



AWS Ground Station Panduan Pengguna Agen

AWS Ground Station



AWS Ground Station: AWS Ground Station Panduan Pengguna Agen

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Ikhtisar	1
Apa AWS Ground Station agennya?	1
Fitur AWS Ground Station Agen	2
Persyaratan agen	3
Diagram VPC	4
Sistem operasi yang didukung	5
Menerima data melalui AWS Ground Station Agen	6
Beberapa aliran data, penerima tunggal	6
Beberapa aliran data, beberapa penerima	7
Pilih instans Amazon EC2 dan pesan inti CPU untuk arsitektur Anda	9
Tipe instans Amazon EC2 yang didukung	9
Perencanaan inti CPU	10
Mengumpulkan informasi arsitektur	11
Contoh penugasan CPU	13
Lampiran: <code>lscpu -p</code> output (penuh) untuk <code>c5.24xlarge</code>	14
Instal agen	17
Gunakan CloudFormation template	17
Langkah 1: Buat AWS sumber daya	17
Langkah 2: Periksa status agen	17
Instal secara manual di EC2	17
Langkah 1: Buat sumber daya AWS	17
Langkah 2: Buat instans EC2	18
Langkah 3: Unduh dan instal agen	18
Langkah 4: Konfigurasi agen	19
Langkah 5: Terapkan Tuning Kinerja	19
Langkah 6: Kelola agen	20
Kelola agen	21
AWS Ground Station Konfigurasi agen	21
AWS Ground Station Agen mulai	21
AWS Ground Station Agen berhenti	22
AWS Ground Station Peningkatan agen	22
AWS Ground Station Downgrade agen	23
AWS Ground Station Agen uninstall	24
AWS Ground Station Status agen	24

AWS Ground Station Info Agen RPM	25
Konfigurasi agen	26
File konfigurasi agen	26
Contoh	26
Kerusakan lapangan	26
Menyetel instans EC2 Anda untuk kinerja	30
Menyetel interupsi perangkat keras dan menerima antrian - berdampak pada CPU dan jaringan	30
Penyatuan interupsi Tune Rx - berdampak pada jaringan	31
Buffer cincin Tune Rx - berdampak pada jaringan	32
Tune CPU C-State - berdampak pada CPU	32
Reserve port ingress - berdampak pada jaringan	33
Boot ulang	33
Lampiran: Parameter yang disarankan untuk lagu interrupt/RPS	33
Bersiaplah untuk mengambil kontak DiGIF	36
Praktik terbaik	37
Praktik terbaik Amazon EC2	37
Penjadwal Linux	37
AWS Ground Station daftar awalan terkelola	37
Batasan kontak tunggal	37
Menjalankan layanan dan proses bersama AWS Ground Station Agen	37
Sebagai contoh menggunakan c5.24xlarge instance	38
Layanan afinisasi (systemd)	38
Proses afinisasi (skrip)	39
Pemecahan masalah	41
Agen gagal memulai	41
Pemecahan masalah	41
AWS Ground Station Log agen	42
Tidak ada kontak yang tersedia	42
Mendapatkan Dukungan	43
Catatan rilis agen	44
Versi Agen Terbaru	44
Versi 1.0.4382.0	44
Versi Agen Usang	44
Versi 1.0.3555.0	44
Versi 1.0.2942.0	45

Versi 1.0.2716.0	45
Versi 1.0.2677.0	46
Validasi instalasi RPM	47
Versi Agen Terbaru	44
Versi 1.0.4382.0	44
Verifikasi RPM	47
Riwayat Dokumen	49
.....	

Ikhtisar

Apa AWS Ground Station agennya?

Dengan AWS Ground Station Agen, tersedia sebagai RPM, Anda dapat menerima aliran data Wideband Digital Intermediate Frequency (DiGIF) sinkron (downlink) selama kontak AWS Ground Station. Anda dapat memilih dua opsi untuk pengiriman data:

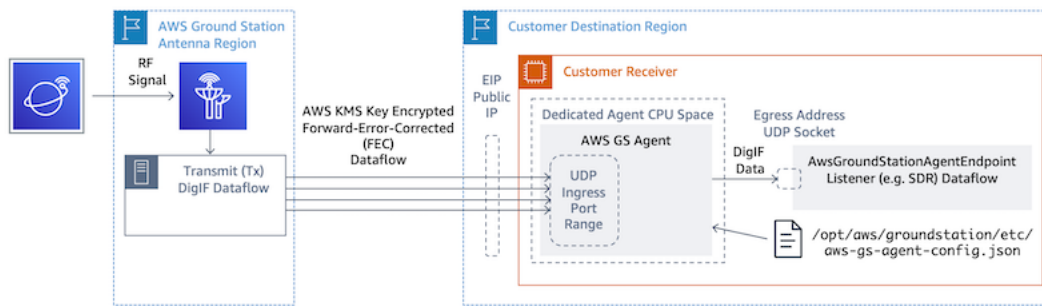
1. Pengiriman data ke instans EC2 - Pengiriman data ke instans EC2 yang Anda miliki. Anda mengelola AWS Ground Station Agen. Opsi ini mungkin paling cocok untuk Anda jika Anda membutuhkan pemrosesan data mendekati waktu nyata. Lihat panduan [Pengiriman Data ke Amazon Elastic Compute Cloud](#) untuk informasi tentang pengiriman data EC2.
2. Pengiriman data ke bucket S3 - Pengiriman data ke bucket AWS S3 yang Anda miliki melalui layanan terkelola Ground Station. Lihat AWS Ground Station panduan [Memulai dengan](#) untuk informasi tentang pengiriman data S3.

Kedua mode pengiriman data mengharuskan Anda membuat satu set sumber daya AWS. Penggunaan CloudFormation untuk membuat sumber daya AWS Anda sangat disarankan untuk memastikan keandalan, akurasi, dan dukungan. Setiap kontak hanya dapat mengirimkan data ke EC2 atau S3 tetapi tidak ke keduanya secara bersamaan.

Note

Karena pengiriman data S3 adalah layanan terkelola Ground Station, panduan ini berfokus pada pengiriman data ke instans EC2 Anda.

Diagram berikut menunjukkan aliran data DiGIF dari Wilayah AWS Ground Station Antena ke instans EC2 Anda dengan Software-Defined Radio (SDR) atau pendengar serupa.



Fitur AWS Ground Station Agen

AWS Ground Station Agen menerima data downlink Digital Intermediate Frequency (DiGIF) dan mengeluarkan data yang didekripsi yang memungkinkan hal-hal berikut:

- Kemampuan downlink DiGIF dari 40 MHz hingga 400 MHz bandwidth.
- Pengiriman data DigiF dengan tingkat tinggi dan jitter rendah ke IP publik (AWS Elastic IP) apa pun di jaringan AWS.
- Pengiriman data yang andal menggunakan Forward Error Correction (FEC).
- Mengamankan pengiriman data menggunakan AWS KMS kunci terkelola pelanggan untuk enkripsi.

Persyaratan agen

Note

Panduan AWS Ground Station Agen ini mengasumsikan bahwa Anda telah masuk ke Ground Station menggunakan panduan [AWS Ground Station Memulai](#).

Instans EC2 penerima AWS Ground Station Agen memerlukan satu set sumber daya AWS yang bergantung untuk mengirimkan data DiGIF secara andal dan aman ke titik akhir Anda.

1. VPC untuk meluncurkan penerima EC2.
2. Kunci AWS KMS untuk enkripsi/dekripsi data.
3. Kunci SSH atau Profil Instans EC2 yang dikonfigurasi untuk [SSM](#) Session Manager.
4. Aturan Grup Jaringan/Keamanan untuk mengizinkan hal berikut:
 1. Lalu lintas UDP dari AWS Ground Station port yang ditentukan dalam grup endpoint aliran data Anda. Agen mencadangkan berbagai port bersebelahan yang digunakan untuk mengirimkan data ke titik akhir aliran data ingress.
 2. Akses SSH ke instans Anda (Catatan: Anda dapat menggunakan AWS Session Manager untuk mengakses instans EC2 Anda).
 3. Baca akses ke bucket S3 yang dapat diakses publik untuk manajemen agen.
 4. Lalu lintas SSL pada port 443 memungkinkan agen untuk berkomunikasi dengan layanan AWS Ground Station
 5. Lalu lintas dari daftar `com.amazonaws.global.groundstation` awalan AWS Ground Station terkelola.

Selain itu, konfigurasi VPC termasuk subnet publik diperlukan. Lihat [Panduan Pengguna VPC](#) untuk latar belakang konfigurasi subnet.

Konfigurasi yang kompatibel:

1. IP Elastis yang terkait dengan instans EC2 Anda di subnet publik.
2. IP Elastis yang terkait dengan ENI di subnet publik, dilampirkan ke instans EC2 Anda (di subnet apa pun di zona ketersediaan yang sama dengan subnet publik).

Anda dapat menggunakan grup keamanan yang sama dengan instans EC2 Anda atau menentukannya dengan setidaknya seperangkat aturan minimum yang terdiri dari:

- Lalu lintas UDP dari AWS Ground Station port yang ditentukan dalam grup endpoint aliran data Anda.

Misalnya templat Pengiriman Data CloudFormation EC2 dengan sumber daya ini yang telah dikonfigurasi sebelumnya, lihat [Satelit siaran publik yang menggunakan AWS Ground Station Agen \(pita lebar\)](#).

Diagram VPC

Diagram: IP Elastis yang terkait dengan instans EC2 Anda di subnet publik

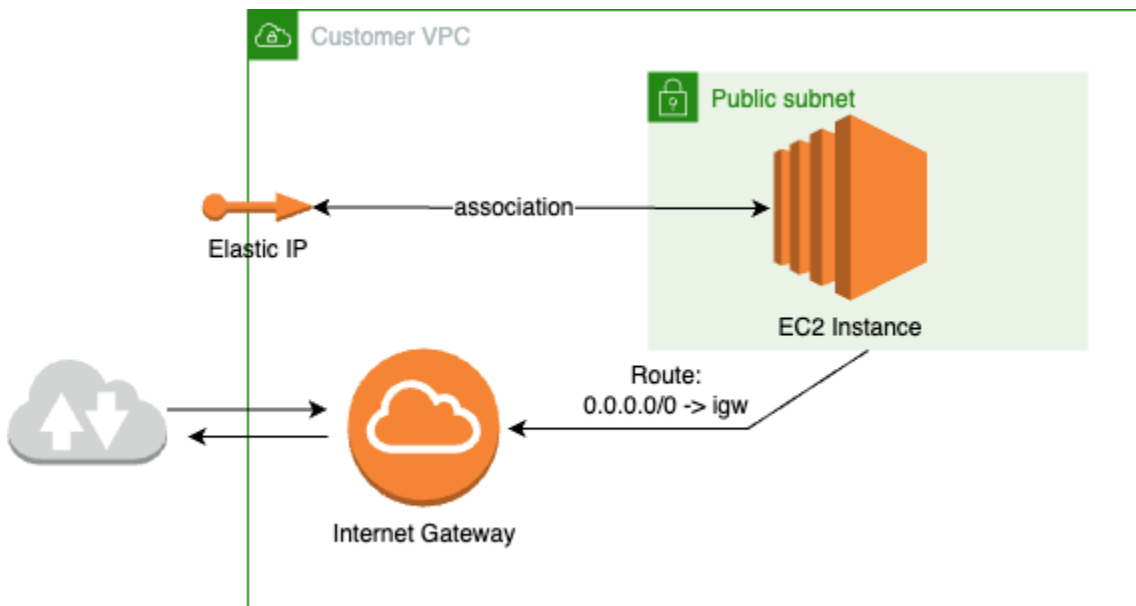
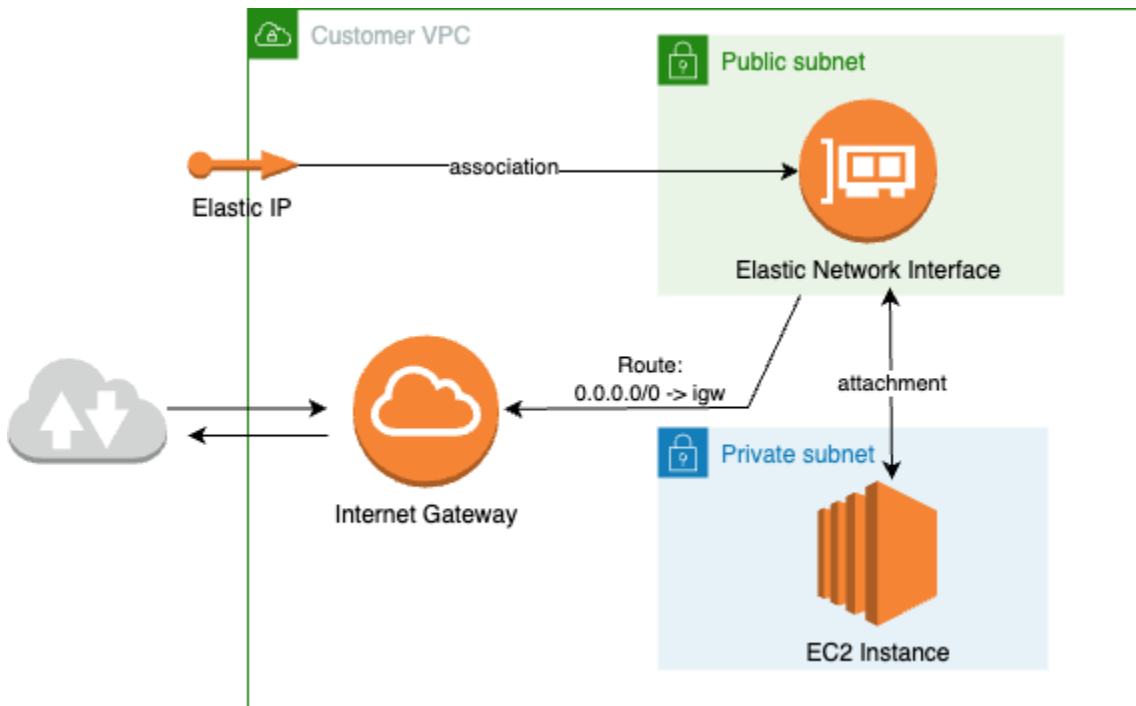


Diagram: IP Elastis yang terkait dengan ENI di subnet publik, dilampirkan ke instans EC2 Anda di subnet pribadi



Sistem operasi yang didukung

Amazon Linux 2 dengan 5.10+ kernel.

Jenis instans yang didukung tercantum dalam [Pilih instans Amazon EC2 dan pesan inti CPU untuk arsitektur Anda](#)

Menerima data melalui AWS Ground Station Agen

Diagram di bawah ini memberikan gambaran umum tentang bagaimana data mengalir AWS Ground Station selama kontak Wideband Digital Intermediate Frequency (DiGIF).

AWS Ground Station Agen akan menangani orkestrasi komponen dataplane untuk kontak. Sebelum menjadwalkan kontak agen harus dikonfigurasi dengan benar, dimulai, dan harus terdaftar (pendaftaran otomatis pada saat agen startup) dengan AWS Ground Station. Selain itu, perangkat lunak penerima data (seperti radio yang ditentukan perangkat lunak) harus berjalan dan dikonfigurasi untuk menerima data di [AwsGroundStationAgentEndpointeGressAddress](#).

Di belakang layar, AWS Ground Station Agen akan menerima tugas dari AWS Ground Station dan membatalkan AWS KMS enkripsi yang diterapkan dalam perjalanan, sebelum meneruskannya ke titik akhir tujuan eGressAddress tempat Software Defined Radio (SDR) Anda mendengarkan. AWS Ground Station Agen dan komponen dasarnya akan menghormati batas CPU yang ditetapkan dalam file konfigurasi untuk memastikannya tidak memengaruhi kinerja aplikasi lain yang berjalan pada instance.

Anda harus memiliki AWS Ground Station Agen yang berjalan pada instance penerima yang terlibat dalam kontak. AWS Ground Station Agen tunggal dapat mengatur beberapa aliran data, seperti yang terlihat di bawah ini, jika Anda lebih suka menerima semua aliran data pada satu instance penerima.

Beberapa aliran data, penerima tunggal

Contoh Skenario:

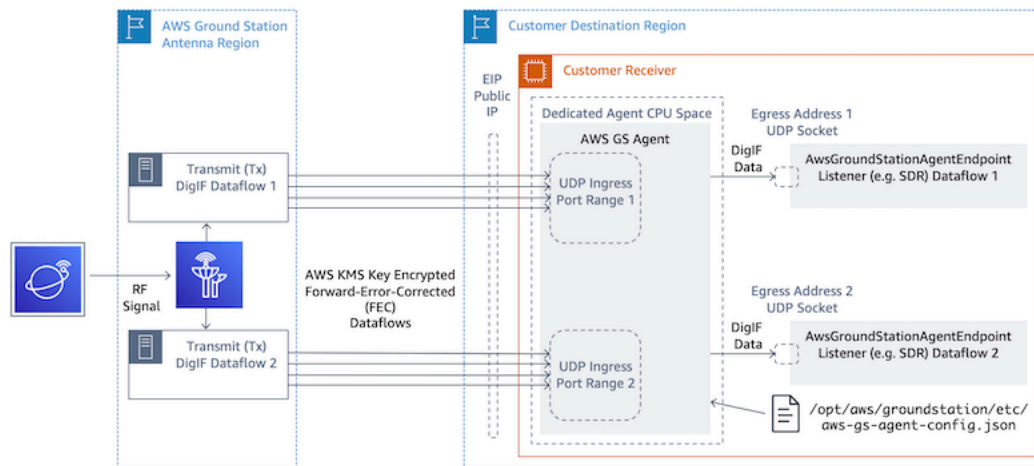
Anda ingin menerima dua downlink antena sebagai aliran data DiGIF pada instance penerima EC2 yang sama. Dua downlink akan menjadi 200 MHz dan 100MHz.

AwsGroundStationAgentEndpoints:

Akan ada dua `AwsGroundStationAgentEndpoint` sumber daya, satu untuk setiap aliran data. Kedua titik akhir akan memiliki alamat IP publik yang sama (`ingressAddress.socketAddress.name`). Ingress `portRange` tidak boleh tumpang tindih, karena aliran data diterima pada instance EC2 yang sama. `egressAddress.socketAddress.port` Keduanya harus unik.

Perencanaan CPU:

- 1 core (2 vCPU) untuk menjalankan AWS Ground Station Agen tunggal pada instance.
- 6 core (12 vCPU) untuk menerima DiGIF Dataflow 1 (200 lookup dalam tabel). MHz [Perencanaan inti CPU](#)
- 4 core (8 vCPU) untuk menerima DiGIF Dataflow 2 (pencarian 100 dalam tabel). MHz [Perencanaan inti CPU](#)
- Total Ruang CPU Agen Khusus = 11 core (22 vCPU) pada soket yang sama.



Beberapa aliran data, beberapa penerima

Contoh Skenario:

Anda ingin menerima dua downlink antenna sebagai aliran data DiGIF pada instans penerima EC2 yang berbeda. Kedua downlink akan menjadi 400MHz.

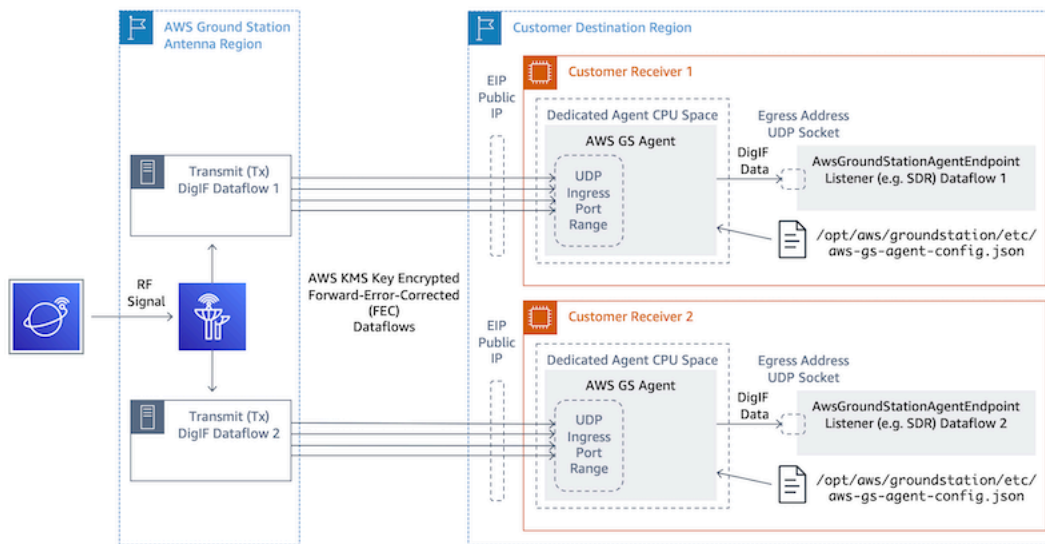
AwsGroundStationAgentEndpoints:

Akan ada dua AwsGroundStationAgentEndpoint sumber daya, satu untuk setiap aliran data. Endpoint akan memiliki alamat IP publik yang berbeda (`ingressAddress.socketAddress.name`). Tidak ada batasan pada nilai port untuk salah satu `ingressAddress` atau `egressAddress` karena aliran data diterima pada infrastruktur terpisah dan tidak akan bertentangan satu sama lain.

Perencanaan CPU:

- Instance Penerima 1
 - 1 core (2 vCPU) untuk menjalankan AWS Ground Station Agen tunggal pada instance.

- 9 core (18 vCPU) untuk menerima DiGIF Dataflow 1 (400 lookup dalam tabel). MHz
[Perencanaan inti CPU](#)
- Total Dedicated Agent CPU Space = 10 core (20 vCPU) pada soket yang sama.
- Instance Penerima 2
 - 1 core (2 vCPU) untuk menjalankan AWS Ground Station Agen tunggal pada instance.
 - 9 core (18 vCPU) untuk menerima DiGIF Dataflow 2 (400 lookup dalam tabel). MHz
[Perencanaan inti CPU](#)
 - Total Dedicated Agent CPU Space = 10 core (20 vCPU) pada soket yang sama.



Pilih instans Amazon EC2 dan pesan inti CPU untuk arsitektur Anda

Tipe instans Amazon EC2 yang didukung

AWS Ground Station Agen memerlukan inti CPU khusus untuk beroperasi karena alur kerja pengiriman data intensif komputasi. Kami mendukung jenis contoh berikut. Lihat [Perencanaan inti CPU](#) untuk memutuskan jenis instance mana yang paling sesuai dengan kasus penggunaan Anda.

Keluarga instans	Tipe instans	Default v CPUs	Inti CPU default	Bandwidth agregat DiGIF maks () MHz
c5	c5.12xlarge	48	24	180
	c5.18xlarge	72	36	380
	c5.24xlarge	96	48	380
c5n	c5n.18xlarge	72	36	400
	c5n.metal	72	36	400
c6i	c6i.24xlarge	96	48	400
	c6i.32xlarge	128	64	400
c7i	c7i.12xlarge	48	24	280
	c7i.24xlarge	96	48	400
p3dn	p3dn.24xlarge	96	48	400
g4dn	g4dn.12xlarge	48	24	400
	g4dn.16xlarge	64	32	400
	g4dn.metal	96	48	400

Keluarga instans	Tipe instans	Default v CPUs	Inti CPU default	Bandwidth agregat DiGIF maks () MHz
p4d	p4d.24xlarge	96	48	400
m5	m5.8xlarge	32	16	100
	m5.12xlarge	48	24	180
	m5.24xlarge	96	48	380
m6i	m6i.32xlarge	128	64	400
r5	r5.24xlarge	96	48	380
	r5.metal	96	48	380
r5n	r5n.24xlarge	96	48	400
	r5n.metal	96	48	400
r6i	r6i.32xlarge	128	64	400

Note

Kolom bandwidth agregat Max DiGIF menunjukkan bandwidth agregat maksimum yang didukung untuk semua aliran data DiGIF yang digabungkan pada setiap jenis instans. Keterbatasan ini disebabkan oleh kapasitas jaringan EC2 yang dialokasikan untuk jenis instans yang diberikan. Nilai-nilai ini mewakili perkiraan konservatif dan harus digunakan saat merencanakan konfigurasi DiGIF Anda. Bandwidth aktual dapat bervariasi berdasarkan beban sistem dan faktor lainnya.

Perencanaan inti CPU

AWS Ground Station Agen memerlukan inti prosesor khusus yang berbagi cache L3 untuk setiap aliran data. Agen ini dirancang untuk memanfaatkan pasangan CPU Hyper-threaded (HT) dan mengharuskan pasangan HT dicadangkan untuk penggunaannya. Pasangan hyper-threaded

adalah sepasang virtual (CPUs vCPU) yang terkandung dalam satu inti. Tabel berikut menyediakan pemetaan laju data aliran data ke jumlah inti yang diperlukan yang disediakan untuk agen untuk aliran data tunggal. Tabel ini mengasumsikan Cascade Lake atau yang lebih baru CPUs dan berlaku untuk semua jenis instans yang didukung. Jika bandwidth Anda berada di antara entri dalam tabel, pilih yang tertinggi berikutnya.

Agan membutuhkan inti cadangan tambahan untuk manajemen dan koordinasi, sehingga total inti yang diperlukan adalah jumlah inti yang dibutuhkan (dari bagan di bawah) untuk setiap aliran data ditambah satu inti tambahan (2 v CPUs).

AntennaDownlink Bandwidth (MHz)	Kecepatan Data DigiF VITA-49,2 yang diharapkan (MB/s)	Jumlah Core (HT CPU Pairs)	Jumlah vCPU
50	1000	3	6
100	2000	4	8
150	3000	5	10
200	4000	6	12
250	5000	6	12
300	6000	7	14
350	7000	8	16
400	8000	9	18

Mengumpulkan informasi arsitektur

Iscpumemberikan informasi tentang arsitektur sistem Anda. Output dasar menunjukkan v mana CPUs (diberi label sebagai "CPU") milik node NUMA mana (dan setiap node NUMA berbagi cache L3). Di bawah ini kami memeriksa `c5.24xlarge` contoh untuk mengumpulkan informasi yang diperlukan untuk mengkonfigurasi AWS Ground Station Agen. Ini termasuk informasi yang berguna seperti jumlah vCPUs, core, dan vCPU-to-node asosiasi.


```

> 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  <-----

```

Cores yang didedikasikan untuk AWS Ground Station Agen harus menyertakan v CPUs untuk setiap inti yang ditetapkan. Semua core untuk aliran data harus ada pada node NUMA yang sama. -pOpsi untuk `lscpu` perintah memberi kita inti ke asosiasi CPU yang diperlukan untuk mengkonfigurasi agen. Bidang yang relevan adalah CPU (yang kami sebut sebagai vCPU), Core, dan L3 (yang menunjukkan cache L3 mana yang dibagikan oleh inti itu). Perhatikan bahwa pada sebagian besar prosesor Intel Node NUMA sama dengan cache L3.

Pertimbangkan subset `lscpu -p` output berikut untuk `a.c5.24xlarge` (disingkat dan diformat untuk kejelasan).

```

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

```

Dari output kita dapat melihat bahwa Core 0 termasuk v CPUs 0 dan 16, Core 1 termasuk v CPUs 1 dan 17, Core 2 termasuk v CPUs 2 dan 18. Dengan kata lain pasangan hyper-threaded adalah: 0 dan 16, 1 dan 17, 2 dan 18.

Contoh penugasan CPU

Sebagai contoh, kita akan menggunakan `c5.24xlarge` instance untuk downlink Dual Polarity Wideband pada 350. MHz Dari tabel di [Perencanaan inti CPU](#) kita tahu bahwa MHz downlink 350 membutuhkan 8 core (16 vCPUs) untuk aliran data tunggal. Ini berarti bahwa pengaturan polaritas ganda ini menggunakan dua aliran data membutuhkan total 16 core (32 vCPUs) ditambah satu inti (2 vCPUs) untuk Agen.

Kami tahu `lscpu` output untuk `c5.24xlarge` include NUMA node0 CPU(s): 0-23,48-71 dan NUMA node1 CPU(s): 24-47,72-95. Karena NUMA node0 memiliki lebih dari yang kita butuhkan, kita hanya akan menetapkan dari core: 0-23 dan 48-71.

Pertama, kita akan memilih 8 core untuk setiap aliran data yang berbagi cache L3 atau NUMA Node. Kemudian kita akan mencari v yang sesuai CPUs (berlabel "CPU") di `lscpu -p` output di [Lampiran: lscpu -p output \(penuh\) untuk c5.24xlarge](#). Contoh proses pemilihan inti mungkin terlihat seperti berikut:

- Cadangan core 0-1 untuk OS.
- Aliran 1: pilih core 2-9 yang dipetakan ke v CPUs 2-9 dan 50-57.
- Aliran 2: pilih core 10-17 yang memetakan ke v CPUs 10-17 dan 58-65.
- Inti agen: pilih inti 18 yang memetakan ke v CPUs 18 dan 66.

Ini menghasilkan v CPUs 2-18 dan 50-66 sehingga daftar untuk menyediakan agen adalah. [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] Anda harus memastikan proses

Anda sendiri tidak berjalan pada ini CPUs seperti yang dijelaskan dalam [Menjalankan layanan dan proses bersama AWS Ground Station Agen](#).

Perhatikan bahwa inti spesifik yang dipilih dalam contoh ini agak sewenang-wenang. Kumpulan inti lainnya akan berfungsi selama memenuhi persyaratan semua berbagi cache L3 untuk setiap aliran data.

Lampiran: `lscpu -p` output (penuh) untuk 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
```

Instal agen

AWS Ground Station Agen dapat diinstal dengan cara-cara berikut:

1. CloudFormation template (disarankan).
2. Instalasi manual di Amazon EC2.

Gunakan CloudFormation template

CloudFormation Template pengiriman data EC2 membuat sumber daya AWS yang diperlukan untuk mengirimkan data ke instans EC2 Anda. CloudFormation Template ini menggunakan AMI AWS Ground Station terkelola yang memiliki AWS Ground Station Agen yang sudah diinstal sebelumnya. Skrip boot instans EC2 yang dibuat kemudian mengisi file konfigurasi agen dan menerapkan tuning kinerja yang diperlukan (). [Menyetel instans EC2 Anda untuk kinerja](#)

Langkah 1: Buat AWS sumber daya

Buat tumpukan sumber daya AWS Anda menggunakan templat [Satelit siaran publik menggunakan AWS Ground Station Agent \(pita lebar\)](#).

Langkah 2: Periksa status agen

Secara default agen dikonfigurasi dan aktif (dimulai). Untuk memeriksa status agen, Anda dapat terhubung ke instans EC2 (SSH atau SSM Session Manager) dan lihat. [AWS Ground Station Status agen](#)

Instal secara manual di EC2

Meskipun Ground Station merekomendasikan penggunaan CloudFormation templat untuk menyediakan Sumber Daya AWS Anda, mungkin ada kasus penggunaan di mana templat standar mungkin tidak cukup. Untuk kasus seperti itu kami sarankan Anda menyesuaikan template sesuai dengan kebutuhan Anda. Jika itu masih tidak memenuhi persyaratan Anda, Anda dapat membuat sumber daya AWS secara manual dan menginstal agen.

Langkah 1: Buat sumber daya AWS

Lihat [Contoh konfigurasi profil misi](#) untuk petunjuk menyiapkan sumber daya AWS yang diperlukan untuk kontak secara manual.

`AwsGroundStationAgentEndpoint` Sumber daya mendefinisikan titik akhir untuk menerima aliran data DiGIF melalui AWS Ground Station Agen dan sangat penting untuk mengambil kontak yang berhasil. Meskipun dokumentasi API terletak di [Referensi API](#), bagian ini akan membahas secara singkat konsep yang relevan dengan AWS Ground Station Agen.

Titik akhir `ingressAddress` adalah tempat AWS Ground Station Agen akan menerima lalu lintas UDP AWS KMS terenkripsi dari Antena. `socketAddressName` ini adalah IP publik dari instans EC2 (dari EIP terlampir). `portRange` harus setidaknya 300 port bersebelahan dalam kisaran yang telah dicadangkan dari penggunaan lain. Lihat [Reserve port ingress - berdampak pada jaringan](#) untuk instruksi. Port ini harus dikonfigurasi untuk memungkinkan lalu lintas masuknya UDP pada grup keamanan untuk VPC tempat instance penerima berjalan.

Titik akhir `egressAddress` adalah tempat Agen akan menyerahkan aliran data DiGIF kepada Anda. Anda harus memiliki aplikasi (misalnya SDR) yang menerima data melalui soket UDP di lokasi ini.

Langkah 2: Buat instans EC2

Berikut ini AMIs didukung:

1. AWS Ground Station AMI - `groundstation-a12-gs-agent-ami-*` di mana `*` adalah tanggal AMI dibangun - dilengkapi dengan agen yang diinstal (disarankan).
2. `amzn2-ami-kernel-5.10-hvm-x86_64-gp2`.

Langkah 3: Unduh dan instal agen

Note

Langkah-langkah di bagian ini harus diselesaikan jika Anda tidak memilih AWS Ground Station Agen AMI pada langkah sebelumnya.

Agen unduhan

AWS Ground Station [Agen tersedia dari bucket S3 khusus wilayah dan dapat diunduh ke instans dukungan EC2 menggunakan baris perintah AWS \(CLI\) dari `s3://groundstation-wb-digif-software-\${AWS::Region}/aws-groundstation-agent/latest/amazon_linux_2_x86_64/aws-groundstation-agent.rpm` mana `\${AWS::Region}` merujuk ke salah satu AWS Ground Station Console dan Wilayah Pengiriman Data yang didukung.](#)

Contoh: Unduh versi rpm terbaru dari AWS region us-east-2 secara lokal ke folder/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
```

Jika Anda perlu mengunduh versi AWS Ground Station Agen tertentu, Anda dapat mengunduhnya dari folder versi tertentu di bucket S3.

Contoh: Unduh versi 1.0.2716.0 rpm dari AWS region us-east-2 secara lokal ke folder /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

Jika Anda ingin mengonfirmasi bahwa RPM yang Anda unduh telah dijual AWS Ground Station, ikuti instruksi untuk [Validasi instalasi RPM](#).

Instal agen

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

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

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

Langkah 4: Konfigurasi agen

Setelah menginstal agen, Anda harus memperbarui file konfigurasi agen. Lihat [Konfigurasi agen](#).

Langkah 5: Terapkan Tuning Kinerja

AWS Ground Station Agen AMI: Jika Anda memilih AWS Ground Station Agen AMI pada langkah sebelumnya maka terapkan penyetelan kinerja berikut.

- [Menyetel interupsi perangkat keras dan menerima antrian - berdampak pada CPU dan jaringan](#)
- [Reserve port ingress - berdampak pada jaringan](#)
- [Boot ulang](#)

Lainnya AMIs: Jika Anda memilih AMI lain di langkah sebelumnya, terapkan semua penyetelan yang tercantum di bawah [Menyetel instans EC2 Anda untuk kinerja](#) dan Reboot instance.

Langkah 6: Kelola agen

Untuk memulai, berhenti dan periksa status agen lihat [Kelola agen](#).

Kelola agen

AWS Ground Station Agen menyediakan kemampuan berikut untuk mengonfigurasi, memulai, menghentikan, meningkatkan, menurunkan versi, dan menghapus instalasi agen menggunakan perangkat perintah Linux bawaan.

Topik

- [AWS Ground Station Konfigurasi agen](#)
- [AWS Ground Station Agen mulai](#)
- [AWS Ground Station Agen berhenti](#)
- [AWS Ground Station Peningkatan agen](#)
- [AWS Ground Station Downgrade agen](#)
- [AWS Ground Station Agen uninstall](#)
- [AWS Ground Station Status agen](#)
- [AWS Ground Station Info Agen RPM](#)

AWS Ground Station Konfigurasi agen

Arahkan ke `/opt/aws/groundstation/etc`, yang harus berisi satu file bernama `aws-gs-agent-config.json`. Lihat [File konfigurasi agen](#)

AWS Ground Station Agen mulai

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

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

Harus menghasilkan output yang menunjukkan agen aktif.

```
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 Agen berhenti

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

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

Harus menghasilkan output yang menunjukkan agen tidak aktif (berhenti).

```
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 Peningkatan agen

1. Unduh versi terbaru agen. Lihat [Agen unduhan](#).
2. Hentikan agennya.

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

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

3. Perbarui agen.

```
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 Downgrade agen

1. Unduh versi agen yang Anda butuhkan. Lihat [Agen unduhan](#).
2. Downgrade agen.

```
# 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 Agen uninstall

Menghapus instalasi agen akan mengganti nama `/opt/aws/groundstation/etc/aws-gs-agent-config.json` to `/opt/aws/groundstation/etc/aws-gs-agent-config.json.rpm.save`. Menginstal agen lagi pada instance yang sama lagi akan menulis nilai default `aws-gs-agent-config.json` dan perlu diperbarui dengan nilai yang benar yang sesuai dengan sumber daya AWS Anda. Lihat [File konfigurasi agen](#).

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

AWS Ground Station Status agen

Status agen aktif (agen sedang berjalan) atau tidak aktif (agen dihentikan).

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

Contoh output menunjukkan bahwa agen diinstal, status tidak aktif (berhenti) dan diaktifkan (mulai layanan saat boot).

```
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 Info Agen RPM

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

Output adalah sebagai berikut:

Note

“Versi” mungkin berbeda berdasarkan versi terbaru yang diterbitkan agen.

```
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
```

Konfigurasi agen

Setelah menginstal agen, Anda harus memperbarui file konfigurasi agen di `/opt/aws/groundstation/etc/aws-gs-agent-config.json`.

File konfigurasi agen

Contoh

```
{
  "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
  ]
}
```

Kerusakan lapangan

kemampuan

Kemampuan ditentukan sebagai Nama Sumber Daya Amazon Grup Titik Akhir Dataflow.

Diperlukan: Benar

Format: Array String

- Nilai: kemampuan ARNs → String

Contoh:

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

pesawat

Bidang ini berisi bidang tambahan yang diperlukan untuk menghitung “perangkat” EC2 saat ini.

Diperlukan: Benar

Format: Objek

Anggota:

- PrivateIP
- PublicIP
- agentCpuCores
- NetworkAdapters

PrivateIP

Bidang ini saat ini tidak digunakan, tetapi disertakan untuk kasus penggunaan di masa mendatang. Jika tidak ada nilai yang disertakan, itu akan default ke ["127.0.0.1"]

Diperlukan: Salah

Format: Array String

- Nilai: Alamat IP → String

Contoh:

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


PublicIP

IP elastis (EIP) per kelompok titik akhir aliran data.

Diperlukan: Benar

Format: Array String

- Nilai: Alamat IP → String

Contoh:

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

agen CPUCores

Ini menentukan inti virtual mana yang dicadangkan untuk aws-gs-agent proses tersebut. Lihat persyaratan [Perencanaan inti CPU](#) untuk menetapkan nilai ini dengan tepat.

Diperlukan: Benar

Format: Array Int

- Nilai: Nomor Inti → int

Contoh:

```
"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, 82  
]
```

NetworkAdapters

Ini sesuai dengan adaptor ethernet, atau antarmuka yang terpasang ENIs, yang akan menerima data.

Diperlukan: Salah

Format: Array String

- Nilai: nama adaptor ethernet (dapat menemukannya dengan menjalankan `ifconfig`)

Contoh:

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

Menyetel instans EC2 Anda untuk kinerja

Note

Jika Anda menyediakan sumber daya AWS menggunakan CloudFormation templat, penyetelan ini akan diterapkan secara otomatis. Jika Anda menggunakan AMI atau membuat instans EC2 secara manual, maka penyetelan kinerja ini harus diterapkan untuk mencapai kinerja yang paling andal.

Ingatlah untuk me-reboot instance Anda setelah menerapkan penyetelan apa pun.

Topik

- [Menyetel interupsi perangkat keras dan menerima antrian - berdampak pada CPU dan jaringan](#)
- [Penyatuan interupsi Tune Rx - berdampak pada jaringan](#)
- [Buffer cincin Tune Rx - berdampak pada jaringan](#)
- [Tune CPU C-State - berdampak pada CPU](#)
- [Reserve port ingress - berdampak pada jaringan](#)
- [Boot ulang](#)

Menyetel interupsi perangkat keras dan menerima antrian - berdampak pada CPU dan jaringan

Bagian ini mengkonfigurasi penggunaan inti CPU systemd, SMP IRQs, Receive Packet Steering (RPS) dan Receive Flow Steering (RFS). Lihat [Lampiran: Parameter yang disarankan untuk lagu interrupt/RPS](#) sekumpulan pengaturan yang direkomendasikan berdasarkan jenis instans yang Anda gunakan.

1. Pin systemd memproses jauh dari inti CPU agen.
2. Permintaan interupsi perangkat keras rute jauh dari inti CPU agen.
3. Konfigurasi RPS untuk mencegah antrian perangkat keras dari kartu antarmuka jaringan tunggal menjadi hambatan dalam lalu lintas jaringan.
4. Konfigurasi RFS untuk meningkatkan hit rate cache CPU dan dengan demikian mengurangi latensi jaringan.

`set_irq_affinity.sh` Skrip yang disediakan oleh RPM mengkonfigurasi semua hal di atas untuk Anda. Tambahkan ke crontab, sehingga diterapkan pada setiap boot:

```
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
```

- Ganti `interrupt_core_list` dengan core yang dicadangkan untuk kernel dan OS - biasanya yang pertama dan kedua bersama dengan pasangan inti hyper-threaded. Ini seharusnya tidak tumpang tindih dengan inti yang dipilih di atas. (Contoh: '0,1,48,49' untuk instance 96-CPU hyper-threaded).
- `rps_core_mask` adalah bit mask heksadesimal yang menentukan mana yang CPUs harus memproses paket masuk, dengan setiap digit mewakili 4. CPUs itu juga harus dipisahkan koma setiap 8 karakter mulai dari kanan. Disarankan untuk mengizinkan semua CPUs dan membiarkan caching menangani penyeimbangan.
 - Untuk melihat daftar parameter yang direkomendasikan untuk setiap jenis instance, lihat [Lampiran: Parameter yang disarankan untuk lagu interrupt/RPS](#).
- Contoh untuk instance 96-CPU:

```
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
```

Penyatuan interupsi Tune Rx - berdampak pada jaringan

Penggabungan interupsi membantu mencegah banjir sistem host dengan terlalu banyak interupsi dan membantu meningkatkan throughput jaringan. Dengan konfigurasi ini, paket dikumpulkan dan satu interupsi tunggal dihasilkan setiap 128 mikrodetik. Tambahkan ke crontab, sehingga diterapkan pada setiap boot:

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

- Ganti `interface` dengan antarmuka jaringan (adaptor ethernet) yang dikonfigurasi untuk menerima data. Biasanya, ini `eth0` karena itulah antarmuka jaringan default yang ditetapkan untuk instans EC2.

Buffer cincin Tune Rx - berdampak pada jaringan

Tingkatkan jumlah entri dering untuk buffer cincin Rx untuk mencegah penurunan paket atau overrun selama koneksi pecah. Tambahkan ke crontab, sehingga diatur dengan benar pada setiap boot:

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

- Ganti `interface` dengan antarmuka jaringan (adaptor ethernet) yang dikonfigurasi untuk menerima data. Biasanya, ini `eth0` karena itulah antarmuka jaringan default yang ditetapkan untuk instans EC2.
- Jika menyiapkan instance `c6i` keluarga, perintah perlu dimodifikasi untuk mengatur buffer ring ke `8192`, bukan `16384`.

Tune CPU C-State - berdampak pada CPU

Atur CPU C-state untuk mencegah idle yang dapat menyebabkan paket hilang selama dimulainya kontak. Membutuhkan contoh reboot.

```
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
```

Reserve port ingress - berdampak pada jaringan

Cadangan semua port dalam rentang port alamat ingress Anda

`AwsGroundStationAgentEndpoint` untuk mencegah konflik dengan penggunaan kernel. Konflik penggunaan port akan menyebabkan kegagalan kontak dan pengiriman data.

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

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

Boot ulang

Setelah semua penyetelan berhasil diterapkan, reboot instance agar penyetelan diterapkan.

```
sudo reboot
```

Lampiran: Parameter yang disarankan untuk lagu interrupt/RPS

Bagian ini menentukan nilai parameter yang disarankan untuk digunakan di bagian tuning Tune Hardware Interrupts and Receive Queues - Dampak CPU dan Jaringan.

Rangkaian	Tipe Instans	<code>{interrupt_core_list}</code>	<code>{rps_core_mask}</code>
c7i	<ul style="list-style-type: none"> • c7i.24xlarge • c7i.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, ffffffff
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, • ffffffff,

Rangkaian	Tipe Instans	$\$ \{interrupt_core_list\}$	$\$ \{rps_core_mask\}$
			fffffff, fffffff
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"> fffffff, fffffff, fffffff ff, ffffffff, fffffff ffff, ffffffff
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, ffffffff, fffffff ff, ffffffff, 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"> fffffff, fffffff, fffffff ffff, ffffffff
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"> fffffff, fffffff, fffffff fffffff, fffffff, fffffff
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"> fffffff, fffffff, fffffff fffffff, fffffff, fffffff

Rangkaian	Tipe Instans	<code>interrup t_core_list</code>	<code>rps_core _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"> ffffff, ffffff, ffffff ffffff, ffffff 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"> ffffff, ffffff, ffffff
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"> ffffff, ffffff, ffffff

Bersiaplah untuk mengambil kontak DiGIF

1. Tinjau Perencanaan Inti CPU untuk aliran data yang diinginkan, dan berikan daftar inti yang dapat digunakan agen. Lihat [Perencanaan inti CPU](#).
2. Tinjau file konfigurasi AWS Ground Station Agen. Lihat [AWS Ground Station Konfigurasi agen](#).
3. Konfirmasikan bahwa penyetelan kinerja yang diperlukan diterapkan. Lihat [Menyetel instans EC2 Anda untuk kinerja](#).
4. Konfirmasikan bahwa Anda mengikuti semua praktik terbaik yang disebut. Lihat [Praktik terbaik](#).
5. Konfirmasikan bahwa AWS Ground Station Agen dimulai sebelum waktu mulai kontak yang dijadwalkan melalui:

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

6. Konfirmasikan bahwa AWS Ground Station Agen sehat sebelum waktu mulai kontak yang dijadwalkan melalui:

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

Verifikasi `awsGroundStationAgentEndpoint` bahwa Anda AKTIF dan SEHAT.
`agentStatus auditResults`

Praktik terbaik

Praktik terbaik Amazon EC2

Ikuti praktik terbaik EC2 saat ini dan pastikan ketersediaan penyimpanan data yang memadai.

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

Penjadwal Linux

Penjadwal Linux dapat memesan ulang paket pada socket UDP jika proses yang sesuai tidak disematkan ke inti tertentu. Setiap thread yang mengirim atau menerima data UDP harus menyematkan dirinya ke inti tertentu selama durasi transmisi data.

AWS Ground Station daftar awalan terkelola

Disarankan untuk menggunakan daftar awalan yang `com.amazonaws.global.groundstation` dikelola AWS saat menentukan aturan jaringan untuk memungkinkan komunikasi dari Antena. Lihat [Bekerja dengan Daftar Awalan Terkelola AWS untuk informasi selengkapnya tentang Daftar Awalan Terkelola AWS](#).

Batasan kontak tunggal

Agen AWS Ground Station mendukung beberapa aliran per kontak, tetapi hanya mendukung satu kontak pada satu waktu. Untuk mencegah masalah penjadwalan, jangan bagikan instance di beberapa grup titik akhir aliran data. Jika konfigurasi agen tunggal dikaitkan dengan beberapa DFEG yang berbeda ARNs, itu akan gagal untuk mendaftar.

Menjalankan layanan dan proses bersama AWS Ground Station Agen

Saat meluncurkan layanan dan proses pada Instans EC2 yang sama dengan AWS Ground Station Agen, penting untuk mengikatnya ke `v` yang CPUs tidak digunakan oleh kernel AWS Ground Station Agen dan Linux karena ini dapat menyebabkan kemacetan dan bahkan kehilangan data selama kontak. Konsep pengikatan ke `v` spesifik CPUs ini dikenal sebagai afinitas.

Inti yang harus dihindari:

- agentCpuCores dari [File konfigurasi agen](#)
- interrupt_core_list dari [Menyetel interupsi perangkat keras dan menerima antrian - berdampak pada CPU dan jaringan](#).
 - Nilai default dapat ditemukan dari [Lampiran: Parameter yang disarankan untuk lagu interrupt/RPS](#)

Sebagai contoh menggunakan **c5.24xlarge** instance

Jika Anda menentukan

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

dan berlari

```
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
```

kemudian hindari core berikut:

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

Layanan afinisasi (systemd)

Layanan yang baru diluncurkan akan secara otomatis berhubungan dengan yang interrupt_core_list disebutkan sebelumnya. Jika kasus penggunaan layanan yang Anda luncurkan memerlukan inti tambahan, atau membutuhkan inti yang kurang padat, ikuti bagian ini.

Periksa afinitas layanan Anda saat ini dikonfigurasi dengan perintah:

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

Jika Anda melihat nilai kosong seperti CPUAffinity=, itu berarti kemungkinan akan menggunakan inti default dari perintah di atas ...bin/set_irq_affinity.sh <using the cores here> ...

Untuk mengganti dan menyetel afinitas tertentu, temukan lokasi file layanan dengan menjalankan:

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

Buka dan modifikasi file (menggunakan `vi`, `nano`, dll.) Dan letakkan `CPUAffinity=<core list>` di `[Service]` bagian seperti:

```
[Unit]
...

[Service]
...
CPUAffinity=2,3

[Install]
...
```

Simpan file dan restart layanan untuk menerapkan afinitas dengan:

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

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

Untuk informasi lebih lanjut kunjungi: [Red Hat Enterprise Linux 8 - Mengelola, memantau, dan memperbarui kernel - Bab 27. Mengkonfigurasi kebijakan Afinitas CPU dan NUMA menggunakan systemd.](#)

Proses afinisasi (skrip)

Sangat disarankan untuk skrip dan proses yang baru diluncurkan untuk dikaitkan secara manual karena perilaku Linux default akan memungkinkan mereka untuk menggunakan inti apa pun pada mesin.

Untuk menghindari konflik inti untuk setiap proses yang berjalan (seperti python, skrip bash, dll.), Luncurkan proses dengan:

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

Jika proses sudah berjalan, gunakan perintah seperti `pidof`, `top`, atau `ps` untuk menemukan ID Proses (PID) dari proses tertentu. Dengan PID Anda dapat melihat afinitas saat ini dengan:

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

dan dapat memodifikasinya dengan:

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

Untuk informasi lebih lanjut tentang `taskset` lihat [taskset](#) - Halaman manual Linux

Pemecahan masalah

Agen gagal memulai

AWS Ground Station Agen mungkin gagal memulai karena beberapa alasan, tetapi skenario yang paling umum mungkin adalah file konfigurasi agen yang salah konfigurasi. Setelah memulai agen (lihat [AWS Ground Station Agen mulai](#)) Anda mungkin mendapatkan status seperti:

```
#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)
```

Pemecahan masalah

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

dapat menghasilkan output dari:

```
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.
```

Kegagalan untuk memulai agen setelah “Memuat Config” menunjukkan masalah dengan konfigurasi agen. Lihat [File konfigurasi agen](#) untuk memverifikasi konfigurasi agen Anda.

AWS Ground Station Log agen

AWS Ground Station Agen menulis informasi tentang eksekusi kontak, kesalahan, dan status kesehatan untuk mencatat file pada instance yang menjalankan agen. Anda dapat melihat file log dengan menghubungkan secara manual ke sebuah instance.

Anda dapat melihat log agen di lokasi berikut.

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

Tidak ada kontak yang tersedia

Penjadwalan kontak membutuhkan AWS Ground Station Agen yang sehat. Harap konfirmasi bahwa AWS Ground Station Agen Anda telah memulai dan bahwa itu sehat dengan menanyakan AWS Ground Station API melalui `get-dataflow-endpoint-group`:

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

Verifikasi `awsGroundStationAgentEndpoint` bahwa Anda **AKTIF** dan **SEHAT**. `agentStatus` `auditResults`

Mendapatkan Dukungan

Hubungi tim Ground Station melalui AWS Support.

1. Sediakan `contact_id` untuk setiap kontak yang terkena dampak. AWS Ground Station Tim tidak dapat menyelidiki kontak tertentu tanpa informasi ini.
2. Berikan detail seputar semua langkah pemecahan masalah yang telah diambil.
3. Berikan pesan kesalahan apa pun yang ditemukan saat menjalankan perintah dalam panduan pemecahan masalah kami.

Catatan rilis agen

Versi Agen Terbaru

Versi 1.0.4382.0

Tanggal Rilis: 11/18/2025

Checksum RPM:

- SHA256: 620fd307124f1276194f2faa0104fe0549427ae18e4f5655444f8c30b919c640
- MD5: 73e06dcad44adaccbe2ab005218abfc7

Perubahan:

- Perbarui perilaku coba lagi klien saat server menunjukkan kelebihan beban.

Versi Agen Usang

Versi 1.0.3555.0

Tanggal Rilis: 03/27/2024

Checksum RPM:

- SHA256: 108f3aceb00e5af549839cd766c56149397e448a6e1e1429c89a9eebb6bc0fc1
- MD5: 65b72fa507fb0af32651adbb18d2e30f

Perubahan:

- Tambahkan metrik Agen untuk versi eksekusi yang dipilih selama startup tugas.
- Tambahkan dukungan file konfigurasi untuk menghindari versi yang dapat dieksekusi tertentu saat versi lain tersedia.
- Tambahkan diagnostik jaringan dan perutean.
- Fitur keamanan tambahan.

- Perbaiki masalah di mana beberapa kesalahan pelaporan metrik ditulis, stdout/journal bukan file log.
- Dengan anggun menangani kesalahan soket jaringan yang tidak dapat dijangkau.
- Ukur kehilangan paket dan latensi antara agen sumber dan tujuan.
- Rilis aws-gs-datapipe versi 2.0 untuk mendukung fitur protokol baru dan kemampuan untuk secara transparan meningkatkan kontak ke protokol baru.

Versi 1.0.2942.0

Tanggal Rilis: 06/26/2023

Tanggal Akhir Support: 31/05/2024

Checksum RPM:

- SHA256: 7d94b642577504308a58bab28f938507f2591d4e1b2c7ea170b77bea97b5a9b6
- MD5: 661ff2b8f11aba5d657a6586b56e0d8f

Perubahan:

- Menambahkan log kesalahan saat Agen RPM diperbarui pada disk dan membutuhkan Agen restart agar perubahan diterapkan.
- Menambahkan validasi penyetelan jaringan untuk memastikan langkah-langkah penyetelan panduan pengguna Agen diikuti dan diterapkan dengan benar.
- Perbaiki bug yang menyebabkan peringatan yang salah di log Agen tentang arsip log.
- Peningkatan deteksi kehilangan paket.
- Instalasi Agen yang Diperbarui untuk mencegah pemasangan atau peningkatan RPM jika Agen sudah berjalan.

Versi 1.0.2716.0

Tanggal Rilis: 03/15/2023

Tanggal Akhir Support: 31/05/2024

Checksum RPM:

- SHA256: cb05b6a77dfcd5c66d81c0072ac550affbcefefc372cc5562ee52fb220844929
- MD5: 65266490c4013b433ec39ee50008116c

Perubahan:

- Aktifkan pengunggahan log saat Agen mengalami kegagalan selama tugas.
- Perbaiki bug kompatibilitas linux dalam skrip penyetelan jaringan yang disediakan.

Versi 1.0.2677.0

Tanggal Rilis: 02/15/2023

Tanggal Akhir Support: 31/05/2024

Checksum RPM:

- SHA256: 77cfe94acb00af7ca637264b17c9b21bd7afdc85b99dffdd627aec9e99397489
- MD5: b8533be7644bb4d12ab84de21341adac

Perubahan:

- Rilis Agen pertama yang tersedia secara umum.

Validasi instalasi RPM

Versi RPM terbaru, MD5 hash divalidasi dari RPM, dan SHA256 hash menggunakan sha256sum ditunjukkan di bawah ini. Nilai-nilai ini, digabungkan, dapat digunakan untuk memvalidasi versi RPM yang digunakan untuk agen stasiun bumi.

Versi Agen Terbaru

Versi 1.0.4382.0

Tanggal Rilis: 11/18/2025

Checksum RPM:

- SHA256: 620fd307124f1276194f2faa0104fe0549427ae18e4f5655444f8c30b919c640
- MD5: 73e06dcad44adaccbe2ab005218abfc7

Perubahan:

- Perbarui perilaku coba lagi klien saat server menunjukkan kelebihan beban.

Verifikasi RPM

Alat yang Anda perlukan untuk dapat memverifikasi instalasi RPM ini adalah:

- [sha256jumlah](#)
- [rpm](#)

Kedua alat datang secara default di Amazon Linux 2. Alat-alat ini akan membantu memvalidasi bahwa RPM yang Anda gunakan adalah versi yang benar. Pertama unduh RPM terbaru dari bucket S3 (lihat [Agen unduhan](#) petunjuk tentang mengunduh RPM). Setelah file ini diunduh, akan ada beberapa hal yang perlu diperiksa:

- Hitung sha256sum dari file RPM. Lakukan tindakan berikut dari baris perintah instance komputasi yang Anda gunakan:

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

Ambil nilai ini dan bandingkan dengan tabel di atas. Ini menunjukkan bahwa file RPM yang diunduh adalah file yang valid untuk digunakan yang AWS Ground Station telah dijual kepada pelanggan. Jika hash tidak cocok, jangan instal RPM, dan hapus dari instance komputasi.

- Periksa MD5 hash file juga, untuk memastikan bahwa RPM belum dikompromikan. Untuk melakukan ini, gunakan alat baris perintah RPM dengan menjalankan perintah berikut:

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

Validasi bahwa MD5 hash yang tercantum di sini sama dengan MD5 hash dari versi yang ada dalam tabel di atas. Setelah kedua hash ini divalidasi terhadap tabel ini yang tercantum dalam AWS Docs, pelanggan dapat dipastikan bahwa RPM yang diunduh dan diinstal adalah versi RPM yang aman dan tanpa kompromi.

Riwayat dokumen untuk Panduan Pengguna AWS Ground Station Agen

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengguna AWS Ground Station Agen.

Perubahan	Deskripsi	Tanggal
Pembaruan Dokumentasi	Dukungan yang dihapus untuk keluarga instance yang lebih lama: m4.	September 30, 2024
Pembaruan Dokumentasi	Menambahkan komentar tentang menyimpan subnet dan EC2 instans Amazon di zona ketersediaan yang sama di Persyaratan Agen .	Juli 18, 2024
Pembaruan Dokumentasi	Pisahkan AWS Ground Station Agen menjadi panduan penggunaannya sendiri. Untuk perubahan sebelumnya, silakan lihat: Riwayat dokumen untuk panduan pengguna AWS Ground Station .	Juli 18, 2024

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.