatus` são definidos em [the section called “AWS Ground Station status de contato”](#).

Alteração de estado do grupo de endpoints do fluxo de dados do Ground Station

Se você quiser executar uma ação quando seu grupo de endpoints de fluxo de dados está sendo usado para receber dados, pode configurar uma regra para automatizar essa ação. Isso permitirá executar ações diferentes em resposta à alteração de estados do status do grupo de endpoints do fluxo de dados. Se você quiser alterar a data de recebimento desses eventos, use um grupo de endpoints de fluxo de dados com diferentes e. [contactPrePassDurationSecondscontactPostPassDurationSeconds](#) Esse evento será enviado para a região do grupo de endpoints do fluxo de dados.

Um exemplo é fornecido abaixo.

```

{
  "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"
  }
}

```

Os possíveis estados do `dataflowEndpointGroupState` incluem PREPASS, PASS, POSTPASS e COMPLETED.

Eventos de efemérides

Mudança de estado da efeméride Ground Station

Se você quiser executar uma ação específica quando uma efeméride estiver mudando de estado, é possível configurar uma regra para automatizar essa ação. Isso permite que você execute ações diferentes em resposta à mudança de estado de uma efeméride. Por exemplo, você pode realizar uma ação quando uma efeméride tiver concluído a validação, e agora está ENABLED. A notificação desse evento será enviada para a região onde a efeméride foi enviada.

Um exemplo é fornecido abaixo.

```

{
  "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-
bccccca005000"
  ],
  "detail": {
    "ephemerisStatus": "ENABLED",
    "ephemerisId": "111111-cccc-bbbb-a555-bccccca005000",
    "satelliteId": "10313191-c9d9-4ecb-a5f2-bc55cab050ec"
  }
}

```

```
}  
}
```

Os possíveis estados do `ephemerisStatus` incluem `ENABLED`, `VALIDATING`, `INVALID ERROR`, `DISABLED` e `EXPIRED`

Registre chamadas de AWS Ground Station API com AWS CloudTrail

AWS Ground Station é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em AWS Ground Station. CloudTrail captura todas as chamadas de API AWS Ground Station como eventos. As chamadas capturadas incluem chamadas do AWS Ground Station console e chamadas de código para as operações AWS Ground Station da API. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para AWS Ground Station. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita AWS Ground Station, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

AWS Ground Station Informações em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre em AWS Ground Station, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos para AWS Ground Station, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para saber mais, consulte:

- [Visão Geral para Criar uma Trilha](#)
- [CloudTrail Serviços e integrações compatíveis](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [recebendo arquivos de CloudTrail log de várias contas](#)

Todas AWS Ground Station as ações são registradas CloudTrail e documentadas na [Referência da AWS Ground Station API](#). Por exemplo, chamadas para o `ReserveContact`, `CancelContact` e `ListConfigs` as ações geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário root ou AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte [Elemento `userIdentity` do CloudTrail](#).

Entendendo as entradas do arquivo de AWS Ground Station log

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ReserveContact` ação.

Exemplo: `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"

```

}

Veja métricas com a Amazon CloudWatch

Durante um contato, captura e envia dados AWS Ground Station automaticamente CloudWatch para análise. Seus dados podem ser visualizados no CloudWatch console da Amazon. Para obter mais informações sobre acesso e CloudWatch métricas, consulte [Usando o Amazon CloudWatch Metrics](#).

O recurso de AWS Ground Station telemetria também pode ser usado para receber métricas quase em tempo real durante os contatos. CloudWatch as métricas não estão disponíveis quase em tempo real e podem ter atrasos na entrega. CloudWatch também agrega métricas em um período de um segundo, potencialmente reduzindo a granularidade dos dados. O recurso de telemetria fornece as métricas individuais e as entrega quase em tempo real diretamente na sua AWS conta. Para obter mais informações, consulte [Trabalhe com telemetria](#).

Important

AWS Ground Station emite CloudWatch métricas para a AWS região associada à localização da estação terrestre do contato, não para a AWS região a partir da qual o contato foi agendado. Para visualizar as métricas de um contato, você deve acessar CloudWatch na região da estação terrestre. Para obter informações sobre qual AWS região está associada a cada localização da estação terrestre, consulte [Encontrando a AWS região para a localização de uma estação terrestre](#). Para receber dados de telemetria na região a partir da qual você agenda seus contatos, você pode usar o recurso de AWS Ground Station telemetria. Consulte [Trabalhe com telemetria](#) para obter mais detalhes.

AWS Ground Station Métricas e dimensões

Quais métricas estão disponíveis?

As métricas a seguir estão disponíveis em AWS Ground Station.

Note

As métricas específicas emitidas dependem dos AWS Ground Station recursos que estão sendo usados. Dependendo da sua configuração, somente um subconjunto das métricas abaixo pode ser emitido.

Métrica	Dimensões da métrica	Description
AzimuthAngle	SatelliteId	<p>O ângulo de azimute da antena. O norte verdadeiro é 0 graus e o leste é 90 graus.</p> <p>Unidades: graus</p>
BitErrorRate	Canal, polarização, SatelliteId	<p>A taxa de erros irre recuperáveis em bits, em um determina do número de transmissões de bits. Erros de bits são gerados por ruídos, distorções ou interferências</p> <p>Unidades: erros de bits por unidade de tempo</p>
BlockErrorRate	Canal, polarização, SatelliteId	<p>A taxa de erros de blocos em um determina do número de blocos recebidos . Erros de blocos são causados por interferência.</p>

Métrica	Dimensões da métrica	Description
		Unidades: blocos com erros/número total de blocos
CarrierFrequencyRecovery_Cn0	Categoria, Config, SatelliteId	Relação entre portadora e densidade de ruído por unidade de largura de banda. Unidades: Decibel-hertz (dB-Hz)
CarrierFrequencyRecovery_Locked	Categoria, Config, SatelliteId	Defina como 1 quando o loop de recuperação da frequência a portadora do demodulador estiver bloqueado e 0 quando desbloqueado. Unidades: sem unidades

Métrica	Dimensões da métrica	Description
CarrierFrequencyRecovery_OffsetFrequency_Hz	Categoria, Config, SatelliteId	<p>O deslocamento entre o centro estimado do sinal e a frequência central ideal. Isso é causado pelo deslocamento Doppler e pelo deslocamento do oscilador local entre a espaçonave e o sistema de antenas.</p> <p>Unidades: hertz (Hz)</p>
ElevationAngle	SatelliteId	<p>O ângulo de elevação da antena. O horizonte é 0 graus e o zênite é 90 graus.</p> <p>Unidades: graus</p>

Métrica	Dimensões da métrica	Description
Es/N0	Canal, polarização, Satellitelid	<p>A razão entre a energia por símbolo e a densidade espectral da potência sonora.</p> <p>Unidades: decibéis (dB)</p>
ReceivedPower	Polarização, Satellitelid	<p>A intensidade do sinal medida no demodulador/decodificador.</p> <p>Unidades: dBm (decibéis miliwatts)</p>
SymbolTimingRecovery_ErrorVectorMagnitude	Categoria, Config, Satellitelid	<p>A magnitude do vetor de erro entre os símbolos recebidos e os pontos ideais da constelação.</p> <p>Unidades: percentual</p>

Métrica	Dimensões da métrica	Description
SymbolTimingRecovery_Locked	Categoria, Config, SatelliteId	<p>Defina como 1 quando o loop de recuperação da frequência da portadora do demodulador estiver bloqueado e 0 quando desbloqueado</p> <p>Unidades: sem unidades</p>
SymbolTimingRecovery_OffsetSymbolRate	Categoria, Config, SatelliteId	<p>O deslocamento entre a taxa de símbolo estimada e a taxa de símbolo de sinal ideal. Isso é causado pelo deslocamento Doppler e pelo deslocamento do oscilador local entre a espaçonave e o sistema de antenas.</p> <p>Unidades: símbolos/segundo</p>

Para quais dimensões são usadas AWS Ground Station?

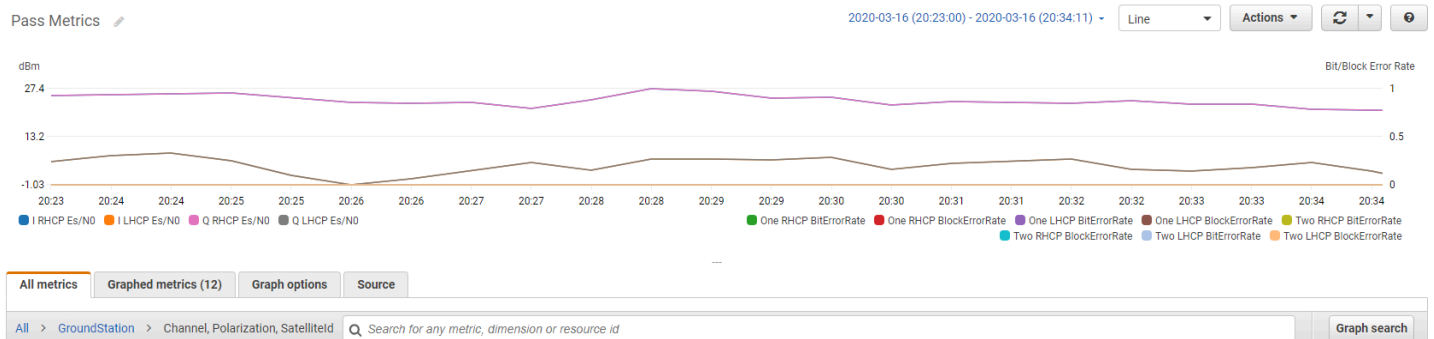
Você pode filtrar AWS Ground Station dados usando as seguintes dimensões.

Dimensão	Description
Category	Demodulação ou decodificação.
Channel	Os canais para cada contato incluem Um, Dois, I (em fase) e Q (quadratura).
Config	Um comando de configuração de decodificação de demod de downlink de antena.
Polarization	A polarização para cada contato inclui PCE (Polarização circular à esquerda) ou PCD (Polarização circular à direita).
SatelliteId	O ID do satélite contém o ARN do satélite para seus contatos.

Visualizar métricas

Ao visualizar as métricas gráficas, é importante observar que a janela de agregação determina como as métricas serão exibidas. As métricas em um contato podem ser exibidas como dados por segundo por um período de três horas após os dados serem recebidos. Seus dados serão agregados pelo CloudWatch Metrics como dados por minuto após o término desse período de 3 horas. Se você precisar visualizar suas métricas em uma medição de dados por segundo, é recomendável visualizá-los dentro do período de 3 horas após o recebimento dos dados ou persisti-los fora das CloudWatch Métricas. Para obter mais informações sobre CloudWatch retenção, consulte [CloudWatch Conceitos da Amazon - Retenção métrica](#).

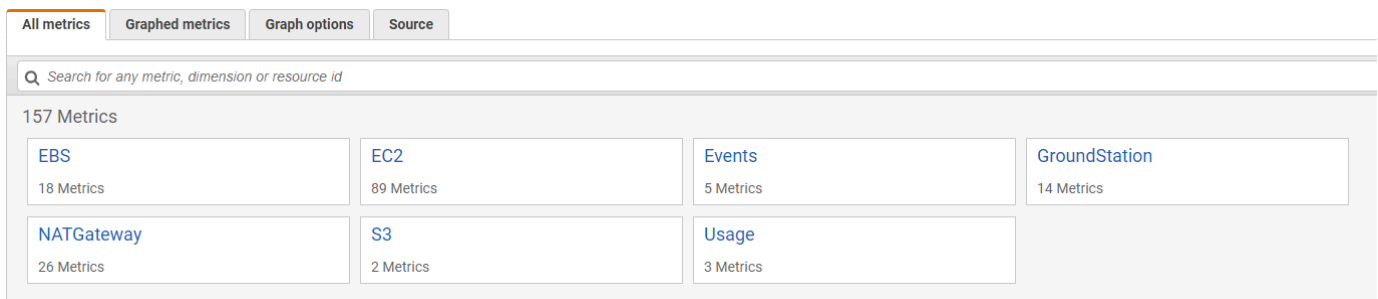
Além disso, os dados capturados dentro dos primeiros 60 segundos não conterão informações suficientes para gerar métricas significativas e provavelmente não serão exibidos. Para visualizar métricas significativas, recomenda-se visualizar os dados após 60 segundos.



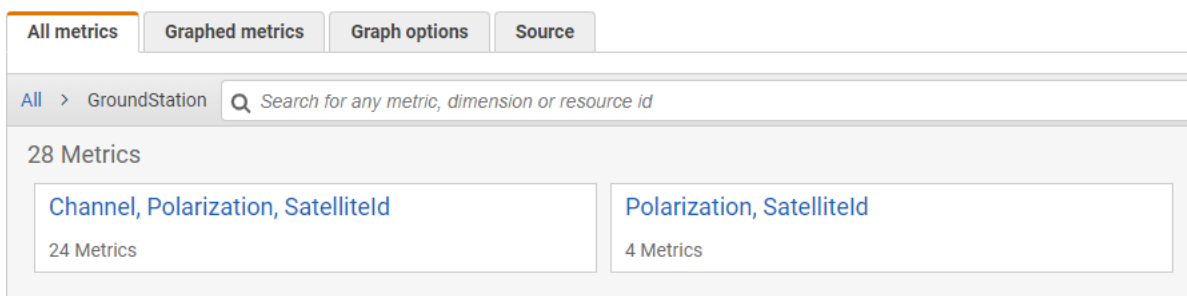
Para obter mais informações sobre a representação gráfica de AWS Ground Station métricas em CloudWatch, consulte [Representação gráfica de](#) métricas.

Para visualizar as métricas usando o console

1. Determine a AWS região associada à localização da sua estação terrestre. AWS Ground Station emite CloudWatch métricas na região associada à localização da estação terrestre do seu contato. Para obter a lista de localizações de estações terrestres e suas AWS regiões associadas, consulte [Encontrando a AWS região para a localização de uma estação terrestre](#).
2. Abra o [console do CloudWatch](#).
3. No painel de navegação, selecione Métricas.
4. Selecione o namespace GroundStation.



5. Selecione as dimensões métricas desejadas (por exemplo, Canal, Polarização Satelliteld).



6. A guia Todas as métricas exibe todas as métricas dessa dimensão no namespace. Você pode fazer o seguinte:
 - a. Para classificar a tabela, use o cabeçalho da coluna.
 - b. Para representar graficamente uma métrica, marque a caixa de seleção associada à métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
 - c. Para filtrar por recurso, escolha o ID do recurso e, em seguida, escolha Adicionar à pesquisa.
 - d. Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Adicionar à pesquisa.

Para visualizar métricas usando AWS CLI

AWS Ground Station emite CloudWatch métricas na região associada à localização da estação terrestre do seu contato. Para obter a lista de localizações de estações terrestres e suas AWS regiões associadas, [Encontrando a AWS região para a localização de uma estação terrestre](#). *ground-station-region-code* Substitua pelo código de AWS região da localização da sua estação terrestre (por exemplo, us-west-2 para Oregon 1, Havaí 1 ou Alasca 1). Todos os AWS CLI comandos subsequentes nesse procedimento devem usar a mesma região.

1. Certifique-se de que AWS CLI esteja instalado. Para obter informações sobre a instalação AWS CLI, consulte [Instalação da AWS CLI versão 2](#).
2. Identifique a AWS região associada à localização da sua estação terrestre.
3. Use o [get-metric-data](#) método da CloudWatch CLI para gerar um arquivo que pode ser modificado para especificar as métricas nas quais você está interessado e, em seguida, ser usado para consultar essas métricas.

Para fazer isso, execute o seguinte: `aws cloudwatch get-metric-data --region ground-station-region-code --generate-cli-skeleton`. Isso gerará uma saída semelhante a:

```
{
  "MetricDataQueries": [
    {
      "Id": "",
```

```

    "MetricStat": {
      "Metric": {
        "Namespace": "",
        "MetricName": "",
        "Dimensions": [
          {
            "Name": "",
            "Value": ""
          }
        ]
      },
      "Period": 0,
      "Stat": "",
      "Unit": "Seconds"
    },
    "Expression": "",
    "Label": "",
    "ReturnData": true,
    "Period": 0,
    "AccountId": ""
  } ],
  "StartTime": "1970-01-01T00:00:00",
  "EndTime": "1970-01-01T00:00:00",
  "NextToken": "",
  "ScanBy": "TimestampDescending",
  "MaxDatapoints": 0,
  "LabelOptions": {
    "Timezone": ""
  }
}

```

4. Liste as CloudWatch métricas disponíveis executando `aws cloudwatch list-metrics --region ground-station-region-code`.

Se você usou recentemente AWS Ground Station, o método deve retornar uma saída que contém entradas como:

```

...
{
  "Namespace": "AWS/GroundStation",
  "MetricName": "ReceivedPower",

```

```

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

```

Note

Se já passaram mais de duas semanas desde a última vez que você usou AWS Ground Station, você precisará inspecionar manualmente a [tabela de métricas disponíveis](#) para encontrar os nomes e dimensões das métricas no namespace da AWS/GroundStation métrica. Para obter mais informações sobre CloudWatch limitações, consulte: [Exibir métricas disponíveis](#)

5. Modifique o arquivo JSON que você criou na etapa 2 para corresponder aos valores necessários da etapa 3, por exemplo `SatelliteId`, e `Polarization` de suas métricas. Além disso, não se esqueça de atualizar os `EndTime` valores de `StartTime`, e para que correspondam ao seu contato. Por exemplo:

```

{
  "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 publica métricas a cada 1 a 60 segundos, dependendo da métrica. As métricas não serão retornadas se o `Period` campo tiver um valor menor que o período de publicação da métrica.

- Execute `aws cloudwatch get-metric-data` com o arquivo de configuração criado nas etapas anteriores. Um exemplo é fornecido abaixo.

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

As métricas serão fornecidas com marca temporal do contato. Um exemplo de saída das métricas do AWS Ground Station é fornecido abaixo.

```
{
```

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

Segurança em AWS Ground Station

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficiará de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança. A AWS oferece ferramentas e recursos específicos para segurança e para ajudar você a atender seus objetivos de segurança. Essas ferramentas e recursos incluem segurança de rede, gerenciamento de configurações, controle de acesso e segurança de dados.

Ao usar AWS Ground Station, recomendamos que você siga as melhores práticas do setor e implemente a end-to-end criptografia. A AWS permite APIs que você integre criptografia e proteção de dados. Para obter mais informações sobre AWS segurança, consulte o whitepaper [Introdução à segurança da AWS](#).

Use os tópicos a seguir para saber como proteger seus recursos do .

Tópicos

- [Identity and Access Management para AWS Ground Station](#)
- [AWS políticas gerenciadas para AWS Ground Station](#)
- [Use funções vinculadas a serviços para Ground Station](#)
- [Criptografia de dados em repouso para AWS Ground Station](#)
- [Criptografia de dados durante o trânsito para AWS Ground Station](#)

Identity and Access Management para AWS Ground Station

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS Ground Station os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticação com identidades](#)
- [Gerenciar o acesso usando políticas](#)

- [Como AWS Ground Station funciona com o IAM](#)
- [Exemplos de políticas baseadas em identidade para AWS Ground Station](#)
- [Solução de problemas AWS Ground Station de identidade e acesso](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte [Solução de problemas AWS Ground Station de identidade e acesso](#)).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte [Como AWS Ground Station funciona com o IAM](#))
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte [Exemplos de políticas baseadas em identidade para AWS Ground Station](#))

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte [Como fazer login em sua Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte [AWS Signature Version 4 para solicitações de API](#) no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte [O que é o IAM Identity Center?](#) no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte [Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte [Casos de uso de usuários do IAM](#) no Guia do usuário do IAM.

Perfis do IAM

Uma [perfil do IAM](#) é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função [mudando de um usuário para uma função do IAM \(console\)](#) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte [Métodos para assumir um perfil](#) no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas

políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte [Políticas de controle de serviço](#) no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

Como AWS Ground Station funciona com o IAM

Antes de usar o IAM para gerenciar o acesso AWS Ground Station, saiba com quais recursos do IAM estão disponíveis para uso AWS Ground Station.

Recursos do IAM que você pode usar com AWS Ground Station

Recurso do IAM	AWS Ground Station apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim

Recurso do IAM	AWS Ground Station apoio
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Não
Perfis vinculados ao serviço	Sim

Para ter uma visão de alto nível de como AWS Ground Station e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte [AWS os serviços que funcionam com o IAM](#) no Guia do usuário do IAM.

Políticas baseadas em identidade para AWS Ground Station

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elemento de política JSON do IAM](#) no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para AWS Ground Station

Para ver exemplos de políticas AWS Ground Station baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para AWS Ground Station](#)

Políticas baseadas em recursos dentro AWS Ground Station

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Ações políticas para AWS Ground Station

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de AWS Ground Station ações, consulte [Ações definidas por AWS Ground Station](#) na Referência de Autorização de Serviço.

As ações de política AWS Ground Station usam o seguinte prefixo antes da ação:

```
groundstation
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

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

```
"groundstation:action2"  
]
```

Para ver exemplos de políticas AWS Ground Station baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS Ground Station](#)

Recursos políticos para AWS Ground Station

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de AWS Ground Station recursos e seus ARNs, consulte [Recursos definidos por AWS Ground Station](#) na Referência de Autorização de Serviço. Para saber com quais ações é possível especificar o ARN de cada atributo, consulte [Ações definidas pelo AWS Ground Station](#).

Para ver exemplos de políticas AWS Ground Station baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS Ground Station](#)

Chaves de condição de política para AWS Ground Station

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem [agentes de condição](#), como "igual a"

ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Para ver uma lista de chaves de AWS Ground Station condição, consulte [Chaves de condição AWS Ground Station](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas por AWS Ground Station](#).

Para ver exemplos de políticas AWS Ground Station baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para AWS Ground Station](#)

ACLs in AWS Ground Station

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com AWS Ground Station

Compatível com ABAC (tags em políticas): sim

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte [Definir permissões com autorização do ABAC](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso por atributo \(ABAC\)](#) no Guia do usuário do IAM.

Usando credenciais temporárias com AWS Ground Station

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte [Credenciais de segurança temporárias no IAM](#) e [Serviços da Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Permissões principais entre serviços para AWS Ground Station

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).

Funções de serviço para AWS Ground Station

Compatível com perfis de serviço: não

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.

Warning

Alterar as permissões de uma função de serviço pode interromper AWS Ground Station a funcionalidade. Edite as funções de serviço somente quando AWS Ground Station fornecer orientação para fazer isso.

Funções vinculadas a serviços para AWS Ground Station

Compatibilidade com perfis vinculados a serviços: sim

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções

vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte [Serviços da AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para AWS Ground Station

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do AWS Ground Station. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte [Criar políticas do IAM \(console\)](#) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos por AWS Ground Station, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição AWS Ground Station na Referência de Autorização de Serviço](#).

Tópicos

- [Práticas recomendadas de política](#)
- [Usando o AWS Ground Station console](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir AWS Ground Station recursos em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões

definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte [Elementos da política JSON do IAM: condição](#) no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usando o AWS Ground Station console

Para acessar o AWS Ground Station console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os AWS Ground Station recursos em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o AWS Ground Station console, anexe também a política AWS Ground Station *ConsoleAccess* ou a política *ReadOnly* AWS gerenciada às entidades. Para obter informações, consulte [Adicionar permissões a um usuário](#) no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
    }
  ]
}
```

Solução de problemas AWS Ground Station de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS Ground Station um IAM.

Tópicos

- [Não estou autorizado a realizar uma ação em AWS Ground Station](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS Ground Station recursos](#)

Não estou autorizado a realizar uma ação em AWS Ground Station

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo `my-example-widget` fictício, mas não tem as permissões `groundstation:GetWidget` fictícias.

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

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso `my-example-widget` usando a ação `groundstation:GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS Ground Station.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no AWS Ground Station. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS Ground Station recursos

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é AWS Ground Station compatível com esses recursos, consulte [Como AWS Ground Station funciona com o IAM](#).
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte [Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui](#) no Guia do usuário do IAM.

- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Como fornecer acesso Contas da AWS a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

AWS políticas gerenciadas para AWS Ground Station

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as [políticas gerenciadas pelo cliente](#) que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para saber mais, consulte [AWS Políticas gerenciadas pela](#) no Guia do usuário do IAM.

AWS política gerenciada: AWSGround StationAgentInstancePolicy

É possível anexar a política `AWSGroundStationAgentInstancePolicy` às suas identidades do IAM.

Essa política concede permissões de AWS Ground Station agente à sua instância do Amazon EC2, permitindo que a instância envie e receba dados durante os contatos da Ground Station. Todas as permissões nesta política são do serviço Ground Station.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `groundstation`— Permite que instâncias de endpoint de fluxo de dados chamem o Ground Station Agent. APIs

Para ver a versão mais recente do documento de política JSON, consulte [AWSGroundStationAgentInstancePolicy](#) o AWS Managed Policy Reference Guide.

AWS política gerenciada: AWSService RoleForGroundStationDataflowEndpointGroupPolicy

Você não pode se associar AWSService RoleForGroundStationDataflowEndpointGroupPolicy às suas entidades do IAM. Essa política está vinculada a uma função vinculada ao serviço que permite AWS Ground Station realizar ações em seu nome. Para mais informações, consulte [Como usar funções vinculadas a serviços](#).

Essa política concede permissões do EC2 que permitem AWS Ground Station encontrar IPv4 endereços públicos.

Detalhes das permissões

Esta política inclui as seguintes permissões.

- `ec2:DescribeAddresses`— AWS Ground Station Permite listar todos os IPs associados EIPs em seu nome.

- `ec2:DescribeNetworkInterfaces`— Permite AWS Ground Station obter informações sobre as interfaces de rede associadas às instâncias do EC2 em seu nome.

Para ver a versão mais recente do documento de política JSON, consulte [AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy](#) o AWS Managed Policy Reference Guide.

AWS Ground Station atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas AWS Ground Station desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico do AWS Ground Station documento.

Alteração	Descrição	Data
AWSGroundStationAgentInstancePolicy : atualizar para uma política existente	AWS Ground Station adicionou novas permissões para permitir que os agentes recuperem a resposta da tarefa URLs para operações aprimoradas da Ground Station.	13 de novembro de 2025
AWSGroundStationAgentInstancePolicy – Nova política	AWS Ground Station adicionou uma nova política para fornecer à instância do endpoint do fluxo de dados permissões para usar o AWS Ground Station Agent.	12 de abril de 2023
AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy – Nova política	AWS Ground Station adicionou uma nova política que concede permissões do EC2 AWS Ground Station para permitir encontrar	2 de novembro de 2022

Alteração	Descrição	Data
	endereços IPv4 públicos associados a EIPs e interfaces de rede associadas a instâncias do EC2.	
AWS Ground Station começou a rastrear as alterações	AWS Ground Station começou a rastrear as alterações nas políticas AWS gerenciadas.	1 de março de 2021

Use funções vinculadas a serviços para Ground Station

AWS Ground Station usa funções [vinculadas ao serviço AWS Identity and Access Management \(IAM\)](#). A função vinculada ao serviço é um tipo exclusivo de perfil do IAM vinculada diretamente ao Ground Station. As funções vinculadas ao serviço são predefinidas pela Ground Station e incluem todas as permissões que o serviço exige para ligar para outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração do Ground Station porque você não precisa adicionar as permissões necessárias manualmente. O Ground Station define as permissões das funções vinculadas ao serviço e, exceto se definido de outra forma, somente o Ground Station pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Para obter informações sobre outros serviços que oferecem suporte a funções vinculadas a serviços, consulte [AWS Serviços que funcionam com IAM](#) e procure os serviços que têm Sim na coluna Funções vinculadas ao serviço. Escolha Sim com um link para visualizar a documentação da função vinculada a esse serviço.

Permissões de perfil vinculado a serviço para o Ground Station

A Ground Station usa a função vinculada ao serviço chamada — A `AWSServiceRoleForGroundStationDataflowEndpointGroup` AWS Ground Station usa essa função vinculada ao serviço para invocar o EC2 para encontrar endereços públicos. IPv4

A função `AWSServiceRoleForGroundStationDataflowEndpointGroup` vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `groundstation.amazonaws.com`

A política de permissões de função denominada `AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy` permite que a Ground Station conclua as seguintes ações nos recursos especificados:

- Ação: `ec2:DescribeAddresses` em `all AWS resources (*)`

A ação permite que a Ground Station liste todos os IPs associados EIPs a.

- Ação: `ec2:DescribeNetworkInterfaces` em `all AWS resources (*)`

A ação permite que a Ground Station obtenha informações sobre as interfaces de rede associadas às instâncias do EC2

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para obter mais informações, consulte [Permissões de função vinculada a serviços](#) no Guia do usuário do IAM.

Criar uma função vinculada a serviços para o Ground Station

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria um `DataflowEndpointGroup` na AWS CLI ou na AWS API, o Ground Station cria a função vinculada ao serviço para você.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um `DataflowEndpointGroup`, o Ground Station cria a função vinculada ao serviço para você novamente.

Você também pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso de Entrega de Dados para o Amazon EC2. Na AWS CLI ou na AWS API, crie uma função vinculada ao serviço com o nome do `groundstation.amazonaws.com` serviço. Para saber mais, consulte [Criar um perfil vinculado a serviço](#) no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Criar uma função vinculada a serviços para o Ground Station

O Ground Station não permite que você edite a função `AWSServiceRoleForGroundStationDataflowEndpointGroup` vinculada ao serviço. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte [Editar uma função vinculada a serviço](#) no Guia do usuário do IAM.

Apagar uma função vinculada a serviços para o Ground Station

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida.

Você pode excluir uma função vinculada ao serviço somente após primeiro excluí-la `DataflowEndpointGroups` usando a função vinculada ao serviço. Isso protege você de revogar inadvertidamente as permissões do seu `DataflowEndpointGroups`. Se uma função vinculada ao serviço for usada com várias `DataflowEndpointGroups`, você deverá excluir todas as `DataflowEndpointGroups` que usam a função vinculada ao serviço antes de excluí-la.

Note

Se o serviço Ground Station estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir os recursos da Ground Station usados pelo `AWSServiceRoleForGroundStationDataflowEndpointGroup`

- Exclua `DataflowEndpointGroups` por meio da AWS CLI ou da API da AWS.

Como excluir manualmente o perfil vinculado ao serviço usando o IAM

Use o console do IAM AWS CLI, o ou a AWS API para excluir a função `AWSServiceRoleForGroundStationDataflowEndpointGroup` vinculada ao serviço. Para saber mais, consulte [Excluir um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Regiões compatíveis com funções vinculadas ao serviço do Ground Station

O Ground Station oferece suporte a perfis vinculados ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte a [Tabela de regiões](#).

Solução de problemas

`NOT_AUTHORIZED_TO_CREATE_SLR`- Isso indica que a função em sua conta que está sendo usada para chamar a `CreateDataflowEndpointGroup` API não tem a `iam:CreateServiceLinkedRole`

permissão. Um administrador com a permissão `iam:CreateServiceLinkedRole` deve criar manualmente a função vinculada ao serviço para sua conta.

Criptografia de dados em repouso para AWS Ground Station

AWS Ground Station fornece criptografia por padrão para proteger seus dados confidenciais em repouso usando chaves AWS de criptografia próprias.

- **AWS chaves próprias** - AWS Ground Station usa essas chaves por padrão para criptografar automaticamente dados pessoais e efemérides diretamente identificáveis. Você não pode visualizar, gerenciar ou usar chaves AWS próprias nem auditar seu uso; no entanto, não é necessário realizar nenhuma ação ou alterar programas para proteger as chaves que criptografam os dados. Para obter mais informações, consulte [AWS-owned keys](#) no [Guia do AWS Key Management Service desenvolvedor](#).

A criptografia de dados em repouso por padrão reduz a sobrecarga operacional e a complexidade envolvidas na proteção de dados confidenciais. Ao mesmo tempo, permite que você crie aplicativos seguros que atendam aos rigorosos requisitos regulatórios e de conformidade de criptografia.

AWS Ground Station impõe criptografia em todos os dados confidenciais em repouso. No entanto, para alguns AWS Ground Station recursos, como efemérides, você pode optar por usar uma chave gerenciada pelo cliente no lugar das chaves gerenciadas padrão. AWS

- **Chaves gerenciadas pelo cliente** -- AWS Ground Station suporta o uso de uma chave simétrica gerenciada pelo cliente que você cria, possui e gerencia no lugar da criptografia existente AWS . Como você tem controle total dessa camada de criptografia, você pode realizar tarefas como:
 - Estabelecer e manter as políticas de chave
 - Estabelecer e manter subsídios e IAM policies
 - Habilitar e desabilitar políticas de chaves
 - Alternar os materiais de criptografia de chave
 - Adicionar etiquetas
 - Criar réplicas de chaves
 - Programar chaves para exclusão

Para obter mais informações, consulte [chave gerenciada pelo cliente](#) no [Guia do desenvolvedor do AWS Key Management Service](#).

A tabela a seguir resume os recursos que oferecem AWS Ground Station suporte ao uso de chaves gerenciadas pelo cliente.

Tipo de dados	AWS criptografia de chave própria	Criptografia de chave gerenciada pelo cliente (opcional)
Dados de efemérides usados para calcular a trajetória de um satélite	Habilitado	Habilitado
Efemérides de elevação de azimute usadas para comandar antenas	Habilitado	Habilitado

Note

AWS Ground Station ativa automaticamente a criptografia em repouso usando Chaves pertencentes à AWS para proteger dados de identificação pessoal sem nenhum custo. No entanto, AWS KMS cobranças são cobradas pelo uso de uma chave gerenciada pelo cliente. Para obter mais informações sobre preços, consulte [definição de preços da AWS Key Management Service](#).

Para obter mais informações sobre AWS KMS, consulte o [Guia do AWS Key Management Service desenvolvedor](#).

Para obter informações específicas para cada tipo de recurso, consulte:

- [Criptografia em repouso para dados de efemérides TLE e OEM](#)
- [Criptografia em repouso para efemérides de elevação de azimute](#)

Criar uma chave gerenciada pelo cliente

Você pode criar uma chave simétrica gerenciada pelo cliente usando o Console de gerenciamento da AWS, ou o. AWS KMS APIs

Para criar uma chave simétrica gerenciada pelo cliente

Siga as etapas para criar uma chave simétrica gerenciada pelo cliente no [Guia do AWS Key Management Service desenvolvedor](#).

Visão geral das principais políticas

As políticas de chaves controlam o acesso à chave gerenciada pelo cliente. Cada chave gerenciada pelo cliente deve ter exatamente uma política de chaves, que contém declarações que determinam quem pode usar a chave e como pode usá-la. Ao criar a chave gerenciada pelo cliente, é possível especificar uma política de chave. Para obter mais informações, consulte [Gerenciamento do acesso às chaves gerenciadas pelo cliente](#) no Guia do AWS Key Management Service desenvolvedor.

Para usar sua chave gerenciada pelo cliente com AWS Ground Station recursos, você deve configurar a política de chaves para conceder as permissões apropriadas ao AWS Ground Station serviço. As permissões específicas e a configuração da política dependem do tipo de recurso que você está criptografando:

- Para dados de efemérides de TLE e OEM, consulte os principais requisitos e [Criptografia em repouso para dados de efemérides TLE e OEM](#) exemplos de políticas específicas.
- Para dados de efemérides de elevação de azimute, consulte os principais requisitos e exemplos [Criptografia em repouso para efemérides de elevação de azimute](#) de políticas específicas.

Note

A configuração da política principal difere entre os tipos de efemérides. Os dados de efemérides TLE e OEM usam concessões para acesso à chave, enquanto as efemérides de elevação de azimute usam permissões diretas de política de chave. Certifique-se de configurar sua política de chaves de acordo com o tipo de recurso específico que você está criptografando.

Para obter mais informações sobre como [especificar permissões em uma política](#) e [solucionar problemas de acesso por chave](#), consulte o Guia do AWS Key Management Service desenvolvedor.

Especificando uma chave gerenciada pelo cliente para AWS Ground Station

Você pode especificar uma chave gerenciada pelo cliente para fornecer criptografia para os seguintes recursos:

- Efemérides (TLE, OEM e elevação de azimute)

Ao criar um recurso, você pode especificar a chave de dados fornecendo um kmsKeyArn

- kmsKeyArn- Um [identificador de chave](#) para uma chave gerenciada pelo AWS KMS cliente

AWS Ground Station contexto de criptografia

Um [contexto de criptografia](#) é um conjunto opcional de pares de valores-chave que contêm informações contextuais adicionais sobre os dados. AWS KMS usa o contexto de criptografia como dados autenticados adicionais para oferecer suporte à criptografia autenticada. Quando você inclui um contexto de criptografia em uma solicitação para criptografar dados, AWS KMS vincula o contexto de criptografia aos dados criptografados. Para descriptografar os dados, você inclui o mesmo contexto de criptografia na solicitação.

AWS Ground Station usa um contexto de criptografia diferente dependendo do recurso que está sendo criptografado e especifica um contexto de criptografia específico para cada concessão de chave criada.

Para obter detalhes do contexto de criptografia específico do recurso, consulte:

- [Criptografia em repouso para dados de efemérides TLE e OEM](#)
- [Criptografia em repouso para efemérides de elevação de azimute](#)

Criptografia em repouso para dados de efemérides TLE e OEM

Principais requisitos de política para efemérides de TLE e OEM

Para usar uma chave gerenciada pelo cliente com dados efemérides, sua política de chaves deve conceder as seguintes permissões ao serviço: AWS Ground Station

- [kms:CreateGrant](#)- Cria uma concessão de acesso em uma chave gerenciada pelo cliente. Concede AWS Ground Station acesso para realizar [operações de concessão](#) na chave gerenciada pelo cliente para leitura e armazenamento de dados criptografados.
- [kms:DescribeKey](#)- Fornece os detalhes da chave gerenciada pelo cliente AWS Ground Station para permitir a validação da chave antes de tentar usar a chave fornecida.

Para obter mais informações sobre [o uso de subsídios](#), consulte o Guia do AWS Key Management Service desenvolvedor.

Permissões de usuário do IAM para criar efemérides com chaves gerenciadas pelo cliente

Quando AWS Ground Station usa uma chave gerenciada pelo cliente em operações criptográficas, ela age em nome do usuário que está criando o recurso de efemérides.

Para criar um recurso de efemérides usando uma chave gerenciada pelo cliente, o usuário deve ter permissões para chamar as seguintes operações na chave gerenciada pelo cliente:

- [kms:CreateGrant](#)- Permite que o usuário crie concessões na chave gerenciada pelo cliente em nome de AWS Ground Station.
- [kms:DescribeKey](#)- Permite que o usuário visualize os detalhes da chave gerenciada pelo cliente para validar a chave.

Você pode especificar essas permissões em uma política de chaves ou em uma política do IAM, se a política de chaves permitir. Essas permissões garantem que os usuários possam AWS Ground Station autorizar o uso da chave gerenciada pelo cliente para operações de criptografia em seu nome.

Como AWS Ground Station usa subsídios AWS KMS para efemérides

AWS Ground Station exige uma [concessão de chave](#) para usar sua chave gerenciada pelo cliente.

Quando você carrega uma efeméride criptografada com uma chave gerenciada pelo cliente, AWS Ground Station cria uma concessão de chave em seu nome enviando uma [CreateGrant](#) solicitação para AWS KMS. As concessões AWS KMS são usadas para dar AWS Ground Station acesso a uma AWS KMS chave em sua conta.

Isso permite AWS Ground Station fazer o seguinte:

- Ligar para [GenerateDataKey](#) para gerar uma chave de dados criptografada e armazená-la, porque a chave de dados não é usada imediatamente para criptografar.
- Chame o [Decrypt](#) para usar a chave de dados criptografada armazenada para acessar os dados criptografados.
- Chame [Encrypt](#) para usar a chave de dados para criptografar dados.
- Configure uma entidade principal aposentada para permitir que o serviço para [RetireGrant](#).

Você pode revogar o acesso à concessão a qualquer momento. Se você fizer isso, AWS Ground Station não conseguirá acessar nenhum dos dados criptografados pela chave gerenciada pelo cliente, o que afeta as operações que dependem desses dados. Por exemplo, se você remover uma concessão de chave de uma efeméride atualmente em uso para um contato, não AWS Ground Station poderá usar os dados de efemérides fornecidos para apontar a antena durante o contato. Isso fará com que o contato termine em um estado de FALHA.

Contexto de criptografia de efemérides

Os principais subsídios para criptografar recursos de efemérides estão vinculados a um ARN de satélite específico.

```
"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

As doações de chaves são reutilizadas para o mesmo par chave-satélite.

Usar o contexto de criptografia para monitoramento

Quando você usa uma chave simétrica gerenciada pelo cliente para criptografar suas efemérides, também pode utilizar o contexto de criptografia em registros de auditoria e logs para identificar como a chave gerenciada pelo cliente está sendo utilizada. O contexto de criptografia também aparece nos [registros gerados pelo AWS CloudTrail ou Amazon CloudWatch Logs](#).

Uso do contexto de criptografia para controlar o acesso à chave gerenciada pelo cliente

Você pode usar o contexto de criptografia nas políticas de chaves e políticas do IAM como `conditions` e controlar o acesso à sua chave simétrica gerenciada pelo cliente. Você também pode usar restrições no contexto de criptografia em uma concessão.

AWS Ground Station usa uma restrição de contexto de criptografia nas concessões para controlar o acesso à chave gerenciada pelo cliente em sua conta ou região. A restrição da concessão exige que as operações permitidas pela concessão usem o contexto de criptografia especificado.

Veja a seguir exemplos de declarações de políticas de chave para conceder acesso a uma chave gerenciada pelo cliente para um contexto de criptografia específico. A condição nesta declaração de política exige que as concessões tenham uma restrição de contexto de criptografia que especifique o contexto de criptografia.

O exemplo a seguir mostra uma política fundamental para dados de efemérides vinculados a um satélite:

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"
            }
        }
    }
}

```

Monitorando suas chaves de criptografia para efemérides

Ao usar uma chave gerenciada pelo AWS Key Management Service cliente com seus recursos de efemérides, você pode usar nossos [CloudWatch registros AWS CloudTrail da Amazon](#) para rastrear solicitações enviadas para AWS Ground Station AWS KMS. Os exemplos a seguir são CloudTrail eventos para [CreateGrant](#), [GenerateDataKey](#), [Decrypt](#) e [DescribeKey](#) para monitorar AWS KMS operações chamadas por AWS Ground Station para acessar dados criptografados pela chave gerenciada pelo cliente.

CreateGrant

Quando você usa uma chave gerenciada pelo AWS KMS cliente para criptografar seus recursos de efemérides, AWS Ground Station envia uma [CreateGrant](#) solicitação em seu nome para acessar a AWS KMS chave em sua conta. A concessão AWS Ground Station criada é específica para o recurso associado à chave gerenciada pelo AWS KMS cliente. Além disso, AWS Ground Station usa a [RetireGrant](#) operação para remover uma concessão quando você exclui um recurso.

O evento de exemplo a seguir registra a [CreateGrant](#) operação de uma efeméride:

```

{
  "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

Quando você usa uma chave gerenciada pelo AWS KMS cliente para criptografar seus recursos de efemérides, AWS Ground Station envia uma [DescribeKey](#) solicitação em seu nome para validar se a chave solicitada existe em sua conta.

O evento de exemplo a seguir registra a operação [DescribeKey](#):

```

{
  "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

Quando você usa uma chave gerenciada pelo AWS KMS cliente para criptografar seus recursos de efemérides, AWS Ground Station envia uma [GenerateDataKey](#) solicitação para gerar uma chave de dados com a qual criptografar seus dados.

O evento de exemplo a seguir registra a [GenerateDataKey](#) operação de uma efeméride:

```

{
  "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

Quando você usa uma chave gerenciada pelo AWS KMS cliente para criptografar seus recursos de efemérides, AWS Ground Station usa a operação [Decrypt para descriptografar](#) as efemérides fornecidas se já estiverem criptografadas com a mesma chave gerenciada pelo cliente. Por

exemplo, se uma efeméride estiver sendo carregada de um bucket do S3 e for criptografada nesse bucket com uma determinada chave.

O evento de exemplo a seguir registra a operação [Decrypt](#) para uma efeméride:

```
{
  "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"
}
```

```
}
```

Criptografia em repouso para efemérides de elevação de azimute

Principais requisitos de política para efemérides de elevação de azimute

Para usar uma chave gerenciada pelo cliente com dados de efemérides de elevação de azimute, sua política de chaves deve conceder as seguintes permissões ao serviço. AWS Ground Station Ao contrário dos dados de efemérides TLE e OEM, que usam concessões, as efemérides de elevação de azimute usam permissões diretas de política de chave para operações de criptografia. Esse é um método mais simples de gerenciar as permissões e usar suas chaves.

- [kms:GenerateDataKey](#)- Gera chaves de dados para criptografar seus dados de efemérides de elevação de azimute.
- [kms:Decrypt](#)- Descriptografa as chaves de dados criptografadas ao acessar seus dados de efemérides de elevação de azimute.

Exemplo de política de chaves que concede AWS Ground Station acesso a uma chave gerenciada pelo cliente

Note

Com efemérides de elevação de azimute, você deve configurar essas permissões diretamente na política de chaves. O diretor do AWS Ground Station serviço regional (por exemplo, `groundstation.region.amazonaws.com`) deve receber essas permissões em suas principais declarações de política. Sem essas declarações adicionadas à política de chaves, não AWS Ground Station será possível armazenar ou acessar suas efemérides personalizadas de elevação de azimute.

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": "*"
  }
]
}

```

Permissões de usuário do IAM para criar efemérides de elevação de azimute com chaves gerenciadas pelo cliente

Quando AWS Ground Station usa uma chave gerenciada pelo cliente em operações criptográficas, ela age em nome do usuário que está criando o recurso de efemérides de elevação de azimute.

Para criar um recurso de efemérides de elevação de azimute usando uma chave gerenciada pelo cliente, o usuário deve ter permissões para chamar as seguintes operações na chave gerenciada pelo cliente:

- [kms:GenerateDataKey](#)- Permite que o usuário gere chaves de dados para criptografar os dados de efemérides de elevação de azimute.
- [kms:Decrypt](#)- Permite que o usuário decifre as chaves de dados ao acessar os dados de efemérides de elevação de azimute.
- [kms:DescribeKey](#)- Permite que o usuário visualize os detalhes da chave gerenciada pelo cliente para validar a chave.

Você pode especificar essas permissões em uma política de chaves ou em uma política do IAM, se a política de chaves permitir. Essas permissões garantem que os usuários possam AWS Ground Station autorizar o uso da chave gerenciada pelo cliente para operações de criptografia em seu nome.

Como AWS Ground Station usa as principais políticas para efemérides de elevação de azimute

Quando você fornece dados de efemérides de elevação de azimute com uma chave gerenciada pelo cliente, AWS Ground Station usa políticas de chaves para acessar sua chave de criptografia. As permissões são concedidas diretamente AWS Ground Station por meio de declarações políticas importantes, e não por meio de doações, como acontece com dados de efemérides de TLE ou OEM.

Se você remover AWS Ground Station o acesso à chave gerenciada pelo cliente, AWS Ground Station não conseguirá acessar nenhum dos dados criptografados por essa chave, o que afeta as operações que dependem desses dados. Por exemplo, se você remover as principais permissões de política para efemérides de elevação de azimute atualmente em uso para um contato, não AWS Ground Station poderá usar os dados de elevação de azimute fornecidos para comandar a antena durante o contato. Isso fará com que o contato termine em um estado de FALHA.

Contexto de criptografia de efemérides de elevação de azimute

[Quando AWS Ground Station usa sua AWS KMS chave para criptografar dados de efemérides de elevação de azimute, o serviço especifica um contexto de criptografia.](#) O contexto de criptografia são dados adicionais autenticados (AAD) AWS KMS usados para garantir a integridade dos dados. Quando um contexto de criptografia é especificado para uma operação de criptografia, o serviço deve especificar esse mesmo contexto para a operação de descriptografia. Caso contrário, ocorrerá uma falha na descriptografia. O contexto de criptografia também é gravado em seus CloudTrail registros para ajudar você a entender por que uma determinada AWS KMS chave foi usada. Seus CloudTrail registros podem conter várias entradas descrevendo o uso de uma AWS KMS chave, mas o contexto de criptografia em cada entrada de registro pode ajudá-lo a determinar o motivo desse uso específico.

AWS Ground Station especifica o seguinte contexto de criptografia ao realizar operações criptográficas com sua chave gerenciada pelo cliente em uma efeméride de elevação de azimute:

```
{
  "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"
  }
}

```

O contexto de criptografia contém:

`aws:groundstation:ground-station-id`

O nome da estação terrestre associada às efemérides de elevação do azimute.

`aws: estação terrestre: arn`

O ARN do recurso de efemérides.

`aws: s3: arn`

O ARN das efemérides armazenadas no Amazon S3.

Usar o contexto de criptografia para controlar o acesso à chave gerenciada pelo cliente

Você pode usar declarações de condição do IAM para controlar o AWS Ground Station acesso à sua chave gerenciada pelo cliente. Adicionar uma declaração de condição `kms:Decrypt` às ações `kms:GenerateDataKey` e restringe para quais estações terrestres a AWS KMS podem ser usadas.

A seguir estão exemplos de declarações de políticas importantes para conceder AWS Ground Station acesso à sua chave gerenciada pelo cliente em uma região específica para uma estação terrestre específica. A condição nesta declaração de política exige que todos criptografem e descriptografem o acesso à chave que especifiquem um contexto de criptografia que corresponda à condição na política de chaves.

Exemplo de política de chaves que concede AWS Ground Station acesso a uma chave gerenciada pelo cliente para uma estação terrestre específica

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"
      }
    }
  }
]
}

```

Exemplo de política de chaves que concede AWS Ground Station acesso a uma chave gerenciada pelo cliente para várias estações terrestres

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"
        ]
      }
    }
  }
]
}

```

Monitorando suas chaves de criptografia para efemérides de elevação de azimute

Ao usar uma chave gerenciada pelo AWS KMS cliente com seus recursos de efemérides de elevação de azimute, você pode usar [CloudTrail](#) ou [CloudWatch registrar para rastrear as solicitações](#) enviadas para AWS Ground Station AWS KMS. Os exemplos a seguir são CloudTrail eventos para [GenerateDataKey](#) [Decrypt](#) para monitorar AWS KMS operações chamadas por AWS Ground Station para acessar dados criptografados pela chave gerenciada pelo cliente.

GenerateDataKey

Quando você usa uma chave gerenciada pelo AWS KMS cliente para criptografar seus recursos de efemérides de elevação de azimute, AWS Ground Station envia uma [GenerateDataKey](#) solicitação para AWS KMS gerar uma chave de dados com a qual criptografar seus dados.

O evento de exemplo a seguir registra a [GenerateDataKey](#) operação para efemérides de elevação de azimute:

```
{
  "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

Quando você usa uma chave gerenciada pelo AWS KMS cliente para criptografar seus recursos de efemérides de elevação de azimute, AWS Ground Station usa a operação [Descriptografar para descriptografar](#) os dados de efemérides de elevação de azimute fornecidos se eles já estiverem criptografados com a mesma chave gerenciada pelo cliente.

O evento de exemplo a seguir registra a operação [Decrypt](#) para efemérides de elevação de azimute:

```

{
  "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"  
}
```

Criptografia de dados durante o trânsito para AWS Ground Station

AWS Ground Station fornece criptografia por padrão para proteger seus dados confidenciais durante o trânsito. Os dados podem ser transmitidos entre as localizações AWS Ground Station da antena e suas instâncias do Amazon EC2 de duas maneiras, dependendo da configuração do perfil da missão.

- AWS Ground Station Agente
- Endpoint de fluxo de dados

Cada método de streaming de dados lida com a criptografia de dados em trânsito de forma diferente. As seções a seguir descrevem cada método.

AWS Ground Station Fluxos de agentes

AWS Ground Station O agente criptografa seus fluxos usando chaves gerenciadas pelo AWS KMS cliente. O AWS Ground Station agente em execução na sua instância do Amazon EC2 descriptografará automaticamente o stream para fornecer dados descriptografados.

A AWS KMS chave usada para criptografar um fluxo é especificada ao criar um `MissionProfile` no [streamsKmsKey](#) parâmetro. Todas as permissões que concedem AWS Ground Station acesso às chaves são tratadas por meio da política de AWS KMS chaves anexada a `streamsKmsKey`

Streams de endpoint de fluxo de dados

Os fluxos de endpoint do Dataflow são criptografados usando o [Datagram Transport Layer Security](#) (DTLS). Isso é feito usando certificados autoassinados e não requer configuração adicional.

Exemplo de configurações de perfil de missão

Os exemplos fornecidos mostram como pegar um satélite de transmissão pública e criar um perfil de missão que o suporte. Os modelos resultantes são fornecidos para ajudá-lo a estabelecer um contato público por satélite de transmissão e para ajudá-lo a tomar decisões sobre seus satélites.

Tópicos

- [JPSS-1 - Satélite de transmissão pública \(PBS\) - Avaliação](#)
- [Satélite de transmissão pública utilizando a entrega de dados do Amazon S3](#)
- [Satélite de transmissão pública utilizando um ponto final de fluxo de dados \(banda estreita\)](#)
- [Satélite de transmissão pública utilizando um endpoint de fluxo de dados \(demodulado e decodificado\)](#)
- [Satélite de transmissão pública utilizando AWS Ground Station Agent \(banda larga\)](#)

JPSS-1 - Satélite de transmissão pública (PBS) - Avaliação

Esta seção de exemplo corresponde ao [Visão geral do processo de integração de clientes](#). Ele fornece uma breve análise de compatibilidade AWS Ground Station e prepara o terreno para os exemplos específicos a seguir.

Conforme mencionado na [Satélites de transmissão pública](#) seção, você pode utilizar satélites selecionados, ou caminhos de comunicação de um satélite, que estão disponíveis publicamente. Nesta seção, descrevemos o [JPSS-1](#) nos termos. AWS Ground Station Para referência, utilizamos o [Joint Polar Satellite System 1 \(JPSS-1\) Spacecraft High Rate Data \(HRD\) para Direct Broadcast Stations \(DBS\) Documento de Controle de Interface \(ICD\) de Radiofrequência \(RF\)](#) para concluir o exemplo. Além disso, vale ressaltar que o JPSS-1 está associado ao NORAD ID 43013.

O satélite JPSS-1 oferece um uplink e três caminhos de comunicação diretos de downlink, conforme visto na Figura 1-1 do ICD. Desses quatro caminhos de comunicação, somente o único caminho de comunicação de downlink de dados de alta taxa (HRD) está disponível para consumo público. Com base nisso, você verá que esse caminho também terá dados muito mais específicos associados a ele. Os quatro caminhos são os seguintes:

- Caminho de comando (uplink) na frequência MHz central de 2067,27 com uma taxa de dados de 2 a 128 kbps. Esse caminho não está acessível ao público.

- Caminho de telemetria (downlink) na frequência MHz central de 2247,5 com uma taxa de dados de 1-524 kbps. Esse caminho não está acessível ao público.
- Caminho SMD (downlink) na frequência GHz central de 26.7034 com uma taxa de dados de 150-300 Mbps. Esse caminho não está acessível ao público.
- A RF para o caminho do HRD (downlink) na frequência MHz central de 7812 com uma taxa de dados de 15 Mbps. Tem uma MHz largura de banda de 30%, e é right-hand-circular-polarized. Quando você integra o JPSS-1 com AWS Ground Station, esse é o caminho de comunicação ao qual você tem acesso. Esse caminho de comunicação contém dados científicos de instrumentos, dados de engenharia de instrumentos, dados de telemetria de instrumentos e dados de manutenção de espaçonaves em tempo real.

Ao compararmos os possíveis caminhos de dados, vemos que os caminhos de comando (uplink), telemetria (downlink) e HRD (downlink) atendem aos recursos de frequência, largura de banda e uso simultâneo multicanal do. AWS Ground Station O caminho SMD não é compatível porque a frequência central está fora do alcance dos receptores existentes. Para obter mais informações sobre os recursos suportados, consulte [AWS Ground Station Capacidades do site](#).

Note

Como o caminho SMD não é compatível com AWS Ground Station ele, ele não será representado nas configurações de exemplo.

Note

Como os caminhos de comando (uplink) e telemetria (downlink) não estão definidos no ICD, nem estão disponíveis para uso público, os valores fornecidos quando usados são nocionais.

Satélite de transmissão pública utilizando a entrega de dados do Amazon S3

Este exemplo se baseia na análise feita na [JPSS-1 - Satélite de transmissão pública \(PBS\) - Avaliação](#) seção do guia do usuário.

Neste exemplo, você precisará assumir um cenário: capturar o caminho de comunicação do HRD como frequência intermediária digital e armazená-lo para processamento em lote no futuro. Isso economiza as amostras brutas de quadratura em fase (I/Q) de radiofrequência (RF) após a digitalização. Quando os dados estiverem em seu bucket do Amazon S3, você poderá demodular e decodificar os dados usando qualquer software que desejar. Consulte o [MathWorks Tutorial](#) para obter um exemplo detalhado de processamento. Depois de usar esse exemplo, você pode considerar a adição de componentes de preços à EC2 vista da Amazon para processar os dados e reduzir seus custos gerais de processamento.

Caminhos de comunicação

Esta seção representa [Planeje seus caminhos de comunicação de fluxo de dados](#) como começar.

Todos os trechos de modelo a seguir pertencem à seção Recursos do CloudFormation modelo.

Resources:

```
# Resources that you would like to create should be placed within the Resources section.
```

Note

Para obter mais informações sobre o conteúdo de um CloudFormation modelo, consulte [as seções Modelo](#).

Considerando nosso cenário para fornecer um único caminho de comunicação para o Amazon S3, você sabe que terá um único caminho de entrega assíncrono. De acordo com a [Entrega assíncrona de dados](#) seção, você deve definir um bucket do Amazon S3.

```
# 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]]]]]]
```

Além disso, você precisará criar as funções e políticas apropriadas AWS Ground Station para permitir o uso do bucket.

```
# 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 configurações

Esta seção representa [Crie configurações](#) como começar.

Você precisará de uma configuração de rastreamento para definir sua preferência de uso do autotrack. Selecionar PREFERRED como trilha automática pode melhorar a qualidade do sinal, mas não é necessário atender à qualidade do sinal devido à qualidade suficiente das efemérides JPSS-1.

```

TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"

```

Com base no caminho de comunicação, você precisará definir uma configuração de downlink de antena para representar a parte do satélite, bem como uma gravação s3 para se referir ao bucket Amazon S3 que você acabou de criar.

```

# 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 perfil da missão

Esta seção representa [Criar perfil de missão](#) como começar.

Agora que você tem as configurações associadas, pode usá-las para criar o fluxo de dados. Você usará os padrões para os demais parâmetros.

```

# 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

```

Juntando tudo

Com os recursos acima, agora você tem a capacidade de agendar contatos JPSS-1 para entrega assíncrona de dados a partir de qualquer um dos seus contatos integrados. [AWS Ground Station Localizações](#)

A seguir está um CloudFormation modelo completo que inclui todos os recursos descritos nesta seção combinados em um único modelo que pode ser usado diretamente em CloudFormation.

O CloudFormation modelo nomeado `AquaSnppJpss-1TerraDigIfS3DataDelivery.yml` contém um bucket Amazon S3 e os AWS Ground Station recursos necessários para agendar contatos e receber dados de transmissão direta de sinal/IP VITA-49.

Se Aqua, SNPP, JPSS-1/NOAA-20 e Terra não estiverem integrados à sua conta, consulte. [Satélite a bordo](#)

Note

Você pode acessar o modelo acessando o bucket Amazon S3 de integração do cliente usando AWS credenciais válidas. Os links abaixo usam um bucket regional do Amazon S3. Altere o código da `us-west-2` região para representar a região correspondente na qual você deseja criar a CloudFormation pilha.

Além disso, as instruções a seguir usam YAML. No entanto, os modelos estão disponíveis no formato YAML ou JSON. Para usar o JSON, substitua a extensão do `.yaml` arquivo por `.json` ao baixar o modelo.

Para baixar o modelo usando AWS CLI, use o seguinte comando:

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml .
```

É possível visualizar e fazer download do modelo no console navegando até o seguinte URL no seu navegador:

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml
```

Você pode especificar o modelo diretamente CloudFormation usando o seguinte link:

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss-1TerraDigIfS3DataDelivery.yml
```

Satélite de transmissão pública utilizando um ponto final de fluxo de dados (banda estreita)

Este exemplo se baseia na análise feita na [JPSS-1 - Satélite de transmissão pública \(PBS\) - Avaliação](#) seção do guia do usuário.

Para concluir este exemplo, você precisará assumir um cenário: capturar o caminho de comunicação do HRD como frequência intermediária digital (DigIF) e processá-lo conforme ele é recebido por um aplicativo de endpoint de fluxo de dados em uma instância da EC2 Amazon usando um SDR.

Caminhos de comunicação

Esta seção representa [Planeje seus caminhos de comunicação de fluxo de dados](#) como começar. Neste exemplo, você criará duas seções em seu CloudFormation modelo: seções de parâmetros e recursos.

Note

Para obter mais informações sobre o conteúdo de um CloudFormation modelo, consulte [as seções Modelo](#).

Para a seção Parâmetros, você adicionará os seguintes parâmetros. Você especificará valores para eles ao criar a pilha por meio do CloudFormation console.

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

Você precisa criar um par de chaves e fornecer o nome para o EC2 EC2Key parâmetro Amazon. Consulte [Criar um par de chaves para sua EC2 instância da Amazon](#). Além disso, você precisará fornecer a ID de AMI específica da região correta ao criar a CloudFormation pilha. Consulte [AWS Ground Station Imagens de máquinas da Amazon \(AMIs\)](#).

Os trechos de modelo restantes pertencem à seção Recursos do CloudFormation modelo.

Resources:

Resources that you would like to create should be placed within the resource section.

Dado nosso cenário para fornecer um único caminho de comunicação para uma EC2 instância, você terá um único caminho de entrega síncrono. De acordo com a [Entrega síncrona de dados](#) seção, você deve instalar e configurar uma EC2 instância da Amazon com um aplicativo de endpoint de fluxo de dados e criar um ou mais grupos de endpoints de fluxo de dados.

```
# The EC2 instance that will send/receive data to/from your satellite using AWS
Ground Station.
```

```
ReceiverInstance:
```

```
Type: AWS::EC2::Instance
```

```
Properties:
```

```
DisableApiTermination: false
```

```
IamInstanceProfile: !Ref GeneralInstanceProfile
```

```
ImageId: !Ref ReceiverAMI
```

```
InstanceType: m5.4xlarge
```

```
KeyName: !Ref EC2Key
```

```
Monitoring: true
```

```
PlacementGroupName: !Ref ClusterPlacementGroup
```

```
SecurityGroupIds:
```

```

- Ref: InstanceSecurityGroup
SubnetId: !Ref ReceiverSubnet
BlockDeviceMappings:
  - DeviceName: /dev/xvda
    Ebs:
      VolumeType: gp2
      VolumeSize: 40
Tags:
  - Key: Name
    Value: !Join [ "-", [ "Receiver" , !Ref "AWS::StackName" ] ]
UserData:
  Fn::Base64:
    |
    #!/bin/bash
    exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console)
2>&1
    echo `date +%F %R:%S` "INFO: Logging Setup" >&2

    GROUND_STATION_DIR="/opt/aws/groundstation"
    GROUND_STATION_BIN_DIR="${GROUND_STATION_DIR}/bin"
    STREAM_CONFIG_PATH="${GROUND_STATION_DIR}/customer_stream_config.json"

    echo "Creating ${STREAM_CONFIG_PATH}"
    cat << STREAM_CONFIG > "${STREAM_CONFIG_PATH}"
    {
      "ddx_streams": [
        {
          "streamName": "Downlink",
          "maximumWanRate": 4000000000,
          "lanConfigDevice": "lo",
          "lanConfigPort": 50000,
          "wanConfigDevice": "eth1",
          "wanConfigPort": 55888,
          "isUplink": false
        }
      ]
    }
    STREAM_CONFIG

    echo "Waiting for dataflow endpoint application to start"
    while netstat -lnt | awk '$4 ~ /:80$/ {exit 1}'; do sleep 10; done

    echo "Configuring dataflow endpoint application streams"

```

```

    python "${GROUND_STATION_BIN_DIR}/configure_streams.py" --configFileName
"${STREAM_CONFIG_PATH}"
    sleep 2
    python "${GROUND_STATION_BIN_DIR}/save_default_config.py"

    exit 0

# The AWS Ground Station Dataflow Endpoint Group that defines the endpoints that AWS
Ground
# Station will use to send/receive data to/from your satellite.
DataflowEndpointGroup:
  Type: AWS::GroundStation::DataflowEndpointGroup
  Properties:
    ContactPostPassDurationSeconds: 180
    ContactPrePassDurationSeconds: 120
    EndpointDetails:
      - Endpoint:
          Name: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ] # needs to
match DataflowEndpointConfig name
          Address:
            Name: !GetAtt ReceiverInstanceNetworkInterface.PrimaryPrivateIpAddress
            Port: 55888
    SecurityDetails:
      SecurityGroupIds:
        - Ref: "DataflowEndpointSecurityGroup"
      SubnetIds:
        - !Ref ReceiverSubnet
      RoleArn: !GetAtt DataDeliveryServiceRole.Arn

# The security group for your EC2 instance.
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: AWS Ground Station receiver instance security group.
    VpcId: !Ref ReceiverVPC
    SecurityGroupIngress:
      # To allow SSH access to the instance, add another rule allowing tcp port 22
from your CidrIp
      - IpProtocol: udp
        FromPort: 55888
        ToPort: 55888
        SourceSecurityGroupId: !Ref DataflowEndpointSecurityGroup
        Description: "AWS Ground Station Downlink Stream"

```

```
# The security group that the ENI created by AWS Ground Station belongs to.
DataflowEndpointSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security Group for AWS Ground Station registration of Dataflow
Endpoint Groups
  VpcId: !Ref ReceiverVPC
  SecurityGroupEgress:
    - IpProtocol: udp
      FromPort: 55888
      ToPort: 55888
      CidrIp: 10.0.0.0/8
      Description: "AWS Ground Station Downlink Stream To 10/8"
    - IpProtocol: udp
      FromPort: 55888
      ToPort: 55888
      CidrIp: 172.16.0.0/12
      Description: "AWS Ground Station Downlink Stream To 172.16/12"
    - IpProtocol: udp
      FromPort: 55888
      ToPort: 55888
      CidrIp: 192.168.0.0/16
      Description: "AWS Ground Station Downlink Stream To 192.168/16"

# The placement group in which your EC2 instance is placed.
ClusterPlacementGroup:
  Type: AWS::EC2::PlacementGroup
  Properties:
    Strategy: cluster

ReceiverVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: "10.0.0.0/16"
  Tags:
    - Key: "Name"
      Value: "AWS Ground Station - PBS to dataflow endpoint Example VPC"
    - Key: "Description"
      Value: "VPC for EC2 instance receiving AWS Ground Station data"

ReceiverSubnet:
  Type: AWS::EC2::Subnet
  Properties:
```

```
# Ensure your CidrBlock will always have at least one available IP address per
dataflow endpoint.
# See https://docs.aws.amazon.com/vpc/latest/userguide/subnet-sizing.html for
subent 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
```

Além disso, você também precisará criar as políticas e funções apropriadas AWS Ground Station para permitir a criação de uma interface de rede elástica (ENI) em sua conta.

```
# 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 configurações

Esta seção representa [Crie configurações](#) como começar.

Você precisará de uma configuração de rastreamento para definir sua preferência de uso do autotrack. Selecionar PREFERRED como trilha automática pode melhorar a qualidade do sinal, mas não é necessário atender à qualidade do sinal devido à qualidade suficiente das efemérides JPSS-1.

```

TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"

```

Com base no caminho de comunicação, você precisará definir uma configuração de antena-downlink para representar a parte do satélite, bem como uma configuração de endpoint de fluxo de dados para se referir ao grupo de endpoints de fluxo de dados que define os detalhes do endpoint.

```

# 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 perfil da missão

Esta seção representa [Criar perfil de missão](#) como começar.

Agora que você tem as configurações associadas, pode usá-las para criar o fluxo de dados. Você usará os padrões para os demais parâmetros.

```

# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnpjPssMissionProfile:
    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

```

Juntando tudo

Com os recursos acima, agora você tem a capacidade de agendar contatos JPSS-1 para entrega síncrona de dados a partir de qualquer um dos seus contatos integrados. AWS Ground Station [AWS Ground Station Localizações](#)

A seguir está um CloudFormation modelo completo que inclui todos os recursos descritos nesta seção combinados em um único modelo que pode ser usado diretamente em CloudFormation.

O CloudFormation modelo nomeado foi AquaSnppJpssTerraDigIF.yml projetado para fornecer acesso rápido para começar a receber dados de frequência intermediária digitalizada (DigIF) para os satélites Aqua, SNPP, JPSS-1/NOAA-20 e Terra. Ele contém uma EC2 instância da Amazon e os CloudFormation recursos necessários para receber dados brutos de transmissão direta do DigIF.

Se Aqua, SNPP, JPSS-1/NOAA-20 e Terra não estiverem integrados à sua conta, consulte. [Satélite a bordo](#)

Note

Você pode acessar o modelo acessando o bucket Amazon S3 de integração do cliente usando AWS credenciais válidas. Os links abaixo usam um bucket regional do Amazon S3. Altere o código da us-west-2 região para representar a região correspondente na qual você deseja criar a CloudFormation pilha.

Além disso, as instruções a seguir usam YAML. No entanto, os modelos estão disponíveis no formato YAML ou JSON. Para usar JSON, substitua a extensão do .yaml arquivo por .json ao baixar o modelo.

Para baixar o modelo usando AWS CLI, use o seguinte comando:

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpssTerraDigIF.yml .
```

É possível visualizar e fazer download do modelo no console navegando até o seguinte URL no seu navegador:

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpssTerraDigIF.yml
```

Você pode especificar o modelo diretamente CloudFormation usando o seguinte link:

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpssTerraDigIF.yml
```

Quais recursos adicionais o modelo define?

O AquaSnppJpssTerraDigIF modelo inclui os seguintes recursos adicionais:

- (Opcional) Acionadores de CloudWatch eventos - AWS Lambda Função que é acionada usando CloudWatch eventos enviados AWS Ground Station antes e depois de um contato. A AWS Lambda função iniciará e, opcionalmente, interromperá sua instância receptora.
- (Opcional) EC2 Verificação de contatos - A opção de usar o Lambda para configurar um sistema de verificação de suas EC2 instâncias da Amazon para contatos com notificação do SNS. É importante observar que isso pode incorrer em cobranças, dependendo do seu uso atual.
- Ground Station Amazon Machine Image Retrieval Lambda: a opção de selecionar qual software está instalado em sua instância e a AMI de sua escolha. As opções de software incluem DDX 2.6.2 Only e DDX 2.6.2 with qRadio 3.6.0. Essas opções continuarão a se expandir à medida que atualizações e recursos adicionais de software forem lançados.
- Perfis de missão adicionais - Perfis de missão para satélites de transmissão pública adicionais (Aqua, SNPP e Terra).
- Configurações adicionais de downlink de antena - Configurações de downlink de antena para satélites de transmissão pública adicionais (Aqua, SNPP e Terra).

Os valores e os parâmetros dos satélites nesse modelo já estão preenchidos. Esses parâmetros facilitam o uso AWS Ground Station imediato desses satélites. Você não precisa configurar seus próprios valores para usar AWS Ground Station ao usar esse modelo. No entanto, é possível personalizar os valores para que o modelo funcione para seu caso de uso.

Onde recebo os meus dados?

O grupo de endpoints do fluxo de dados é configurado para usar a interface de rede da instância do receptor que parte do modelo cria. A instância receptora usa um aplicativo de endpoint de fluxo de dados para receber o stream de dados da AWS Ground Station porta definida pelo endpoint de fluxo de dados. Após serem recebidos, os dados estarão disponíveis para consumo por meio da porta UDP 50000 no adaptador de loopback da instância do receptor. Para obter

mais informações sobre como configurar um grupo de endpoints de fluxo de dados, consulte.

[AWS::GroundStation::DataflowEndpointGroup](#)

Satélite de transmissão pública utilizando um endpoint de fluxo de dados (demodulado e decodificado)

Este exemplo se baseia na análise feita na [JPSS-1 - Satélite de transmissão pública \(PBS\) - Avaliação](#) seção do guia do usuário.

Para concluir este exemplo, você precisará assumir um cenário: você deseja capturar o caminho de comunicação do HRD como dados de transmissão direta demodulados e decodificados usando um endpoint de fluxo de dados. Este exemplo é um bom ponto de partida se você planeja processar os dados usando o software NASA Direct Readout Labs (RT-STPS e IPOPP).

Caminhos de comunicação

Esta seção representa [Planeje seus caminhos de comunicação de fluxo de dados](#) como começar. Neste exemplo, você criará duas seções em seu CloudFormation modelo: seções de parâmetros e recursos.

Note

Para obter mais informações sobre o conteúdo de um CloudFormation modelo, consulte [as seções Modelo](#).

Para a seção Parâmetros, você adicionará os seguintes parâmetros. Você especificará valores para eles ao criar a pilha por meio do CloudFormation console.

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

Você precisa criar um par de chaves e fornecer o nome para o EC2 EC2Key parâmetro da Amazon. Consulte [Criar um par de chaves para sua EC2 instância da Amazon](#). Além disso, você precisará fornecer a ID de AMI específica da região correta ao criar a CloudFormation pilha. Consulte [AWS Ground Station Imagens de máquinas da Amazon \(AMIs\)](#).

Os trechos de modelo restantes pertencem à seção Recursos do CloudFormation modelo.

Resources:

Resources that you would like to create should be placed within the resource section.

Dado nosso cenário para fornecer um único caminho de comunicação para uma EC2 instância, você terá um único caminho de entrega síncrono. De acordo com a [Entrega síncrona de dados](#) seção, você deve instalar e configurar uma EC2 instância da Amazon com um aplicativo de endpoint de fluxo de dados e criar um ou mais grupos de endpoints de fluxo de dados.

```
# The EC2 instance that will send/receive data to/from your satellite using AWS
Ground Station.
```

```
ReceiverInstance:
```

```
Type: AWS::EC2::Instance
```

```
Properties:
```

```
DisableApiTermination: false
```

```
IamInstanceProfile: !Ref GeneralInstanceProfile
```

```
ImageId: !Ref ReceiverAMI
```

```
InstanceType: m5.4xlarge
```

```
KeyName: !Ref EC2Key
```

```
Monitoring: true
```

```
PlacementGroupName: !Ref ClusterPlacementGroup
```

```

SecurityGroupIds:
  - Ref: InstanceSecurityGroup
SubnetId: !Ref ReceiverSubnet
BlockDeviceMappings:
  - DeviceName: /dev/xvda
    Ebs:
      VolumeType: gp2
      VolumeSize: 40
Tags:
  - Key: Name
    Value: !Join [ "-", [ "Receiver" , !Ref "AWS::StackName" ] ]
UserData:
  Fn::Base64:
    |
    #!/bin/bash
    exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console)
2>&1
    echo `date +%F %R:%S` ` "INFO: Logging Setup" >&2

    GROUND_STATION_DIR="/opt/aws/groundstation"
    GROUND_STATION_BIN_DIR="${GROUND_STATION_DIR}/bin"
    STREAM_CONFIG_PATH="${GROUND_STATION_DIR}/customer_stream_config.json"

    echo "Creating ${STREAM_CONFIG_PATH}"
    cat << STREAM_CONFIG > "${STREAM_CONFIG_PATH}"
    {
      "ddx_streams": [
        {
          "streamName": "Downlink",
          "maximumWanRate": 4000000000,
          "lanConfigDevice": "lo",
          "lanConfigPort": 50000,
          "wanConfigDevice": "eth1",
          "wanConfigPort": 55888,
          "isUplink": false
        }
      ]
    }
    STREAM_CONFIG

    echo "Waiting for dataflow endpoint application to start"
    while netstat -lnt | awk '$4 ~ /:80$/ {exit 1}'; do sleep 10; done

    echo "Configuring dataflow endpoint application streams"

```

```

python "${GROUND_STATION_BIN_DIR}/configure_streams.py" --configFileName
"${STREAM_CONFIG_PATH}"
    sleep 2
python "${GROUND_STATION_BIN_DIR}/save_default_config.py"

exit 0

```

```

# The AWS Ground Station Dataflow Endpoint Group that defines the endpoints that AWS
Ground
# Station will use to send/receive data to/from your satellite.
DataflowEndpointGroup:
  Type: AWS::GroundStation::DataflowEndpointGroup
  Properties:
    ContactPostPassDurationSeconds: 180
    ContactPrePassDurationSeconds: 120
    EndpointDetails:
      - Endpoint:
          Name: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ] # needs to
match DataflowEndpointConfig name
          Address:
            Name: !GetAtt ReceiverInstanceNetworkInterface.PrimaryPrivateIpAddress
            Port: 55888
    SecurityDetails:
      SecurityGroupIds:
        - Ref: "DataflowEndpointSecurityGroup"
      SubnetIds:
        - !Ref ReceiverSubnet
      RoleArn: !GetAtt DataDeliveryServiceRole.Arn

# The security group 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
```

Você também precisará das políticas, funções e perfis apropriados para permitir AWS Ground Station a criação de uma interface de rede elástica (ENI) em sua conta.

```
# 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 configurações

Esta seção representa [Crie configurações](#) o guia do usuário.

Você precisará de uma configuração de rastreamento para definir sua preferência de uso do autotrack. Selecionar PREFERRED como trilha automática pode melhorar a qualidade do sinal, mas não é necessário atender à qualidade do sinal devido à qualidade suficiente das efemérides JPSS-1.

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
    ConfigData:
      TrackingConfig:
        Autotrack: "PREFERRED"
```

Com base no caminho de comunicação, você precisará definir uma antenna-downlink-demod-decodeconfiguração para representar a parte do satélite, bem como uma configuração de endpoint de fluxo de dados para se referir ao grupo de endpoints de fluxo de dados que define os detalhes do endpoint.

Note

Para obter detalhes sobre como definir os valores para `DemodulationConfig`, `eDecodeConfig`, consulte [Configuração de decodificação de demodulação de downlink de antena](#).

```
# 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"
    }
},
    "NrzDecoder":{
        "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 perfil da missão

Esta seção representa [Criar perfil de missão](#) o guia do usuário.

Agora que você tem as configurações associadas, pode usá-las para criar o fluxo de dados. Você usará os padrões para os demais parâmetros.

```

# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnpjpsMissionProfile:
  Type: AWS::GroundStation::MissionProfile
  Properties:
    Name: "37849 SNPP And 43013 JPSS"

```

```
ContactPrePassDurationSeconds: 120
ContactPostPassDurationSeconds: 60
MinimumViableContactDurationSeconds: 180
TrackingConfigArn: !Ref TrackingConfig
DataflowEdges:
  - Source: !Join [ "/", [ !Ref JpssDownlinkDemodDecodeAntennaConfig,
    "UncodedFramesEgress" ] ]
    Destination: !Ref DownlinkDemodDecodeEndpointConfig
```

Juntando tudo

Com os recursos acima, agora você tem a capacidade de agendar contatos JPSS-1 para entrega síncrona de dados a partir de qualquer um dos seus contatos integrados. [AWS Ground Station Localizações](#)

A seguir está um CloudFormation modelo completo que inclui todos os recursos descritos nesta seção combinados em um único modelo que pode ser usado diretamente em CloudFormation.

O CloudFormation modelo nomeado foi `AquaSnppJpss.yml` projetado para fornecer acesso rápido para começar a receber dados dos satélites Aqua, SNPP e JPSS-1/NOAA-20. Ele contém uma EC2 instância da Amazon e os AWS Ground Station recursos necessários para agendar contatos e receber dados de transmissão direta demodulados e decodificados.

Se Aqua, SNPP, JPSS-1/NOAA-20 e Terra não estiverem integrados à sua conta, consulte. [Satélite a bordo](#)

Note

Você pode acessar o modelo acessando o bucket Amazon S3 de integração do cliente usando AWS credenciais válidas. Os links abaixo usam um bucket regional do Amazon S3. Altere o código da `us-west-2` região para representar a região correspondente na qual você deseja criar a CloudFormation pilha.

Além disso, as instruções a seguir usam YAML. No entanto, os modelos estão disponíveis no formato YAML ou JSON. Para usar o JSON, substitua a extensão do `.yml` arquivo por `.json` ao baixar o modelo.

Para baixar o modelo usando AWS CLI, use o seguinte comando:

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yml .
```

É possível visualizar e fazer download do modelo no console navegando até o seguinte URL no seu navegador:

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/AquaSnppJpss.yml
```

Você pode especificar o modelo diretamente CloudFormation usando o seguinte link:

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/AquaSnppJpss.yml
```

Quais recursos adicionais o modelo define?

O AquaSnppJpss modelo inclui os seguintes recursos adicionais:

- (Opcional) Acionadores de CloudWatch eventos - AWS Lambda Função que é acionada usando CloudWatch eventos enviados AWS Ground Station antes e depois de um contato. A AWS Lambda função iniciará e, opcionalmente, interromperá sua instância receptora.
- (Opcional) EC2 Verificação de contatos - A opção de usar o Lambda para configurar um sistema de verificação de suas EC2 instâncias da Amazon para contatos com notificação do SNS. É importante observar que isso pode incorrer em cobranças, dependendo do seu uso atual.
- Ground Station Amazon Machine Image Retrieval Lambda: a opção de selecionar qual software está instalado em sua instância e a AMI de sua escolha. As opções de software incluem DDX 2.6.2 Only e DDX 2.6.2 with qRadio 3.6.0. Se você quiser usar o DigiF Data Delivery de banda larga e o AWS Ground Station Agent, consulte. [Satélite de transmissão pública utilizando AWS Ground Station Agent \(banda larga\)](#) Essas opções continuarão a se expandir à medida que atualizações e recursos adicionais de software forem lançados.
- Perfis de missão adicionais - Perfis de missão para satélites de transmissão pública adicionais (Aqua, SNPP e Terra).
- Configurações adicionais de downlink de antena - Configurações de downlink de antena para satélites de transmissão pública adicionais (Aqua, SNPP e Terra).

Os valores e os parâmetros dos satélites nesse modelo já estão preenchidos. Esses parâmetros facilitam o uso AWS Ground Station imediato desses satélites. Você não precisa configurar seus

próprios valores para usá-los AWS Ground Station ao usar esse modelo. No entanto, é possível personalizar os valores para que o modelo funcione para seu caso de uso.

Onde recebo os meus dados?

O grupo de endpoints do fluxo de dados é configurado para usar a interface de rede da instância do receptor que parte do modelo cria. A instância receptora usa um aplicativo de endpoint de fluxo de dados para receber o stream de dados da AWS Ground Station porta definida pelo endpoint de fluxo de dados. Após serem recebidos, os dados estarão disponíveis para consumo por meio da porta UDP 50000 no adaptador de loopback da instância do receptor. Para obter mais informações sobre como configurar um grupo de endpoints de fluxo de dados, consulte.

[AWS::GroundStation::DataflowEndpointGroup](#)

Satélite de transmissão pública utilizando AWS Ground Station Agent (banda larga)

Este exemplo se baseia na análise feita na [JPSS-1 - Satélite de transmissão pública \(PBS\) - Avaliação](#) seção do guia do usuário.

Para concluir este exemplo, você precisará assumir um cenário: capturar o caminho de comunicação do HRD como frequência intermediária digital de banda larga (DigIF) e processá-lo conforme recebido pelo agente AWS Ground Station em uma instância da Amazon EC2 usando um SDR.

Note

O sinal real do caminho de comunicação JPSS HRD tem uma largura de banda de 30 MHz, mas você configurará a configuração de downlink da antena para tratá-lo como um sinal com uma MHz largura de banda de 100% para que ele possa fluir pelo caminho correto a ser recebido pelo Agente neste exemplo. AWS Ground Station

Caminhos de comunicação

Esta seção representa [Planeje seus caminhos de comunicação de fluxo de dados](#) como começar. Neste exemplo, você precisará de uma seção adicional em seu CloudFormation modelo que não tenha sido usada nos outros exemplos, a seção Mapeamentos.

Note

Para obter mais informações sobre o conteúdo de um CloudFormation modelo, consulte [as seções Modelo](#).

Você começará configurando uma seção de mapeamentos em seu CloudFormation modelo para as listas de AWS Ground Station prefixos por região. Isso permite que as listas de prefixos sejam facilmente referenciadas pelo grupo de segurança de EC2 instâncias da Amazon. Para obter mais informações sobre como usar uma lista de prefixos, consulte [Configuração de VPC com agente AWS Ground Station](#).

Mappings:**PrefixListId:****us-east-2:**

groundstation: pl-087f83ba4f34e3bea

us-west-2:

groundstation: pl-0cc36273da754ebdc

us-east-1:

groundstation: pl-0e5696d987d033653

eu-central-1:

groundstation: pl-03743f81267c0a85e

sa-east-1:

groundstation: pl-098248765e9effc20

ap-northeast-2:

groundstation: pl-059b3e0b02af70e4d

ap-southeast-1:

groundstation: pl-0d9b804fe014a6a99

ap-southeast-2:

groundstation: pl-08d24302b8c4d2b73

me-south-1:

groundstation: pl-02781422c4c792145

eu-west-1:

groundstation: pl-03fa6b266557b0d4f

eu-north-1:

groundstation: pl-033e44023025215c0

af-south-1:

groundstation: pl-0382d923a9d555425

Para a seção Parâmetros, você adicionará os seguintes parâmetros. Você especificará valores para eles ao criar a pilha por meio do CloudFormation console.

Parameters:**EC2Key:**

Description: The SSH key used to access the EC2 receiver instance. Choose any SSH key if you are not creating an EC2 receiver instance. For instructions on how to create an SSH key see <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html>

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

AZ:

Description: "The AvailabilityZone that the resources of this stack will be created in. (e.g. us-east-2a)"

Type: AWS::EC2::AvailabilityZone::Name

ReceiverAMI:

Description: The Ground Station Agent AMI ID you want to use. Please note that AMIs are region specific. For instructions on how to retrieve an AMI see <https://docs.aws.amazon.com/ground-station/latest/ug/dataflows.ec2-configuration.html#dataflows.ec2-configuration.amis>

Type: AWS::EC2::Image::Id

Note

Você precisa criar um par de chaves e fornecer o nome para o EC2 EC2Key parâmetro da Amazon. Consulte [Criar um par de chaves para sua EC2 instância da Amazon](#). Além disso, você precisará fornecer a ID de AMI específica da região correta ao criar a CloudFormation pilha. Consulte [AWS Ground Station Imagens de máquinas da Amazon \(AMIs\)](#).

Os trechos de modelo restantes pertencem à seção Recursos do CloudFormation modelo.

Resources:

Resources that you would like to create should be placed within the Resources section.

Dado nosso cenário para fornecer um único caminho de comunicação para uma EC2 instância da Amazon, você sabe que terá um único caminho de entrega síncrona. De acordo com a [Entrega síncrona de dados](#) seção, você deve instalar e configurar uma EC2 instância da Amazon com o AWS Ground Station Agent e criar um ou mais grupos de endpoints de fluxo de dados. Você começará configurando primeiro a Amazon VPC para o AWS Ground Station agente.

ReceiverVPC:

Type: AWS::EC2::VPC

Properties:

EnableDnsSupport: 'true'

EnableDnsHostnames: 'true'

CidrBlock: 10.0.0.0/16

Tags:

- Key: "Name"

Value: "AWS Ground Station Example - PBS to AWS Ground Station Agent VPC"

- Key: "Description"

Value: "VPC for EC2 instance receiving AWS Ground Station data"

PublicSubnet:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref ReceiverVPC

MapPublicIpOnLaunch: 'true'

AvailabilityZone: !Ref AZ

CidrBlock: 10.0.0.0/20

Tags:

- Key: "Name"

Value: "AWS Ground Station Example - PBS to AWS Ground Station Agent Public

Subnet"

- Key: "Description"

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

RouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref ReceiverVPC

Tags:

- Key: Name

Value: AWS Ground Station Example - RouteTable

RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

```
RouteTableId: !Ref RouteTable
SubnetId: !Ref PublicSubnet
```

Route:

```
Type: AWS::EC2::Route
DependsOn: InternetGateway
Properties:
  RouteTableId: !Ref RouteTable
  DestinationCidrBlock: '0.0.0.0/0'
  GatewayId: !Ref InternetGateway
```

InternetGateway:

```
Type: AWS::EC2::InternetGateway
Properties:
  Tags:
    - Key: Name
      Value: AWS Ground Station Example - Internet Gateway
```

GatewayAttachment:

```
Type: AWS::EC2::VPCGatewayAttachment
Properties:
  VpcId: !Ref ReceiverVPC
  InternetGatewayId: !Ref InternetGateway
```

Note

Para obter mais informações sobre as configurações de VPC suportadas pelo AWS Ground Station agente, consulte Requisitos do [AWS Ground Station agente - diagramas de VPC](#).

Em seguida, você configurará a EC2 instância Receiver Amazon.

```
# 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

```

Você também precisará das políticas, funções e perfis apropriados para permitir AWS Ground Station a criação da interface de rede elástica (ENI) em sua conta.

```

# 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 configurações

Esta seção representa [Crie configurações](#) como começar.

Você precisará de uma configuração de rastreamento para definir sua preferência de uso do autotrack. Selecionar PREFERRED como trilha automática pode melhorar a qualidade do sinal, mas não é necessário atender à qualidade do sinal devido à qualidade suficiente das efemérides JPSS-1.

```
TrackingConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "JPSS Tracking Config"
  ConfigData:
    TrackingConfig:
      Autotrack: "PREFERRED"
```

Com base no caminho de comunicação, você precisará definir uma configuração de antena-downlink para representar a parte do satélite, bem como uma configuração de endpoint de fluxo de dados para se referir ao grupo de endpoints de fluxo de dados que define os detalhes do endpoint.

```
# The AWS Ground Station Antenna Downlink Config that defines the frequency spectrum
used to
# downlink data from your satellite.
SnpJPSSDownlinkDigIfAntennaConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "SNPP JPSS Downlink WBDigIF Antenna Config"
  ConfigData:
    AntennaDownlinkConfig:
      SpectrumConfig:
        Bandwidth:
          Units: "MHz"
          Value: 100
        CenterFrequency:
          Units: "MHz"
          Value: 7812
        Polarization: "RIGHT_HAND"
```

```
# The AWS Ground Station Dataflow Endpoint Config that defines the endpoint used to
downlink data
# from your satellite.
DownlinkDigIfEndpointConfig:
  Type: AWS::GroundStation::Config
  Properties:
    Name: "Aqua SNPP JPSS Terra Downlink DigIF Endpoint Config"
    ConfigData:
      DataflowEndpointConfig:
        DataflowEndpointName: !Join [ "-", [ !Ref "AWS::StackName" , "Downlink" ] ]
        DataflowEndpointRegion: !Ref AWS::Region
```

AWS Ground Station perfil da missão

Esta seção representa [Criar perfil de missão](#) como começar.

Agora que você tem as configurações associadas, pode usá-las para criar o fluxo de dados. Você usará os padrões para os demais parâmetros.

```
# The AWS Ground Station Mission Profile that groups the above configurations to
define how to
# uplink and downlink data to your satellite.
SnpjpsMissionProfile:
  Type: AWS::GroundStation::MissionProfile
  Properties:
    Name: !Sub 'JPSS WBDigIF gs-agent EC2 Delivery'
    ContactPrePassDurationSeconds: 120
    ContactPostPassDurationSeconds: 120
    MinimumViableContactDurationSeconds: 180
    TrackingConfigArn: !Ref TrackingConfig
    DataflowEdges:
      - Source: !Ref SnpjpsDownlinkDigIfAntennaConfig
        Destination: !Ref DownlinkDigIfEndpointConfig
    StreamsKmsKey:
      KmsKeyArn: !GetAtt GroundStationDataDeliveryKmsKey.Arn
    StreamsKmsRole: !GetAtt GroundStationKmsKeyRole.Arn
```

Juntando tudo

Com os recursos acima, agora você tem a capacidade de agendar contatos JPSS-1 para entrega síncrona de dados a partir de qualquer um dos seus contatos integrados. [AWS Ground Station Localizações](#)

A seguir está um CloudFormation modelo completo que inclui todos os recursos descritos nesta seção combinados em um único modelo que pode ser usado diretamente em CloudFormation.

O CloudFormation modelo nomeado foi `DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml` projetado para fornecer acesso rápido para começar a receber dados de frequência intermediária digitalizada (DigIF) para os satélites Aqua, SNPP, JPSS-1/NOAA-20 e Terra. Ele contém uma EC2 instância da Amazon e os CloudFormation recursos necessários para receber dados brutos de transmissão direta do DigIF usando o AWS Ground Station Agent.

Se Aqua, SNPP, JPSS-1/NOAA-20 e Terra não estiverem integrados à sua conta, consulte. [Satélite a bordo](#)

Note

Você pode acessar o modelo acessando o bucket Amazon S3 de integração do cliente usando AWS credenciais válidas. Os links abaixo usam um bucket regional do Amazon S3. Altere o código da `us-west-2` região para representar a região correspondente na qual você deseja criar a CloudFormation pilha.

Além disso, as instruções a seguir usam YAML. No entanto, os modelos estão disponíveis no formato YAML ou JSON. Para usar o JSON, substitua a extensão do `.yml` arquivo por `.json` ao baixar o modelo.

Para baixar o modelo usando AWS CLI, use o seguinte comando:

```
aws s3 cp s3://groundstation-cloudformation-templates-us-west-2/agent/ec2_delivery/
DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml .
```

É possível visualizar e fazer download do modelo no console navegando até o seguinte URL no seu navegador:

```
https://s3.console.aws.amazon.com/s3/object/groundstation-cloudformation-templates-us-west-2/agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

Você pode especificar o modelo diretamente CloudFormation usando o seguinte link:

```
https://groundstation-cloudformation-templates-us-west-2.s3.us-west-2.amazonaws.com/agent/ec2_delivery/DirectBroadcastSatelliteWbDigIfEc2DataDelivery.yml
```

Quais recursos adicionais o modelo define?

O `DirectBroadcastSatelliteWbDigIfEc2DataDelivery` modelo inclui os seguintes recursos adicionais:

- Interface de rede elástica da instância do receptor - (condicional) Uma interface de rede elástica é criada na sub-rede especificada por, `PublicSubnetId` fornecida. Isso é necessário se a instância do receptor estiver em uma sub-rede privada. A interface de rede elástica será associada ao EIP e anexada à instância do receptor.
- IP elástico da instância do receptor - Um IP elástico que se AWS Ground Station conectará a. Isso se conecta à instância do receptor ou à interface de rede elástica.
- Uma das seguintes associações de IP elástico:
 - Associação entre instância do receptor e IP elástico - A associação do IP elástico à sua instância do receptor, se não `PublicSubnetId` for especificada. Isso requer essa `SubnetId` referência a uma sub-rede pública.
 - Interface de rede elástica da instância receptora com associação de IP elástico - A associação do IP elástico à interface de rede elástica da instância receptora, se `PublicSubnetId` for especificada.
- (Opcional) Acionadores de CloudWatch eventos - AWS Lambda Função que é acionada usando CloudWatch eventos enviados AWS Ground Station antes e depois de um contato. A AWS Lambda função iniciará e, opcionalmente, interromperá sua instância receptora.
- (Opcional) Amazon EC2 Verification for Contacts - A opção de usar o Lambda para configurar um sistema de verificação de suas EC2 instâncias da Amazon para contatos com notificação do SNS. É importante observar que isso pode incorrer em cobranças, dependendo do seu uso atual.
- Perfis de missão adicionais - Perfis de missão para satélites de transmissão pública adicionais (Aqua, SNPP e Terra).
- Configurações adicionais de downlink de antena - Configurações de downlink de antena para satélites de transmissão pública adicionais (Aqua, SNPP e Terra).

Os valores e os parâmetros dos satélites nesse modelo já estão preenchidos. Esses parâmetros facilitam o uso AWS Ground Station imediato desses satélites. Você não precisa configurar seus próprios valores para usá-los AWS Ground Station ao usar esse modelo. No entanto, é possível personalizar os valores para que o modelo funcione para seu caso de uso.

Onde recebo os meus dados?

O grupo de endpoints do fluxo de dados é configurado para usar a interface de rede da instância do receptor que parte do modelo cria. A instância receptora usa o AWS Ground Station Agente para receber o fluxo de dados da AWS Ground Station porta definida pelo endpoint do fluxo de dados. Para obter mais informações sobre como configurar um grupo de endpoints de fluxo de dados, consulte [AWS::GroundStation::DataflowEndpointGroup](#) Para obter mais informações sobre o AWS Ground Station agente, consulte [O que é o AWS Ground Station agente?](#)

Solução de problemas

A documentação a seguir pode ajudá-lo a solucionar problemas que podem ocorrer durante o uso AWS Ground Station.

Tópicos

- [Solucionar problemas de contatos que entregam dados para o Amazon EC2](#)
- [Solucionar problemas de contatos com FALHA](#)
- [Solucionar problemas nas atualizações de contatos](#)
- [Solucionar problemas de contatos FAILED_TO_SCHEDULE](#)
- [Solucione o problema que DataflowEndpointGroups não está em um estado SAUDÁVEL](#)
- [Solucionar problemas de efemérides inválidas](#)
- [Solucionar problemas de contatos que não receberam dados](#)
- [Solucionar problemas de telemetria](#)

Solucionar problemas de contatos que entregam dados para o Amazon EC2

Se você não conseguir concluir um AWS Ground Station contato com sucesso, precisará verificar se sua instância do Amazon EC2 está em execução, verificar se seu aplicativo de endpoint de fluxo de dados está em execução e verificar se o stream do seu aplicativo de endpoint de fluxo de dados está configurado corretamente.

Note

DataDefender (DDX) é um exemplo de um aplicativo de endpoint de fluxo de dados atualmente suportado pelo AWS Ground Station

Pré-requisito

Os procedimentos a seguir presumem que já há uma instância do Amazon EC2 configurada. Para configurar uma instância do Amazon EC2 em AWS Ground Station, consulte [Getting Started](#).

Etapa 1: verificar se sua instância do EC2 está em execução

O procedimento a seguir mostra como encontrar sua instância do Amazon EC2 no console e iniciá-la se ela não estiver em execução.

1. Localize a instância do Amazon EC2 que foi usada para o contato que você está solucionando. Use as seguintes etapas:
 - a. Em seu CloudFormationpainel, selecione a pilha que contém sua instância do Amazon EC2.
 - b. Escolha a guia Recursos e localize a instância do Amazon EC2 na coluna ID lógico. Verifique se a instância foi criada na coluna Status.
 - c. Na coluna ID físico, selecione o link para a instância do Amazon EC2. Isso levará você ao console de gerenciamento do Amazon EC2.
2. No console de gerenciamento do Amazon EC2, certifique-se de que seu Estado da instância do Amazon EC2 esteja em execução.
3. Se a instância estiver sendo executada, siga para a próxima etapa. Se a instância não estiver em execução, inicie-a usando a seguinte etapa:
 - Com a instância do Amazon EC2 selecionada, escolha Ações > Estado da instância > Iniciar.

Etapa 2: Determinar o tipo de aplicativo de fluxo de dados usado


Se você estiver usando o AWS Ground Station Agente para entrega de dados, redirecione para a seção [AWS Ground Station Agente de Solução de Problemas](#). Caso contrário, se você estiver usando o aplicativo DataDefender (DDX), continue [the section called “Etapa 3: verificar se o aplicativo de fluxo de dados está em execução”](#) usando.

Etapa 3: verificar se o aplicativo de fluxo de dados está em execução

A verificação do status de DataDefender exige que você se conecte à sua instância no Amazon EC2. Para obter mais detalhes sobre como se conectar à sua instância, consulte [Conecte-se à sua instância Linux](#).

O procedimento a seguir fornece etapas de solução de problemas usando comandos em um cliente SSH.

1. Abra um terminal ou prompt de comando e conecte a instância do Amazon EC2 usando o SSH. Encaminhe a porta 80 do host remoto para visualizar a interface do usuário DataDefender da web. Os comandos a seguir demonstram como usar o SSH para conectar a uma instância do Amazon EC2 por meio de um bastion com encaminhamento de porta habilitado.

 Note

Substitua <SSH KEY>, <BASTION HOST> e <HOST> pela chave ssh específica, nome do bastion host e nome do host da instância do Amazon EC2.

No 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>
```

No 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. Verifique se DataDefender (também chamado de DDX) está em execução fazendo grepping (verificando) um processo em execução chamado ddx na saída. O comando para grepping (verificação) para um processo em execução e um exemplo de saída bem-sucedida é fornecido a seguir.

```
[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
```

Se DataDefender estiver em execução, vá para [the section called “Etapa 4: verificar se o stream do aplicativo de fluxo de dados está configurado”](#) Caso contrário, continue para a próxima etapa.

3. Comece a DataDefender usar o comando show abaixo.

```
sudo service rtlogic-ddx start
```

Se DataDefender estiver em execução depois de usar o comando, vá para [the section called “Etapa 4: verificar se o stream do aplicativo de fluxo de dados está configurado”](#) Caso contrário, continue para a próxima etapa.

4. Inspecione os arquivos a seguir usando os comandos abaixo para ver se houve algum erro durante a instalação e configuração DataDefender.

```
cat /var/log/user-data.log
cat /opt/aws/groundstation/.startup.out
```

Note

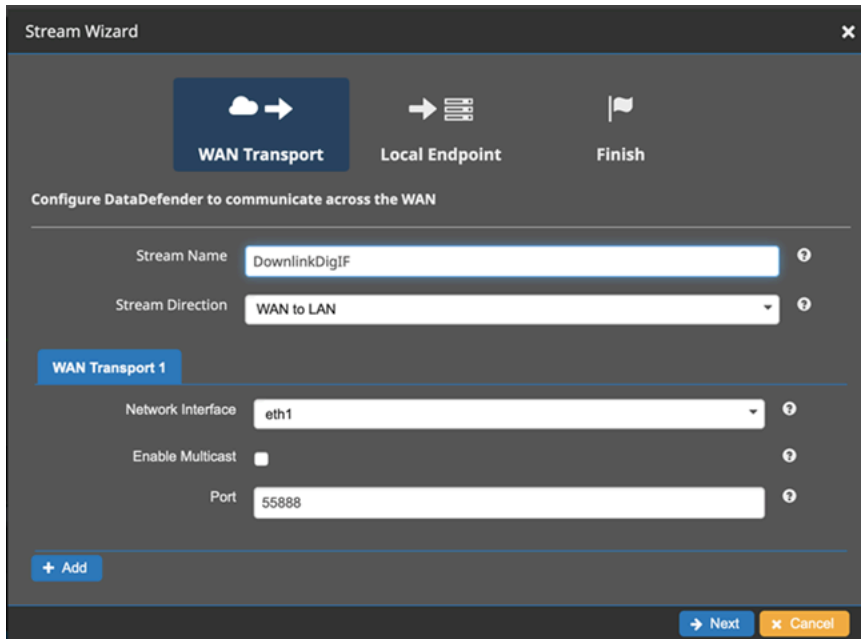
Um problema comum descoberto ao inspecionar esses arquivos é que a instância do Amazon EC2 está executando não tem acesso ao Amazon S3 para fazer download dos arquivos de instalação. Caso descubra em seus registros que o problema é esse, verifique as configurações da Amazon VPC e do grupo de segurança da instância do EC2 para garantir que não estejam bloqueando o acesso ao Amazon S3.

Se DataDefender estiver em execução após verificar as configurações da Amazon VPC, continue. [the section called “Etapa 4: verificar se o stream do aplicativo de fluxo de dados está configurado”](#) Se o problema persistir, [entre em contato com o AWS Support](#) e envie os arquivos de log com uma descrição do problema.

Etapa 4: verificar se o stream do aplicativo de fluxo de dados está configurado

1. Em um navegador da Web, acesse sua interface de usuário DataDefender da Web inserindo o seguinte endereço na barra de endereço: localhost:8080. Depois, pressione Enter.
2. No DataDefenderpainel, escolha Ir para detalhes.
3. Selecione o fluxo na lista de fluxos e selecione Editar fluxo.
4. Na caixa de diálogo Assistente de fluxo, faça o seguinte:

- a. No painel Transporte de WAN, certifique-se de que WAN para LAN está selecionado para a Direção do fluxo.
- b. Na caixa Porta, certifique-se de que a porta WAN selecionada para o grupo de endpoints do fluxo de dados esteja presente. Por padrão, essa porta é 55888. Em seguida, escolha Próximo.



The screenshot shows the 'Stream Wizard' window with three tabs: 'WAN Transport', 'Local Endpoint', and 'Finish'. The 'WAN Transport' tab is active. Below the tabs, the text reads 'Configure DataDefender to communicate across the WAN'. The configuration fields are as follows:

- Stream Name: DownlinkDigIF
- Stream Direction: WAN to LAN
- WAN Transport 1:
 - Network Interface: eth1
 - Enable Multicast:
 - Port: 55888

At the bottom, there is a '+ Add' button and 'Next' and 'Cancel' buttons.

- c. No painel Endpoint local, certifique-se de que uma porta válida esteja presente na caixa Porta. Por padrão, essa porta é 50000. Essa é a porta na qual você receberá seus dados depois DataDefender de recebê-los do AWS Ground Station serviço. Em seguida, escolha Próximo.

The screenshot shows the 'Stream Wizard' interface with the 'Local Endpoint' step selected. The configuration for 'Local Endpoint 1' is as follows:

- Network Interface: lo
- Protocol: UDP
- Enable Multicast:
- Local Consumer: 127.0.0.1
- Port: 50000

Navigation buttons at the bottom include 'Previous', 'Next', and 'Cancel'.

- d. Selecione Concluir no menu restante se tiver alterado qualquer valor. Caso contrário, é possível cancelar no menu Assistente de fluxo.

Agora você garantiu que sua instância do Amazon EC2 está sendo executada DataDefender e configurada adequadamente para receber dados do. AWS Ground Station Avance para [the section called “Etapa 5: Certifique-se de ter endereços IP disponíveis suficientes na sub-rede da \(s\) instância \(s\) do receptor”](#).

Etapa 5: Certifique-se de ter endereços IP disponíveis suficientes na sub-rede da (s) instância (s) do receptor

O procedimento a seguir mostra como encontrar o número de endereços IP disponíveis em uma instância do receptor Amazon EC2 no console.

1. Para cada instância de receptor do Amazon EC2 usada para o contato, você está solucionando problemas. Use as seguintes etapas:
 - a. Em seu CloudFormationpainel, selecione a pilha que contém sua instância do Amazon EC2.
 - b. Escolha a guia Recursos e localize a instância do Amazon EC2 na coluna ID lógico. Verifique se a instância foi criada na coluna Status.
 - c. Na coluna ID físico, selecione o link para a instância do Amazon EC2. Isso levará você ao console de gerenciamento do Amazon EC2.

2. No console de gerenciamento do Amazon EC2, localize e clique no link do ID da sub-rede no resumo da instância do receptor do Amazon EC2. Isso levará você ao console de gerenciamento correspondente da Amazon VPC.
3. Selecione a sub-rede correspondente no console de gerenciamento da Amazon VPC e verifique os detalhes da sua sub-rede para ver os endereços disponíveis. IPv4 Se esse número não for pelo menos igual aos endpoints de fluxo de dados que usam essa instância receptora do Amazon EC2, faça o seguinte:
 - a. Atualize a sub-rede correspondente do seu CloudFormation modelo CidrBlock para que seja dimensionada corretamente. Para obter mais detalhes sobre o tamanho da sub-rede, consulte Blocos CIDR de [sub-rede](#).
 - b. Reimplante sua pilha com seu modelo atualizado CloudFormation .

Se você continuar enfrentando problemas, [entre em contato com o AWS Support](#).

Solucionar problemas de contatos com FALHA

Um contato terá o status de contato do terminal FALHOU quando AWS Ground Station detectar um problema com a configuração do seu recurso. Os casos de uso comuns que podem causar o status FAILED nos contatos são fornecidos abaixo, junto com as etapas para ajudar a solucionar problemas.

Note

Este guia é especificamente para o status de contato FAILED e não se destina a outros status de falha, como AWS_FAILED, AWS_CANCELLED ou FAILED_TO_SCHEDULE. Consulte mais informações sobre os status de contato em [the section called “AWS Ground Station status de contato”](#)

Casos de uso FALHADOS do endpoint do Dataflow

Veja a seguir uma lista de casos de uso comuns que podem resultar em um status de contato FALHADO para fluxos de dados baseados em endpoints de fluxo de dados:

- O endpoint do Dataflow nunca se conecta — a conexão entre a AWS Ground Station Antenna e seu grupo de endpoints do Dataflow para um ou mais fluxos de dados nunca foi estabelecida.

- O endpoint do Dataflow se conecta tardiamente — a conexão entre a AWS Ground Station Antenna e seu grupo de endpoints do Dataflow para um ou mais fluxos de dados foi estabelecida após o horário de início do contato.
- A sub-rede do endpoint do Dataflow está sem endereços IP disponíveis. A solução de entrega de dados AWS Ground Station da Dataflow não consegue criar uma ENI na sua rede privada porque não tem nenhum endereço IP disponível na sub-rede da instância receptora.
- A sub-rede do endpoint do Dataflow é AWS Ground Station inválida - a solução de entrega de dados da Dataflow não consegue criar uma ENI na sua rede privada devido à incapacidade de acessar a sub-rede fornecida especificada no Dataflow Endpoint Group.

Para qualquer caso de falha de endpoint de fluxo de dados, é recomendável examinar o seguinte:

- Confirme se a instância do Amazon EC2 do receptor foi iniciada com sucesso, antes da hora de início do contato.
- Confirme se o software do endpoint de fluxo de dados estava funcionando durante o contato.
- Verifique se você tem pelo menos um endereço IP disponível por endpoint de fluxo de dados por sub-rede de instância receptora.
- Garanta que as sub-redes associadas ao seu grupo de endpoints do Dataflow, por meio de fluxos de dados configurados em [Configurar e configurar a Amazon VPC](#), permaneçam ativas e disponíveis para AWS Ground Station

Consulte etapas de solução de problemas mais específicas na seção sobre [Solucionar problemas de contatos que entregam dados para o Amazon EC2](#).

AWS Ground Station Casos de uso com FALHA do agente

Veja a seguir a lista de casos de uso comuns que podem resultar em um status de contato FAILED para fluxos de dados baseados no agente:

- AWS Ground Station Status do agente nunca reportado — O agente responsável por orquestrar a entrega de dados em seu grupo de endpoints do Dataflow para um ou mais fluxos de dados nunca reportou o status com êxito. AWS Ground Station Essa atualização de status deve ocorrer alguns segundos após a hora de término do contato.
- AWS Ground Station Agente iniciado tarde - O agente responsável por orquestrar a entrega de dados em seu grupo de endpoints do Dataflow para um ou mais fluxos de dados foi iniciado tarde, após o horário de início do contato.

Para qualquer caso de falha no fluxo de dados do AWS Ground Station Agente, é recomendável examinar o seguinte:

- Confirme se a instância do Amazon EC2 do receptor foi iniciada com sucesso, antes da hora de início do contato.
- Confirme se a aplicação do agente estava funcionando no início e durante o contato.
- Confirme se a aplicação do agente e a instância do Amazon EC2 não foram encerradas em até 15 segundos após o término do contato. Isso dá ao agente tempo suficiente para informar o status ao AWS Ground Station.

Consulte etapas de solução de problemas mais específicas na seção sobre [Solucionar problemas de contatos que entregam dados para o Amazon EC2](#).

Solucionar problemas nas atualizações de contatos

Quando você chama a [UpdateContact](#) API, AWS Ground Station executa a validação síncrona na solicitação. Se a validação for aprovada, a atualização será processada de forma assíncrona para propagar as alterações na região da antena. Os erros de validação síncrona são retornados diretamente na resposta HTTP. Falhas assíncronas são `failureCodes` relatadas por meio de `failureMessage` campos na versão do contato, que você pode ver [DescribeContactVersion](#) chamando a versão que não foi atualizada.

Para obter mais informações sobre controle de versão de contatos, consulte [Atualizar contatos e controle de versão de contatos](#)

Erros de validação síncrona

Os erros a seguir são retornados diretamente na resposta HTTP quando a [UpdateContact](#) solicitação falha na validação.

ResourceNotFoundException: Contato não encontrado

Causa comum

O especificado `contactId` não existe ou pertence a uma AWS conta diferente.

Resolução

1. Verifique se o `contactId` está correto.

2. Confirme se você está usando as credenciais da AWS conta proprietária do contato.
3. Use [ListContacts](#) para encontrar o correto `contactId`.

ConflictException: Não é possível atualizar o contato

Causa comum

O contato está em um estado que não permite atualizações. A [UpdateContact](#) API só pode ser chamada quando o contato está no PASS estado SCHEDULEDPREPASS, ou. Esse erro também ocorre se outra atualização já estiver em andamento (a versão mais recente do contato está no UPDATING estado).

Resolução

1. Ligue [DescribeContact](#) para verificar o status atual do contato.
2. Se o contato estiver em um estado terminal (por exemplo,, COMPLETEDFAILED, ouCANCELLED), ele não poderá ser atualizado. Um contato só pode ser atualizado quando está no PASS estado SCHEDULEDPREPASS, ou. Para obter uma lista completa dos estados dos terminais, consulte [AWS Ground Station status de contato](#).
3. Se outra atualização estiver em andamento, aguarde até que a atualização atual alcance ACTIVE ou o FAILED_TO_UPDATE status antes de enviar outra atualização. Você pode pesquisar a [DescribeContactVersion](#) API ou usar os utilitários de conveniência do ContactUpdated garçom fornecidos por some e o. AWS SDKs AWS Command Line Interface

InvalidParameterException: parâmetros de solicitação inválidos

Causa comum

A solicitação contém parâmetros inválidos. As causas comuns incluem:

- Ausente ou vazio `clientToken`.
- Vários tipos de `ProgramTrackSettings` (azimute/elevação, OEM e TLE) incluídos em uma única solicitação. Somente um tipo é permitido por solicitação.
- `satelliteArn` Definir como nulo sem ser aprovado para [efemérides de elevação de azimute na estação terrestre do contato](#).
- Falta `AzElProgramTrackSettings` quando `satelliteArn` é nulo.

- Fornecendo um `ephemerisId` que não esteja associado ao especificado `satelliteArn`.
- O satélite não tem uma janela de visibilidade válida da estação terrestre para o intervalo de tempo de contato.
- O satélite não está embarcado na estação terrestre ou não tem o licenciamento exigido pelo perfil da missão.
- O perfil da missão inclui [Configuração de decodificação de demodulação de downlink de antena](#) configurações que não são compatíveis com atualizações de contatos.

Resolução

1. Examine a mensagem de erro na resposta para obter detalhes sobre qual parâmetro é inválido.
2. Certifique-se de fornecer exatamente um tipo de solicitação `ProgramTrackSettings` por solicitação.
3. Se estiver usando ângulos de azimuth/elevation apontamento sem `umsatelliteArn`, confirme se sua conta foi aprovada para esse recurso na estação terrestre. Para obter mais informações, consulte [Forneça dados de efemérides de elevação de azimuth](#).
4. Verifique se as efemérides referenciadas estão associadas ao satélite correto e cobrem o intervalo de tempo de contato.

ResourceLimitExceededException: limite máximo de versão atingido

Causa comum

O contato atingiu o número máximo de versões (128). Cada chamada para [UpdateContact](#) cria uma nova versão, e um contato não pode exceder esse limite.

Resolução

1. Este limite não pode ser aumentado. Se precisar fazer mais alterações, cancele o contato e reserve um novo.

Códigos de falha assíncronos

Os códigos de falha a seguir aparecem no `failureCodes` campo de uma versão de contato com um `FAILED_TO_UPDATE` status. Use [DescribeContactVersion](#) para recuperar esses detalhes. O `failureMessage` campo fornece contexto adicional sobre a falha.

Código de falha	Causa comum	Resolução
INTERNAL_ERROR	Ocorreu um erro interno inesperado ao processar a atualização.	Tente fazer a atualização novamente. Se o problema persistir, entre em contato com o AWS Support .
INVALID_SATELLITE_ARN	O ARN do satélite fornecido na solicitação de atualização não é válido ou não existe.	Verifique o ARN do satélite e confirme se o satélite está registrado em sua conta.
INVALID_UPDATE_CONTACT_REQUEST	A solicitação de atualização contém parâmetros inválidos que não foram detectados durante a validação síncrona.	Revise <code>failureMessage</code> os detalhes e corrija os parâmetros da solicitação.
EPHEMERIS_NOT_FOUND	As efemérides referenciadas nas substituições de rastreamento não existem.	Verifique <code>ephemerisId</code> e confirme se as efemérides não foram excluídas.
EPHEMERIS_TIME_RANGE_INVALID	As efemérides não cobrem o intervalo de tempo do contato.	Faça o upload de uma nova efeméride que cubra todo o intervalo de tempo de contato. Se o intervalo de tempo das efemérides não puder ser estendido, cancele o contato e reserve um novo durante o período das efemérides. Para obter mais informações, consulte Forneça dados de efemérides personalizados .
EPHEMERIS_NOT_ENABLED	A efeméride referenciada não está em um estado. ENABLED	Verifique o status das efemérides e ative-o antes de tentar novamente a atualização.

Código de falha	Causa comum	Resolução
SATELLITE _DOES_NOT _MATCH_EPHEMERIS	A efeméride não está associada ao satélite especificado na solicitação de atualização.	Certifique-se de que <code>ephemerisId</code> pertence ao satélite especificado em <code>satelliteArn</code> .
NOT_ONBOARD ED_TO_AZ EL_EPHEMERIS	Sua conta não foi aprovada para usar dados de efemérides de elevação de azimute na estação terrestre do contato. A efeméride de elevação de azimute é um recurso restrito disponível para um número limitado de casos de uso especializados.	Se efemérides de elevação de azimute forem necessárias para seu caso de uso, abra um AWS Support ticket por meio do AWS Support Center Console para solicitar acesso. Como alternativa, considere usar dados de efemérides TLE ou dados de efemérides de OEM, se eles se adequarem ao seu caso de uso.
AZEL_EPHEMERIS_NOT_FOUND	As efemérides de elevação de azimute referenciadas na solicitação não existem.	Verifique <code>ephemerisId</code> e confirme se as efemérides de elevação de azimute não foram excluídas.
AZEL_EPHEMERIS_WRONG_GROUND_STATION	A efeméride de elevação de azimute foi criada para uma estação terrestre diferente daquela usada pelo contato.	Faça upload de uma nova efeméride de elevação de azimute para a estação terrestre correta ou use uma efeméride existente que corresponda à estação terrestre do contato.
AZEL_EPHEMERIS_INVALID_STATUS	As efemérides de elevação de azimute não estão em um estado válido para uso.	Verifique o status das efemérides. Deve estar em um ENABLED estado. Se a efeméride falhar na validação, faça o upload de uma versão corrigida.

Código de falha	Causa comum	Resolução
AZEL_EPHE MERIS_TIM E_RANGE_INVALID	As efemérides de elevação do azimute não cobrem o intervalo de tempo do contato.	Faça o upload de uma nova efeméride de elevação de azimute que cubra toda a faixa de tempo de contato. Se o intervalo de tempo das efemérides não puder ser estendido, cancele o contato e reserve um novo durante o período das efemérides.

Verificando o status de uma atualização

Depois de ligar `UpdateContact`, a nova versão do contato começa no `UPDATING` estado. Durante esse período, `DescribeContact` continua retornando a versão anteriormente ativa do contato. A nova versão não aparece `DescribeContact` até ser propagada para a antena e atingir `ACTIVE` o status. Para verificar o status de uma versão específica, use [DescribeContactVersion](#).

Para determinar se uma atualização foi bem-sucedida ou falhou:

1. Ligue [DescribeContactVersion](#) com o `contactId` e `versionId` retorne pela `UpdateContact` resposta.
2. Verifique o `version.status` campo. Um status de `ACTIVE` significa que a atualização foi aplicada com sucesso. Um status de `FAILED_TO_UPDATE` significa que a atualização falhou.
3. Se o status for `FAILED_TO_UPDATE`, verifique os `version.failureMessage` campos `version.failureCodes` e para obter detalhes sobre o que deu errado.

Tip

Alguns AWS SDKs e outros AWS Command Line Interface apoiam um `ContactUpdated` garçom que faz pesquisas automaticamente `DescribeContactVersion` até que a versão alcance `ACTIVE` nosso status. `FAILED_TO_UPDATE` Por exemplo, ele AWS Command Line Interface fornece um comando [aws groundstation wait contact-updated](#). Use o garçom em vez de implementar sua própria lógica de pesquisa.

Solucionar problemas de contatos FAILED_TO_SCHEDULE

Um contato terminará em um estado FAILED_TO_SCHEDULE quando AWS Ground Station detectar um problema na configuração do seu recurso ou no sistema interno. Um contato que termina em um estado FAILED_TO_SCHEDULE, opcionalmente, fornecerá um contexto adicional. `errorMessage` Para obter informações sobre como descrever contatos, consulte a [DescribeContactAPI](#).

Os casos de uso comuns que podem causar contatos FAILED_TO_SCHEDULE são fornecidos abaixo, junto com as etapas para ajudar a solucionar problemas.

Note

Este guia é específico para o status de contato FAILED_TO_SCHEDULE - e não se destina a outros status de falha, como, ou FAILED. AWS_FAILEDAWS_CANCELLED Consulte mais informações sobre os status de contato em [the section called “AWS Ground Station status de contato”](#)

As configurações especificadas em sua Antenna Downlink Demod Decode Config não são suportadas.

O [perfil da missão](#) usado para agendar esse contato tinha uma [antenna-downlink-demod-decode configuração](#) que não era válida.

AntennaDownlinkDemodDecode Configuração existente anteriormente

- Se suas antenna-downlink-demod-decode configurações foram alteradas recentemente, volte para uma versão em funcionamento anterior antes de tentar agendar.
- Se essa foi uma alteração intencional em uma configuração existente ou em uma configuração existente anteriormente que não está mais sendo agendada com sucesso, siga a próxima etapa sobre como integrar uma nova configuração. AntennaDownlinkDemodDecode

AntennaDownlinkDemodDecode Configuração recém-criada

Entre em contato AWS Ground Station diretamente para integrar sua nova configuração. Crie um caso com o [AWS Support](#), incluindo `contactId` aquele que terminou no estado FAILED_TO_SCHEDULE

Etapas gerais de solução de problemas

Se as etapas de solução de problemas anteriores não resolverem seu problema:

- Tente agendar o contato novamente ou agende outro contato usando o mesmo perfil de missão. Para obter informações sobre como reservar um contato, consulte [ReserveContact](#).
- [Se você continuar recebendo o status FAILED_TO_SCHEDULE para esse perfil de missão, entre em contato com o AWS Support](#)

Solucione o problema que DataflowEndpointGroups não está em um estado SAUDÁVEL

Abaixo estão listados os motivos pelos quais seus grupos de endpoints de fluxo de dados podem não estar em um estado HEALTHY, bem como a ação corretiva apropriada a ser tomada.

- NO_REGISTERED_AGENT: inicie sua instância do EC2, que registrará o agente. Observe que você deve ter um arquivo de configuração de controlador válido para que essa chamada seja bem-sucedida. Consulte o [AWS Ground Station Agente de uso](#) para obter detalhes sobre como configurar esse arquivo.
- INVALID_IP_OWNERSHIP- Use a DeleteDataflowEndpointGroup API para excluir o Dataflow Endpoint Group e, em seguida, use a CreateDataflowEndpointGroup API para recriar o Dataflow Endpoint Group usando endereços IP e portas associados à instância do EC2.
- UNVERIFIED_IP_OWNERSHIP: o endereço IP ainda não foi validado. A validação ocorre periodicamente, então isso deve se resolver sozinho.
- NOT_AUTHORIZED_TO_CREATE_SLR: a conta não está autorizada a criar a função vinculada ao serviço necessária. Verifique as etapas de solução de problemas em [Use funções vinculadas a serviços para Ground Station](#)

Solucionar problemas de efemérides inválidas

Quando você carrega dados de efemérides para AWS Ground Station, eles passam por um fluxo de trabalho de validação assíncrona. Se a validação falhar, o status da efeméride mudará para. INVALID A mensagem de erro na [DescribeEphemeris](#) resposta fornece informações detalhadas para ajudar você a identificar e resolver o problema.

Entendendo os erros de validação de efemérides

Quando uma efeméride falha na validação, a resposta da [DescribeEphemeris](#) API inclui dois campos para ajudar a diagnosticar o problema:

errorCode

Um código legível por máquina que identifica o erro de validação específico. Isso pode ser usado para tratamento de erros programáticos.

errorMessage

Uma descrição legível do erro de validação com detalhes específicos sobre o que deu errado e orientações sobre como corrigi-lo.

Exemplo de [DescribeEphemeris](#) resposta para uma efeméride inválida:

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

Erros comuns de validação para efemérides TLE

A seguir estão os erros de validação comuns encontrados ao fazer o upload de efemérides TLE:

Número de catálogo de satélites incompatível

Erro: “O número do catálogo do satélite presente nas efemérides não corresponde ao número do catálogo do satélite associado”

Solução: verifique se o número do ID/satellite catálogo NORAD em suas linhas TLE corresponde ao número do catálogo do satélite do seu satélite. Use 00000 para satélites sem um número de catálogo atribuído.

Movimento médio inválido

Erro: “O movimento médio das efemérides fornecidas difere muito das efemérides de referência mais recentes”

Solução: verifique se os dados do TLE estão corretos e representam uma órbita válida. A Ground Station usa efemérides do Space-Track como referência durante a validação.

Erros comuns de validação para efemérides OEM

A seguir estão os erros comuns de validação encontrados ao fazer o upload de efemérides de OEM:

Quadro de referência inválido

Erro: “O REF_FRAME não é suportado”

Solução: atualize seu arquivo OEM para usar um dos quadros de referência compatíveis: EME2000 ou ITRF2000.

Campos obrigatórios ausentes

Erro: “Campo de metadados ausente: INTERPOLAÇÃO”

Solução: adicione os campos INTERPOLATION e INTERPOLATION_DEGREE à sua seção de metadados OEM. Eles são necessários AWS Ground Station para gerar ângulos precisos de apontamento da antena.

Sistema de horário não suportado

Erro: “O TIME_SYSTEM não é suportado”

Solução: Certifique-se de que seu arquivo OEM use UTC como sistema de horário.

Versão OEM não suportada

Erro: “O CCSDS_OEM_VERS não é suportado”

Solução: Certifique-se de que seu arquivo OEM use o CCSDS OEM versão 2.0.

Erros comuns de validação para efemérides de elevação de azimute

A seguir estão os erros de validação comuns encontrados ao carregar efemérides de elevação de azimute:

azimuth/elevation Dados ausentes

Erro: "Nenhum TimeAzEl campo estava presente em pelo menos um AzElSegment"

Solução: certifique-se de que cada segmento em seus dados de elevação de azimute contenha pelo menos um par com marcação temporal azimuth/elevation .

Faixa de ângulo de azimute inválida (graus)

Erro: "AzEl az deve ser maior ou igual a -180 e menor ou igual a 360 graus"

Solução: verifique se os ângulos de azimute estão dentro de $[-180, 360]$ graus.

Faixa de ângulo de elevação inválida (graus)

Erro: "AzEl el deve ser maior ou igual a -90 e menor ou igual a 90 graus"

Solução: verifique se os ângulos de elevação estão dentro de $[-90, 90]$ graus.

Faixa de ângulo de azimute inválida (radianos)

Erro: "AzEl az deve ser maior ou igual a $-\pi$ e menor ou igual a 2π radianos"

Solução: verifique se os ângulos de azimute estão dentro de $[-\pi, 2\pi]$ radianos.

Faixa de ângulo de elevação inválida (radianos)

Erro: "AzEl el deve ser maior ou igual a $-\pi/2$ e menor ou igual a $\pi/2$ radianos"

Solução: verifique se os ângulos de elevação estão dentro de $[-\pi/2, \pi/2]$ radianos.

Valores de tempo não monotônicos

Erro: "Os TimeAzEl itens dentro de um AzElSegment devem estar temporariamente em ordem"

Solução: Certifique-se de que os valores de tempo em cada segmento estejam aumentando estritamente.

Segmentos fora de ordem

Erro: "AzElSegments deve estar temporariamente em ordem"

Solução: Certifique-se de que os segmentos estejam organizados em ordem cronológica.

Segmentos sobrepostos

Erro: “O intervalo de tempo de pelo menos um segmento se sobrepõe aos outros intervalos de tempo do segmento”

Solução: garanta que cada segmento tenha um intervalo de tempo exclusivo e sem sobreposição. O `endTime` de um segmento não deve exceder o `startTime` do próximo segmento.

Etapas de solução de problemas

Se sua efeméride falhar na validação, siga estas etapas para resolver o problema:

1. Ligue [DescribeEphemeris](#) com seu ID de efeméride para recuperar o `e.errorMessage`
2. Revise a mensagem de erro para obter detalhes específicos sobre qual verificação de validação falhou.
3. Corrija os problemas identificados em seus dados de efemérides.
4. Faça o upload de uma nova efeméride com os dados corrigidos usando [CreateEphemeris](#)
5. Monitore o novo status de efemérides até que ele atinja o estado. `ENABLED`
6. Exclua a efeméride inválida usando [DeleteEphemeris](#) se ela não for mais necessária.

Referência completa do código de erro

As seções a seguir fornecem um mapeamento abrangente de todos os `errorCode` valores que podem ser retornados quando a validação de efemérides falha, organizado pela categoria de alto nível. `invalidReason`

Motivo inválido: **METADATA_INVALID**

Esses erros ocorrem quando os campos de metadados obrigatórios estão ausentes, formatados incorretamente ou contêm valores não suportados nos dados de efemérides.

Código de erro	Mensagem de erro
ID_SATCAT_ID INCOMPATÍVEL	O número do catálogo do satélite presente nas efemérides do TLE não corresponde ao número do catálogo do satélite associado

Código de erro	Mensagem de erro
VERSÃO_OEM NÃO SUPORTADA	As CCSDS_OEM_VERS efemérides do OEM não são suportadas. Valores suportados: [2..0]
ORIGINADOR_AUSENTE	O campo do ORIGINATOR cabeçalho está ausente nas efemérides do OEM
FALTA_DATA_DE-CRIAÇÃO	O campo do CREATION_DATE cabeçalho está ausente nas efemérides do OEM
NOME_DO_OBJETO_AUSENTE	O campo de OBJECT_NAME metadados está ausente nas efemérides do OEM
ID DO OBJETO AUSENTE	O campo de OBJECT_ID metadados está ausente nas efemérides do OEM
REF_FRAME_NÃO SUPORTADO	As REF_FRAME efemérides do OEM não são suportadas. Valores suportados: [EME2000,ITRF2000]
REF_FRAME_EPOCH_NÃO SUPORTADO	O campo de REF_FRAME_EPOCH metadados nas efemérides do OEM não é suportado. Remova esse campo das efemérides
TIME_SYSTEM_NÃO SUPORTADO	As TIME_SYSTEM efemérides do OEM não são suportadas. Valores suportados: [UTC]
CENTER_BODY_NÃO SUPORTADO	As CENTER_BODY efemérides do OEM não são suportadas. Valores suportados: [Earth]
INTERPOLAÇÃO_AUSENTE	O campo de INTERPOLATION metadados está ausente nas efemérides do OEM
GRAUS_DE_INTERPOLAÇÃO_INVÁLIDO	O grau de interpolação nas efemérides do OEM deve ser maior que 0 para o método de interpolação
AZ_EL_SEGMENT_LIST_MISSING	O azElSegmentList campo está faltando
TEMPO_AZ_INSUFICIENTE_AZ_EL	Nenhum TimeAzEl campo estava presente em pelo menos um azElSegmentList

Motivo inválido: **TIME_RANGE_INVALID**

Esses erros ocorrem quando as efemérides contêm intervalos de tempo inválidos, incluindo problemas com start/end horários, ordenação de segmentos, segmentos sobrepostos ou inconsistências temporais.

Código de erro	Mensagem de erro
HORÁRIO_DE_IN_FUTURO	O horário de início das efemérides está no futuro, mas deve estar no passado
HORÁRIO DE TÉRMINO NO PASSADO	O fim do horário das efemérides está no passado, mas deve estar no futuro
TEMPO DE EXPIRAÇÃO MUITO CEDO	O tempo de expiração fornecido é anterior ao horário de término da efeméride
START_TIME_METADATA_MUITO CEDO	O valor dos START_TIME metadados é anterior ao primeiro momento presente nos dados de efemérides do OEM
STOP_TIME_METADATA_MUITO TARDE	O valor dos STOP_TIME metadados é posterior à última vez presente nos dados de efemérides do OEM
AZ_EL_SEGMENT_END_TIME_BEFORE_START_TIME	O endTime de pelo menos um segmento de dados está antes do segmento startTime
AZ_EL_SEGMENT_TIME_S_OVERLAP	O intervalo de tempo de pelo menos um segmento se sobrepõe aos outros intervalos de tempo do segmento
AZ_EL_SEGMENTS_OUT_OF_ORDER	Os segmentos não estão ordenados temporalmente
TIME_AZ_EL_ITEMS_OUT_OF_ORDER	Os TimeAzEl itens dentro de um AzElSegment devem estar temporariamente em ordem
AZ_EL_SEGMENT_REFERENCE_EPOCH_INVÁLIDO	A época de referência para um segmento é inválida ou formatada incorretamente

Código de erro	Mensagem de erro
AZ_EL_SEGMENT_START_TIME_INVALIDO	A hora de início no intervalo de tempo válido de um segmento não começa após o primeiro segmento
AZ_EL_SEGMENT_END_TIME_INVALIDO	A hora de término no intervalo de tempo válido de um segmento não termina após o último segmento
AZ_EL_SEGMENT_VALID_TIME_RANGE_INVALIDO	O intervalo de tempo válido para um segmento é inválido
AZ_EL_SEGMENT_END_TIME_TOO_LATE	A hora de término de um segmento excede a duração máxima permitida a partir da época de referência
AZ_EL_TOTAL_DURATION_EXCEEDED	A duração total em todos os segmentos excede a duração máxima permitida do ângulo de apontamento

Motivo inválido: **TRAJECTORY_INVALID**

Esses erros ocorrem quando as efemérides contêm dados de trajetória inválidos, incluindo problemas com parâmetros orbitais, faixas de ângulos ou unidades.

Código de erro	Mensagem de erro
MEAN_MOTION_INVALIDO	O movimento médio das efemérides TLE fornecidas difere muito das efemérides de referência mais recentes. Nota: A Ground Station usa efemérides do Space-Track como referência durante a validação
TIME_AZ_EL_AZ_RADIAN_RANGE_INVALIDO	AzEl az deve ser maior ou igual a $-\pi$ e menor ou igual a 2π radianos
TIME_AZ_EL_EL_RADIAN_RANGE_INVALIDO	AzEl el deve ser maior ou igual a $-\pi/2$ e menor ou igual a $\pi/2$ radianos
TIME_AZ_EL_AZ_DEGREE_RANGE_INVALIDO	AzEl az deve ser maior ou igual a -180 e menor ou igual a 360 graus

Código de erro	Mensagem de erro
TIME_AZ_EL_EL_DEGR EE_RANGE_INVÁLIDO	AzEl <u>e1</u> deve ser maior ou igual a -90 graus e menor ou igual a 90 graus
TIME_AZ_EL_ANGLE_UNITS_INVÁLIDO	Unidades de AzEl ângulo inválidas

Motivo inválido: **KMS_KEY_INVALID**

Esses erros ocorrem quando há problemas com a chave do AWS Key Management Service (KMS) usada para criptografar os dados de efemérides.

Código de erro	Mensagem de erro
PERMISSÕES_KMS_INSUFICIENTE S	A Ground Station não tem permissões suficientes para acessar a chave KMS dessa efeméride

Motivo inválido: **VALIDATION_ERROR**

Esses erros ocorrem quando há problemas gerais de validação com os dados de efemérides que não se enquadram em outras categorias específicas.

Código de erro	Mensagem de erro
INTERNAL_ERROR	Ocorreu um erro interno durante a validação das efemérides
FORMATO_DO_ARQUIVO _INVÁLIDO	O formato do arquivo de efemérides é inválido ou está corrompido. Verifique se o arquivo está em conformidade com o formato esperado para o tipo de efemérides

Solucionar problemas de contatos que não receberam dados

É possível que um contato pareça bem-sucedido, mas ainda não tenha recebido nenhum dado. Isso pode significar que você recebe arquivos PCAP vazios ou nenhum arquivo PCAP se estiver usando

a entrega de dados do S3. Isso pode acontecer por vários motivos. A seguir, são abordadas algumas das causas e como resolvê-las.

Configuração de downlink incorreta

Cada contato que recebe dados de um satélite terá um associado [Configuração de downlink de antena](#) ou [Configuração de decodificação de demodulação de downlink de antena](#). Se a configuração especificada não concordar com o sinal transmitido por um satélite, não AWS Ground Station será capaz de receber o sinal transmitido. Isso fará com que nenhum dado seja recebido por AWS Ground Station.

Para corrigir isso, verifique se as configurações que você está usando concordam com o sinal transmitido pelo seu satélite. Por exemplo, verifique se você definiu a frequência central correta, a largura de banda, a polarização e, se necessário, os parâmetros de demodulação e decodificação.

Manobra de satélite

Há momentos em que um satélite pode realizar uma manobra que desativa temporariamente alguns de seus sistemas de comunicação. A manobra também pode alterar significativamente a localização do satélite no céu. AWS Ground Station não conseguirá receber um sinal de um satélite que não esteja transmitindo um sinal ou se a efeméride usada fizer com que a AWS Ground Station antena aponte para um local no céu onde o satélite não esteja presente.

[Se você estiver tentando se comunicar com um satélite de transmissão pública operado pela NOAA, poderá encontrar uma mensagem descrevendo uma interrupção ou manobra na página de mensagens de alerta por satélite da NOAA.](#) A mensagem pode incluir um cronograma de quando se espera que a transmissão de dados seja retomada, ou isso pode ser publicado em uma mensagem subsequente.

Se você estiver se comunicando com seus próprios satélites, é sua responsabilidade entender suas operações de satélite e como isso pode afetar a comunicação com eles. AWS Ground Station Se você estiver realizando uma manobra que afetará a trajetória do satélite, isso pode incluir o fornecimento de dados de efemérides personalizados atualizados. Para obter mais informações sobre o fornecimento de dados de efemérides personalizados, consulte. [Entenda como AWS Ground Station usa efemérides](#)

AWS Ground Station interrupção

Se AWS Ground Station fizer com que um contato falhe ou o cancele, AWS Ground Station definirá o status do contato como `AWS_FAILED`, ou `AWS_CANCELLED`. Para obter mais informações sobre o

ciclo de vida do contato, consulte. [Entenda o ciclo de vida do contato](#) Em alguns casos, AWS Ground Station pode haver uma falha que impede que os dados sejam entregues à sua conta, mas não faz com que o contato tenha um `AWS_CANCELLED`status `AWS_FAILED`ou. Quando isso acontece, AWS Ground Station deve publicar um evento específico da conta em seu painel AWS Health. Para obter mais informações sobre o painel AWS Health, consulte [AWS Health User Guide](#).

Solucionar problemas de telemetria

Use as informações a seguir para solucionar problemas comuns com a telemetria.

Problemas comuns de configuração

Erros de permissão do IAM

Sintomas

Ao ligar `CreateConfig` para criar um `TelemetrySinkConfig`, você recebe um erro:

```
Unable to write to Kinesis Data Streams stream. Ensure that Ground Station has kinesis:PutRecord permissions for the given stream
```

Causas

- A função do IAM especificada no `TelemetrySinkConfig` não tem as permissões necessárias para gravar no stream do Kinesis Data Streams.
- A política de confiança na função do IAM não AWS Ground Station permite assumir a função.
- O `TelemetrySinkConfig` ARN do stream do Kinesis Data Streams em está incorreto ou o stream não existe.

Soluções

1. Verifique se a função do IAM existe e tem as permissões corretas. Revise [Etapa 2: criar um TelemetrySinkConfig](#) e garanta que todas as etapas foram seguidas.
2. Verifique se AWS Ground Station pode assumir sua função do IAM:

```
aws iam get-role --role-name GroundStationTelemetryRole
```

Verifique se a política de confiança está incluída `groundstation.amazonaws.com` como principal de serviço confiável.

3. Verifique se a função do IAM tem as permissões necessárias do Kinesis:

```
aws iam list-attached-role-policies --role-name GroundStationTelemetryRole
```

Certifique-se de que a política inclua `kinesis:DescribeStreamkinesis:PutRecord`, e `kinesis:PutRecords` permissões para sua transmissão.

4. Verifique se o stream do Kinesis Data Streams existe e se o ARN está correto:

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

5. Se estiver usando criptografia gerenciada pelo cliente, verifique se a função do IAM tem `kms:GenerateDataKey` permissão para sua AWS KMS chave.

PassRole erros de permissão

Sintomas

Ao ligar `CreateConfig`, você recebe um erro sobre não ter permissão para transmitir a função do IAM.

Solução

Certifique-se de que seu usuário ou função do IAM tenha `iam:PassRole` permissão para a função do IAM de telemetria. Adicione a seguinte política ao seu usuário ou função:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iam:GetRole",  
        "iam:PassRole"  
      ],  
      "Resource": "arn:aws:iam::9999999999:role/your-stream-name"  
    }  
  ]  
}
```

```
}  
]  
}
```

Problemas de configuração do stream do Kinesis Data Streams

Sintomas

A entrega da telemetria falha ou é intermitente.

Causas

- O stream do Kinesis Data Streams tem capacidade insuficiente para a taxa de transferência de telemetria.
- O stream está sendo usado por outros aplicativos, causando limitação de gravação.

Soluções

1. Verifique o status da transmissão:

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

2. Monitore a limitação de gravação usando CloudWatch métricas:

```
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. Se a limitação for detectada, considere:

- Mudar para o modo de capacidade sob demanda para escalabilidade automática.
- Usando um stream dedicado para AWS Ground Station telemetria.
- Se estiver usando o modo provisionado, aumente o número de fragmentos.

Problemas de entrega de telemetria

Nenhum dado de telemetria aparecendo

Sintomas

Depois de agendar um contato com um perfil de missão habilitado para telemetria, nenhum dado de telemetria aparece em seu stream do Kinesis Data Streams.

Possíveis causas e soluções

O perfil da missão não tem a telemetria ativada

Verifique se o perfil da missão usado para o contato inclui `telemetrySinkConfigArn`:

```
aws groundstation get-mission-profile \  
  --mission-profile-id 12345678-1234-1234-1234-123456789012 \  
  --region us-east-2
```

Verifique a saída do `telemetrySinkConfigArn` campo. Se não estiver presente, o perfil da missão não tem a telemetria ativada.

Problema de permissões de função do IAM

Analise as etapas de solução de problemas de permissão do IAM em [Erros de permissão do IAM](#).

O stream do Kinesis Data Streams não existe ou está na região errada

Verifique se o fluxo existe na região correta:

```
aws kinesis describe-stream \  
  --stream-name your-stream-name \  
  --region us-east-2
```

O contato ainda não foi iniciado

A entrega da telemetria começa no horário de início do contato. Verifique se o contato foi iniciado verificando o status do contato:

```
aws groundstation describe-contact \  
  --contact-id 12345678-1234-1234-1234-123456789012 \  
  --region us-east-2
```

Dados de telemetria intermitentes

Sintomas

Os dados de telemetria são fornecidos de forma inconsistente com lacunas ou registros ausentes.

Causas possíveis

- Problemas de capacidade ou limitação de streaming do Kinesis Data Streams. Consulte [Problemas de configuração do stream do Kinesis Data Streams](#).
- Problemas de conectividade de rede entre AWS Ground Station e seu stream do Kinesis Data Streams.

Soluções

- Monitore as CloudWatch métricas de stream do Kinesis Data Streams em busca de limitações ou erros.
- Certifique-se de que seu stream esteja usando o modo de capacidade sob demanda ou tenha capacidade provisionada suficiente.
- Use um stream dedicado para AWS Ground Station telemetria para evitar contenções com outros aplicativos.

Problemas de formato de dados

Erros de análise de JSON

Sintomas

Seu aplicativo encontra erros ao analisar registros de telemetria como JSON.

Soluções

- Verifique a decodificação em Base64 - Os dados no stream do Kinesis Data Streams são codificados em Base64. Certifique-se de decodificar os dados antes de analisá-los como JSON. Para obter mais informações, consulte [Lendo dados do stream do Kinesis Data Streams](#).
- Verifique se há registros vazios - AWS Ground Station pode enviar registros de validação vazios ao criar um TelemetrySinkConfig. Seu aplicativo deve lidar com registros vazios ou malformados normalmente.

- Implemente a análise com reconhecimento de versão - analise primeiro os `telemetryVersion` campos `telemetryTypeAndVersion` e `telemetryType`, e para determinar o esquema apropriado para cada registro.

Tipos ou versões de telemetria desconhecidos

Sintomas

Seu aplicativo encontra tipos ou versões de telemetria que não reconhece.

Solução

Esse é o comportamento esperado, pois novos tipos de telemetria e versões de esquema podem ser introduzidos ao longo do tempo. Sua inscrição deve:

- Registre tipos e versões desconhecidos para monitoramento.
- Continue processando tipos e versões conhecidos.
- Implemente um tratamento elegante para esquemas desconhecidos.

Para obter mais informações sobre o controle de versão do esquema, consulte [Controle de versão e evolução do esquema](#)

Como obter ajuda

Se você continuar enfrentando problemas depois de seguir as etapas de solução de problemas, entre em contato com o AWS Support.

Informações a serem fornecidas

Ao entrar em contato com o suporte, forneça as seguintes informações:

- Contato IDs com problemas
- ID do perfil de missão usado
- `TelemetrySinkConfig` ARN
- ARN do stream do Kinesis Data Streams
- ARN da função do IAM e políticas anexadas
- Mensagens de erro do CloudWatch Logs ou do seu aplicativo

- Registros de data e hora em que os problemas ocorreram
- Etapas de solução de problemas já realizadas

Para obter AWS Ground Station suporte geral, consulte o [Guia AWS Ground Station do usuário](#).

Cotas e limites

Você pode visualizar as regiões suportadas, seus endpoints associados e cotas em [AWS Ground Station endpoints](#) e cotas.

Você pode usar o [console do Service Quotas](#), a [API da AWS](#) e a [CLI da AWS](#) para solicitar aumentos de cota, quando necessário.

Termos de serviço

Para saber os termos do AWS Ground Station serviço, consulte os [Termos de Serviço da AWS](#).

Histórico do documento para o guia AWS Ground Station do usuário

A tabela a seguir descreve as mudanças importantes em cada versão do Guia AWS Ground Station do usuário.

Alteração	Descrição	Data
Novo recurso	Documentação adicionada para antenas AWS Ground Station dedicadas. Para obter mais informações, consulte Antenas AWS Ground Station dedicadas .	14 de abril de 2026
Novo recurso	Foi adicionada documentação para a UpdateContact API e o controle de versão de contatos. Para obter mais informações, consulte Atualizar contatos e controle de versão de contatos .	14 de abril de 2026
Novo recurso	Documentação adicionada para ListAntennas ListGroundStationReservations APIs e. Para obter mais informações, consulte AWS Ground Station Localizações e Exibir reservas de estações terrestres .	14 de abril de 2026
Atualização da documentação	Adicionou funcionalidades adicionais à CancelContact API e inclui informações sobre essa funcionalidade e implicações de medição.	10 de dezembro de 2025

	Para obter mais informações, consulte Compreender a medição de contatos .	
Atualização da documentação	Esclareceu que CloudWatch as métricas são emitidas na região associada à estação terrestre do contato. Links quebrados corrigidos.	2 de dezembro de 2025
Política AWS gerenciada atualizada	AWS Ground Station atualizou a política gerenciada <code>AWSGroundStationAgentInstancePolicy</code> para incluir permissões adicionais para recuperar a resposta URLs da tarefa. Para obter informações, consulte AWS Ground Station atualizações nas políticas AWS gerenciadas .	13 de novembro de 2025
Novo recurso	Atualizou o guia do usuário para incluir efemérides de elevação de azimute. Para obter mais informações, consulte Fornecer dados de efemérides de elevação de azimute	22 de outubro de 2025
Atualização da documentação	A entrega de dados entre regiões não exige mais configurações ou aprovações especiais. Para obter mais informações, consulte Usar a entrega de dados entre regiões .	11 de setembro de 2025

Atualização da documentação	Foi adicionado um esclarecimento sobre a utilização de contatos dos recursos configurados.	04 de abril de 2025
Novo recurso	Atualizou o guia do usuário para incluir gêmeos AWS Ground Station digitais.	6 de agosto de 2024
Atualização da documentação	Várias seções do guia do usuário foram atualizadas, incluindo novos diagramas, exemplos e muito mais.	18 de julho de 2024
Atualização da documentação	Feed RSS adicionado ao Guia do usuário.	18 de julho de 2024
Atualização da documentação	Divida o Guia do Usuário do AWS Ground Station Agente em um Guia do Usuário separado.	18 de julho de 2024
Novo recurso	Agora, os contatos podem ser programados em até 30 segundos fora dos intervalos de tempo de visibilidade. Os tempos de visibilidade estão incluídos nas DescribeContact respostas.	26 de março de 2024
Atualização da documentação	Organização aprimorada e adição da seção “Seleção de instância do EC2 e planejamento de CPU”.	6 de março de 2024

Atualização da documentação	Foram adicionadas novas práticas recomendadas ao Guia do Usuário do AWS Ground Station Agente para executar serviços e processos junto com o AWS Ground Station Agente.	23 de fevereiro de 2024
Atualização da documentação	Foi adicionada a página de notas de versão do agente.	21 de fevereiro de 2024
Atualização do modelo	Foi adicionado suporte para sub-rede pública separada no DataDelivery modelo DirectBroadcastSatelliteWbDigiFec 2.	14 de fevereiro de 2024
Atualização da documentação	Foi adicionada referência à AWS Notificações de Usuários na documentação de monitoramento.	6 de agosto de 2023
Atualização da documentação	Foram adicionadas instruções para marcar satélites com um nome a ser exibido no console do AWS Ground Station .	26 de julho de 2023
Novo recurso	Foi adicionado o Guia do Usuário do AWS Ground Station Agente para o lançamento do DigiF Data Delivery de banda larga.	12 de abril de 2023
Nova política AWS gerenciada	AWS Ground Station adicionou uma nova política chamada AWSGroundStationAgentInstancePolicy.	12 de abril de 2023

Novo recurso	Atualizou o guia do usuário para o lançamento do CPE Preview.	9 de novembro de 2022
Nova política AWS gerenciada	AWS Ground Station adicionou a AWSServiceRoleForGroundStationDataflowEndpointGroup service-linked-role (SLR) que inclui uma nova política chamada AWSServiceRoleForGroundStationDataflowEndpointGroupPolicy.	2 de novembro de 2022
Novo recurso	Atualizou o guia do usuário para incluir a integração com AWS CLI o.	17 de abril de 2020
Novo recurso	Atualizou o guia do usuário para incluir a integração com o CloudWatch Metrics.	24 de fevereiro de 2020
Novo modelo	Satélites de transmissão pública (AquaSnppJpss modelo) adicionados ao Guia do AWS Ground Station usuário.	19 de fevereiro de 2020
Novo recurso	Guia do usuário atualizado para incluir a entrega de dados entre regiões.	5 de fevereiro de 2020
Atualização da documentação	Exemplos e descrições atualizados para monitoramento AWS Ground Station com CloudWatch eventos.	4 de fevereiro de 2020

Atualização da documentação	Os locais de modelo foram atualizados e as seções Conceitos básicos e Solução de problemas foram revisadas.	19 de dezembro de 2019
Nova seção de solução de problemas	Uma seção de Solução de problemas foi adicionada ao Guia do Usuário do AWS Ground Station .	7 de novembro de 2019
Novo tópico de introdução	Atualizou o tópico Introdução, que inclui os CloudFormation modelos mais atuais.	1 de julho de 2019
Versão Kindle	Versão Kindle do Guia do Usuário do AWS Ground Station publicada.	20 de junho de 2019
Serviço e guia novos	Esta é a versão inicial AWS Ground Station e o Guia AWS Ground Station do Usuário.	23 de maio de 2019

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o [AWS glossário](#) na Glossário da AWS Referência.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.