



AWS Ground Station Guia do usuário do agente

# AWS Ground Station



# AWS Ground Station: AWS Ground Station Guia do usuário do agente

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

Visão geral do .....	1
O que é o AWS Ground Station agente? .....	1
Características do AWS Ground Station agente .....	2
Requisitos do agente .....	3
Diagramas da VPC .....	4
Sistema operacional com suporte .....	5
Receba dados por meio do AWS Ground Station agente .....	6
Vários fluxos de dados, um único receptor .....	6
Vários fluxos de dados, vários receptores .....	7
Selecione a instância do Amazon EC2 e reserve núcleos de CPU para sua arquitetura .....	9
Tipos de instância do Amazon EC2 compatíveis .....	9
Planejamento do núcleo da CPU .....	11
Coleta de informações de arquitetura .....	12
Exemplo de atribuição de CPU .....	14
Apêndice: <code>lscpu -p</code> saída (completa) para <code>c5.24xlarge</code> .....	15
Instalar o agente do .....	18
Use um CloudFormation modelo .....	18
Etapa 1: criar AWS recursos .....	18
Etapa 2: verificar o status do agente .....	18
Instale manualmente no EC2 .....	18
Etapa 1: criar recursos da AWS .....	18
Etapa 2: criar uma instância do EC2 .....	19
Etapa 3: baixar e instalar o agente .....	19
Etapa 4: configurar o agente .....	21
Etapa 5: aplicar ajuste de desempenho .....	21
Etapa 6: gerenciar o agente .....	21
Gerencie o agente .....	22
AWS Ground Station Configuração do agente .....	22
AWS Ground Station Início do agente .....	22
AWS Ground Station Agente: parada .....	23
AWS Ground Station Atualização do agente .....	23
AWS Ground Station Rebaixamento do agente .....	24
AWS Ground Station Desinstalação do agente .....	25
AWS Ground Station Status do agente .....	25

AWS Ground Station Informações de RPM do agente .....	26
Configurar o agente .....	27
Arquivo de configuração do agente .....	27
Exemplo .....	27
Detalhamento do campo .....	27
Ajuste sua instância do EC2 para obter desempenho .....	31
Ajuste interrupções de hardware e filas de recebimento - afeta a CPU e a rede .....	31
Tune Rx interrompe a coalescência - afeta a rede .....	32
Tune Rx ring buffer - afeta a rede .....	33
Ajuste a CPU C-State - afeta a CPU .....	33
Portas de entrada de reserva - impacta a rede .....	34
Reinicializar .....	34
Apêndice: Parâmetros recomendados para ajuste interrupt/RPS .....	34
Prepare-se para receber um contato DigIF .....	37
Práticas recomendadas .....	38
Melhores práticas do Amazon EC2 .....	38
Agendador Linux .....	38
AWS Ground Station lista de prefixos gerenciada .....	38
Limitação de contato único .....	38
Executando serviços e processos junto com o AWS Ground Station agente .....	38
Como exemplo, usando uma c5.24xlarge instância .....	39
Serviços de afinização (systemd) .....	39
Processos de afinização (scripts) .....	40
Solução de problemas .....	42
O agente falha ao iniciar .....	42
Solução de problemas .....	42
AWS Ground Station Registros do agente .....	43
Nenhum contato disponível .....	43
Obter suporte .....	44
Notas de versão do agente .....	45
Versão mais recente do agente .....	45
Versão 1.0.4382.0 .....	45
Versões obsoletas do agente .....	45
Versão 1.0.3555.0 .....	45
Versão 1.0.2942.0 .....	46
Versão 1.0.2716.0 .....	46

---

Versão 1.0.2677.0 .....	47
Validação da instalação RPM .....	48
Versão mais recente do agente .....	45
Versão 1.0.4382.0 .....	45
Verifique o RPM .....	48
Histórico do documento .....	50
.....	li

# Visão geral do

## O que é o AWS Ground Station agente?

Com o AWS Ground Station Agent, disponível como RPM, você pode receber (downlink) fluxos de dados síncronos de frequência intermediária digital (DigiF), incluindo banda estreita e banda larga, durante os contatos do AWS Ground Station. Você pode selecionar duas opções para entrega de dados:

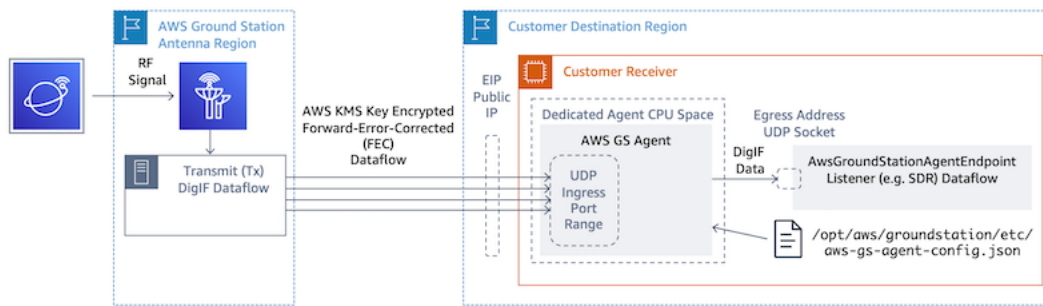
1. Entrega de dados para uma instância do EC2 - Entrega de dados para uma instância do EC2 que você possui. Você gerencia o AWS Ground Station Agente. Essa opção pode ser mais adequada se você precisar de processamento de dados quase em tempo real. Consulte o guia [Data Delivery to Amazon Elastic Compute Cloud](#) para obter informações sobre a entrega de dados do EC2.
2. Entrega de dados para um bucket S3 — Entrega de dados para um bucket AWS S3 que você possui por meio de um serviço gerenciado da Ground Station. Consulte o AWS Ground Station guia de [introdução 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 o EC2 ou S3, mas não para ambos simultaneamente.

### Note

Como a entrega de dados do S3 é um serviço gerenciado da Ground Station, este guia se concentra na entrega de dados para sua(s) instância(s) do EC2.

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



## Características do AWS Ground Station agente

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

- O DigiF tem capacidade de downlink de até 400% de largura MHz de banda, incluindo fluxos de dados de banda estreita ( $\leq 40$  MHz) e banda larga ( $> 40$ ) MHz.
- Entrega de dados DigiF de alta taxa e baixa instabilidade para qualquer IP público (AWS Elastic IP) na rede da AWS.
- Entrega confiável de dados usando Forward Error Correction (FEC).
- Entrega segura de dados usando uma AWS KMS chave gerenciada pelo cliente para criptografia.

# Requisitos do agente

## Note

Este guia do AWS Ground Station agente pressupõe que você tenha embarcado na Ground Station usando o guia de [AWS Ground Station introdução](#).

A instância EC2 do AWS Ground Station agente receptor requer um conjunto de recursos dependentes da AWS para entregar dados DigiF de forma confiável e segura aos seus endpoints.

1. Uma VPC na qual executar o receptor do EC2.
2. Uma chave do AWS KMS para criptografia/descriptografia de dados.
3. Uma chave SSH ou perfil de instância do EC2 configurado para o [SSM Session Manager](#).
4. Regras de rede/grupo de segurança para permitir o seguinte:
  1. Tráfego UDP proveniente das portas especificadas AWS Ground Station no seu grupo de endpoints de fluxo de dados. O agente reserva uma variedade de portas contíguas usadas para entregar dados ao(s) endpoint(s) do fluxo de dados de entrada.
  2. Acesso SSH à sua instância (nota: como alternativa, você pode usar o AWS Session Manager para acessar sua instância EC2).
  3. Acesso de leitura a um bucket do S3 acessível ao público para gerenciamento de agentes.
  4. Tráfego SSL na porta 443, permitindo que o agente se comunique com o AWS Ground Station serviço.
  5. Tráfego da lista com `.amazonaws.global.groundstation` de prefixos AWS Ground Station gerenciados.

Além disso, é necessária uma configuração de VPC incluindo uma sub-rede pública. Consulte informações básicas sobre a configuração da sub-rede no [Guia do usuário da VPC](#).

Configurações compatíveis:

1. Um IP elástico associado à instância do EC2 em uma sub-rede pública.
2. Um IP elástico associado a uma ENI em uma sub-rede pública, anexado à sua instância do EC2 (em qualquer sub-rede na mesma zona de disponibilidade da sub-rede pública).

É possível usar o mesmo grupo de segurança da instância do EC2 ou especificar um com pelo menos o conjunto mínimo de regras que consiste em:

- Tráfego UDP proveniente das portas especificadas AWS Ground Station no seu grupo de endpoints de fluxo de dados.

Por exemplo, modelos de entrega de dados do CloudFormation EC2 com esses recursos pré-configurados, consulte [Satélite de transmissão pública utilizando o AWS Ground Station agente \(banda larga\)](#).

## Diagramas da VPC

Diagrama: um IP elástico associado à instância do EC2 em uma sub-rede pública

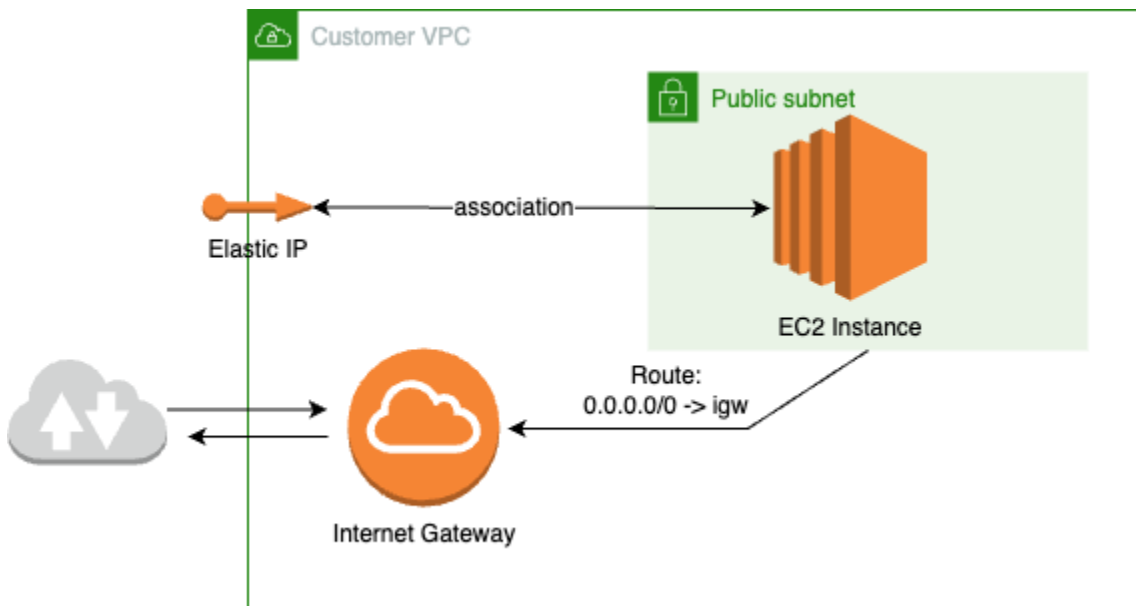
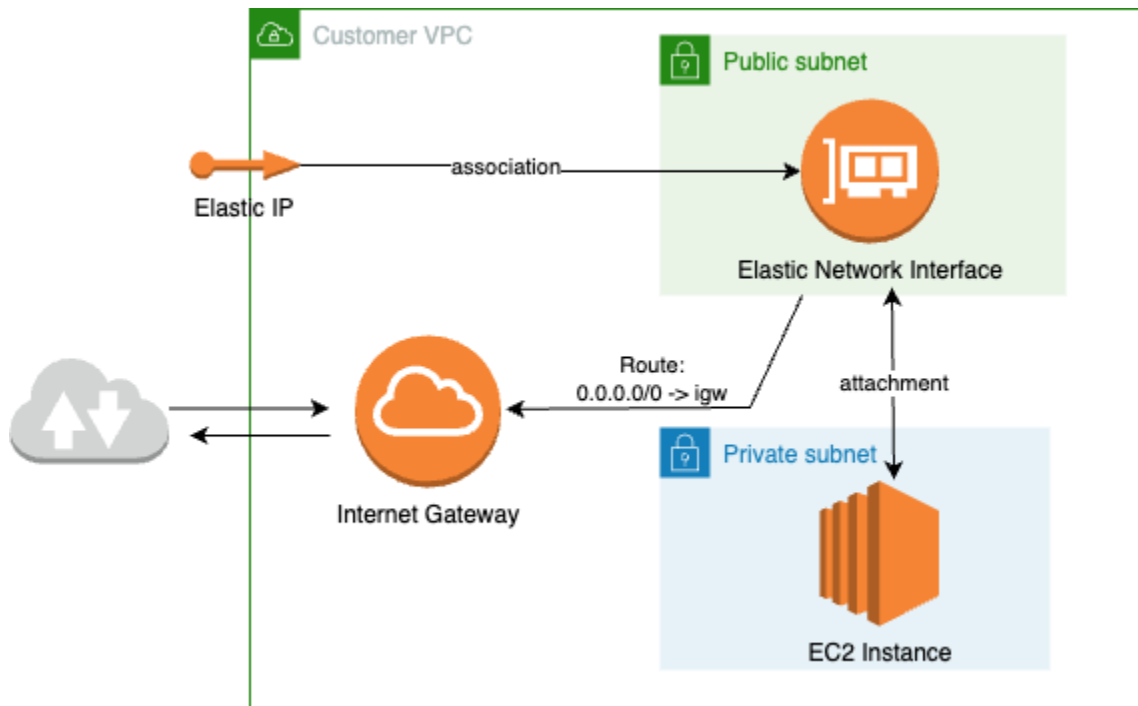


Diagrama: um IP elástico associado a uma ENI em uma sub-rede pública, anexado à instância do EC2 em uma sub-rede privada



## Sistema operacional com suporte

Amazon Linux 2 com kernel 5.10+.

Os tipos de instâncias compatíveis estão listados em [Selecione a instância do Amazon EC2 e reserve núcleos de CPU para sua arquitetura](#)

# Receba dados por meio do AWS Ground Station agente

Os diagramas abaixo fornecem uma visão geral de como os dados fluem AWS Ground Station durante os contatos de frequência intermediária digital de banda larga (DigIF).

O AWS Ground Station agente cuidará da orquestração dos componentes do plano de dados para um contato. Antes de agendar um contato, o agente deve estar corretamente configurado, iniciado e registrado (o registro é automático na inicialização do agente) com AWS Ground Station. Além disso, o software de recebimento de dados (como um rádio definido por software) deve estar em execução e configurado para receber dados no endereço de [AwsGroundStationAgentEndpoints](#)saída.

Nos bastidores, o AWS Ground Station Agente receberá tarefas AWS Ground Station e desfará a AWS KMS criptografia aplicada em trânsito, antes de encaminhá-la para o endereço de saída do endpoint de destino onde seu Software Defined Radio (SDR) está ouvindo. O AWS Ground Station Agente e seus componentes subjacentes respeitarão os limites da CPU definidos no arquivo de configuração para garantir que isso não afete o desempenho de outros aplicativos em execução na instância.

Você deve ter o AWS Ground Station Agente em execução na instância receptora envolvida no contato. Um único AWS Ground Station agente é capaz de orquestrar vários fluxos de dados, conforme mostrado abaixo, se você preferir receber todos os fluxos de dados em uma única instância receptora.

## Vários fluxos de dados, um único receptor

Exemplo de cenário:

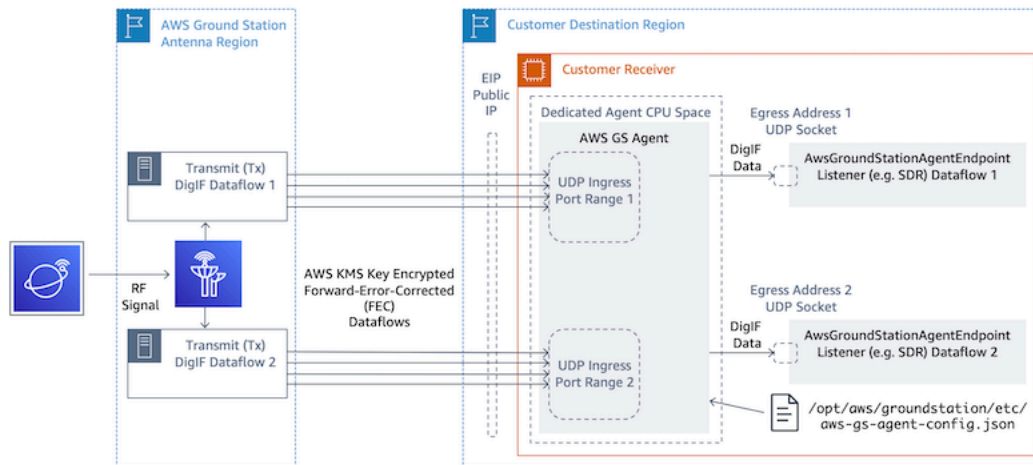
Você gostaria de receber dois downlinks de antena como fluxos de dados DigIF na mesma instância do receptor EC2. Os dois downlinks serão 200 MHz e 100MHz.

`AwsGroundStationAgentEndpoints`:

Haverá dois recursos `AwsGroundStationAgentEndpoint`, um para cada fluxo de dados. Ambos os endpoints terão o mesmo endereço IP público (`ingressAddress.socketAddress.name`). As entradas `portRange` não devem se sobrepor, pois os fluxos de dados estão sendo recebidos na mesma instância do EC2. Ambos `egressAddress.socketAddress.port` devem ser únicos.

Planejamento de CPU:

- 1 núcleo (2 vCPUs) para executar o único AWS Ground Station agente na instância.
- 6 núcleos (12 vCPUs) para receber o DigiF Dataflow 1 (pesquisa de 200 na tabela). MHz  
[Planejamento do núcleo da CPU](#)
- 4 núcleos (8 vCPUs) para receber o DigiF Dataflow 2 (100 consultas na tabela). MHz  
[Planejamento do núcleo da CPU](#)
- Espaço total de CPU dedicado para agentes = 11 núcleos (22 vCPU) no mesmo soquete.



## Vários fluxos de dados, vários receptores

Exemplo de cenário:

Você gostaria de receber dois downlinks de antena como fluxos de dados DigiF em diferentes instâncias do receptor EC2. Ambos os downlinks serão 400MHz.

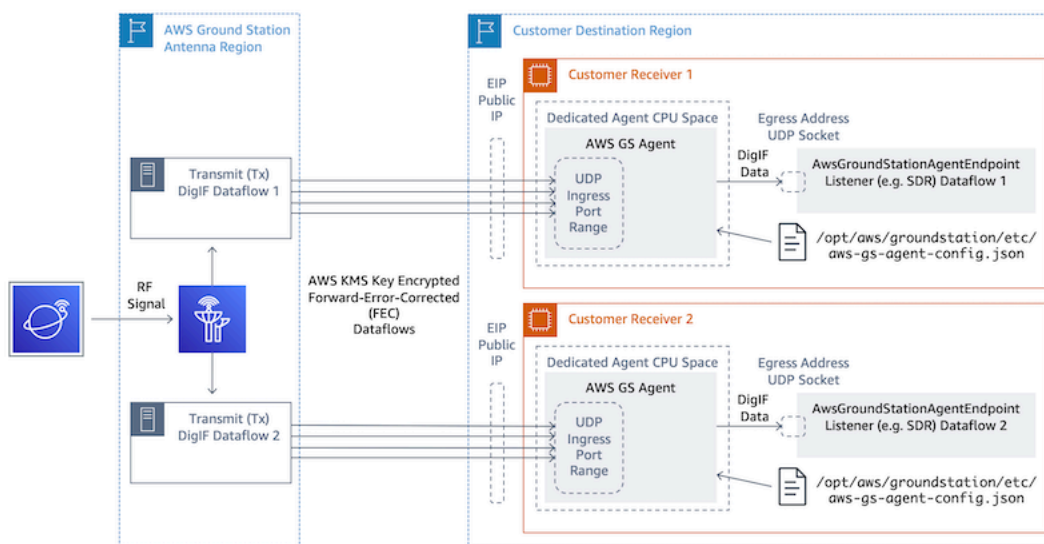
`AwsGroundStationAgentEndpoints`:

Haverá dois recursos `AwsGroundStationAgentEndpoint`, um para cada fluxo de dados. Ambos os endpoints terão o mesmo endereço IP público (`ingressAddress.socketAddress.name`). Não há restrição quanto aos valores das portas para qualquer um deles, `ingressAddress` ou `egressAddress`, pois os fluxos de dados são recebidos em uma infraestrutura separada e não entrarão em conflito uns com os outros.

Planejamento de CPU:

- Instância do receptor 1

- 1 núcleo (2 vCPUs) para executar o único AWS Ground Station agente na instância.
- 9 núcleos (18 vCPUs) para receber o DigiF Dataflow 1 (consulta de 400 na tabela). MHz [Planejamento do núcleo da CPU](#)
- Espaço total de CPU dedicado para agentes = 10 núcleos (20 vCPU) no mesmo soquete.
- Instância do receptor 2
  - 1 núcleo (2 vCPUs) para executar o único AWS Ground Station agente na instância.
  - 9 núcleos (18 vCPUs) para receber o DigiF Dataflow 2 (consulta de 400 na tabela). MHz [Planejamento do núcleo da CPU](#)
  - Espaço total de CPU dedicado para agentes = 10 núcleos (20 vCPU) no mesmo soquete.



# Selecione a instância do Amazon EC2 e reserve núcleos de CPU para sua arquitetura

## Tipos de instância do Amazon EC2 compatíveis

O AWS Ground Station Agente requer núcleos de CPU dedicados para operar devido aos fluxos de trabalho de entrega de dados com uso intensivo de computação. Nós oferecemos suporte aos seguintes tipos de instâncias. Consulte [Planejamento do núcleo da CPU](#) para decidir qual tipo de instância é mais adequado ao seu caso de uso.

Família de instâncias	Tipo de instância	Padrão v CPUs	Núcleos de CPU padrão	Largura de banda agregada máxima do DigiF () MHz
c5	c5.12xlarge	48	24	180
	c5.18xlarge	72	36	380
	c5.24xlarge	96	48	380
c5n	c5n.18xlarge	72	36	400
	c5n.metal	72	36	400
c6i	c6i.24xlarge	96	48	400
	c6i.32xlarge	128	64	400
c7i	c7i.12xlarge	48	24	280
	c7i.24xlarge	96	48	400
p3dn	p3dn.24xlarge	96	48	400
g4dn	g4dn.12xlarge	48	24	400
	g4dn.16xlarge	64	32	400

Família de instâncias	Tipo de instância	Padrão v CPUs	Núcleos de CPU padrão	Largura de banda agregada máxima do DigiF () MHz
	g4dn.metal	96	48	400
p4d	p4d.24xlarge	96	48	400
m5	m5.8xlarge	32	16	100
	m5.12xlarge	48	24	180
	m5.24xlarge	96	48	380
m6i	m6i.32xlarge	128	64	400
r5	r5.24xlarge	96	48	380
	r5.metal	96	48	380
r5n	r5n.24xlarge	96	48	400
	r5n.metal	96	48	400
r6i	r6i.32xlarge	128	64	400

#### Note

A coluna Largura de banda agregada máxima do DigiF mostra a largura de banda agregada máxima suportada para todos os fluxos de dados DigiF combinados em cada tipo de instância. Essas limitações se devem à capacidade de rede do EC2 alocada para os tipos de instância específicos. Esses valores representam estimativas conservadoras e devem ser usados ao planejar suas configurações DigiF. A largura de banda real pode variar com base na carga do sistema e em outros fatores.

## Planejamento do núcleo da CPU

O AWS Ground Station Agente exige núcleos de processador dedicados que compartilhem cache L3 para cada fluxo de dados. O agente foi projetado para aproveitar pares de CPU Hyper-threaded (HT) e exige que os pares HT sejam reservados para seu uso. Um par hyper-threaded é um par virtual ( CPUs vCPU) contido em um único núcleo. As tabelas a seguir fornecem um mapeamento da taxa de dados do fluxo de dados para o número necessário de núcleos reservados para o agente em um único fluxo de dados. Essas tabelas pressupõem o Cascade Lake ou versões mais recentes CPUs e são válidas para qualquer tipo de instância compatível. Se sua largura de banda estiver entre as entradas em uma tabela, selecione a próxima mais alta.

AWS Ground Station usa duas estratégias de entrega diferentes, dependendo da largura de banda do fluxo de dados:

- Banda estreita — larguras de banda de 40 e abaixo. MHz
- Banda larga — larguras de banda acima de 40. MHz

Essas estratégias têm requisitos básicos diferentes. Notavelmente, fluxos de dados de banda estreita podem exigir mais núcleos do que fluxos de dados de banda larga em larguras de banda comparáveis ou até maiores. A estratégia é selecionada automaticamente com base na AntennaDownlink largura de banda configurada.

### Important

Cada fluxo de dados exige seu próprio conjunto dedicado de núcleos. Se você tiver vários fluxos de dados, pesquise os principais requisitos para cada fluxo de dados de forma independente usando a tabela apropriada (banda estreita ou banda larga) e some-os. Não compartilhe núcleos entre fluxos de dados. O agente precisa de um núcleo reservado adicional para gerenciamento e coordenação, portanto, o total de núcleos necessários será a soma dos núcleos necessários para cada fluxo de dados mais um único núcleo adicional (2 v CPUs).

### Requisitos principais de banda estreita ( $\leq 40$ ) MHz

AntennaDownlink Largura de banda ( ) MHz	Taxa de dados VITA-49,2 DigiF esperada (MB/s)	Número de núcleos (pares de CPU HT)	Total de vCPU
≤25	≤500	3	6
30	600	4	8
35	700	4	8
40	800	4	8

### Requisitos básicos de banda larga (>40) MHz

AntennaDownlink Largura de banda ( ) MHz	Taxa de dados VITA-49,2 DigiF esperada (MB/s)	Número de núcleos (pares de CPU HT)	Total de vCPU
50	1000	3	6
100	2000	4	8
150	3000	5	10
200	4000	6	12
250	5000	6	12
300	6000	7	14
350	7000	8	16
400	8000	9	18

## Coleta de informações de arquitetura

`lscpu` fornece informações sobre a arquitetura do seu sistema. A saída básica mostra quais v CPUs (rotulados como "CPU") pertencem a quais nós NUMA (e cada nó NUMA compartilha um cache L3).

Abaixo, examinamos uma `c5.24xlarge` instância para coletar as informações necessárias para configurar o AWS Ground Station Agente. Isso inclui informações úteis como número de vCPUs, núcleos e vCPU-to-node associação.

```
> lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 96
On-line CPU(s) list: 0-95
Thread(s) per core: 2          <-----
Core(s) per socket: 24
Socket(s): 2
NUMA node(s): 2
Vendor ID: GenuineIntel
CPU family: 6
Model: 85
Model name: Intel(R) Xeon(R) Platinum 8275CL CPU @ 3.00GHz
Stepping: 7
CPU MHz: 3601.704
BogoMIPS: 6000.01
Hypervisor vendor: KVM
Virtualization type: full
L1d cache: 32K
L1i cache: 32K
L2 cache: 1024K
L3 cache: 36608K
NUMA node0 CPU(s): 0-23,48-71   <-----
NUMA node1 CPU(s): 24-47,72-95  <-----
```

Os núcleos dedicados ao AWS Ground Station Agente devem incluir ambos v CPUs para cada núcleo atribuído. Todos os núcleos de um fluxo de dados devem existir no mesmo nó NUMA. A opção do `lscpu` comando nos fornece as associações básicas com a CPU necessárias para configurar o agente. Os campos relevantes são CPU (que é o que chamamos de vCPU), Core e L3 (que indica qual cache L3 é compartilhado por esse núcleo). Observe que na maioria dos processadores Intel, o NUMA Node é igual ao cache L3.

Considere o seguinte subconjunto da `lscpu -p` saída para a `c5.24xlarge` (abreviado e formatado para maior clareza).

```

CPU,Core,Socket,Node,,L1d,L1i,L2,L3
0  0  0  0  0  0  0  0
1  1  0  0  1  1  1  0
2  2  0  0  2  2  2  0
3  3  0  0  3  3  3  0
...
16 0  0  0  0  0  0  0
17 1  0  0  1  1  1  0
18 2  0  0  2  2  2  0
19 3  0  0  3  3  3  0

```

Na saída, podemos ver que o Core 0 inclui v CPUs 0 e 16, o Core 1 inclui v CPUs 1 e 17, o Core 2 inclui v CPUs 2 e 18. Em outras palavras, os pares hiperencadeados são: 0 e 16, 1 e 17, 2 e 18.

## Exemplo de atribuição de CPU

Como exemplo, usaremos uma `c5.24xlarge` instância para um downlink de banda larga de polaridade dupla em 350. MHz A partir da tabela, [Planejamento do núcleo da CPU](#) sabemos que um MHz downlink 350 requer 8 núcleos (16 vCPUs) para um único fluxo de dados. Isso significa que essa configuração de polaridade dupla usando dois fluxos de dados requer um total de 16 núcleos (32 vCPUs) mais um núcleo (2 vCPUs) para o Agente.

Conhecemos a `lscpu` saída de `c5.24xlarge` NUMA `node0 CPU(s): 0-23,48-71` includes NUMA `node1 CPU(s): 24-47,72-95` e. Como o NUMA `node0` tem mais do que precisamos, atribuiremos apenas a partir dos núcleos: 0-23 e 48-71.

Primeiro, selecionaremos 8 núcleos para cada fluxo de dados que compartilham um cache L3 ou um Nó NUMA. Em seguida, procuraremos o v correspondente CPUs (denominado “CPU”) na `lscpu -p` saída em [Apêndice: lscpu -p saída \(completa\) para c5.24xlarge](#). Um exemplo de processo de seleção principal pode ter a seguinte aparência:

- Reserve os núcleos 0-1 para o sistema operacional.
- Fluxo 1: selecione os núcleos 2-9 que são mapeados para v CPUs 2-9 e 50-57.
- Fluxo 2: selecione os núcleos 10-17 que são mapeados para v CPUs 10-17 e 58-65.
- Núcleo do agente: selecione o núcleo 18, que mapeia para as CPUs versões 18 e 66.

Isso resulta em v CPUs 2-18 e 50-66, então a lista para fornecer ao agente é. [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66] Você deve garantir que seus próprios processos não estejam sendo executados neles CPUs , conforme descrito em [Executando serviços e processos junto com o AWS Ground Station agente](#).

Observe que os núcleos específicos selecionados neste exemplo são um tanto arbitrários. Outros conjuntos de núcleos funcionariam desde que satisfizessem a exigência de todos compartilharem um cache L3 para cada fluxo de dados.

## Apêndice: **lscpu -p** saída (completa) para c5.24xlarge

```
> lscpu -p
# The following is the parsable format, which can be fed to other
# programs. Each different item in every column has an unique ID
# starting from zero.
# CPU,Core,Socket,Node,,L1d,L1i,L2,L3
0,0,0,0,,0,0,0,0
1,1,0,0,,1,1,1,0
2,2,0,0,,2,2,2,0
3,3,0,0,,3,3,3,0
4,4,0,0,,4,4,4,0
5,5,0,0,,5,5,5,0
6,6,0,0,,6,6,6,0
7,7,0,0,,7,7,7,0
8,8,0,0,,8,8,8,0
9,9,0,0,,9,9,9,0
10,10,0,0,,10,10,10,0
11,11,0,0,,11,11,11,0
12,12,0,0,,12,12,12,0
13,13,0,0,,13,13,13,0
14,14,0,0,,14,14,14,0
15,15,0,0,,15,15,15,0
16,16,0,0,,16,16,16,0
17,17,0,0,,17,17,17,0
18,18,0,0,,18,18,18,0
19,19,0,0,,19,19,19,0
20,20,0,0,,20,20,20,0
21,21,0,0,,21,21,21,0
22,22,0,0,,22,22,22,0
23,23,0,0,,23,23,23,0
```

```
24,24,1,1,,24,24,24,1
25,25,1,1,,25,25,25,1
26,26,1,1,,26,26,26,1
27,27,1,1,,27,27,27,1
28,28,1,1,,28,28,28,1
29,29,1,1,,29,29,29,1
30,30,1,1,,30,30,30,1
31,31,1,1,,31,31,31,1
32,32,1,1,,32,32,32,1
33,33,1,1,,33,33,33,1
34,34,1,1,,34,34,34,1
35,35,1,1,,35,35,35,1
36,36,1,1,,36,36,36,1
37,37,1,1,,37,37,37,1
38,38,1,1,,38,38,38,1
39,39,1,1,,39,39,39,1
40,40,1,1,,40,40,40,1
41,41,1,1,,41,41,41,1
42,42,1,1,,42,42,42,1
43,43,1,1,,43,43,43,1
44,44,1,1,,44,44,44,1
45,45,1,1,,45,45,45,1
46,46,1,1,,46,46,46,1
47,47,1,1,,47,47,47,1
48,0,0,0,,0,0,0,0
49,1,0,0,,1,1,1,0
50,2,0,0,,2,2,2,0
51,3,0,0,,3,3,3,0
52,4,0,0,,4,4,4,0
53,5,0,0,,5,5,5,0
54,6,0,0,,6,6,6,0
55,7,0,0,,7,7,7,0
56,8,0,0,,8,8,8,0
57,9,0,0,,9,9,9,0
58,10,0,0,,10,10,10,0
59,11,0,0,,11,11,11,0
60,12,0,0,,12,12,12,0
61,13,0,0,,13,13,13,0
62,14,0,0,,14,14,14,0
63,15,0,0,,15,15,15,0
64,16,0,0,,16,16,16,0
65,17,0,0,,17,17,17,0
66,18,0,0,,18,18,18,0
67,19,0,0,,19,19,19,0
```

```
68,20,0,0,,20,20,20,0
69,21,0,0,,21,21,21,0
70,22,0,0,,22,22,22,0
71,23,0,0,,23,23,23,0
72,24,1,1,,24,24,24,1
73,25,1,1,,25,25,25,1
74,26,1,1,,26,26,26,1
75,27,1,1,,27,27,27,1
76,28,1,1,,28,28,28,1
77,29,1,1,,29,29,29,1
78,30,1,1,,30,30,30,1
79,31,1,1,,31,31,31,1
80,32,1,1,,32,32,32,1
81,33,1,1,,33,33,33,1
82,34,1,1,,34,34,34,1
83,35,1,1,,35,35,35,1
84,36,1,1,,36,36,36,1
85,37,1,1,,37,37,37,1
86,38,1,1,,38,38,38,1
87,39,1,1,,39,39,39,1
88,40,1,1,,40,40,40,1
89,41,1,1,,41,41,41,1
90,42,1,1,,42,42,42,1
91,43,1,1,,43,43,43,1
92,44,1,1,,44,44,44,1
93,45,1,1,,45,45,45,1
94,46,1,1,,46,46,46,1
95,47,1,1,,47,47,47,1
```

# Instalar o agente do

O AWS Ground Station Agente pode ser instalado das seguintes formas:

1. CloudFormation modelo (recomendado).
2. Instalação manual no Amazon EC2.

## Use um CloudFormation modelo

O CloudFormation modelo de entrega de dados do EC2 cria os recursos necessários da AWS para entregar dados à sua instância do EC2. Esse CloudFormation modelo usa a AMI AWS Ground Station gerenciada que tem o AWS Ground Station Agente pré-instalado. Em seguida, o script de inicialização da instância EC2 criada preenche o arquivo de configuração do agente e aplica o ajuste de desempenho necessário ([Ajuste sua instância do EC2 para obter desempenho](#)).

### Etapa 1: criar AWS recursos

Crie sua pilha de recursos da AWS usando o modelo de [satélite de transmissão pública usando o AWS Ground Station Agent \(banda larga\)](#).

### Etapa 2: verificar o status do agente

Por padrão, o agente está configurado e ativo (iniciado). Para verificar o status do agente, você pode se conectar à instância EC2 (SSH ou SSM Session Manager) e consultar [AWS Ground Station Status do agente](#).

## Instale manualmente no EC2

Embora a Ground Station recomende o uso de CloudFormation modelos para provisionar seus recursos da AWS, pode haver casos de uso em que o modelo padrão pode não ser suficiente. Nesses casos, recomendamos que você personalize o modelo de acordo com suas necessidades. Se isso ainda não atender aos requisitos, você poderá criar recursos da AWS manualmente e instalar o agente.

### Etapa 1: criar recursos da AWS

Consulte [exemplos de configurações de perfil de missão](#) para obter instruções sobre como configurar manualmente os recursos da AWS necessários para um contato.

O `AwsGroundStationAgentEndpointrecurso` define um endpoint para receber um fluxo de dados DigiF AWS Ground Station via Agent e é fundamental para obter um contato bem-sucedido. Embora a documentação da API esteja localizada na [Referência da API](#), esta seção discutirá brevemente os conceitos relevantes para o AWS Ground Station Agente.

O endpoint `ingressAddress` é onde o AWS Ground Station agente receberá tráfego UDP AWS KMS criptografado da antena. `socketAddress name` é o IP público da instância do EC2 (do EIP anexado). `portRange` deve ter pelo menos 300 portas contíguas em um intervalo que tenha sido reservado para qualquer outro uso. Para obter instruções, consulte [Portas de entrada de reserva - impacta a rede](#). Essas portas devem ser configuradas para permitir o tráfego de entrada UDP no grupo de segurança da VPC em que a instância receptora está sendo executada.

O endpoint `egressAddress` é onde o agente entregará o fluxo de dados DigiF para você. Você deve ter um aplicativo (por exemplo, SDR) recebendo os dados por meio de um soquete UDP neste local.

## Etapa 2: criar uma instância do EC2

O seguinte AMIs é suportado:

1. AWS Ground Station A AMI - `groundstation-a12-gs-agent-ami-*` onde \* é a data em que a AMI foi criada - vem com o agente instalado (recomendado).
2. `amzn2-ami-kernel-5.10-hvm-x86_64-gp2`.

## Etapa 3: baixar e instalar o agente

### Note

As etapas desta seção devem ser concluídas se você não tiver escolhido a AMI do AWS Ground Station agente na etapa anterior.

## Baixar o agente

O AWS Ground Station Agente está disponível em buckets S3 específicos da região e pode ser baixado em instâncias EC2 compatíveis usando a linha de comando (CLI) da AWS. Recomendamos sempre fazer o download do `latest` caminho para garantir que você esteja executando a versão mais atual do agente. O caminho de download é `s3://groundstation-wb-digif-software-`

`${AWS::Region}/aws-groundstation-agent/latest/amazon_linux_2_x86_64/aws-groundstation-agent.rpm` onde `${AWS::Region}` se refere a uma das [regiões de entrega de dados e console do AWS Ground Station](#) compatíveis.

### Note

Sempre use o `latest` caminho em vez de codificar um número de versão específico em seus scripts de download. O uso do `latest` caminho garante que suas instâncias recebam automaticamente patches de segurança e correções de bugs quando são provisionadas.

Exemplo: baixe a versão rpm mais recente da região `us-east-2` da AWS localmente para a pasta `/tmp`.

```
aws s3 --region us-east-2 cp s3://groundstation-wb-digif-software-us-east-2/aws-groundstation-agent/latest/amazon_linux_2_x86_64/aws-groundstation-agent.rpm /tmp
```

Se precisar baixar uma versão específica do AWS Ground Station Agente para fins de solução de problemas ou reversão, você pode baixá-la da pasta específica da versão no bucket do S3. Isso não é recomendado para operações normais.

Exemplo: baixe a versão rpm `1.0.2716.0` da região `us-east-2` da AWS localmente para a pasta `/tmp`.

```
aws s3 --region us-east-2 cp s3://groundstation-wb-digif-software-us-east-2/aws-groundstation-agent/1.0.2716.0/amazon_linux_2_x86_64/aws-groundstation-agent.rpm /tmp
```

### Note

Se você quiser confirmar que o RPM que você baixou foi vendido AWS Ground Station, siga as instruções para [Validação da instalação RPM](#).

## Instalar agente

```
sudo yum install ${MY_RPM_FILE_PATH}
```

Example: Assumes agent is in the "/tmp" directory  
sudo yum install /tmp/aws-groundstation-agent.rpm

## Etapa 4: configurar o agente

Depois de instalar o agente, você deve atualizar o arquivo de configuração do agente. Consulte [Configurar o agente](#).

## Etapa 5: aplicar ajuste de desempenho

AWS Ground Station AMI do agente: se você escolheu a AMI do AWS Ground Station agente na etapa anterior, aplique os seguintes ajustes de desempenho.

- [Ajuste interrupções de hardware e filas de recebimento - afeta a CPU e a rede](#)
- [Portas de entrada de reserva - impacta a rede](#)
- [Reinicializar](#)

Outros AMIs: se você escolheu qualquer outra AMI na etapa anterior, aplique todos os ajustes listados abaixo [Ajuste sua instância do EC2 para obter desempenho](#) e reinicie a instância.

## Etapa 6: gerenciar o agente

Para começar, pare e verifique o status do agente, consulte [Gerencie o agente](#).

# Gerencie o agente

O AWS Ground Station Agente fornece os seguintes recursos para configurar, iniciar, interromper, atualizar, rebaixar e desinstalar o agente usando ferramentas de comando Linux integradas.

## Tópicos

- [AWS Ground Station Configuração do agente](#)
- [AWS Ground Station Início do agente](#)
- [AWS Ground Station Agente: parada](#)
- [AWS Ground Station Atualização do agente](#)
- [AWS Ground Station Rebaixamento do agente](#)
- [AWS Ground Station Desinstalação do agente](#)
- [AWS Ground Station Status do agente](#)
- [AWS Ground Station Informações de RPM do agente](#)

## AWS Ground Station Configuração do agente

Navegue até `/opt/aws/groundstation/etc`, que deve conter um único arquivo chamado `aws-gs-agent-config.json`. Consulte [Arquivo de configuração do agente](#)

## AWS Ground Station Início do agente

```
#start
sudo systemctl start aws-groundstation-agent

#check status
systemctl status aws-groundstation-agent
```

Deve produzir uma saída mostrando que o agente está ativo.

```
aws-groundstation-agent.service - aws-groundstation-agent
```

```
Loaded: loaded (/usr/lib/systemd/system/aws-groundstation-agent.service; enabled;
        vendor preset: disabled)
Active: active (running) since Tue 2023-03-14 00:39:08 UTC; 1 day 13h ago
Docs: https://aws.amazon.com/ground-station/
Main PID: 8811 (aws-gs-agent)
CGroup: /system.slice/aws-groundstation-agent.service
##8811 /opt/aws/groundstation/bin/aws-gs-agent production
```

## AWS Ground Station Agente: parada

```
#stop
sudo systemctl stop aws-groundstation-agent

#check status
systemctl status aws-groundstation-agent
```

Deve produzir uma saída mostrando que o agente está inativo (parado).

```
aws-groundstation-agent.service - aws-groundstation-agent
Loaded: loaded (/usr/lib/systemd/system/aws-groundstation-agent.service; enabled;
        vendor preset: disabled)
Active: inactive (dead) since Thu 2023-03-09 15:35:08 UTC; 6min ago
Docs: https://aws.amazon.com/ground-station/
Process: 84182 ExecStart=/opt/aws/groundstation/bin/launch-aws-gs-agent (code=exited,
        status=0/SUCCESS)
Main PID: 84182 (code=exited, status=0/SUCCESS)
```

## AWS Ground Station Atualização do agente

1. Faça download da versão mais recente do agente. Consulte [Baixar o agente](#).
2. Interrompa o agente.

```
#stop
sudo systemctl stop aws-groundstation-agent
```

```
#confirm inactive (stopped) state
systemctl status aws-groundstation-agent
```

### 3. Atualizar o agente.

```
sudo yum update ${MY_RPM_FILE_PATH}

# check the new version has been installed correctly by comparing the agent version
with the starting agent version
yum info aws-groundstation-agent

# reload the systemd configuration
sudo systemctl daemon-reload

# restart the agent
sudo systemctl restart aws-groundstation-agent

# check agent status
systemctl status aws-groundstation-agent
```

## AWS Ground Station Rebaixamento do agente

1. Baixe a versão do agente de que você precisa. Consulte [Baixar o agente](#).
2. Rebaixe o agente.

```
# get the starting agent version
yum info aws-groundstation-agent

# stop the agent service
sudo systemctl stop aws-groundstation-agent

# downgrade the rpm
sudo yum downgrade ${MY_RPM_FILE_PATH}

# check the new version has been installed correctly by comparing the agent version
with the starting agent version
```

```
yum info aws-groundstation-agent

# reload the systemd configuration
sudo systemctl daemon-reload

# restart the agent
sudo systemctl restart aws-groundstation-agent

# check agent status
systemctl status aws-groundstation-agent
```

## AWS Ground Station Desinstalação do agente

A desinstalação do agente renomeará `/opt/aws/groundstation/etc/aws-gs-agent-config.json` to `/opt/aws/groundstation/etc/aws-gs-agent-config.json.rpmsave`. Instalar o agente novamente na mesma instância gravará valores padrão para `aws-gs-agent-config.json` e precisará ser atualizado com os valores corretos correspondentes aos seus recursos da AWS. Consulte [Arquivo de configuração do agente](#).

```
sudo yum remove aws-groundstation-agent
```

## AWS Ground Station Status do agente

O status do agente é ativo (o agente está em execução) ou inativo (o agente está parado).

```
systemctl status aws-groundstation-agent
```

Um exemplo de saída mostra que o agente está instalado, inativo (parado) e habilitado (inicia o serviço na inicialização).

```
aws-groundstation-agent.service - aws-groundstation-agent
Loaded: loaded (/usr/lib/systemd/system/aws-groundstation-agent.service; enabled;
       vendor preset: disabled)
```

```
Active: inactive (dead) since Thu 2023-03-09 15:35:08 UTC; 6min ago
Docs: https://aws.amazon.com/ground-station/
Process: 84182 ExecStart=/opt/aws/groundstation/bin/launch-aws-gs-agent (code=exited,
status=0/SUCCESS)
Main PID: 84182 (code=exited, status=0/SUCCESS)
```

## AWS Ground Station Informações de RPM do agente

```
yum info aws-groundstation-agent
```

A saída é a seguinte:

### Note

A “versão” pode ser diferente com base na versão mais recente publicada pelo agente.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name           : aws-groundstation-agent
Arch           : x86_64
Version        : 1.0.2677.0
Release        : 1
Size           : 51 M
Repo           : installed
Summary        : Client software for AWS Ground Station
URL            : https://aws.amazon.com/ground-station/
License        : Proprietary
Description    : This package provides client applications for use with AWS Ground Station
```

# Configurar o agente

Depois de instalar o agente, você deve atualizar o arquivo de configuração do agente em `/opt/aws/groundstation/etc/aws-gs-agent-config.json`.

## Arquivo de configuração do agente

### Exemplo

```
{
  "capabilities": [
    "arn:aws:groundstation:eu-central-1:123456789012:dataflow-endpoint-group/
bb6c19ea-1517-47d3-99fa-3760f078f100"
  ],
  "device": {
    "privateIps": [
      "127.0.0.1"
    ],
    "publicIps": [
      "1.2.3.4"
    ],
    "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
  ]
}
```

## Detalhamento do campo

### Capacidades

Os recursos são especificados como nomes de recursos da Amazon do Dataflow Endpoint Group.

Obrigatório: verdadeiro

Formato: String Array

- Valores: capacidade ARNs → Cadeia

Exemplos:

```
"capabilities": [  
  "arn:aws:groundstation:${AWS::Region}:${AWS::AccountId}:dataflow-endpoint-group/  
  ${DataflowEndpointGroupId}"  
]
```

## Dispositivo

Esse campo contém campos adicionais necessários para enumerar o “dispositivo” EC2 atual.

Obrigatório: verdadeiro

Formato: objeto

Membros:

- `privatelp`
- `publiclp`
- `agentCpuCores`
- `networkAdapters`

### `privatelp`

No momento, esse campo não é usado, mas está incluído para futuros casos de uso. Se nenhum valor for incluído, o padrão será ["127.0.0.1"]

Obrigatório: falso

Formato: String Array

- Valores: Endereços IP → String

Exemplo:

```
"privateIps": [  
  "127.0.0.1"  
],
```

## publicIps

IP elástico (EIP) por grupo de endpoint de fluxo de dados.

Obrigatório: verdadeiro

Formato: String Array

- Valores: Endereços IP → String

Exemplo:

```
"publicIps": [  
  "9.8.7.6"  
],
```

## agente CPUCores

Isso especifica quais núcleos virtuais são reservados para o aws-gs-agent processo. Consulte [Planejamento do núcleo da CPU](#) para ver os requisitos para definir esse valor adequadamente.

Obrigatório: verdadeiro

Formato: Int Array

- Valores: números principais → int

Exemplo:

```
"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  
]
```

## networkAdapters

Isso corresponde aos adaptadores Ethernet, ou interfaces conectadas ENIs, que receberão dados.

Obrigatório: falso

Formato: String Array

- Valores: nomes dos adaptadores Ethernet (pode encontrá-los executando `ifconfig`)

Exemplo:

```
"networkAdapters": [  
  "eth0"  
]
```

# Ajuste sua instância do EC2 para obter desempenho

## Note

Se você provisionou seus recursos da AWS usando CloudFormation modelos, esses ajustes serão aplicados automaticamente. Se você usou uma AMI ou criou manualmente sua instância do EC2, esses ajustes de desempenho devem ser aplicados para obter o desempenho mais confiável.

Lembre-se de reinicializar sua instância depois de aplicar qualquer ajuste.

## Tópicos

- [Ajuste interrupções de hardware e filas de recebimento - afeta a CPU e a rede](#)
- [Tune Rx interrompe a coalescência - afeta a rede](#)
- [Tune Rx ring buffer - afeta a rede](#)
- [Ajuste a CPU C-State - afeta a CPU](#)
- [Portas de entrada de reserva - impacta a rede](#)
- [Reinicializar](#)

## Ajuste interrupções de hardware e filas de recebimento - afeta a CPU e a rede

Esta seção configura o uso principal da CPU do systemd, SMP IRQs, Receive Packet Steering (RPS) e Receive Flow Steering (RFS). Consulte [Apêndice: Parâmetros recomendados para ajuste interrupt/RPS](#) para ver um conjunto de configurações recomendadas com base no tipo de instância que você está usando.

1. Afaste os processos do systemd dos núcleos da CPU do agente.
2. Redirecione as solicitações de interrupção de hardware para fora dos núcleos da CPU do agente.
3. Configure o RPS para evitar que a fila de hardware de uma única placa de interface de rede se torne um gargalo no tráfego da rede.

- Configure o RFS para aumentar a taxa de acertos de cache da CPU e, assim, reduzir a latência da rede.

O script `set_irq_affinity.sh` fornecido pelo RPM configura todas as opções acima para você. Adicione ao `crontab`, para que ele seja aplicado em cada inicialização:

```
echo "@reboot sudo /opt/aws/groundstation/bin/set_irq_affinity.sh  
'${interrupt_core_list}' '${rps_core_mask}' >> /var/log/user-data.log 2>&1" >>/var/  
spool/cron/root
```

- `interrupt_core_list` Substitua por núcleos reservados para o kernel e o sistema operacional - normalmente o primeiro e o segundo, juntamente com pares de núcleos hiperencadeados. Isso não deve se sobrepôr aos núcleos selecionados acima. (Por exemplo: '0,1,48,49' para uma instância hyper-threaded de 96 CPUs).
- `rps_core_mask` é uma máscara de bits hexadecimal que especifica quais CPUs devem processar os pacotes recebidos, com cada dígito representando 4 CPUs. Também deve ser separado por vírgula a cada oito caracteres, começando pela direita. É recomendável permitir tudo CPUs e deixar o cache lidar com o balanceamento.
  - Para ver a lista de parâmetros recomendados para cada tipo de instância, consulte [Apêndice: Parâmetros recomendados para ajuste interrupt/RPS](#).
- Exemplo de instância de 96 CPUs:

```
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
```

## Tune Rx interrompe a coalescência - afeta a rede

A coalescência de interrupções ajuda a evitar inundar o sistema host com muitas interrupções e ajuda a aumentar o throughput da rede. Com essa configuração, os pacotes são coletados e uma única interrupção é gerada a cada 128 microssegundos. Adicione ao `crontab`, para que ele seja aplicado em cada inicialização:

```
echo "@reboot sudo ethtool -C ${interface} rx-usecs 128 tx-usecs 128 >>/var/log/user-data.log 2>&1" >>/var/spool/cron/root
```

- Substitua `interface` pela interface de rede (adaptador Ethernet) configurada para receber dados. Normalmente, essa é `eth0` a interface de rede padrão atribuída a uma instância do EC2.

## Tune Rx ring buffer - afeta a rede

Aumente o número de entradas de anel para o buffer de anel Rx para evitar quedas ou sobrecargas de pacotes durante conexões intermitentes. Adicione ao crontab, para que fique configurado corretamente em cada inicialização:

```
echo "@reboot sudo ethtool -G ${interface} rx 16384 >>/var/log/user-data.log 2>&1" >>/var/spool/cron/root
```

- Substitua `interface` pela interface de rede (adaptador Ethernet) configurada para receber dados. Normalmente, essa é `eth0` a interface de rede padrão atribuída a uma instância do EC2.
- Ao configurar uma instância `c6i` familiar, o comando precisa ser modificado para definir o buffer de anel como `8192`, em vez de `16384`.

## Ajuste a CPU C-State - afeta a CPU

Defina o estado C da CPU para evitar a ociosidade, o que pode causar a perda de pacotes durante o início de um contato. Requer reinicialização da instância.

```
echo "GRUB_CMDLINE_LINUX_DEFAULT=\"console=tty0 console=ttyS0,115200n8 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1 processor.max_cstate=1 max_cstate=1\" >/etc/default/grub" >>/etc/default/grub
echo "GRUB_TIMEOUT=0" >>/etc/default/grub
grub2-mkconfig -o /boot/grub2/grub.cfg
```

## Portas de entrada de reserva - impacta a rede

Reserve todas as portas no intervalo de portas do endereço de entrada de `AwsGroundStationAgentEndpoint` para evitar conflitos com o uso do kernel. O conflito de uso da porta levará à falha no contato e na entrega de dados.

```
echo "net.ipv4.ip_local_reserved_ports=${port_range_min}-${port_range_max}" >> /etc/sysctl.conf
```

- Exemplo: `echo "net.ipv4.ip_local_reserved_ports=42000-43500" >> /etc/sysctl.conf.`

## Reinicializar

Depois que todos os ajustes forem aplicados com êxito, reinicialize a instância para que os ajustes entrem em vigor.

```
sudo reboot
```

## Apêndice: Parâmetros recomendados para ajuste interrupt/RPS

Esta seção determina os valores de parâmetros recomendados para uso na seção Ajustar interrupções de hardware e filas de recebimento: impactam a CPU e a rede.

Família	Tipo de instância	<code>interru pt_core_list</code>	<code>rps_cor e_mask</code>
c7i	<ul style="list-style-type: none"> <li>c7i.24xlarge</li> <li>c7i.12xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> <li>0,1,24,25</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,ffffff,ffffff</li> <li>fff,ffffff</li> </ul>

Família	Tipo de instância	$\{\text{interrupt\_core\_list}\}$	$\{\text{rps\_core\_mask}\}$
c6i	<ul style="list-style-type: none"> <li>c6i.32xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,64,65</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,ffffff,ffffff,ffffff</li> </ul>
c5	<ul style="list-style-type: none"> <li>c5.24xlarge</li> <li>c5.18xlarge</li> <li>c5.12xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> <li>0,1,36,37</li> <li>0,1,24,25</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,ffffff,ffffff</li> <li>ff,ffffffff,ffffff</li> <li>ffff,ffffff</li> </ul>
c5n	<ul style="list-style-type: none"> <li>c5n.metal</li> <li>c5n.18xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,36,37</li> <li>0,1,36,37</li> </ul>	<ul style="list-style-type: none"> <li>ff,ffffffff,ffffff</li> <li>ff,ffffffff,ffffff</li> </ul>
m5	<ul style="list-style-type: none"> <li>m5.24xlarge</li> <li>m5.12xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> <li>0,1,24,25</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,ffffff,ffffff</li> <li>ffff,ffffff</li> </ul>
r5	<ul style="list-style-type: none"> <li>r5.metal</li> <li>r5.24xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> <li>0,1,48,49</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,ffffff,ffffff</li> <li>ffffff,ffffff,ffffff</li> </ul>

Família	Tipo de instância	<code>interru</code> <code>pt_core_list</code>	<code>rps_cor</code> <code>e_mask</code>
r5n	<ul style="list-style-type: none"> <li>r5n.metal</li> <li>r5n.24xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> <li>0,1,48,49</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,</li> <li>ffffff,</li> <li>ffffff</li> <li>ffffff,</li> <li>ffffff,</li> <li>ffffff</li> </ul>
g4dn	<ul style="list-style-type: none"> <li>g4dn.metal</li> <li>g4dn.16xlarge</li> <li>g4dn.12xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> <li>0,1,32,33</li> <li>0,1,24,25</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,</li> <li>ffffff,</li> <li>ffffff</li> <li>ffffff,</li> <li>ffffff</li> <li>fff,ffffff</li> </ul>
p4d	<ul style="list-style-type: none"> <li>p4d.24xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,</li> <li>ffffff,</li> <li>ffffff</li> </ul>
p3dn	<ul style="list-style-type: none"> <li>p3dn.24xlarge</li> </ul>	<ul style="list-style-type: none"> <li>0,1,48,49</li> </ul>	<ul style="list-style-type: none"> <li>ffffff,</li> <li>ffffff,</li> <li>ffffff</li> </ul>

## Prepare-se para receber um contato DigIF

1. Analise o planejamento do núcleo da CPU para ver os fluxos de dados desejados e forneça uma lista dos núcleos que o agente pode usar. Consulte [Planejamento do núcleo da CPU](#).
2. Revise o arquivo de configuração do AWS Ground Station agente. Consulte [AWS Ground Station Configuração do agente](#).
3. Confirme se o ajuste de desempenho necessário foi aplicado. Consulte [Ajuste sua instância do EC2 para obter desempenho](#).
4. Confirme se você está seguindo todas as melhores práticas mencionadas. Consulte [Práticas recomendadas](#).
5. Confirme se o AWS Ground Station agente foi iniciado antes do horário de início do contato agendado por meio de:

```
systemctl status aws-groundstation-agent
```

6. Confirme se o AWS Ground Station agente está íntegro antes do horário de início do contato agendado por meio de:

```
aws groundstation get-dataflow-endpoint-group --dataflow-endpoint-group-id  
${DATAFLOW-ENDPOINT-GROUP-ID} --region ${REGION}
```

Verifique se o `agentStatus` do seu `awsGroundStationAgentEndpoint` está **ATIVO** e se o `auditResults` está **SAUDÁVEL**.

# Práticas recomendadas

## Melhores práticas do Amazon EC2

Siga as melhores práticas atuais do EC2 e garanta disponibilidade suficiente de armazenamento de dados.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-best-practices.html>

## Agendador Linux

O agendador Linux pode reordenar pacotes em soquetes UDP se os processos correspondentes não estiverem fixados em um núcleo específico. Qualquer thread que envie ou receba dados UDP deve se fixar em um núcleo específico durante a transmissão de dados.

## AWS Ground Station lista de prefixos gerenciada

É recomendável utilizar a lista de prefixos com `.amazonaws.global.groundstation` gerenciada pela AWS ao especificar as regras de rede para permitir a comunicação da antena. Consulte [Trabalhar com as listas de prefixos gerenciados pela AWS](#) para obter mais informações sobre Listas de Prefixos Gerenciadas pela AWS.

## Limitação de contato único

O agente do AWS Ground Station oferece suporte a vários streams por contato, mas suporta apenas um único contato por vez. Para evitar problemas de agendamento, não compartilhe uma instância em vários grupos de endpoints de fluxo de dados. Se uma única configuração de agente estiver associada a vários DFEG diferentes ARNs, ela falhará no registro.

## Executando serviços e processos junto com o AWS Ground Station agente

Ao iniciar serviços e processos na mesma instância EC2 do AWS Ground Station Agente, é importante vinculá-los a v que CPUs não estão em uso pelo AWS Ground Station Agente e pelo kernel Linux, pois isso pode causar gargalos e até mesmo perda de dados durante os contatos. Esse conceito de vinculação a v específico CPUs é conhecido como afinidade.

Núcleos a serem evitados:

- agentCpuCores de [Arquivo de configuração do agente](#).
- interrupt\_core\_list do [Ajuste interrupções de hardware e filas de recebimento - afeta a CPU e a rede](#).
  - Os valores padrão podem ser encontrados em [Apêndice: Parâmetros recomendados para ajuste interrupt/RPS](#)

## Como exemplo, usando uma **c5.24xlarge** instância

Se você especificou

```
"agentCpuCores": [24,25,26,27,72,73,74,75]"
```

e fugiu

```
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
```

em seguida, evite os seguintes núcleos:

```
0,1,24,25,26,27,48,49,72,73,74,75
```

## Serviços de afinização (systemd)

Os serviços recém-lançados serão automaticamente afinizados com os `interrupt_core_list` mencionados anteriormente. Se o caso de uso dos serviços lançados exigir núcleos adicionais ou precisar de núcleos menos congestionados, siga esta seção.

Verifique para qual afinidade seu serviço está configurado atualmente com o comando:

```
systemctl show --property CPUAffinity <service name>
```

Se você ver um valor vazio como `CPUAffinity=`, isso significa que ele provavelmente usará os núcleos padrão do comando acima `...bin/set_irq_affinity.sh <using the cores here> ...`

Para substituir e definir uma afinidade específica, encontre a localização do arquivo de serviço executando:

```
systemctl show -p FragmentPath <service name>
```

Abra e modifique o arquivo (usando `vim`, etc.) e coloque o `CPUAffinity=<core list>` na `[Service]` seção como:

```
[Unit]
...

[Service]
...
CPUAffinity=2,3

[Install]
...
```

Salve o arquivo e reinicie o serviço para aplicar a afinidade com:

```
systemctl daemon-reload
systemctl restart <service name>

# Additionally confirm by re-running
systemctl show --property CPUAffinity <service name>
```

Para obter mais informações, visite: [Red Hat Enterprise Linux 8 - Gerenciando, monitorando e atualizando o kernel - Capítulo 27. Configurando políticas de CPU Affinity e NUMA usando systemd.](#)

## Processos de afinização (scripts)

É altamente recomendável que scripts e processos recém-lançados sejam afinizados manualmente, pois o comportamento padrão do Linux permitirá que eles usem qualquer núcleo na máquina.

Para evitar conflitos fundamentais em qualquer processo em execução (como python, scripts bash etc.), inicie o processo com:

```
taskset -c <core list> <command>  
# Example: taskset -c 8 ./bashScript.sh
```

Se o processo já estiver em execução, use comandos como `pidof` ou `ps` para encontrar a ID do processo (PID) do processo específico. Com o PID, você pode ver a afinidade atual com:

```
taskset -p <pid>
```

e pode modificá-lo com:

```
taskset -p <core mask> <pid>  
# Example: taskset -p c 32392 (which sets it to cores 0xc -> 0b1100 -> cores 2,3)
```

Para obter mais informações sobre o conjunto de tarefas, consulte conjunto de [tarefas - página do manual Linux](#)

# Solução de problemas

## O agente falha ao iniciar

O AWS Ground Station Agente pode falhar ao iniciar devido a vários motivos, mas o cenário mais comum pode ser um arquivo de configuração do agente mal configurado. Depois de iniciar o agente (consulte [AWS Ground Station Início do agente](#)), você pode obter um status como:

```
#agent is automatically retrying a restart
aws-groundstation-agent.service - aws-groundstation-agent
Loaded: loaded (/usr/lib/systemd/system/aws-groundstation-agent.service; enabled;
        vendor preset: disabled)
Active: activating (auto-restart) (Result: exit-code) since Fri 2023-03-10 01:48:14
        UTC; 23s ago
Docs: https://aws.amazon.com/ground-station/
Process: 43038 ExecStart=/opt/aws/groundstation/bin/launch-aws-gs-agent (code=exited,
        status=101)
Main PID: 43038 (code=exited, status=101)

#agent has failed to start
aws-groundstation-agent.service - aws-groundstation-agent
Loaded: loaded (/usr/lib/systemd/system/aws-groundstation-agent.service; enabled;
        vendor preset: disabled)
Active: failed (Result: start-limit) since Fri 2023-03-10 01:50:15 UTC; 13s ago
Docs: https://aws.amazon.com/ground-station/
Process: 43095 ExecStart=/opt/aws/groundstation/bin/launch-aws-gs-agent (code=exited,
        status=101)
Main PID: 43095 (code=exited, status=101)
```

## Solução de problemas

```
sudo journalctl -u aws-groundstation-agent | grep -i -B 3 -A 3 'Loading Config' | tail
-6
```

pode resultar em uma saída de:

```
launch-aws-gs-agent[43095]: Running with options Production(ProductionOptions
 { endpoint: None, region: None })
launch-aws-gs-agent[43095]: Loading Config
launch-aws-gs-agent[43095]: System has 96 logical cores
systemd[1]: aws-groundstation-agent.service: main process exited, code=exited,
 status=101/n/a
systemd[1]: Unit aws-groundstation-agent.service entered failed state.
```

A falha ao iniciar o agente após “Carregar Config” indica um problema com a configuração do agente. Consulte [Arquivo de configuração do agente](#) para verificar a configuração do seu agente.

## AWS Ground Station Registros do agente

AWS Ground Station O agente grava informações sobre execuções de contatos, erros e status de saúde em arquivos de log na instância que executa o agente. Você pode visualizar os arquivos de log conectando-se manualmente a uma instância.

É possível visualizar logs do agente em instâncias nos locais a seguir.

```
/var/log/aws/groundstation
```

## Nenhum contato disponível

O agendamento de contatos requer um AWS Ground Station agente saudável. Confirme se seu AWS Ground Station agente foi iniciado e se está íntegro consultando a AWS Ground Station API por meio `get-dataflow-endpoint-group` de:

```
aws groundstation get-dataflow-endpoint-group --dataflow-endpoint-group-id ${DATAFLOW-
ENDPOINT-GROUP-ID} --region ${REGION}
```

Verifique se o `agentStatus` do seu `awsGroundStationAgentEndpoint` está **ATIVO** e se o `auditResults` está **SAUDÁVEL**.

## Obter suporte

Entre em contato com a equipe da Ground Station por meio do AWS Support.

1. Forneça `contact_id` todos os contatos afetados. A AWS Ground Station equipe não pode investigar um contato específico sem essas informações.
2. Forneça detalhes sobre todas as etapas de solução de problemas já tomadas.
3. Forneça todas as mensagens de erro encontradas ao executar os comandos em nossa orientação de solução de problemas.

# Notas de versão do agente

## Versão mais recente do agente

### Versão 1.0.4382.0

Data de lançamento: 18/11/2025

Somas de verificação de RPM:

- SHA256: 620fd307124f1276194f2faa0104fe0549427ae18e4f5655444f8c30b919c640
- MD5: 73e06dcad44adaccbe2ab005218abfc7

Alterações:

- Atualize o comportamento de nova tentativa do cliente quando o servidor indicar sobrecarga.

## Versões obsoletas do agente

### Versão 1.0.3555.0

Data de lançamento: 27/03/2024

Somas de verificação de RPM:

- SHA256: 108f3aceb00e5af549839cd766c56149397e448a6e1e1429c89a9eebb6bc0fc1
- MD5: 65b72fa507fb0af32651adbb18d2e30f

Alterações:

- Adicione a métrica do Agente para a versão executável selecionada durante a inicialização da tarefa.
- Adicione suporte ao arquivo de configuração para evitar versões executáveis específicas quando outras versões estiverem disponíveis.
- Adicione diagnósticos de rede e roteamento.

- Recursos de segurança adicionais.
- Corrige o problema em que alguns erros de relatórios de métricas eram gravados stdout/journal em vez do arquivo de log.
- Lide com facilidade com erros de soquete inacessíveis na rede.
- Meça a perda e a latência de pacotes entre os agentes de origem e destino.
- Lance a aws-gs-datapipe versão 2.0 para oferecer suporte aos novos recursos do protocolo e à capacidade de atualizar contatos de forma transparente para o novo protocolo.

## Versão 1.0.2942.0

Data de lançamento: 26/06/2023

Data de fim do Support: 31/05/2024

Somas de verificação de RPM:

- SHA256: 7d94b642577504308a58bab28f938507f2591d4e1b2c7ea170b77bea97b5a9b6
- MD5: 661ff2b8f11aba5d657a6586b56e0d8f

Alterações:

- Foram adicionados registros de erros para quando o Agent RPM é atualizado no disco e precisa ser reiniciado o Agente para que as alterações entrem em vigor.
- Foi adicionada a validação do ajuste de rede para garantir que as etapas de ajuste do guia do usuário do Agente sejam seguidas e aplicadas corretamente.
- Corrija o erro que causava avisos errôneos nos registros do Agente sobre o arquivamento de registros.
- Detecção aprimorada de perda de pacotes.
- Instalação atualizada do Agente para evitar a instalação ou atualização do RPM se o Agente já estiver em execução.

## Versão 1.0.2716.0

Data de lançamento: 15/03/2023

Data de fim do Support: 31/05/2024

### Somas de verificação de RPM:

- SHA256: cb05b6a77dfcd5c66d81c0072ac550affbcefefc372cc5562ee52fb220844929
- MD5: 65266490c4013b433ec39ee50008116c

### Alterações:

- Ative o upload de registros quando o agente tiver falhas durante a tarefa.
- Corrija o bug de compatibilidade do Linux nos scripts de ajuste de rede fornecidos.

## Versão 1.0.2677.0

Data de lançamento: 15/02/2023

Data de fim do Support: 31/05/2024

### Somas de verificação de RPM:

- SHA256: 77cfe94acb00af7ca637264b17c9b21bd7afdc85b99dffdd627aec9e99397489
- MD5: b8533be7644bb4d12ab84de21341adac

### Alterações:

- Primeira versão do Agent disponível ao público em geral.

## Validação da instalação RPM

A versão mais recente do RPM, o MD5 hash validado a partir do RPM e o SHA256 hash usando sha256sum são mostrados abaixo. Esses valores, combinados, podem ser usados para validar a versão RPM que está sendo usada para o agente da estação terrestre.

### Versão mais recente do agente

#### Versão 1.0.4382.0

Data de lançamento: 18/11/2025

Somas de verificação de RPM:

- SHA256: 620fd307124f1276194f2faa0104fe0549427ae18e4f5655444f8c30b919c640
- MD5: 73e06dcad44adaccbe2ab005218abfc7

Alterações:

- Atualize o comportamento de nova tentativa do cliente quando o servidor indicar sobrecarga.

## Verifique o RPM

As ferramentas necessárias para verificar essa instalação do RPM são:

- [sha256sum](#)
- [rpm](#)

Ambas as ferramentas vêm por padrão no Amazon Linux 2. Essas ferramentas ajudarão a validar se o RPM que você está usando é a versão correta. Primeiro, baixe o RPM mais recente do bucket S3 (consulte [Baixar o agente](#) para obter instruções sobre como baixar o RPM). Depois que esse arquivo for baixado, haverá algumas coisas a serem verificadas:

- Calcule o sha256sum do arquivo RPM. Execute a ação a seguir na linha de comando da instância de computação que você está usando:

```
sha256sum aws-groundstation-agent.rpm
```

Pegue esse valor e compare-o com a tabela acima. Isso mostra que o arquivo RPM baixado é um arquivo válido para uso e que o AWS Ground Station distribuiu aos clientes. Se os hashes não corresponderem, não instale o RPM e o exclua da instância de computação.

- Verifique também o MD5 hash do arquivo para garantir que o RPM não tenha sido comprometido. Para fazer isso, use a ferramenta de linha de comando RPM executando o seguinte comando:

```
rpm -Kv ./aws-groundstation-agent.rpm
```

Verifique se o MD5 hash listado aqui é o MD5 mesmo da versão que está na tabela acima. Depois que esses dois hashes forem validados em relação a essa tabela listada no AWS Docs, o cliente poderá ter certeza de que o RPM baixado e instalado é a versão segura e sem comprometimentos do RPM.

# Histórico de documentos do Guia do usuário do AWS Ground Station agente

A tabela a seguir descreve as mudanças importantes em cada versão do Guia do Usuário do AWS Ground Station Agente.

Alteração	Descrição	Data
<a href="#">Atualização da documentação</a>	Suporte removido para a família de instâncias mais antiga: m4.	30 de setembro de 2024
<a href="#">Atualização da documentação</a>	Foi adicionado um comentário sobre como manter a sub-rede e a EC2 instância da Amazon na mesma zona de disponibilidade nos <a href="#">Requisitos do agente</a> .	18 de julho de 2024
<a href="#">Atualização da documentação</a>	Divida o AWS Ground Station Agente em seu próprio guia do usuário. Para alterações anteriores, consulte: <a href="#">Histórico do documento para o guia do usuário do AWS Ground Station</a> .	18 de julho de 2024

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.