



AWS Ground Station Guía del usuario del agente

AWS Ground Station



AWS Ground Station: AWS Ground Station Guía del usuario del agente

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Información general	1
¿Qué es el AWS Ground Station agente?	1
Características del agente AWS Ground Station	2
Requisitos del agente	3
VPCdiagramas	4
Sistemas operativos compatible	5
Entrega de datos mediante AWS Ground Station un agente	6
Varios flujos de datos, un solo receptor	6
Múltiples flujos de datos, múltiples receptores	7
Selección y CPU planificación de EC2 instancias de Amazon	9
Tipos de EC2 instancias de Amazon compatibles	9
CPUplanificación básica	10
Recopilación de información sobre la arquitectura	11
CPUejemplo de asignación	12
Apéndice: <code>lscpu -p salida (completa)</code> para <code>c5.24xlarge</code>	13
Instalación del agente	17
Uso de AWS CloudFormation la plantilla	17
Paso 1: Crear recursos AWS	17
Paso 2: Compruebe el estado del agente	17
Instalación manual en EC2	17
Paso 1: Crear AWS recursos	17
Paso 2: Crear una EC2 instancia	18
Paso 2: descargar e instalar el agente	18
Paso 4: configurar el agente	20
Paso 5: aplicar ajustes de rendimiento	20
Paso 6: administrar el agente	20
Administrar el agente	21
AWS Ground Station Configuración del agente	21
AWS Ground Station Inicio del agente	21
AWS Ground Station Agente, deténgase	22
AWS Ground Station Actualización del agente	22
AWS Ground Station Bajar de categoría de agente	23
AWS Ground Station Desinstalación del agente	24
AWS Ground Station Estado del agente	24

AWS Ground Station RPMInformación del agente	25
Configuración del agente	26
Archivo de configuración del agente	26
Ejemplo	26
Desglose de campos	26
EC2ajuste del rendimiento de la instancia	30
Ajuste las interrupciones del hardware y las colas de recepción: impactos y red CPU	30
La fusión de interrupciones de Tune Rx afecta a la red	31
Tune el búfer Rx Rx: afecta a la red	32
Tune CPU C-State: impactos CPU	32
Los puertos de entrada de reserva afectan a la red	33
Reboot	33
Apéndice: Parámetros recomendados para interrumpir RPS o ajustar	33
Prepárese para tener un contacto en DigiF	36
Prácticas recomendadas	37
EC2Mejores prácticas de Amazon	37
Programador de Linux	37
AWS Ground Station lista de prefijos gestionada	37
Limitación de contacto único	37
Ejecutar servicios y procesos junto con el AWS Ground Station agente	37
Como ejemplo, usar una c5.24xlarge instancia	38
Servicios de afinización (systemd)	38
Procesos de afinización (scripts)	39
Resolución de problemas	41
El agente no se puede iniciar	41
Resolución de problemas	41
AWS Ground Station Registros del agente	42
No hay contactos disponibles	42
Cómo obtener asistencia	43
Notas de lanzamiento del agente	44
Última versión del agente	44
Versión 1.0.3555.0	44
Versiones de agentes obsoletas	44
Versión 1.0.2942.0	44
Versión 1.0.2716.0	45
Versión 1.0.2677.0	46

RPMvalidación de la instalación	47
Versión más reciente del agente	44
Versión 1.0.3555.0	44
Compruebe la RPM	48
Historial de documentos	49
.....	

Información general

¿Qué es el AWS Ground Station agente?

El AWS Ground Station Agent, disponible como agenteRPM, le permite recibir flujos de datos de frecuencia intermedia digital de banda ancha (DigiF) síncronos (enlace descendente) durante los contactos con la estación Ground. AWS Puede seleccionar dos opciones para la entrega de datos:

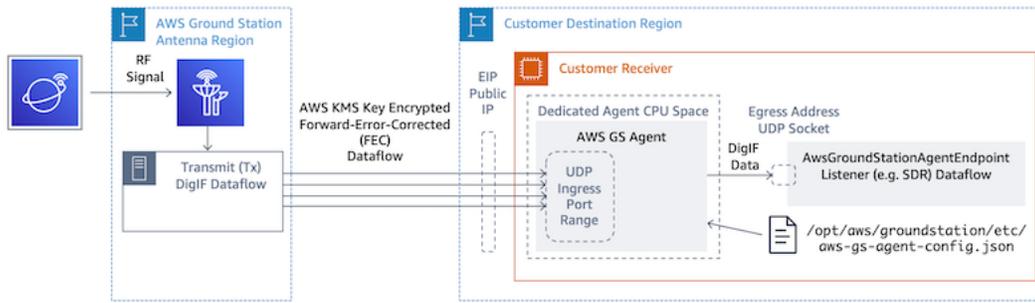
1. Entrega de datos a una EC2 instancia: entrega de datos a una EC2 instancia de tu propiedad. Usted administra el AWS Ground Station agente. Esta opción puede ser la más adecuada si necesita un procesamiento de datos casi en tiempo real. Consulte la guía [Data Delivery to Amazon Elastic Compute Cloud](#) para obtener información sobre la entrega de EC2 datos.
2. Entrega de datos a un depósito de S3: entrega de datos a un depósito de AWS S3 de su propiedad a través de un servicio gestionado de Ground Station. Consulte la AWS Ground Station guía de [introducción para obtener](#) información sobre la entrega de datos en S3.

Ambos modos de entrega de datos requieren la creación de un conjunto de AWS recursos. Se recomienda encarecidamente el uso de CloudFormation para crear sus AWS recursos a fin de garantizar la fiabilidad, la precisión y la compatibilidad. Cada contacto solo puede enviar datos a EC2 o S3, pero no a ambos simultáneamente.

Note

Dado que la entrega de datos de S3 es un servicio gestionado por Ground Station, esta guía se centra en la entrega de datos a sus EC2 instancias.

El siguiente diagrama muestra un flujo de datos DigiF desde una región de AWS Ground Station antena a su EC2 instancia con su radio definida por software () o un oyente similar. SDR



Características del agente AWS Ground Station

El AWS Ground Station agente recibe datos de enlace descendente de frecuencia intermedia digital (DigiF) y saca los datos descifrados que permiten lo siguiente:

- Capacidad de enlace descendente DigiF de 40 MHz a 400 MHz de ancho de banda.
- Entrega de datos DigiF de alta velocidad y baja fluctuación a cualquier IP pública AWS (IP elástica) de la AWS red.
- Entrega de datos fiable mediante Forward Error Correction (FEC).
- Entrega segura de datos mediante una AWS KMS clave de cifrado gestionada por el cliente.

Requisitos del agente

Note

En esta guía para AWS Ground Station agentes se supone que se ha incorporado a Ground Station mediante la guía de [AWS Ground Station introducción](#).

La EC2 instancia AWS Ground Station Agent Receiver requiere un conjunto de AWS recursos dependientes para entregar los datos DigiF de manera confiable y segura a sus puntos finales.

1. A VPC en el que lanzar el EC2 receptor.
2. Una AWS KMS clave para el cifrado/descifrado de datos.
3. [Una SSH clave o un perfil de EC2 instancia configurado para SSM el administrador de sesiones](#).
4. Reglas de red/grupo de seguridad que permiten lo siguiente:
 1. UDPtráfico procedente de los puertos especificados AWS Ground Station en el grupo de puntos finales de su flujo de datos. El agente reserva un rango de puertos contiguos que se utilizan para entregar datos a los puntos de conexión del flujo de datos de entrada.
 2. SSHacceso a su instancia (Nota: también puede usar el Administrador de AWS sesiones para acceder a su EC2 instancia).
 3. Acceso de lectura a un bucket de S3 de acceso público para la administración de agentes.
 4. SSLtráfico en el puerto 443 que permite al agente comunicarse con el AWS Ground Station servicio.
 5. Tráfico de la lista `com.amazonaws.global.groundstation` de prefijos AWS Ground Station gestionada.

Además, se requiere una VPC configuración que incluya una subred pública. Consulte la [Guía del VPC usuario](#) para obtener información básica sobre la configuración de subredes.

Configuraciones compatibles:

1. Una IP elástica asociada a su EC2 instancia en una subred pública.
2. Una IP elástica asociada a una ENI subred pública, conectada a la EC2 instancia (en cualquier subred de la misma zona de disponibilidad que la subred pública).

Puede usar el mismo grupo de seguridad que su EC2 instancia o especificar uno con, al menos, el conjunto mínimo de reglas compuesto por:

- UDP tráfico procedente de los puertos especificados AWS Ground Station en el grupo de puntos finales de su flujo de datos.

Para ver ejemplos de plantillas de entrega de AWS CloudFormation EC2 datos con estos recursos preconfigurados, consulte [Public broadcast satellite using AWS Ground Station Agent \(banda ancha\)](#).

VPC diagramas

Diagrama: una IP elástica asociada a tu EC2 instancia en una subred pública

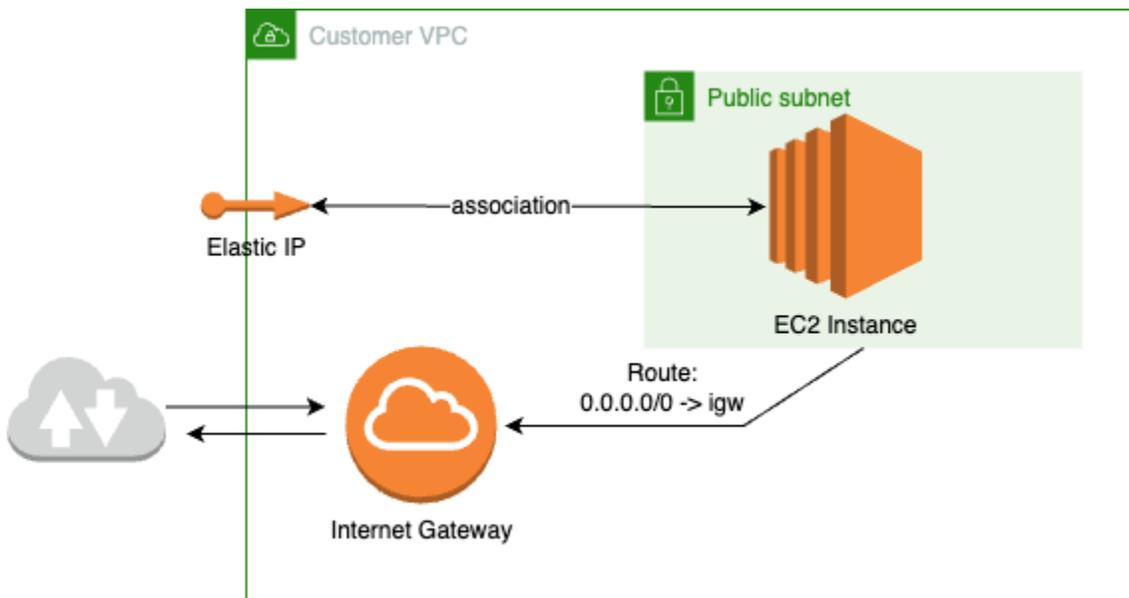
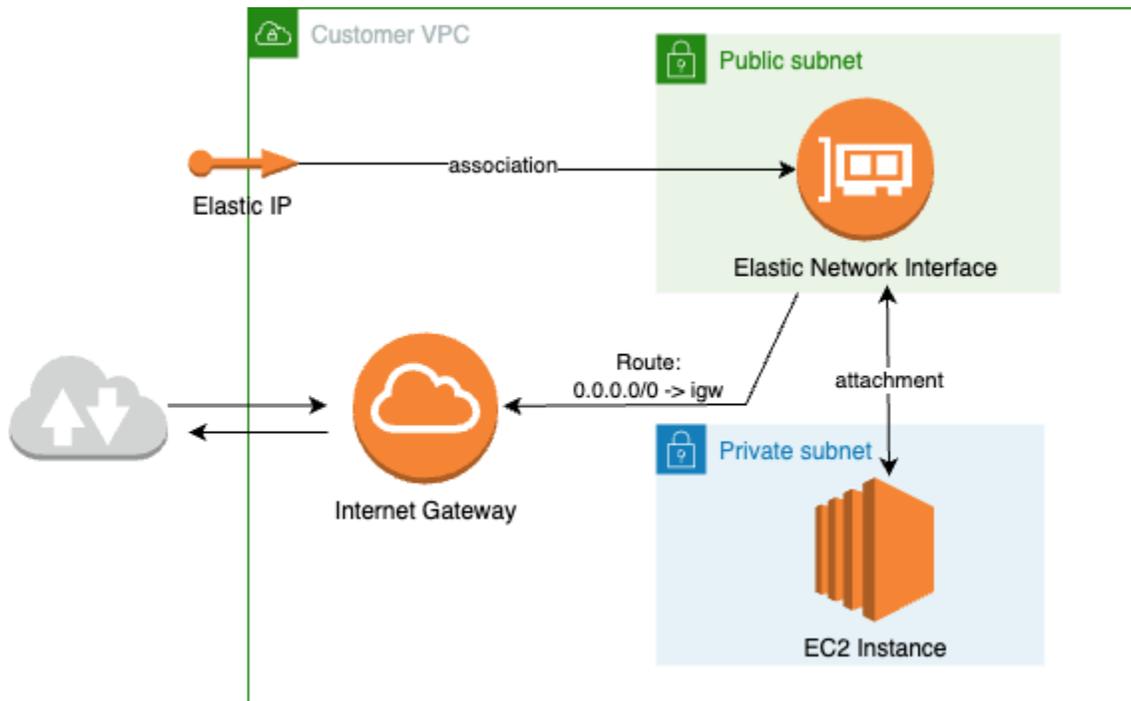


Diagrama: una IP elástica asociada a una de una ENI subred pública y conectada a tu EC2 instancia en una subred privada



Sistemas operativos compatible

Amazon Linux 2 con kernel 4.14.186

Los tipos de instancias compatibles se muestran en [Selección y CPU planificación de EC2 instancias de Amazon](#)

Entrega de datos mediante AWS Ground Station un agente

Los siguientes diagramas proporcionan una descripción general de cómo fluyen los datos AWS Ground Station durante los contactos de frecuencia intermedia digital de banda ancha (DigiF).

El AWS Ground Station agente se encargará de organizar los componentes del plano de datos de un contacto. Antes de programar un contacto, el agente debe estar correctamente configurado, iniciado y registrado (el registro es automático al iniciar el agente) en él. AWS Ground Station Además, el software de recepción de datos (como una radio definida por software) debe estar funcionando y configurado para recibir datos en el [AwsGroundStationAgentEndpoint](#) egressAddress.

Entre bastidores, el AWS Ground Station agente recibirá tareas AWS Ground Station y deshará el AWS KMS cifrado que se aplicó durante el tránsito, antes de reenviarlo al egressAddress punto final de destino en el que está escuchando su radio definida por software (SDR). El AWS Ground Station agente y sus componentes subyacentes respetarán los CPU límites establecidos en el archivo de configuración para garantizar que esto no afecte al rendimiento de otras aplicaciones que se ejecuten en la instancia.

Debe tener el AWS Ground Station agente ejecutándose en la instancia receptora implicada en el contacto. Un solo AWS Ground Station agente puede organizar varios flujos de datos, como se indica a continuación, si prefiere recibir todos los flujos de datos en una sola instancia receptora.

Varios flujos de datos, un solo receptor

Escenario de ejemplo:

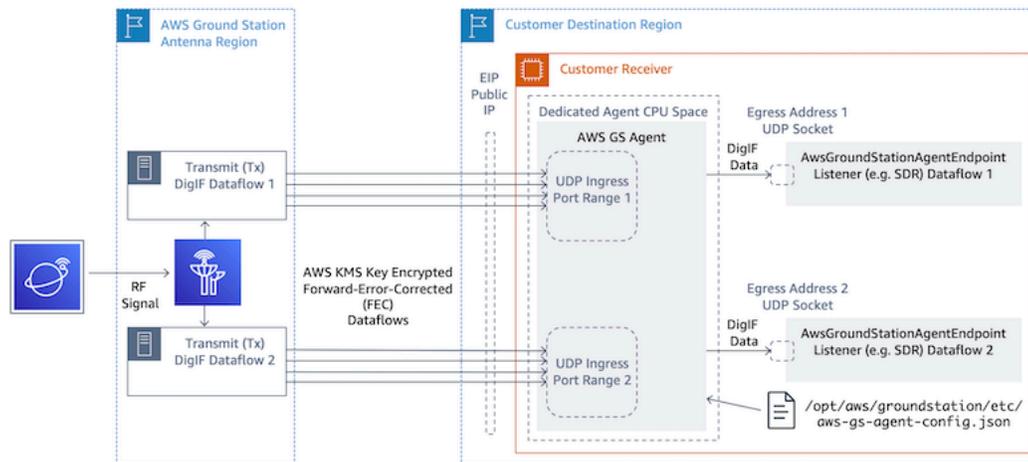
Le gustaría recibir dos enlaces descendentes de antena como flujos de datos DigiF en la misma instancia del receptor. EC2 Los dos enlaces descendentes serán 200 y 100. MHz MHz

AwsGroundStationAgentEndpoints:

Habrán dos recursos de `AwsGroundStationAgentEndpoint`, uno para cada flujo de datos. Ambos puntos de conexión tendrán la misma dirección IP pública (`ingressAddress.socketAddress.name`). Las entradas `portRange` no deben superponerse, ya que los flujos de datos se reciben en la misma instancia. EC2 Ambos `egressAddress.socketAddress.port` deben ser únicos.

CPUPlanificación:

- 1 núcleo (2 vCPU) para ejecutar el AWS Ground Station agente único en la instancia.
- 6 núcleos (12 vCPU) para recibir DigiF Dataflow 1 (MHzconsulta 200 en la tabla). [CPUplanificación básica](#)
- 4 núcleos (8 vCPU) para recibir DigiF Dataflow 2 (MHzconsulta 100 en la tabla). [CPUplanificación básica](#)
- CPUEspacio total de agente dedicado = 11 núcleos (22 vCPU) en el mismo conector.



Múltiples flujos de datos, múltiples receptores

Escenario de ejemplo:

Le gustaría recibir dos enlaces descendentes de antena como flujos de datos DigiF en diferentes instancias del receptor. EC2 Ambos enlaces descendentes serán 400. MHz

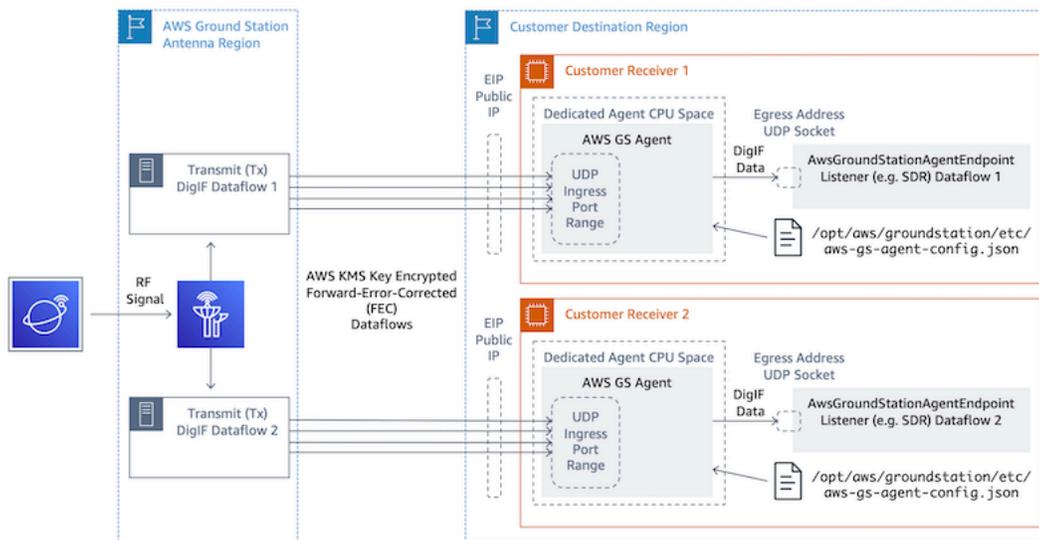
AwsGroundStationAgentEndpoints:

Habrán dos recursos de `AwsGroundStationAgentEndpoint` recursos, uno para cada flujo de datos. Ambos puntos de conexión tendrán la misma dirección IP pública (`ingressAddress.socketAddress.name`). No hay restricciones en los valores de los puertos para `ingressAddress` y `egressAddress`, ya que los flujos de datos se reciben en infraestructuras separadas y no entrarán en conflicto entre sí.

CPUPlanificación:

- Instancia de receptor 1
 - 1 núcleo (2 vCPU) para ejecutar el AWS Ground Station agente único en la instancia.

- 9 núcleos (18 vCPU) para recibir DigiF Dataflow 1 (MHzconsulta 400 en la tabla).
[CPUplanificación básica](#)
- CPUEspacio total de agente dedicado = 10 núcleos (20 vCPU) en el mismo conector.
- Instancia de receptor 2
 - 1 núcleo (2 vCPU) para ejecutar el único AWS Ground Station agente en la instancia.
 - 9 núcleos (18 vCPU) para recibir DigiF Dataflow 2 (MHzconsulta 400 en la tabla).
[CPUplanificación básica](#)
 - CPUEspacio total de agente dedicado = 10 núcleos (20 vCPU) en el mismo conector.



Selección y CPU planificación de EC2 instancias de Amazon

Tipos de EC2 instancias de Amazon compatibles

El AWS Ground Station agente requiere CPU núcleos dedicados para funcionar debido a los flujos de trabajo de entrega de datos que requieren un uso intensivo de cómputo. Admitimos los siguientes tipos de instancias. Consulte [CPU planificación básica](#) para decidir qué tipo de instancia se adapta mejor a su caso de uso.

Tipo de instancia	Predeterminado vCPUs	CPUNúcleos predeterminados
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
m4.16xlarge	64	32
m5.12xlarge	48	24
m5.24xlarge	96	48
m6i.32xlarge	128	64
p3dn.24xlarge	96	48
p4d.24xlarge	96	48

Tipo de instancia	Predeterminado vCPUs	CPUNúcleos predeterminados
r5.24xlarge	96	48
r5.metal	96	48
r5n.24xlarge	96	48
r5n.metal	96	48
r6i.32xlarge	128	64

CPUplanificación básica

El AWS Ground Station agente requiere núcleos de procesador dedicados que compartan la caché L3 para cada flujo de datos. El agente está diseñado para aprovechar los pares Hyper-Threading (HT) y requiere que los CPU pares HT estén reservados para su uso. Un par hiperproceso es un par de virtuales CPUs (vCPU) que se encuentran dentro de un único núcleo. La siguiente tabla proporciona un mapeo de la velocidad de datos del flujo de datos al número requerido de núcleos reservados para el agente en un solo flujo de datos. En esta tabla se presupone Cascade Lake o una versión más reciente CPUs y es válida para cualquier tipo de instancia compatible. Si tu ancho de banda se encuentra entre las entradas de la tabla, selecciona la siguiente más alta.

El agente necesita un núcleo reservado adicional para la administración y la coordinación, por lo que el total de núcleos necesarios será la suma de los núcleos necesarios (según el gráfico siguiente) para cada flujo de datos más un núcleo adicional (2 vCPUs).

AntennaDownlink Ancho de banda () MHz	Velocidad de datos DigiF esperada de VITA -49.2 (MB/s)	Número de núcleos (pares HT) CPU	Total v CPU
50	1 000	3	6
100	2000	4	8
150	3 000	5	10

AntennaDownlink Ancho de banda () MHz	Velocidad de datos DigiF esperada de VITA -49.2 (MB/s)	Número de núcleos (pares HT) CPU	Total v CPU
200	4000	6	12
250	5000	6	12
300	6000	7	14
350	7000	8	16
400	8000	9	18

Recopilación de información sobre la arquitectura

`lscpu` proporciona información sobre la arquitectura del sistema. El resultado básico muestra cuáles vCPUs (etiquetados como "CPU«) pertenecen a qué NUMA nodos (y cada NUMA nodo comparte una caché L3). A continuación, examinamos una `c5.24xlarge` instancia para recopilar la información necesaria para configurar el AWS Ground Station agente. Esto incluye información útil como el número de núcleos vCPUs, y la asociación CPU entre v y el nodo.

```
> 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  <-----

```

Los núcleos dedicados al AWS Ground Station agente deben incluir ambos vCPUs para cada núcleo asignado. Todos los núcleos de un flujo de datos deben estar en el mismo NUMA nodo. La `-p` opción del `lscpu` comando nos proporciona el núcleo de las CPU asociaciones necesarias para configurar el agente. Los campos relevantes son CPU (que es lo que denominamos vCPU), Core y L3 (que indica qué caché L3 comparte ese núcleo). Tenga en cuenta que en la mayoría de los procesadores Intel, el NUMA nodo es igual a la caché L3.

Considere el siguiente subconjunto de la `lscpu -p` salida para a `c5.24xlarge` (abreviado y formateado para mayor claridad).

```

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

```

En el resultado podemos ver que Core 0 incluye vCPUs 0 y 16, Core 1 incluye vCPUs 1 y 17, Core 2 incluye vCPUs 2 y 18. En otras palabras, los pares hiperhilos son: 0 y 16, 1 y 17, 2 y 18.

CPUejemplo de asignación

Como ejemplo, usaremos una `c5.24xlarge` instancia para un enlace descendente de banda ancha de doble polaridad a 350. MHz De la tabla siguiente, [CPUplanificación básica](#) sabemos que un MHz

enlace descendente de 350 grados requiere 8 núcleos (16vCPUs) para el flujo de datos único. Esto significa que esta configuración de doble polaridad que utiliza dos flujos de datos requiere un total de 16 núcleos (32vCPUs) más un núcleo (2vCPUs) para el agente.

Sabemos que el `lscpu` resultado de incluye y. `c5.24xlarge` NUMA node0 CPU(s): 0-23, 48-71 NUMA node1 CPU(s): 24-47, 72-95 Como el NUMA node0 tiene más de lo que necesitamos, solo asignaremos desde los núcleos: 0-23 y 48-71.

En primer lugar, seleccionaremos 8 núcleos para cada flujo de datos que compartan un nodo o caché L3. NUMA Luego buscaremos lo correspondiente vCPUs (denominado «CPU») en la `lscpu -p` salida de. [Apéndice: `lscpu -p` salida \(completa\) para `c5.24xlarge`](#) Un ejemplo de proceso de selección básica podría tener el siguiente aspecto:

- Reserve los núcleos 0-1 para el sistema operativo.
- Flujo 1: seleccione los núcleos 2-9 que se asignen a vCPUs 2-9 y 50-57.
- Flujo 2: seleccione los núcleos 10-17 que se asignen a vCPUs 10-17 y 58-65.
- Núcleo de agente: selecciona el núcleo 18, que se asigna a vCPUs 18 y 66.

Esto da como resultado vCPUs 2-18 y 50-66, por lo que la lista que se debe proporcionar al agente es. [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] Debe asegurarse de que sus propios procesos no se ejecuten en ellos, tal y CPUs como se describe en. [Ejecutar servicios y procesos junto con el AWS Ground Station agente](#)

Tenga en cuenta que los núcleos específicos seleccionados en este ejemplo son un tanto arbitrarios. Otros conjuntos de núcleos funcionarían siempre que satisfagan el requisito de que todos compartan una caché L3 para cada flujo de datos.

Apéndice: `lscpu -p` salida (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
```

Instalación del agente

El AWS Ground Station agente se puede instalar de las siguientes maneras:

1. AWS CloudFormation plantilla (recomendada).
2. Instalación manual en AmazonEC2.

Uso de AWS CloudFormation la plantilla

La AWS CloudFormation plantilla de entrega de EC2 datos crea los AWS recursos necesarios para entregar los datos a la EC2 instancia. Esta AWS CloudFormation plantilla utiliza la plantilla AWS Ground Station gestionada AMI que tiene el AWS Ground Station agente preinstalado. A continuación, el script de arranque de la EC2 instancia creada rellena el archivo de configuración del agente y aplica los ajustes de rendimiento necesarios ([EC2ajuste del rendimiento de la instancia](#)).

Paso 1: Crear recursos AWS

Cree su pila de AWS recursos utilizando la plantilla [Public Broadcast Satellite utilizando AWS Ground Station Agent \(banda ancha\)](#).

Paso 2: Compruebe el estado del agente

De forma predeterminada, el agente está configurado y activo (iniciado). Para comprobar el estado del agente, puede conectarse a la EC2 instancia (SSHo al administrador de SSM sesiones) y ver [AWS Ground Station Estado del agente](#).

Instalación manual en EC2

Si bien Ground Station recomienda usar CloudFormation plantillas para aprovisionar sus AWS recursos, puede haber casos de uso en los que la plantilla estándar no sea suficiente. En estos casos, le recomendamos que personalice la plantilla para adaptarla a sus necesidades. Si eso sigue sin cumplir con sus requisitos, puede crear sus AWS recursos e instalar el agente manualmente.

Paso 1: Crear AWS recursos

Consulta los [ejemplos de configuraciones de perfiles de misión](#) para obtener instrucciones sobre cómo configurar manualmente los AWS recursos necesarios para un contacto.

El `AwsGroundStationAgentEndpointrecurso` define un punto final para recibir un flujo de datos DigiF a través del AWS Ground Station agente y es fundamental para establecer un contacto exitoso. Si bien la API documentación se encuentra en la [APIReferencia](#), en esta sección se analizarán brevemente los conceptos relevantes para el agente. AWS Ground Station

El punto final `ingressAddress` es donde el AWS Ground Station agente recibirá el UDP tráfico AWS KMS cifrado de la antena. `socketAddressnameEs` la IP pública de la EC2 instancia (de la adjuntaEIP). `portRange` debe tener al menos 300 puertos contiguos en un rango que se haya reservado para cualquier otro uso. Para obtener instrucciones, consulte [Los puertos de entrada de reserva afectan a la red](#). Estos puertos deben configurarse para permitir la UDP entrada de tráfico en el grupo de seguridad en el que VPC se ejecuta la instancia receptora.

El punto final `egressAddress` es donde el agente le entregará el flujo de datos de DigiF. Debería tener una aplicación (por ejemploSDR) que reciba los datos a través de un UDP conector en esta ubicación.

Paso 2: Crear una EC2 instancia

Se admiten los siguientes AMIs:

1. AWS Ground Station AMI- `groundstation-a12-gs-agent-ami-*` donde `*` es la fecha en AMI que se creó - viene con el agente instalado (recomendado).
2. `amzn2-ami-kernel-5.10-hvm-x86_64-gp2`.

Paso 2: descargar e instalar el agente

Note

Si no eligió el AWS Ground Station agente en el paso anterior, debe completar los pasos AMI de esta sección.

Descargar el agente

El AWS Ground Station agente está disponible en buckets S3 específicos de la región y se puede descargar en las EC2 instancias de soporte mediante la línea de AWS comandos (CLI), desde `s3://groundstation-wb-digif-software- $\{\text{AWS::Region}\}$ /aws-groundstation-`

agent/latest/amazon_linux_2_x86_64/aws-groundstation-agent.rpm donde \$ {AWS: :Region} hace referencia a una de las regiones de [AWSGround Station Console y Data Delivery](#) compatibles.

Ejemplo: descargue localmente la última versión de rpm de AWS la región us-east-2 a la carpeta /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
```

Si necesita descargar una versión específica del AWS Ground Station agente, puede descargarla desde la carpeta específica de la versión en el bucket de S3.

Ejemplo: descargue la versión 1.0.2716.0 del rpm de la región AWS us-east-2 localmente a la carpeta /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

Si quiere confirmar que lo que ha descargado ha sido vendido porRPM, siga las instrucciones para. AWS Ground Station [RPMvalidación de la instalación](#)

Instalar el agente

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

Example: Assumes agent is in the "/tmp" directory

```
sudo yum install /tmp/aws-groundstation-agent.rpm
```

Paso 4: configurar el agente

Tras instalar el agente, debe actualizar el archivo de configuración del agente. Consulte [Configuración del agente](#).

Paso 5: aplicar ajustes de rendimiento

AWS Ground Station Agente AMI: Si eligió el AWS Ground Station agente AMI en el paso anterior, aplique los siguientes ajustes de rendimiento.

- [Ajuste las interrupciones del hardware y las colas de recepción: impactos y red CPU](#)
- [Los puertos de entrada de reserva afectan a la red](#)
- [Reboot](#)

Otros AMIs: Si eligió cualquier otro AMI en el paso anterior, aplique todos los ajustes que se enumeran en la lista [EC2ajuste del rendimiento de la instancia](#) y reinicie la instancia.

Paso 6: administrar el agente

Para iniciar, detener y comprobar el estado del agente, consulte [Administrar el agente](#).

Administrar el agente

AWS Ground Station El agente proporciona las siguientes funciones para configurar, iniciar, detener, actualizar, degradar y desinstalar el agente mediante las herramientas de comandos integradas en Linux.

Temas

- [AWS Ground Station Configuración del agente](#)
- [AWS Ground Station Inicio del agente](#)
- [AWS Ground Station Agente, deténgase](#)
- [AWS Ground Station Actualización del agente](#)
- [AWS Ground Station Bajar de categoría de agente](#)
- [AWS Ground Station Desinstalación del agente](#)
- [AWS Ground Station Estado del agente](#)
- [AWS Ground Station RPM Información del agente](#)

AWS Ground Station Configuración del agente

Navegue hasta `/opt/aws/groundstation/etc`, que debe contener un único archivo denominado `aws-gs-agent-config.json`. Consulte [Archivo de configuración del agente](#)

AWS Ground Station Inicio del agente

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

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

Debería producir un resultado que muestre que el agente está activo.

```
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, deténgase

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

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

Debería producir un resultado que muestre que el agente está inactivo (detenido).

```
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 Actualización del agente

1. Descargue la versión más reciente del agente. Consulte [Descargar el agente](#).
2. Detenga el agente de .

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

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

3. Actualización del 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 Bajar de categoría de agente

1. Descarga la versión de agente que necesites. Consulte [Descargar el agente](#).
2. Descargue el 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 Desinstalación del agente

Al desinstalar el agente, se cambiará el nombre de `/opt/aws/groundstation/etc/.json` a `/opt/aws/groundstation/etc/.json.rpmsaveaws-gs-agent-config.aws-gs-agent-config`. Si vuelves a instalar el agente en la misma instancia, se escribirán los valores predeterminados de `.json` y tendrás que actualizarlos con los valores correctos correspondientes a tus recursos. `aws-gs-agent-config` AWS. Consulte [Archivo de configuración del agente](#).

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

AWS Ground Station Estado del agente

El estado del agente es activo (el agente está en ejecución) o inactivo (el agente está detenido).

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

Un ejemplo de resultado muestra que el agente está instalado, en estado inactivo (detenido) y activado (inicia el servicio al arrancar).

```
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 RPM Información del agente

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

La salida es la siguiente:

Note

La versión puede variar según la última versión publicada por el 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
```

Configuración del agente

Tras instalar el agente, debe actualizar el archivo de configuración del agente en `/opt/aws/groundstation/etc/aws-gs-agent-config.json`.

Archivo de configuración del agente

Ejemplo

```
{
  "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
  ]
}
```

Desglose de campos

capacidades

Las capacidades se especifican como nombres de recursos de Amazon de Dataflow Endpoint Group.

Obligatorio: true

Formato: matriz de cadenas

- Valores: capacidad ARNs → Cadena

Ejemplos:

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

dispositivo

Este campo contiene los campos adicionales necesarios para enumerar el EC2 «dispositivo» actual.

Obligatorio: true

Formato: objeto

Miembros:

- privateIps
- publicIps
- agentCpuCores
- networkAdapters

privateIps

Este campo no se utiliza actualmente, pero se incluye para futuros casos de uso. Si no se incluye ningún valor, el valor predeterminado será [«127.0.0.1»]

Obligatorio: false

Formato: matriz de cadenas

- Valores: Direcciones IP → Cadena

Ejemplo:

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

publicIps

IP elástica (EIP) por grupo de puntos finales del flujo de datos.

Obligatorio: true

Formato: matriz de cadenas

- Valores: Direcciones IP → Cadena

Ejemplo:

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

agentCPUCores

Esto especifica qué núcleos virtuales están reservados para el aws-gs-agent proceso. Consulte [CPUplanificación básica](#) para conocer los requisitos para establecer este valor de forma adecuada.

Obligatorio: true

Formato: matriz int

- Valores: Números básicos → int

Ejemplo:

```
"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

Esto corresponde a los adaptadores Ethernet, o las interfaces conectadas ENIs, que recibirán los datos.

Obligatorio: false

Formato: matriz de cadenas

- Valores: nombres de adaptadores de Ethernet (puede encontrarlos ejecutando `ifconfig`)

Ejemplo:

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

EC2ajuste del rendimiento de la instancia

Note

Si aprovisionaste tus AWS recursos mediante CloudFormation plantillas, estos ajustes se aplican automáticamente. Si ha utilizado una EC2 instancia AMI o la ha creado manualmente, debe aplicar estos ajustes de rendimiento para lograr el rendimiento más fiable.

Recuerde reiniciar la instancia después de aplicar cualquier ajuste.

Temas

- [Ajuste las interrupciones del hardware y las colas de recepción: impactos y red CPU](#)
- [La fusión de interrupciones de Tune Rx afecta a la red](#)
- [Tune el búfer Rx Rx: afecta a la red](#)
- [Tune CPU C-State: impactos CPU](#)
- [Los puertos de entrada de reserva afectan a la red](#)
- [Reboot](#)

Ajuste las interrupciones del hardware y las colas de recepción: impactos y red CPU

En esta sección se configura el uso CPU principal de systemd SMPIRQs, Receive Packet Steering (RPS) y Receive Flow Steering (). RFS Consulte [Apéndice: Parámetros recomendados para interrumpir RPS o ajustar](#) para ver el conjunto de ajustes recomendados en función del tipo de instancia que utilice.

1. Aleje los procesos de systemd de los núcleos de los agentes. CPU
2. Redirija las solicitudes de interrupción de hardware lejos de los CPU núcleos de los agentes.
3. RPSConfigúrelo para evitar que la cola de hardware de una sola tarjeta de interfaz de red se convierta en un cuello de botella en el tráfico de la red.
4. Configure RFS para aumentar la tasa de aciertos de la CPU memoria caché y, por lo tanto, reducir la latencia de la red.

El `set_irq_affinity.sh` script proporcionado por el RPM usuario configura todo lo anterior por usted. Añádalo a `crontab` para que se aplique en cada arranque:

```
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` Sustitúyalos por núcleos reservados para el núcleo y el sistema operativo, normalmente el primero y el segundo, junto con los pares de núcleos con hipersubprocesos. Esto no debe superponerse con los núcleos seleccionados anteriormente. (Por ejemplo, «0,1,48,49» para una instancia de 96 con hipersubprocesos). CPU
- `rps_core_maskes` una máscara de bits hexadecimal que especifica quién CPUs debe procesar los paquetes entrantes, en la que cada dígito representa 4. CPUs También debe estar separada por comas cada 8 caracteres empezando por la derecha. Se recomienda permitir todo CPUs y dejar que el almacenamiento en caché se encargue del equilibrio.
 - Para ver la lista de parámetros recomendados para cada tipo de instancia, consulte [Apéndice: Parámetros recomendados para interrumpir RPS o ajustar](#).
- Ejemplo para 96 CPU instancias:

```
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
```

La fusión de interrupciones de Tune Rx afecta a la red

La fusión de interrupciones ayuda a evitar que el sistema de host se inunde con demasiadas interrupciones y a aumentar el rendimiento de la red. Con esta configuración, se recopilan los paquetes y se genera una única interrupción cada 128 microsegundos. Añádalo a `crontab` para que se aplique en cada arranque:

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

- Sustituya `interface` por la interfaz de red (adaptador Ethernet) configurada para recibir datos. Por lo general, esto `eth0` se debe a que es la interfaz de red predeterminada asignada a una EC2 instancia.

Tune el búfer Rx Rx: afecta a la red

Aumente el número de entradas de anillo en el búfer de anillo Rx para evitar que los paquetes se caigan o se sobrecarguen durante las conexiones interrumpidas. Añádalo al `crontab` para que quede correctamente configurado en cada arranque:

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

- Sustituya `interface` por la interfaz de red (adaptador Ethernet) configurada para recibir datos. Por lo general, esto `eth0` se debe a que es la interfaz de red predeterminada asignada a una EC2 instancia.
- Si configura una instancia `c6i.32xlarge`, es necesario modificar el comando para configurar el búfer circular en 8192, en lugar de en 16384

Tune CPU C-State: impactos CPU

Configure el CPU estado C para evitar el ralentí, lo que puede provocar la pérdida de paquetes durante el inicio de un contacto. Requiere reinicio de instancias.

```
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
```

Los puertos de entrada de reserva afectan a la red

Reserve todos los puertos del rango de puertos de su dirección de entrada de `AwsGroundStationAgentEndpoint` para evitar conflictos con el uso del núcleo. Un conflicto en el uso de los puertos provocará un fallo en el contacto y en la entrega de datos.

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

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

Reboot

Cuando todas las afinaciones se hayan aplicado correctamente, reinicie la instancia para que se apliquen los cambios.

```
sudo reboot
```

Apéndice: Parámetros recomendados para interrumpir RPS o ajustar

En esta sección se determinan los valores de los parámetros recomendados para su uso en la sección de ajuste de las interrupciones del hardware y las colas de recepción: impactos y red. CPU

Familia	Tipo de instancia	<code>interru</code> <code>pt_core_list</code>	<code>rps_cor</code> <code>e_mask</code>
C6i	<ul style="list-style-type: none"> • c6i.32xlarge 	<ul style="list-style-type: none"> • 0,1,64,65 	<ul style="list-style-type: none"> • ffffffff, fffffff, fffffff, fffffff

Familia	Tipo de instancia	\${interru pt_core_list}	\${rps_cor e_mask}
c5	<ul style="list-style-type: none"> • c5.24xlarge • c5.18xlarge • c5.12xlarge 	<ul style="list-style-type: none"> • 0,1,48,49 • 0,1,36,37 • 0,1,24,25 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff • ff,ffffff ff,fffffff • ffff,fffffff
c5n	<ul style="list-style-type: none"> • c5n.metal • c5n.18xlarge 	<ul style="list-style-type: none"> • 0,1,36,37 • 0,1,36,37 	<ul style="list-style-type: none"> • ff,ffffff ff,fffffff • ff,ffffff ff,fffffff
m5	<ul style="list-style-type: none"> • m5.24xlarge • m5.12xlarge 	<ul style="list-style-type: none"> • 0,1,48,49 • 0,1,24,25 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff • ffff,fffffff
r5	<ul style="list-style-type: none"> • r5.metal • r5.24xlarge 	<ul style="list-style-type: none"> • 0,1,48,49 • 0,1,48,49 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff • ffffffff, ffffffff, ffffffff
r5n	<ul style="list-style-type: none"> • r5n.metal • r5n.24xlarge 	<ul style="list-style-type: none"> • 0,1,48,49 • 0,1,48,49 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff • ffffffff, ffffffff, ffffffff

Familia	Tipo de instancia	<code>interru</code> <code>pt_core_list</code>	<code>rps_cor</code> <code>e_mask</code>
G4dn	<ul style="list-style-type: none"> • g4dn.metal • g4dn.16xlarge • g4dn.12xlarge 	<ul style="list-style-type: none"> • 0,1,48,49 • 0,1,32,33 • 0,1,24,25 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff • ffffffff, ffffffff • ffff,fffffff
p4d	<ul style="list-style-type: none"> • p4d.24xlarge 	<ul style="list-style-type: none"> • 0,1,48,49 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff
p3dn	<ul style="list-style-type: none"> • p3dn.24xlarge 	<ul style="list-style-type: none"> • 0,1,48,49 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff

Prepárese para tener un contacto en DigiF

1. Revise la planificación CPU básica para ver los flujos de datos deseados y proporcione una lista de los núcleos que el agente puede usar. Consulte [CPUplanificación básica](#).
2. Revise el archivo de configuración del AWS Ground Station agente. Consulte [AWS Ground Station Configuración del agente](#).
3. Confirme que se han aplicado los ajustes de rendimiento necesarios. Consulte [EC2ajuste del rendimiento de la instancia](#).
4. Confirme que está siguiendo todas las prácticas recomendadas indicadas. Consulte [Prácticas recomendadas](#).
5. Confirme que el AWS Ground Station agente se haya iniciado antes de la hora de inicio programada del contacto a través de:

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

6. Confirme que el AWS Ground Station agente está en buen estado antes de la hora de inicio programada del contacto mediante:

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

Compruebe que lo `agentStatus` suyo `awsGroundStationAgentEndpoint` es `ACTIVE` y lo `auditResults` es `HEALTHY`.

Prácticas recomendadas

EC2 Mejores prácticas de Amazon

Siga las EC2 mejores prácticas actuales y garantice una disponibilidad suficiente de almacenamiento de datos.

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

Programador de Linux

El programador de Linux puede reordenar los paquetes en los UDP sockets si los procesos correspondientes no están anclados a un núcleo específico. Cualquier subprocesso que envíe o reciba UDP datos debe estar anclado a un núcleo específico durante la transmisión de datos.

AWS Ground Station lista de prefijos gestionada

Se recomienda utilizar la lista de `com.amazonaws.global.groundstation` AWS prefijos gestionada al especificar las reglas de red para permitir la comunicación desde la antena. Consulte [Trabajar con listas de AWS prefijos gestionadas para obtener más información sobre las listas de prefijos AWS gestionadas](#).

Limitación de contacto único

El AWS Ground Station Agent admite múltiples transmisiones por contacto, pero solo admite un contacto a la vez. Para evitar problemas de programación, no comparta una instancia entre varios grupos de puntos de conexión de flujos de datos. Si la configuración de un solo agente está asociada a varias configuraciones diferentes DFEGARNs, no se registrará.

Ejecutar servicios y procesos junto con el AWS Ground Station agente

Al lanzar servicios y procesos en la misma EC2 instancia que el AWS Ground Station agente, es importante vincularlos para que vCPUs no los utilicen el AWS Ground Station agente y el núcleo de Linux, ya que esto puede provocar cuellos de botella e incluso la pérdida de datos durante los contactos. Este concepto de unión a algo específico vCPUs se conoce como afinidad.

Núcleos que se deben evitar:

- `agentCpuCores` de [Archivo de configuración del agente](#)
- `interrupt_core_list` de [Ajuste las interrupciones del hardware y las colas de recepción: impactos y red CPU](#).
 - Los valores predeterminados se pueden encontrar en [Apéndice: Parámetros recomendados para interrumpir RPS o ajustar](#)

Como ejemplo, usar una **c5.24xlarge** instancia

Si especificó

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

y corrió

```
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
```

luego evita los siguientes núcleos:

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

Servicios de afinización (systemd)

Los servicios recién lanzados se afinizarán automáticamente con los mencionados anteriormente.

`interrupt_core_list` Si el caso de uso de sus servicios lanzados requiere núcleos adicionales o necesita núcleos menos congestionados, siga esta sección.

Compruebe la afinidad con la que está configurado su servicio actualmente con el comando:

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

Si ves un valor vacío `CPUAffinity=`, por ejemplo, significa que probablemente usará los núcleos predeterminados del comando anterior `...bin/set_irq_affinity.sh <using the cores here> ...`

Para anular y establecer una afinidad específica, busca la ubicación del archivo de servicio ejecutando:

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

Abra y modifique el archivo (usando `vi`, etc.) y colóquelo `CPUAffinity=<core list>` en la `[Service]` sección de la siguiente manera:

```
[Unit]
...

[Service]
...
CPUAffinity=2,3

[Install]
...
```

Guarde el archivo y reinicie el servicio para aplicar la afinidad con:

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

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

Para obtener más información, visite: [Red Hat Enterprise Linux 8: Administración, supervisión y actualización del núcleo, capítulo 27. Configuración de CPU Affinity y NUMA políticas mediante systemd.](#)

Procesos de afinización (scripts)

Se recomienda encarecidamente afinizar manualmente los scripts y procesos recién lanzados, ya que el comportamiento predeterminado de Linux les permitirá utilizar cualquier núcleo de la máquina.

Para evitar conflictos fundamentales en cualquier proceso en ejecución (como python, scripts bash, etc.), inicie el proceso con:

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

Si el proceso ya se está ejecutando, utilice comandos como `pidof` o `ps` busque el ID de proceso (PID) del proceso específico. Con el PID puede ver la afinidad actual con:

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

y puede modificarla con:

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

Para obtener más información sobre el conjunto de tareas, consulte la página de manual de [taskset: Linux](#)

Resolución de problemas

El agente no se puede iniciar

Es posible que el AWS Ground Station agente no se inicie debido a varios motivos, pero el escenario más común podría ser un archivo de configuración del agente mal configurado. Tras iniciar el agente (consulte [AWS Ground Station Inicio del agente](#)), es posible que obtenga un estado como el siguiente:

```
#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)
```

Resolución de problemas

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

podría dar como resultado una salida 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.
```

Si no se inicia el agente después de Loading Config, se debe a un problema con la configuración del agente. Consulte [Archivo de configuración del agente](#) para verificar la configuración del agente.

AWS Ground Station Registros del agente

AWS Ground Station El agente escribe información sobre las ejecuciones, los errores y el estado de los contactos en los archivos de registro de la instancia que ejecuta el agente. Puede ver los archivos de registro conectándose manualmente a una instancia.

Puede ver los registros del agente en la siguiente ubicación.

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

No hay contactos disponibles

Para programar contactos se necesita un AWS Ground Station agente en buen estado. Confirme que su AWS Ground Station agente ha empezado y que está en buen estado consultando la siguiente AWS Ground Station API vía get-dataflow-endpoint-group:

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

Verifica que lo agentStatus tuyo awsGroundStationAgentEndpoint es ACTIVE y lo auditResults esHEALTHY.

Cómo obtener asistencia

Póngase en contacto con el equipo de Ground Station a través de AWS Support.

1. Indique `contact_id` para los contactos afectados. El AWS Ground Station equipo no puede investigar a un contacto específico sin esta información.
2. Proporcione detalles sobre todos los pasos de solución de problemas que ya se hayan tomado.
3. Escribe cualquier mensaje de error que encuentres al ejecutar los comandos en nuestra guía de solución de problemas.

Notas de lanzamiento del agente

Última versión del agente

Versión 1.0.3555.0

Fecha de lanzamiento: 27/03/2024

Fecha de fin de soporte: 31/08/2024

RPMSumas de verificación:

- SHA256: 108f3aceb00e5af549839cd766c56149397e448a6e1e1429c89a9eebb6bc0fc1
- MD5: 65b72fa507fb0af32651adbb18d2e30f

Cambios:

- Agregue la métrica del agente para la versión ejecutable seleccionada durante el inicio de la tarea.
- Agregue soporte para archivos de configuración para evitar versiones ejecutables específicas cuando haya otras versiones disponibles.
- Agregue diagnósticos de red y enrutamiento.
- Funciones de seguridad adicionales.
- Se solucionó el problema por el que algunas métricas que informaban de errores se escribían en stdout/journal en lugar de en un archivo de registro.
- Maneje correctamente los errores de sockets inalcanzables de la red.
- Mida la pérdida de paquetes y la latencia entre los agentes de origen y destino.
- Publique aws-gs-datapipe la versión 2.0 para admitir las nuevas funciones del protocolo y la capacidad de actualizar los contactos al nuevo protocolo de forma transparente.

Versiones de agentes obsoletas

Versión 1.0.2942.0

Fecha de lanzamiento: 26/06/2023

Fecha de fin de soporte: 31/05/2024

RPMSumas de verificación:

- SHA256: 7d94b642577504308a58bab28f938507f2591d4e1b2c7ea170b77bea97b5a9b6
- MD5: 661ff2b8f11aba5d657a6586b56e0d8f

Cambios:

- Se agregaron registros de errores para cuando el agente RPM se actualiza en el disco y es necesario reiniciarlo para que los cambios surtan efecto.
- Se agregó la validación del ajuste de la red para garantizar que se sigan y apliquen correctamente los pasos de ajuste de la guía del usuario del agente.
- Se ha corregido un error que provocaba advertencias erróneas en los registros del agente sobre el archivado de registros.
- Detección de pérdida de paquetes mejorada.
- Se actualizó la instalación del agente para evitar la instalación o la actualización RPM si el agente ya está en ejecución.

Versión 1.0.2716.0

Fecha de lanzamiento: 15/03/2023

Fecha de fin de soporte: 31/05/2024

RPMSumas de verificación:

- SHA256: cb05b6a77dfcd5c66d81c0072ac550affbcefefc372cc5562ee52fb220844929
- MD5: 65266490c4013b433ec39ee50008116c

Cambios:

- Habilite la carga de registros cuando el agente experimente errores durante la asignación de tareas.
- Corrige un error de compatibilidad con Linux en los scripts de ajuste de red proporcionados.

Versión 1.0.2677.0

Fecha de lanzamiento: 15/02/2023

Fecha de fin de soporte: 31/05/2024

RPMSumas de verificación:

- SHA256: 77cfe94acb00af7ca637264b17c9b21bd7afdc85b99dffdd627aec9e99397489
- MD5: b8533be7644bb4d12ab84de21341adac

Cambios:

- Primera versión de Agent disponible de forma general.

RPMvalidación de la instalación

A continuación se muestran la RPM versión más recienteRPM, el MD5 hash validado desde y el SHA256 hash utilizado con sha256sum. Estos valores, combinados, se pueden usar para validar la RPM versión que se está utilizando para el agente de la estación terrestre.

Versión más reciente del agente

Versión 1.0.3555.0

Fecha de lanzamiento: 27/03/2024

Fecha de fin de soporte: 31/08/2024

RPMSumas de verificación:

- SHA256: 108f3aceb00e5af549839cd766c56149397e448a6e1e1429c89a9eebb6bc0fc1
- MD5: 65b72fa507fb0af32651adbb18d2e30f

Cambios:

- Agregue la métrica del agente para la versión ejecutable seleccionada durante el inicio de la tarea.
- Agregue soporte para archivos de configuración para evitar versiones ejecutables específicas cuando haya otras versiones disponibles.
- Agregue diagnósticos de red y enrutamiento.
- Funciones de seguridad adicionales.
- Se solucionó el problema por el que algunas métricas que informaban de errores se escribían en stdout/journal en lugar de en un archivo de registro.
- Maneje correctamente los errores de sockets inalcanzables de la red.
- Mida la pérdida de paquetes y la latencia entre los agentes de origen y destino.
- Publique aws-gs-datapipe la versión 2.0 para admitir las nuevas funciones del protocolo y la capacidad de actualizar los contactos al nuevo protocolo de forma transparente.

Compruebe la RPM

Las herramientas que necesitará para poder verificar esta RPM instalación son:

- [sha256sum](#)
- [RPM](#)

Ambas herramientas vienen de forma predeterminada en Amazon Linux 2. Estas herramientas le ayudarán a comprobar que la RPM que está utilizando es la versión correcta. En primer lugar, descargue la versión más reciente RPM del bucket de S3 (consulte las instrucciones [Descargar el agente](#) para descargarla RPM). Una vez descargado el archivo, habrá que comprobar algunas cosas:

- Calcule la suma sha256 del RPM archivo. Realice la siguiente acción desde la línea de comandos de la instancia de procesamiento que esté utilizando:

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

Tome este valor y compárelo con la tabla anterior. Esto demuestra que el RPM archivo que se descarga es un archivo válido para su uso y que AWS Ground Station ha distribuido a los clientes. Si los hashes no coinciden, no los instale y RPM elimínelos de la instancia de procesamiento.

- Compruebe también el MD5 hash del archivo para asegurarte de que no se RPM ha visto comprometido. Para ello, utilice la herramienta de línea de RPM comandos ejecutando el siguiente comando:

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

Valide que el MD5 hash que se muestra aquí sea el mismo que el MD5 hash de la versión que aparece en la tabla anterior. Una vez que estos dos hashes se hayan validado con la tabla que aparece en AWS Docs, el cliente puede estar seguro de RPM que la versión descargada e instalada es segura y sin restricciones. RPM

Historial de documentos de la Guía del usuario del AWS Ground Station agente

En la siguiente tabla se describen los cambios importantes de cada versión de la Guía del usuario del AWS Ground Station agente.

Cambio	Descripción	Fecha
Actualización de la documentación	Se agregó un comentario sobre cómo mantener la subred y la EC2 instancia de Amazon en la misma zona de disponibilidad en Agent Requirements .	18 de julio de 2024
Actualización de la documentación	Divida el AWS Ground Station agente en su propia guía de usuario. Para ver los cambios anteriores, consulte el historial de documentos de la guía del usuario de AWS Ground Station .	18 de julio de 2024

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.