AWS Ground Station Guia do usuário do agente

# AWS Ground Station

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

# Table of Contents

# Visão geral

## O que é o AWS Ground Station agente?

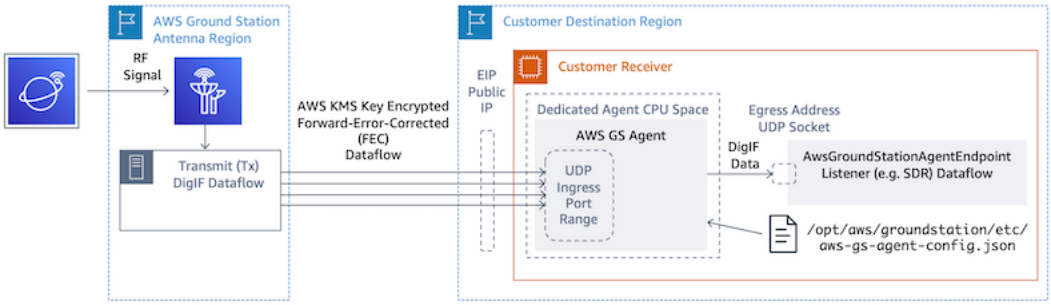
Com o AWS Ground Station Agente, disponível como umRPM, você pode receber (downlink) fluxos de dados síncronos de frequência intermediária digital de banda larga (DigIF) durante os contatos da Ground Station. AWS Você pode selecionar duas opções para entrega de dados:

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 o guia Data Delivery to Amazon Elastic Compute Cloud para obter informações sobre a entrega EC2 de dados.
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 AWS recursos. O uso de CloudFormation para criar seus AWS recursos 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.

> **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) EC2 instância (s).

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 () ou ouvinte similar. SDR




# Características do 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.

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

1. A VPC para iniciar o EC2 receptor.
2. Uma AWS KMS chave para criptografia/decodificação de dados.
3. Uma SSH chave ou perfil de EC2 instância configurado para o Gerenciador de SSM Sessões.
4. Regras de rede/grupo de segurança para permitir o seguinte:
    1. UDPtráfego AWS Ground Station nas portas especificadas em 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. SSHacesso à sua instância (Observação: como alternativa, você pode usar o Gerenciador de AWS Sessões para acessar sua EC2 instância).
    3. Acesso de leitura a um bucket do S3 acessível ao público para gerenciamento de agentes.
    4. SSLtráfego 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 VPC configuração que inclua uma sub-rede pública. Consulte o Guia do VPC usuário para obter informações básicas sobre a configuração da sub-rede.

Configurações compatíveis:

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

Você pode usar o mesmo grupo de segurança da sua EC2 instância ou especificar um com pelo menos o conjunto mínimo de regras que consiste em:

- UDPtráfego AWS Ground Station nas portas especificadas em seu grupo de endpoints de fluxo de dados.

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

## VPCdiagramas

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

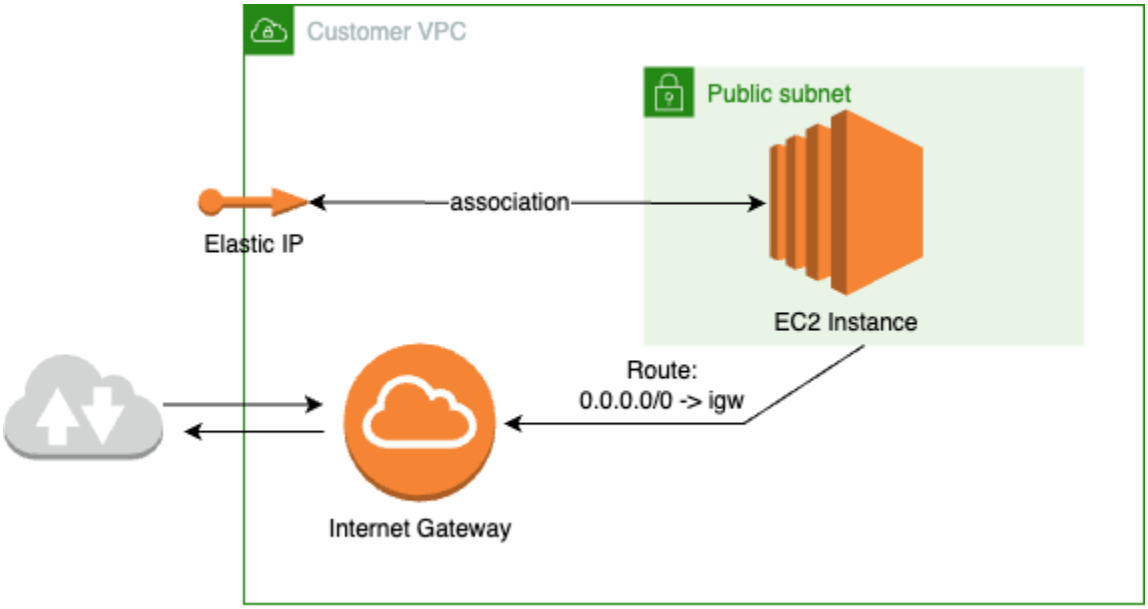
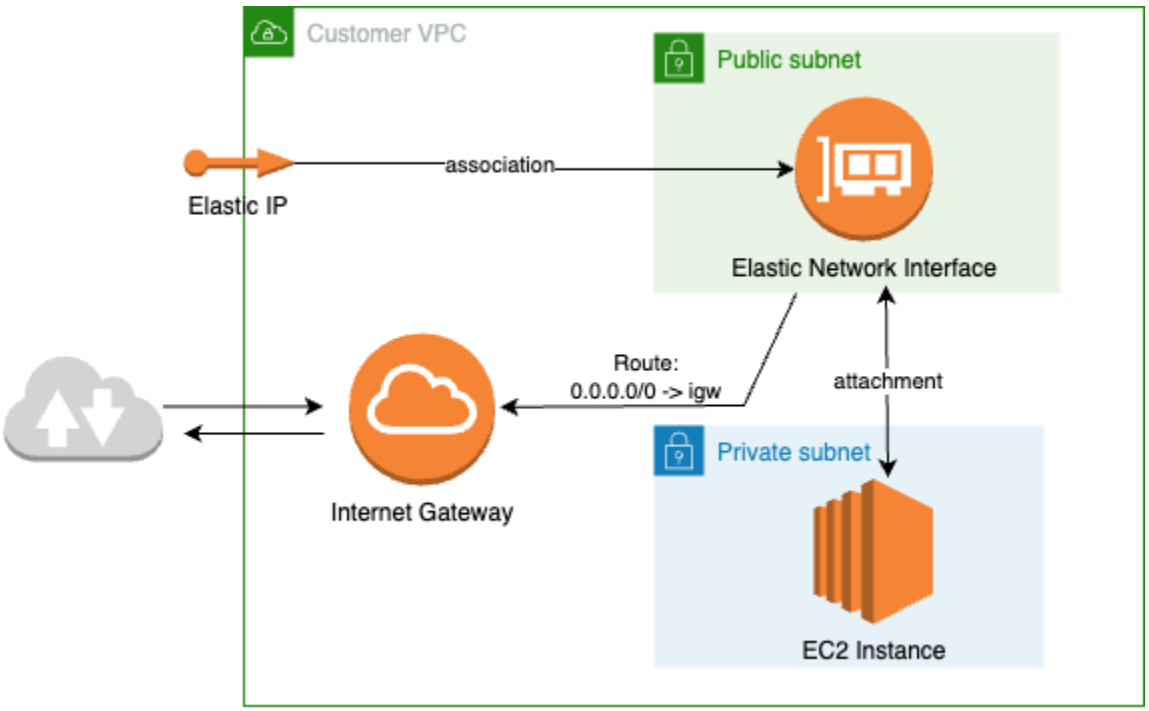



Diagrama: um IP elástico associado a um ENI em uma sub-rede pública, anexado à sua EC2 instância em uma sub-rede privada




# Sistema operacional com suporte

Amazon Linux 2 com kernel 5.10+.

Os tipos de instâncias compatíveis estão listados em Selecione a EC2 instância da Amazon e reserve CPU núcleos 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 AwsGroundStationAgentEndpointegressAddress.

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 endpoint de destino em egressAddress que seu Software Defined Radio () SDR está ouvindo. O AWS Ground Station Agente e seus componentes subjacentes respeitarão os CPU limites 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.

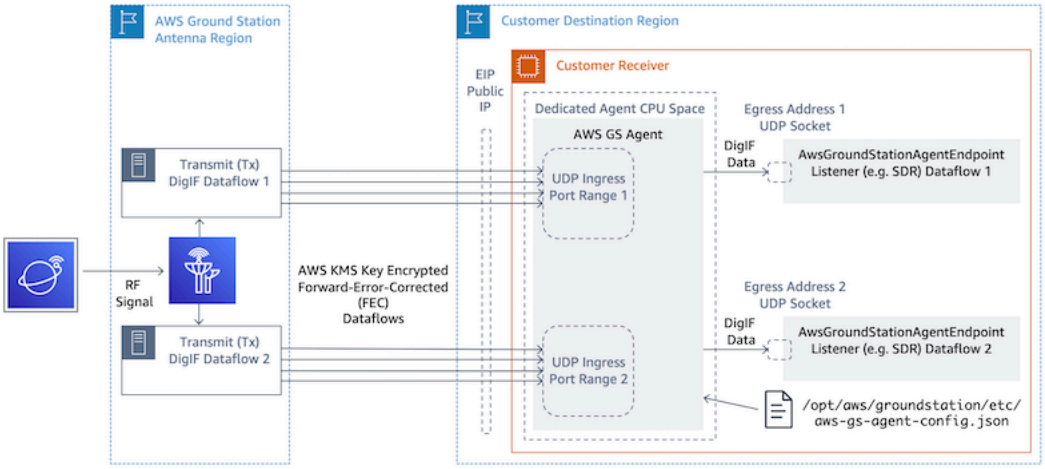
AwsGroundStationAgentEndpoints:

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

CPUPlanejamento:

- 1 núcleo (2 vCPU) para executar o único AWS Ground Station agente na instância.

- 6 núcleos (12 vCPU) para receber o DigiF Dataflow 1 (MHzpesquisa de 200 na tabela). CPUplanejamento básico
- 4 núcleos (8 vCPU) para receber o DigiF Dataflow 2 (100 consultas na tabela)MHz. CPUplanejamento básico
- CPUEspaço total 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.

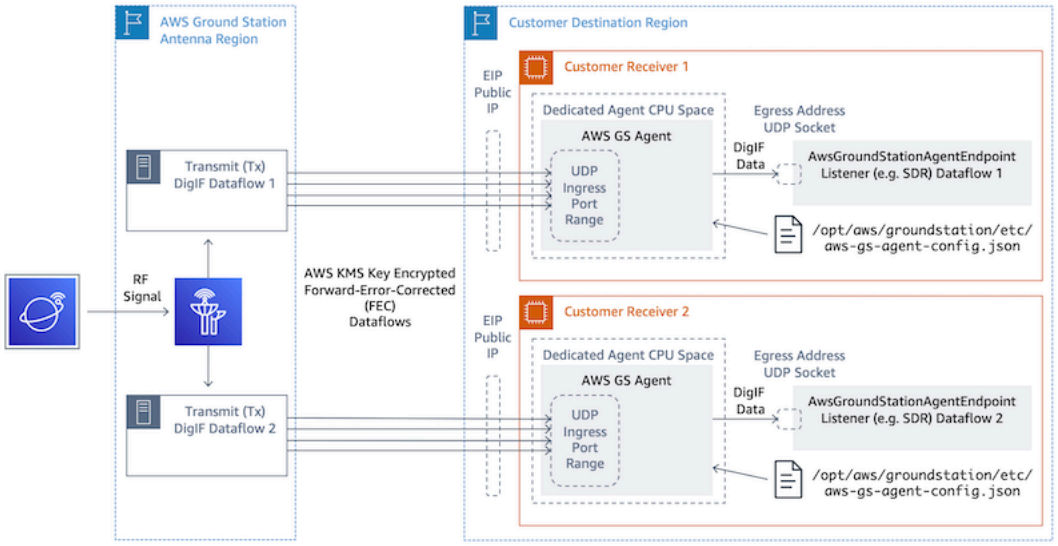
AwsGroundStationAgentEndpoints:

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.

CPUPlanejamento:

- Instância do receptor 1
  - 1 núcleo (2 vCPU) para executar o único AWS Ground Station agente na instância.

  - 9 núcleos (18 vCPU) para receber o DigiF Dataflow 1 (MHzconsulta de 400 na tabela). CPUplanejamento básico
  - CPUEspaço total dedicado para agentes = 10 núcleos (20 vCPU) no mesmo soquete.
- Instância do receptor 2
  - 1 núcleo (2 vCPU) para executar o único AWS Ground Station agente na instância.
  - 9 núcleos (18 vCPU) para receber o DigiF Dataflow 2 (MHzconsulta de 400 na tabela). CPUplanejamento básico
  - CPUEspaço total dedicado para agentes = 10 núcleos (20 vCPU) no mesmo soquete.



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

## Tipos de EC2 instância da Amazon compatíveis

O AWS Ground Station Agente requer CPU núcleos 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 CPUplanejamento básico para decidir qual tipo de instância é mais adequado ao seu caso de uso.

| Tipo de instância | Padrão vCPUs | CPUNúcleos padrão |
|---|---|---|
| c5.12xlarge | 48 | 24 |
| c5.18xlarge | 72 | 36 |
| c5.24xlarge | 96 | 48 |
| c5n.18xlarge | 72 | 36 |
| c5n.metal | 72 | 36 |
| c6i.32xlarge | 128 | 64 |
| g4dn.12xlarge | 48 | 24 |
| g4dn.16xlarge | 64 | 32 |
| g4dn.metal | 96 | 48 |
| m5.12xlarge | 48 | 24 |
| m5.24xlarge | 96 | 48 |
| m6i.32xlarge | 128 | 64 |
| p3dn.24xlarge | 96 | 48 |
| p4d.24xlarge | 96 | 48 |

| Tipo de instância | Padrão vCPUs | CPUNúcleos padrão |
|---|---|---|
| r5.24xlarge | 96 | 48 |
| r5.metal | 96 | 48 |
| r5n.24xlarge | 96 | 48 |
| r5n.metal | 96 | 48 |
| r6i.32xlarge | 128 | 64 |

# CPUplanejamento básico

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 Hyper-threaded (HT) e exige que CPU os pares HT sejam reservados para seu uso. Um par hiperencadeado é um par de virtuais CPUs (vCPU) contidos em um único núcleo. A tabela a seguir fornece 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. Essa tabela pressupõe o Cascade Lake ou mais recente CPUs e é válida para qualquer tipo de instância compatível. Se sua largura de banda estiver entre as entradas na tabela, selecione a próxima mais alta.

O agente precisa de um núcleo reservado adicional para gerenciamento e coordenação, portanto, o total de núcleos necessários será a soma dos núcleos necessários (da tabela abaixo) para cada fluxo de dados mais um único núcleo adicional (2 vCPUs).

| AntennaDownlink Largura de banda () MHz | Taxa de dados VITA esperada de -49,2 DigiF (MB/s) | Número de núcleos (CPUpares HT) | Total v CPU |
|---|---|---|---|
| 50 | 1000 | 3 | 6 |
| 100 | 2000 | 4 | 8 |
| 150 | 3000 | 5 | 10 |

| AntennaDownlink Largura de banda () MHz | Taxa de dados VITA esperada de -49,2 DigiF (MB/s) | Número de núcleos (CPUpares HT) | Total v CPU |
|---|---|---|---|
| 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 vCPUs (rotulados como "CPU“) pertencem a quais NUMA nós (e cada NUMA nó 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 númerovCPUs, 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 vCPUs para cada núcleo atribuído. Todos os núcleos de um fluxo de dados devem existir no mesmo NUMA nó. A `-p` opção do `lscpu` comando nos fornece o núcleo das CPU associações 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 Núcleo 0 inclui vCPUs 0 e 16, o Núcleo 1 inclui vCPUs 1 e 17, o Núcleo 2 inclui vCPUs 2 e 18. Em outras palavras, os pares hiperencadeados são: 0 e 16, 1 e 17, 2 e 18.

# CPUexemplo de tarefa

Como exemplo, usaremos uma `c5.24xlarge` instância para um downlink de banda larga de polaridade dupla em 350. MHz A partir da tabela abaixo, CPUplanejamento básico sabemos que um MHz downlink 350 requer 8 núcleos (16vCPUs) 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 (32vCPUs) mais um núcleo (2vCPUs) 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 Node. NUMA Em seguida, procuraremos o correspondente vCPUs (rotulado como “CPU”) na `lscpu -p` saída emApêndice: `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 vCPUs 2-9 e 50-57.
- Fluxo 2: selecione os núcleos 10-17 que são mapeados para vCPUs 10-17 e 58-65.
- Núcleo do agente: selecione o núcleo 18, que mapeia para vCPUs 18 e 66.

Isso resulta em vCPUs 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 nelesCPUs, conforme descrito emExecutando 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. AWS CloudFormation modelo (recomendado).
2. Instalação manual na AmazonEC2.

## Use um AWS CloudFormation modelo

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

### Etapa 1: criar AWS recursos

Crie sua pilha de AWS recursos usando o modelo de satélite de transmissão pública utilizando 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 EC2 instância (SSHou ao Gerenciador de SSM Sessões) e verAWS Ground Station Status do agente.

## Instale manualmente em EC2

Embora a Ground Station recomende o uso de CloudFormation modelos para provisionar seus AWS recursos, 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 seus requisitos, você pode criar manualmente seus AWS recursos e instalar o agente.

### Etapa 1: criar AWS recursos

Consulte exemplos de configurações de perfil de missão para obter instruções sobre como configurar manualmente os AWS recursos 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 API documentação esteja localizada na APIReferência, esta seção discutirá brevemente os conceitos relevantes para o AWS Ground Station Agente.

O endpoint `ingressAddress` é onde o AWS Ground Station agente receberá UDP tráfego AWS KMS criptografado da antena. `socketAddressname`É o IP público da EC2 instância (do anexoEIP). `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 UDP entrada no grupo de segurança em VPC que a instância do receptor está sendo executada.

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

## Etapa 2: criar EC2 instância

Os seguintes AMIs são compatíveis:

1. AWS Ground Station AMI- `groundstation-al2-gs-agent-ami-*` onde * é a data em AMI que foi criado - 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 escolheu o AWS Ground Station Agente AMI na etapa anterior.

### Baixar o agente

O AWS Ground Station agente está disponível em buckets S3 específicos da região e pode ser baixado em EC2 instâncias de suporte usando a linha de AWS comando (CLI) de `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.

Exemplo: baixe a versão rpm mais recente da AWS região us-east-2 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, você pode baixá-la da pasta específica da versão no bucket do S3.

Exemplo: baixe a versão 1.0.2716.0 do rpm da região AWS us-east-2 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 se o RPM download foi vendido por AWS Ground Station, siga as instruções paraRPMvalidação de instalação.

### 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 Agente AMI: Se você escolheu o AWS Ground Station Agente AMI na etapa anterior, aplique os seguintes ajustes de desempenho.

- Ajuste interrupções de hardware e filas de recebimento - impactos CPU e rede
- Portas de entrada de reserva - impacta a rede
- Reinicializar

Outros AMIs: Se você escolheu qualquer outro AMI na etapa anterior, aplique todos os ajustes listados abaixo Ajuste sua EC2 instância para 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

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, pare.

```
#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 fará com que rename /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 AWS recursos. 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 RPMInformações 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 → String

Exemplos:

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

## Dispositivo

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

Obrigatório: verdadeiro

Formato: objeto

Membros:

- privateIps
- publicIps
- agentCpuCores
- networkAdapters

## privateIps

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 endpoints de fluxo de dados.

Obrigatório: verdadeiro

Formato: String Array

- Valores: Endereços IP → String

Exemplo:

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

## agentCPUCores

Isso especifica quais núcleos virtuais são reservados para o aws-gs-agent processo. Consulte CPUplanejamento básico 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 conectadosENIs, que receberão dados.

Obrigatório: falso

Formato: String Array

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

Exemplo:

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

# Ajuste sua EC2 instância para desempenho

> **Note**
>
> Se você provisionou seus AWS recursos usando CloudFormation modelos, esses ajustes serão aplicados automaticamente. Se você usou uma EC2 instância AMI ou criou manualmente, 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 - impactos CPU e rede

Esta seção configura o uso CPU principal do systemd SMPIRQs, Receive Packet Steering (RPS) e Receive Flow Steering (). RFS Consulte Apêndice: Parâmetros recomendados para interrupção/sintonia 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 CPU núcleos dos agentes.
2. Redirecione as solicitações de interrupção de hardware para fora dos CPU núcleos dos agentes.
3. Configure RPS para evitar que a fila de hardware de uma única placa de interface de rede se torne um gargalo no tráfego da rede.
4. Configure RFS para aumentar a taxa de acerto do CPU cache e, assim, reduzir a latência da rede.

O `set_irq_affinity.sh` script 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 sobrepor aos núcleos selecionados acima. (Ex: '0,1,48,49' para uma instância hyper-threaded de 96-). CPU
- `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 que o cache cuide do balanceamento.
  - Para ver a lista de parâmetros recomendados para cada tipo de instância, consulte Apêndice: Parâmetros recomendados para interrupção/sintonia RPS.
- Exemplo para 96 CPU instâncias:

```
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 EC2 instância.

## 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 ele seja 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 EC2 instância.
- Ao configurar uma instância c6i.32xlarge, o comando precisa ser modificado para definir o buffer de anel como `8192` em vez de `16384`.

## Tune CPU C-State - impactos CPU

Defina o CPU estado C para evitar a inatividade, o que pode causar 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
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 interrupção/sintonia RPS

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

| Família | Tipo de instância | ${interrupt_core_list} | ${rps_core_mask} |
|---|---|---|---|
| c6i | • c6i.32xlarge | • 0,1,64,65 | • ffffffff, ffffffff, ffffffff, ffffffff |
| c5 | • c5.24xlarge<br>• c5.18xlarge<br>• c5.12xlarge | • 0,1,48,49<br>• 0,1,36,37<br>• 0,1,24,25 | • ffffffff, ffffffff, ffffffff<br>• ff,ffffffff,ffffffff |

| Família | Tipo de instância | ${interrupt_core_list} | ${rps_core_mask} |
|---|---|---|---|
| | | | • ffff,ffffffff |
| c5n | • c5n.metal<br>• c5n.18xlarge | • 0,1,36,37<br>• 0,1,36,37 | • ff,ffffffff,ffffffff<br>• ff,ffffffff,ffffffff |
| m5 | • m5.24xlarge<br>• m5.12xlarge | • 0,1,48,49<br>• 0,1,24,25 | • ffffffff, ffffffff, ffffffff<br>• ffff,ffffffff |
| r5 | • r5.metal<br>• r5.24xlarge | • 0,1,48,49<br>• 0,1,48,49 | • ffffffff, ffffffff, ffffffff<br>• ffffffff, ffffffff, ffffffff |
| r5n | • r5n.metal<br>• r5n.24xlarge | • 0,1,48,49<br>• 0,1,48,49 | • ffffffff, ffffffff, ffffffff<br>• ffffffff, ffffffff, ffffffff |
| g4dn | • g4dn.metal<br>• g4dn.16xlarge<br>• g4dn.12xlarge | • 0,1,48,49<br>• 0,1,32,33<br>• 0,1,24,25 | • ffffffff, ffffffff, ffffffff<br>• ffffffff, ffffffff<br>• ffff,ffffffff |

| Família | Tipo de instância | ${interrupt_core_list} | ${rps_core_mask} |
| --- | --- | --- | --- |
| p4d | • p4d.24xlarge | • 0,1,48,49 | • ffffffff, ffffffff, ffffffff |
| p3dn | • p3dn.24xlarge | • 0,1,48,49 | • ffffffff, ffffffff, ffffffff |

# Prepare-se para receber um contato DigIF

1. Analise o planejamento CPU básico para ver os fluxos de dados desejados e forneça uma lista dos núcleos que o agente pode usar. Consulte CPUplanejamento básico.
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 EC2 instância para 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 `agentStatus` o seu `awsGroundStationAgentEndpoint` é ACTIVE e o `auditResults` éHEALTHY.

# Práticas recomendadas

## EC2Melhores práticas da Amazon

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

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

## Agendador Linux

O programador Linux pode reordenar pacotes em UDP soquetes se os processos correspondentes não estiverem fixados em um núcleo específico. Qualquer thread que envie ou receba UDP dados 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` AWS -managed ao especificar as regras de rede para permitir a comunicação da antena. Consulte Trabalhando com listas de prefixos AWS gerenciadas para obter mais informações sobre listas de prefixos AWS gerenciadas.

## Limitação de contato único

O AWS Ground Station Agent suporta vários fluxos 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 várias configurações diferentes DFEGARNs, ela falhará no registro.

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

Ao iniciar serviços e processos na mesma EC2 instância do AWS Ground Station agente, é importante vinculá-los a vCPUs não serem usados pelo AWS Ground Station agente e pelo kernel Linux, pois isso pode causar gargalos e até perda de dados durante os contatos. Esse conceito de vinculação ao específico vCPUs 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 - impactos CPU e rede.
  - Os valores padrão podem ser encontrados em Apêndice: Parâmetros recomendados para interrupção/sintonia RPS

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

Se você especificou

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

e correu

```
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 `vinano`, 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 CPU afinidade e NUMA políticas usando o 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 `pidoftop`, 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 via: get-dataflow-endpoint-group

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

Verifique se `agentStatus` o seu `awsGroundStationAgentEndpoint` é ACTIVE e o `auditResults` éHEALTHY.

# 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 1.0.3555.0

Data de lançamento: 27/03/2024

Data de fim do Support: 31/08/2024

RPMSomas de verificação:

- 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 em stdout/journal em vez de no 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.

## Versões obsoletas do agente

### Versão 1.0.2942.0

Data de lançamento: 26/06/2023

Data de fim do Support: 31/05/2024

RPMSomas de verificação:

- SHA256: `7d94b642577504308a58bab28f938507f2591d4e1b2c7ea170b77bea97b5a9b6`
- MD5: `661ff2b8f11aba5d657a6586b56e0d8f`

Alterações:

- Foram adicionados registros de erros para quando o Agente RPM é atualizado no disco e precisa ser reiniciado 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 impedir 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

RPMSomas de verificação:

- 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

RPMSomas de verificação:

- SHA256: `77cfe94acb00af7ca637264b17c9b21bd7afdc85b99dffdd627aec9e99397489`
- MD5: `b8533be7644bb4d12ab84de21341adac`

Alterações:

- Primeira versão do Agent disponível ao público em geral.

# RPMvalidação de instalação

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

## Versão mais recente do agente

### Versão 1.0.3555.0

Data de lançamento: 27/03/2024

Data de fim do Support: 31/08/2024

RPMSomas de verificação:

- 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 em stdout/journal em vez de no 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.

# Verifique o RPM

As ferramentas necessárias para verificar essa RPM instalação são:

- sha256sum
- rpm

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

- Calcule a soma sha256 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 RPM arquivo baixado é um arquivo válido para ser usado e que a AWS Ground Station vendeu aos clientes. Se os hashes não corresponderem, não instale o RPM e exclua-o da instância de computação.
- Verifique também o MD5 hash do arquivo, para garantir que ele não RPM tenha sido comprometido. Para fazer isso, use a ferramenta de linha de RPM comando 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 à tabela listada no AWS Docs, o cliente poderá ter certeza de que o RPM que foi 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 |
| --- | --- | --- |
| Atualização da documentação | Suporte removido para a família de instâncias mais antiga: m4. | 30 de setembro de 2024 |
| Atualização da documentação | Foi adicionado um comentário sobre como manter a sub-rede e a EC2 instância da Amazon na mesma zona de disponibilidade nos Requisitos do agente. | 18 de julho de 2024 |
| Atualização da documentação | Divida o AWS Ground Station Agente em seu próprio guia do usuário. Para alterações anteriores, consulte: Histórico de documentos do guia do usuário da AWS Ground Station. | 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.