



AWS Ground Station Benutzerhandbuch für Agenten

AWS Ground Station



AWS Ground Station: AWS Ground Station Benutzerhandbuch für Agenten

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Übersicht	1
Was ist der AWS Ground Station Agent?	1
Funktionen des Agenten AWS Ground Station	2
Anforderungen an den Agenten	3
VPCDiagramme	4
Unterstütztes Betriebssystem	5
Datenzustellung über den AWS Ground Station Agenten	6
Mehrere Datenflüsse, ein einziger Empfänger	6
Mehrere Datenflüsse, mehrere Empfänger	7
Auswahl und CPU Planung von EC2 Amazon-Instances	9
Unterstützte EC2 Amazon-Instance-Typen	9
CPUKernplanung	10
Sammeln von Architekturinformationen	11
CPUBeispiel für eine Zuweisung	13
Anhang: <code>lscpu -p</code> Ausgabe (vollständig) für <code>c5.24xlarge</code>	13
Den Agenten installieren	17
AWS CloudFormation Vorlage verwenden	17
Schritt 1: Ressourcen erstellen AWS	17
Schritt 2: Überprüfen Sie den Agentenstatus	17
Manuelle Installation auf EC2	17
Schritt 1: AWS Ressourcen erstellen	17
Schritt 2: EC2 Instanz erstellen	18
Schritt 3: Laden Sie den Agenten herunter und installieren Sie ihn	18
Schritt 4: Konfigurieren Sie den Agenten	20
Schritt 5: Leistungsoptimierung anwenden	20
Schritt 6: Den Agenten verwalten	20
Den Agenten verwalten	21
AWS Ground Station Konfiguration des Agenten	21
AWS Ground Station Der Agent wird gestartet	21
AWS Ground Station Der Agent wird gestoppt	22
AWS Ground Station Agenten-Upgrade	22
AWS Ground Station Downgrade des Agenten	23
AWS Ground Station Deinstallation des Agenten	24
AWS Ground Station Status des Agenten	24

AWS Ground Station RPM Informationen zum Agenten	25
Den Agenten konfigurieren	26
Agenten-Konfigurationsdatei	26
Beispiel	26
Aufschlüsselung der Felder	26
EC2 Optimierung der Instanzleistung	30
Optimieren Sie Hardware-Interrupts und Empfangswarteschlangen — Auswirkungen auf CPU das Netzwerk	30
Tune Rx Interrupt Coalescing — wirkt sich auf das Netzwerk aus	31
Tune Rx Ring Buffer — wirkt sich auf das Netzwerk aus	32
Tune C-State — Auswirkungen CPU CPU	32
Eingangsports reservieren — wirkt sich auf das Netzwerk aus	33
Neustart	33
Anhang: Empfohlene Parameter für Interrupt/Tune RPS	33
Bereiten Sie sich darauf vor, einen DigiF-Kontakt aufzunehmen	36
Bewährte Methoden	37
EC2 Bewährte Methoden von Amazon	37
Linux-Scheduler	37
AWS Ground Station verwaltete Präfixliste	37
Beschränkung auf einen einzigen Kontakt	37
Dienste und Prozesse werden zusammen mit dem AWS Ground Station Agenten ausgeführt	37
Als Beispiel mit einer c5.24xlarge Instanz	38
Affinialisierung von Diensten (systemd)	38
Affinialisierung von Prozessen (Skripten)	40
Fehlerbehebung	41
Der Agent kann nicht gestartet werden	41
Fehlerbehebung	41
AWS Ground Station Agent-Protokolle	42
Keine Kontakte verfügbar	42
Supportanfragen	43
Versionshinweise für Agenten	44
Aktuelle Agent-Version	44
Version 1.0.3555.0	44
Veraltete Agent-Versionen	44
Version 1.0.2942.0	44
Version 1.0.2716.0	45

Version 1.0.2677.0	46
RPMÜberprüfung der Installation	47
Aktuelle Agent-Version	44
Version 1.0.3555.0	44
Überprüfen Sie die RPM	48
Dokumentverlauf	49
.....	

Übersicht

Was ist der AWS Ground Station Agent?

Der AWS Ground Station Agent, der als verfügbar istRPM, ermöglicht Ihnen den Empfang (Downlink) synchroner Breitband-Datenflüsse mit digitaler Zwischenfrequenz (DigiF) bei Kontakten mit der Ground Station. AWS Sie können zwei Optionen für die Datenübermittlung wählen:

1. Datenlieferung an eine EC2 Instanz — Datenlieferung an eine EC2 Instanz, die Sie besitzen. Sie verwalten den AWS Ground Station Agenten. Diese Option eignet sich möglicherweise am besten für Sie, wenn Sie eine Datenverarbeitung nahezu in Echtzeit benötigen. Informationen [zur Datenbereitstellung finden Sie im Leitfaden EC2 Datenlieferung an Amazon Elastic Compute Cloud](#).
2. Datenlieferung an einen S3-Bucket — Datenlieferung an einen AWS S3-Bucket, den Sie besitzen, über einen von der Ground Station verwalteten Service. Informationen zur S3-Datenbereitstellung finden Sie im AWS Ground Station Leitfaden [Erste Schritte mit](#).

Für beide Arten der Datenübermittlung müssen Sie eine Reihe von AWS Ressourcen erstellen. Die Verwendung von CloudFormation zur Erstellung Ihrer AWS Ressourcen wird dringend empfohlen, um Zuverlässigkeit, Genauigkeit und Unterstützbarkeit zu gewährleisten. Jeder Kontakt kann nur Daten an EC2 oder S3, aber nicht an beide gleichzeitig liefern.

Note

Da es sich bei S3 Data Delivery um einen von der Ground Station verwalteten Service handelt, konzentriert sich dieser Leitfaden auf die Datenbereitstellung für Ihre EC2 Instance (en).

Das folgende Diagramm zeigt einen DigiF-Datenfluss von einer AWS Ground Station Antennenregion zu Ihrer EC2 Instance mit Ihrem softwaredefinierten Radio (SDR) oder einem ähnlichen Listener.



Funktionen des Agenten AWS Ground Station

Der AWS Ground Station Agent empfängt Downlink-Daten (Digital Intermediate Frequency, DigiF) und sendet entschlüsselte Daten aus, die Folgendes ermöglichen:

- DigiF-Downlink-Fähigkeit von 40 MHz bis 400 MHz Bandbreite.
- DigiF-Datenübermittlung mit hoher Rate und geringem Jitter an jede öffentliche IP (AWSElastic IP) im AWS Netzwerk.
- Zuverlässige Datenübermittlung mit Forward Error Correction (FEC).
- Sichere Datenübermittlung mit einem vom Kunden verwalteten AWS KMS Schlüssel zur Verschlüsselung.

Anforderungen an den Agenten

Note

In diesem AWS Ground Station Agenten-Handbuch wird davon ausgegangen, dass Sie sich mithilfe des Handbuchs [AWS Ground Station Erste Schritte](#) an der Ground Station angemeldet haben.

Die AWS Ground Station EC2 Agent-Empfänger-Instanz benötigt eine Reihe von abhängigen AWS Ressourcen, um DigiF-Daten zuverlässig und sicher an Ihre Endpunkte zu liefern.

1. A, VPC in dem der Receiver gestartet werden soll. EC2
2. Ein AWS KMS Schlüssel für die Verschlüsselung/Entschlüsselung von Daten.
3. [Ein für Session Manager konfigurierter SSH Schlüssel oder EC2 Instanzprofil. SSM](#)
4. Netzwerk-/Sicherheitsgruppenregeln, die Folgendes ermöglichen:
 1. UDP-Datenverkehr von AWS Ground Station den Ports, die in Ihrer Datenfluss-Endpunktgruppe angegeben sind. Der Agent reserviert eine Reihe von zusammenhängenden Ports, die für die Übertragung von Daten an die Endpunkte des eingehenden Datenflusses verwendet werden.
 2. SSH-Zugriff auf Ihre Instance (Hinweis: Sie können alternativ AWS Session Manager verwenden, um auf Ihre Instance zuzugreifen). EC2
 3. Lesezugriff auf einen öffentlich zugänglichen S3-Bucket für die Agentenverwaltung.
 4. SSL-Verkehr auf Port 443, der es dem Agenten ermöglicht, mit dem AWS Ground Station Dienst zu kommunizieren.
 5. Verkehr aus der Liste der AWS Ground Station verwalteten Präfixe `com.amazonaws.global.groundstation`.

Darüber hinaus ist eine VPC Konfiguration mit einem öffentlichen Subnetz erforderlich. Hintergrundinformationen zur Subnetzkonfiguration finden Sie im [VPC Benutzerhandbuch](#).

Kompatible Konfigurationen:

1. Eine Elastic IP, die Ihrer EC2 Instance in einem öffentlichen Subnetz zugeordnet ist.

2. Eine Elastic IP, die einem ENI in einem öffentlichen Subnetz zugeordnet ist und mit Ihrer EC2 Instance verbunden ist (in einem beliebigen Subnetz in derselben Availability Zone wie das öffentliche Subnetz).

Sie können dieselbe Sicherheitsgruppe wie Ihre EC2 Instance verwenden oder eine mit mindestens den folgenden Mindestregeln angeben:

- UDP-Datenverkehr von AWS Ground Station den Ports, die in Ihrer Datenfluss-Endpointgruppe angegeben sind.

Vorlagen für die AWS CloudFormation EC2 Datenübermittlung, bei denen diese Ressourcen vorkonfiguriert sind, finden Sie beispielsweise unter [Öffentlicher Rundfunksatellit mit AWS Ground Station Agent \(Breitband\)](#).

VPC-Diagramme

Diagramm: Eine Elastic IP, die Ihrer EC2 Instance in einem öffentlichen Subnetz zugeordnet ist

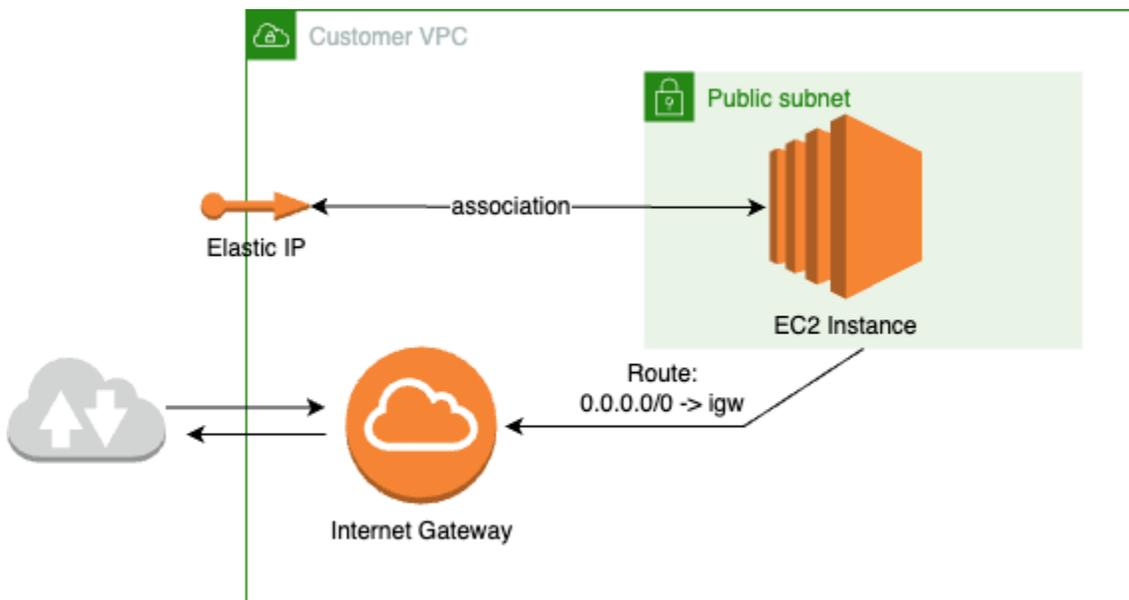
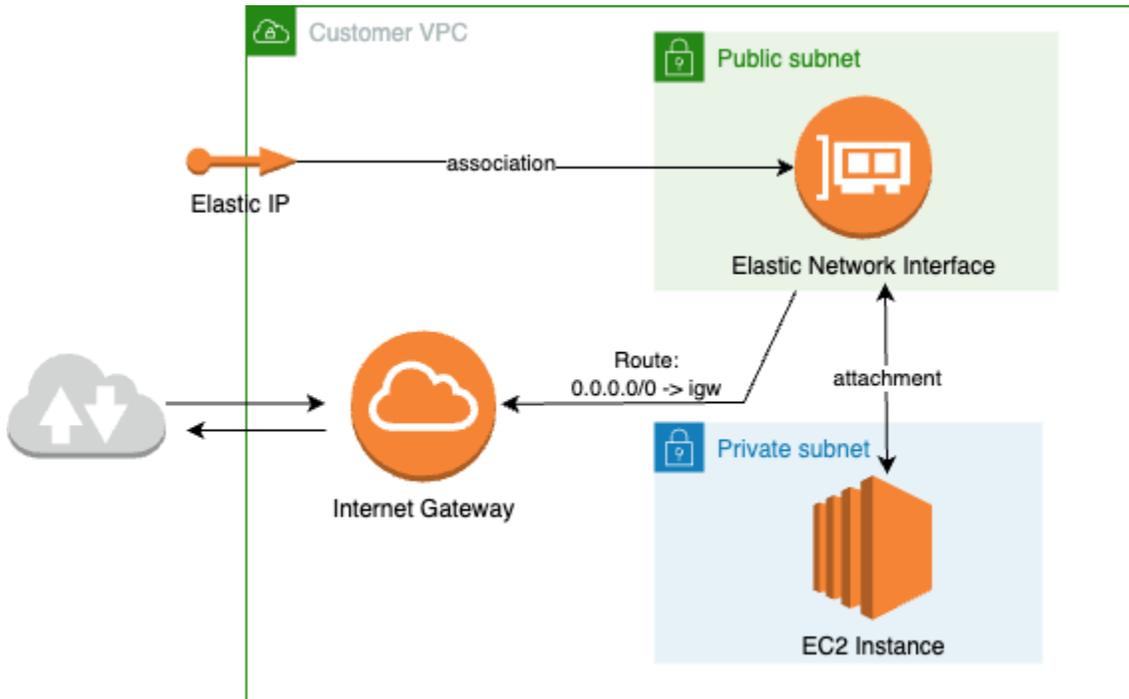


Diagramm: Eine Elastic IP, die mit einem ENI in einem öffentlichen Subnetz verknüpft ist und mit Ihrer EC2 Instance in einem privaten Subnetz verbunden ist



Unterstütztes Betriebssystem

Amazon Linux 2 mit Kernel 5.10+.

Die unterstützten Instance-Typen sind unter aufgeführt [Auswahl und CPU Planung von EC2 Amazon-Instances](#)

Datenzustellung über den AWS Ground Station Agenten

Die folgenden Diagramme geben einen Überblick darüber, wie Daten AWS Ground Station bei Kontakten mit Breitband-Digitalinterfrequenz (DigiF) durchfließen.

Der AWS Ground Station Agent kümmert sich um die Orchestrierung der Datenebenenkomponenten für einen Kontakt. Vor der Planung eines Kontakts muss der Agent korrekt konfiguriert, gestartet und registriert sein (die Registrierung erfolgt automatisch beim Start des Agenten). AWS Ground Station Darüber hinaus muss die Datenempfangssoftware (z. B. ein softwaredefiniertes Radio) ausgeführt und so konfiguriert sein, dass sie Daten am empfängt [AwsGroundStationAgentEndpoint](#)gressAddress.

Im Hintergrund empfängt der AWS Ground Station Agent Aufgaben von der AWS KMS Verschlüsselung, die während der Übertragung angewendet wurde, AWS Ground Station und macht sie rückgängig, bevor er sie an den Zielendpunkt weiterleitet, egressAddress an dem Ihr softwaredefiniertes Radio (SDR) zuhört. Der AWS Ground Station Agent und die zugrundeliegenden Komponenten respektieren die in der Konfigurationsdatei festgelegten CPU Grenzen, um sicherzustellen, dass die Leistung anderer Anwendungen, die auf der Instanz ausgeführt werden, nicht beeinträchtigt wird.

Der AWS Ground Station Agent muss auf der Empfängerinstanz laufen, die an dem Kontakt beteiligt war. Ein einzelner AWS Ground Station Agent kann mehrere Datenflüsse orchestrieren, wie unten dargestellt, wenn Sie es vorziehen, alle Datenflüsse auf einer einzigen Empfängerinstanz zu empfangen.

Mehrere Datenflüsse, ein einziger Empfänger

Beispielszenario:

Sie möchten zwei Antennen-Downlinks als DigiF-Datenflüsse auf derselben Empfängerinstanz empfangen. EC2 Die beiden Downlinks werden 200 und 100 sein. MHz MHz

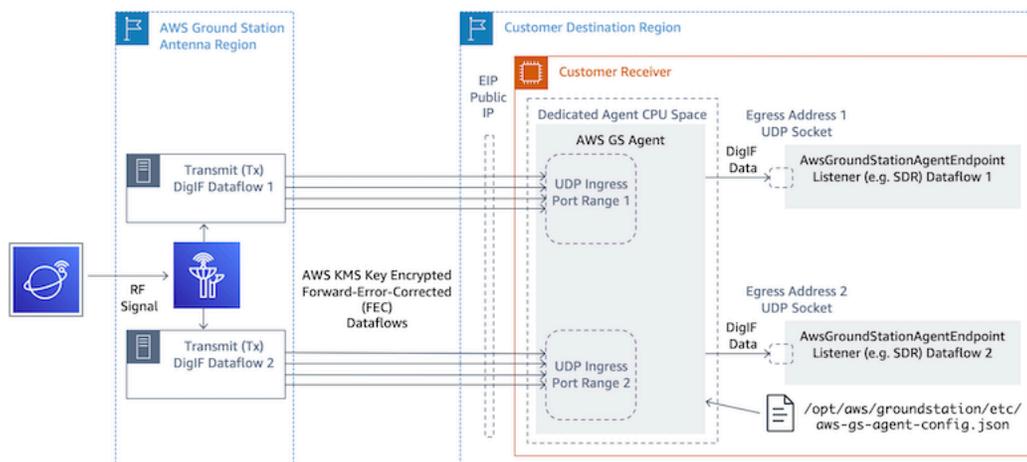
AwsGroundStationAgentEndpoints:

Es wird zwei AwsGroundStationAgentEndpoint Ressourcen geben, eine für jeden Datenfluss. Beide Endpunkte werden dieselbe öffentliche IP-Adresse () haben. ingressAddress.socketAddress.name Die Eingänge sollten portRange sich nicht

überschneiden, da die Datenflüsse auf derselben Instanz empfangen werden. EC2 Beide müssen `egressAddress.socketAddress.port` einzigartig sein.

CPUPlanung:

- 1 Kern (2 vCPU) zum Ausführen des Single AWS Ground Station Agents auf der Instance.
- 6 Kerne (12 VCPU) für den Empfang von DigIF Dataflow 1 (200 Abfragen in [CPUKernplanung](#) der MHz Tabelle).
- 4 Kerne (8 VCPU) für den Empfang von DigIF Dataflow 2 (100 Abfragen in [CPUKernplanung](#) der MHz Tabelle).
- Gesamter dedizierter CPU Agentenspeicher = 11 Kerne (22 VCPU) auf demselben Socket.



Mehrere Datenflüsse, mehrere Empfänger

Beispielszenario:

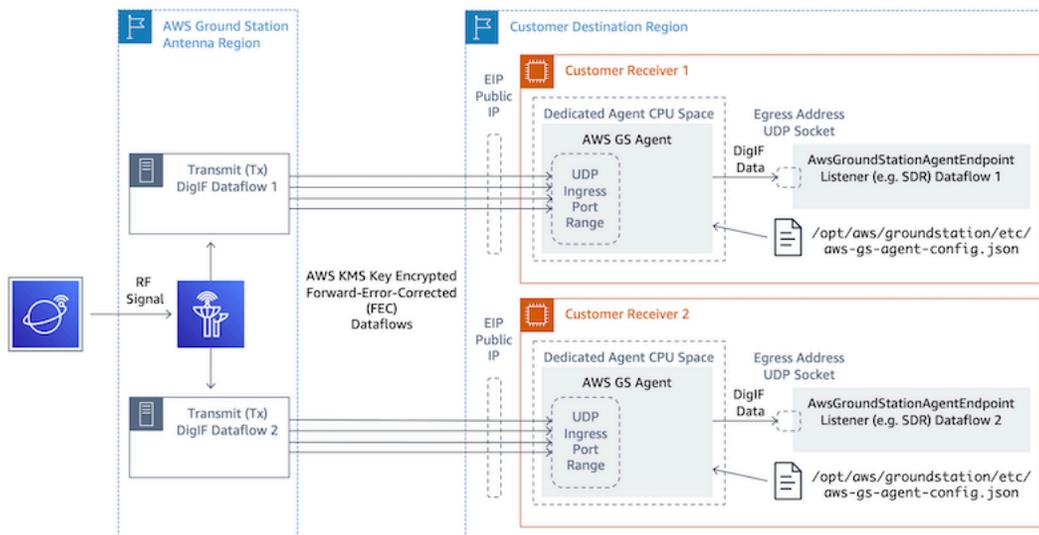
Sie möchten zwei Antennen-Downlinks als DigiF-Datenflüsse an verschiedenen Empfängerinstanzen empfangen. EC2 Beide Downlinks werden 400 sein. MHz

AwsGroundStationAgentEndpoints:

Es wird zwei `AwsGroundStationAgentEndpoint` Ressourcen geben, eine für jeden Datenfluss. Die Endpunkte werden eine andere öffentliche IP-Adresse () haben. `ingressAddress.socketAddress.name` Es gibt keine Beschränkung der Portwerte für beide `ingressAddress` oder, `egressAddress` da die Datenflüsse auf einer separaten Infrastruktur empfangen werden und nicht miteinander in Konflikt geraten.

CPUPlanung:

- Empfänger-Instanz 1
 - 1 Kern (2 VCPU) zum Ausführen des einzelnen AWS Ground Station Agenten auf der Instanz.
 - 9 Kerne (18 VCPU) für den Empfang von DigIF Dataflow 1 (MHz400-Suche in [CPUKernplanung](#) der Tabelle).
 - Gesamter dedizierter CPU Agentenspeicher = 10 Kerne (20 VCPU) auf demselben Socket.
- Empfänger-Instanz 2
 - 1 Kern (2 vCPU) zum Ausführen des einzelnen AWS Ground Station Agenten auf der Instanz.
 - 9 Kerne (18 VCPU) für den Empfang von DigIF Dataflow 2 (MHz400-Suche in [CPUKernplanung](#) der Tabelle).
 - Gesamter dedizierter CPU Agentenspeicher = 10 Kerne (20 VCPU) auf demselben Socket.



Auswahl und CPU Planung von EC2 Amazon-Instances

Unterstützte EC2 Amazon-Instance-Typen

Aufgrund der rechenintensiven Workflows zur Datenbereitstellung benötigt der AWS Ground Station Agent dedizierte CPU Kerne für den Betrieb. Wir unterstützen die folgenden Instance-Typen.

[CPUKernplanung](#) Entscheiden Sie selbst, welcher Instance-Typ am besten zu Ihrem Anwendungsfall passt.

Instance-Typ	Standard vCPUs	CPUStandardkerne
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.12xgroß	48	24
g4dn.16xgroß	64	32
g4dn.metal	96	48
m4.16xlarge	64	32
m5.12xlarge	48	24
m5.24xlarge	96	48
m6i.32xlarge	128	64
p3dn.24xgroß	96	48
p4d.24xgroß	96	48

Instance-Typ	Standard vCPUs	CPUSandardkerne
r5.24xlarge	96	48
r5.metal	96	48
r5n.24xlarge	96	48
r5n.metal	96	48
r6i.32xlarge	128	64

CPUKernplanung

Der AWS Ground Station Agent benötigt dedizierte Prozessorkerne, die sich den L3-Cache für jeden Datenfluss teilen. Der Agent ist für die Nutzung von Hyper-Threaded (HT) CPU -Paaren konzipiert und erfordert, dass HT-Paare für seine Verwendung reserviert werden. Ein Hyperthread-Paar ist ein Paar virtueller CPUs (vCPU), die in einem einzigen Kern enthalten sind. Die folgende Tabelle bietet eine Zuordnung der Datenfluss-Datenrate zur erforderlichen Anzahl von Kernen, die für den Agenten für einen einzelnen Datenfluss reserviert sind. In dieser Tabelle wird von Cascade Lake oder einer neueren Version ausgegangen CPUs und sie ist für jeden unterstützten Instance-Typ gültig. Wenn Ihre Bandbreite zwischen den Einträgen in der Tabelle liegt, wählen Sie die nächsthöhere aus.

Der Agent benötigt einen zusätzlichen reservierten Kern für Verwaltung und Koordination. Die Gesamtzahl der benötigten Kerne entspricht also der Summe der benötigten Kerne (aus der Tabelle unten) für jeden Datenfluss plus einem einzelnen zusätzlichen Kern (2 vCPUs).

AntennaDownlink Bandbreite () MHz	Erwartete VITA -49,2 DigiF-Dat enrate (MB/s)	Anzahl der Kerne (HT-Paare) CPU	Insgesamt v CPU
50	1000	3	6
100	2000	4	8
150	3000	5	10

AntennaDownlink Bandbreite (MHz)	Erwartete VITA -49,2 DigiF-Dat enrate (MB/s)	Anzahl der Kerne (HT-Paare) CPU	Insgesamt v CPU
200	4000	6	12
250	5000	6	12
300	6 000	7	14
350	7000	8	16
400	8000	9	18

Sammeln von Architekturinformationen

lscpu bietet Informationen über die Architektur Ihres Systems. Die grundlegende Ausgabe zeigt, welche vCPUs (als "CPU," gekennzeichnet) zu welchen NUMA Knoten gehören (und jeder NUMA Knoten teilt sich einen L3-Cache). Im Folgenden untersuchen wir eine c5.24xlarge Instanz, um die für die Konfiguration des AWS Ground Station Agenten erforderlichen Informationen zu sammeln. Dazu gehören nützliche Informationen wie die Anzahl der vCPUs Kerne und die Zuordnung von vCPU zu Knoten.

```
> 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   <-----

```

Dem AWS Ground Station Agenten zugewiesene Kerne sollten beide Kerne vCPUs für jeden zugewiesenen Kern enthalten. Alle Kerne für einen Datenfluss müssen auf demselben NUMA Knoten vorhanden sein. Die `-p` Option für den `lscpu` Befehl liefert uns den Kern der CPU Verknüpfungen, die für die Konfiguration des Agenten erforderlich sind. Die relevanten Felder sind CPU (was wir als `V` bezeichnen CPU), Core und L3 (was angibt, welcher L3-Cache von diesem Core gemeinsam genutzt wird). Beachten Sie, dass bei den meisten Intel-Prozessoren der NUMA Node dem L3-Cache entspricht.

Betrachten Sie die folgende Teilmenge der `lscpu -p` Ausgabe als Beispiel `c5.24xlarge` (aus Gründen der Übersichtlichkeit abgekürzt und formatiert).

```

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

```

Aus der Ausgabe können wir ersehen, dass Core 0 vCPUs 0 und 16 enthält, Core 1 vCPUs 1 und 17, Core 2 2 und vCPUs 18. Mit anderen Worten, die Hyper-Thread-Paare sind: 0 und 16, 1 und 17, 2 und 18.

CPU-Beispiel für eine Zuweisung

Als Beispiel verwenden wir eine `c5.24xlarge` Instanz für einen Breitband-Downlink mit doppelter Polarität bei 350. MHz. Aus der Tabelle in [wissen CPU-Kernplanung](#) wissen wir, dass ein MHz 350-Downlink 8 Kerne (16vCPUs) für den einzelnen Datenfluss benötigt. Das bedeutet, dass für dieses Setup mit doppelter Polarität, bei dem zwei Datenflüsse verwendet werden, insgesamt 16 Kerne (32vCPUs) plus ein Kern (2vCPUs) für den Agenten erforderlich sind.

Wir wissen, dass die `lscpu` Ausgabe für `c5.24xlarge` beinhaltet: `NUMA node0 CPU(s): 0-23, 48-71` `NUMA node1 CPU(s): 24-47, 72-95`. Da `NUMA node0` mehr hat, als wir benötigen, werden wir nur von den Kernen aus zuweisen: 0-23 und 48-71.

Zunächst wählen wir 8 Kerne für jeden Datenfluss aus, die sich einen L3-Cache oder Node teilen. `NUMA` Dann suchen wir in der Ausgabe in nach dem entsprechenden vCPUs (mit „CPU“ beschriftet). `lscpu -p` [Anhang: lscpu -p Ausgabe \(vollständig\) für c5.24xlarge](#) Ein Beispiel für einen zentralen Auswahlprozess könnte wie folgt aussehen:

- Reservieren Sie die Kerne 0-1 für das Betriebssystem.
- Flow 1: Wählen Sie die Kerne 2-9 aus, die vCPUs 2-9 und 50-57 zugeordnet sind.
- Flow 2: Wählen Sie die Kerne 10-17 aus, die 10-17 und 58-65 zugeordnet vCPUs sind.
- Agentenkern: Wählen Sie Kern 18 aus, der 18 und 66 zugeordnet vCPUs ist.

Das Ergebnis sind vCPUs 2-18 und 50-66, sodass die Liste, die der Agent bereitstellen muss, lautet. `[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]` Sie sollten sicherstellen, dass Ihre eigenen Prozesse nicht auf diesen Systemen ausgeführt werden, CPUs wie unter beschrieben. [Dienste und Prozesse werden zusammen mit dem AWS Ground Station Agenten ausgeführt](#)

Beachten Sie, dass die in diesem Beispiel ausgewählten spezifischen Kerne etwas willkürlich sind. Andere Gruppen von Kernen würden funktionieren, solange sie die Anforderung erfüllen, dass sich alle für jeden Datenfluss einen L3-Cache teilen.

Anhang: `lscpu -p` Ausgabe (vollständig) für `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
```

Den Agenten installieren

Der AWS Ground Station Agent kann auf folgende Weise installiert werden:

1. AWS CloudFormation Vorlage (empfohlen).
2. Manuelle Installation bei AmazonEC2.

AWS CloudFormation Vorlage verwenden

Die EC2 AWS CloudFormation Datenlieferungsvorlage erstellt die erforderlichen AWS Ressourcen, um Daten an Ihre EC2 Instanz zu liefern. Diese AWS CloudFormation Vorlage verwendet die AWS Ground Station verwaltete VorlageAMI, auf der der AWS Ground Station Agent vorinstalliert ist. Das Boot-Skript der erstellten EC2 Instanz füllt dann die Agenten-Konfigurationsdatei auf und wendet die erforderliche Leistungsoptimierung an ([EC2Optimierung der Instanzleistung](#)).

Schritt 1: Ressourcen erstellen AWS

Erstellen Sie Ihren AWS Ressourcenstapel mithilfe der Vorlage [Public Broadcast Satellite AWS using Ground Station Agent \(Breitband\)](#).

Schritt 2: Überprüfen Sie den Agentenstatus

Standardmäßig ist der Agent konfiguriert und aktiv (gestartet). Um den Agentenstatus zu überprüfen, können Sie eine Verbindung zur EC2 Instanz (SSH oder zum SSM Sitzungsmanager) herstellen und nachschauen [AWS Ground Station Status des Agenten](#).

Manuelle Installation auf EC2

Ground Station empfiehlt zwar die Verwendung von CloudFormation Vorlagen für die Bereitstellung Ihrer AWS Ressourcen, es kann jedoch Anwendungsfälle geben, in denen die Standardvorlage möglicherweise nicht ausreicht. In solchen Fällen empfehlen wir Ihnen, die Vorlage an Ihre Bedürfnisse anzupassen. Wenn das immer noch nicht Ihren Anforderungen entspricht, können Sie Ihre AWS Ressourcen manuell erstellen und den Agenten installieren.

Schritt 1: AWS Ressourcen erstellen

Anweisungen zum manuellen Einrichten der für einen Kontakt erforderlichen AWS Ressourcen finden Sie unter [Beispielkonfigurationen für Missionsprofile](#).

Die `AwsGroundStationAgentEndpointResource` definiert einen Endpunkt für den Empfang eines DigiF-Datenflusses über den AWS Ground Station Agenten und ist entscheidend für die erfolgreiche Kontaktaufnahme. Die API Dokumentation befindet sich zwar in der [APIReferenz](#), in diesem Abschnitt werden jedoch kurz Konzepte behandelt, die für den Agenten relevant sind. AWS Ground Station

Auf dem Endpunkt empfängt der AWS Ground Station Agent AWS KMS verschlüsselten UDP Datenverkehr von der Antenne. `ingressAddress` Das `socketAddress` name ist die öffentliche IP der EC2 Instanz (von der angehängtenEIP). Es `portRange` sollten mindestens 300 zusammenhängende Ports in einem Bereich sein, der für jegliche andere Nutzung reserviert wurde. Detaillierte Anweisungen finden Sie unter [Eingangsports reservieren — wirkt sich auf das Netzwerk aus](#). Diese Ports müssen so konfiguriert sein, dass sie UDP eingehenden Datenverkehr in der Sicherheitsgruppe zulassen, auf der die VPC Empfängerinstanz ausgeführt wird.

Auf dem Endpunkt übergibt `egressAddress` der Agent den DigiF-Datenfluss an Sie. Sie sollten eine Anwendung haben (z. B.SDR), die die Daten über einen UDP Socket an diesem Standort empfängt.

Schritt 2: EC2 Instanz erstellen

Die folgenden AMIs werden unterstützt:

1. AWS Ground Station AMI- `groundstation-a12-gs-agent-ami-*` wobei `*` für das Datum steht, an dem die AMI Datei erstellt wurde. Im Lieferumfang ist der Agent bereits installiert (empfohlen).
2. `amzn2-ami-kernel-5.10-hvm-x86_64-gp2`.

Schritt 3: Laden Sie den Agenten herunter und installieren Sie ihn

Note

Die Schritte in diesem Abschnitt müssen abgeschlossen werden, wenn Sie AMI im vorherigen Schritt den AWS Ground Station Agenten nicht ausgewählt haben.

Laden Sie den Agenten herunter

Der AWS Ground Station Agent ist in regionsspezifischen S3-Buckets verfügbar und kann mithilfe der AWS Befehlszeile (CLI) auf EC2 Support-Instances heruntergeladen werden, `s3://`

groundstation-wb-digif-software-`${AWS::Region}`/aws-groundstation-agent/latest/amazon_linux_2_x86_64/aws-groundstation-agent.rpm wobei `${AWS::Region}` auf eine der unterstützten [AWS Ground Station Console und Data Delivery](#) Regions verweist.

Beispiel: Laden Sie die neueste RPM-Version aus der AWS Region us-east-2 lokal in den /tmp-Ordner herunter.

```
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
```

Wenn Sie eine bestimmte Version des AWS Ground Station Agenten herunterladen müssen, können Sie sie aus dem versionsspezifischen Ordner im S3-Bucket herunterladen.

Beispiel: Laden Sie Version 1.0.2716.0 von rpm aus der AWS Region us-east-2 lokal in den /tmp-Ordner herunter.

```
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

Wenn Sie überprüfen möchten, ob das von RPM Ihnen heruntergeladene Produkt verkauft wurde, folgen Sie den Anweisungen für [AWS Ground Station RPM Überprüfung der Installation](#)

Installieren Sie den Agenten

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

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

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

Schritt 4: Konfigurieren Sie den Agenten

Nach der Installation des Agenten müssen Sie die Agenten-Konfigurationsdatei aktualisieren. Siehe [Den Agenten konfigurieren](#).

Schritt 5: Leistungsoptimierung anwenden

AWS Ground Station Agent AMI: Wenn Sie AMI im vorherigen Schritt den AWS Ground Station Agenten ausgewählt haben, wenden Sie die folgenden Leistungsoptimierungen an.

- [Optimieren Sie Hardware-Interrupts und Empfangswarteschlangen — Auswirkungen auf CPU das Netzwerk](#)
- [Eingangsports reservieren — wirkt sich auf das Netzwerk aus](#)
- [Neustart](#)

Andere AMIs: Wenn Sie AMI im vorherigen Schritt eine andere Option ausgewählt haben, wenden Sie alle unter aufgeführten Optimierungen an [EC2Optimierung der Instanzleistung](#) und starten Sie die Instanz neu.

Schritt 6: Den Agenten verwalten

Informationen zum Starten, Beenden und Überprüfen des Agentenstatus finden Sie unter [Den Agenten verwalten](#).

Den Agenten verwalten

AWS Ground Station Der Agent bietet die folgenden Funktionen zum Konfigurieren, Starten, Stoppen, Aktualisieren, Herabstufen und Deinstallieren des Agenten mithilfe der integrierten Linux-Befehlstools.

Topics

- [AWS Ground Station Konfiguration des Agenten](#)
- [AWS Ground Station Der Agent wird gestartet](#)
- [AWS Ground Station Der Agent wird gestoppt](#)
- [AWS Ground Station Agenten-Upgrade](#)
- [AWS Ground Station Downgrade des Agenten](#)
- [AWS Ground Station Deinstallation des Agenten](#)
- [AWS Ground Station Status des Agenten](#)
- [AWS Ground Station RPM Informationen zum Agenten](#)

AWS Ground Station Konfiguration des Agenten

Navigieren Sie zu `/opt/aws/groundstation/etc`, das eine einzelne Datei namens `aws-gs-agent-config.json` enthalten sollte. Siehe [Agenten-Konfigurationsdatei](#)

AWS Ground Station Der Agent wird gestartet

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

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

Sollte eine Ausgabe erzeugen, die zeigt, dass der Agent aktiv ist.

```
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 Der Agent wird gestoppt

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

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

Sollte eine Ausgabe erzeugen, die zeigt, dass der Agent inaktiv (gestoppt) ist.

```
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 Agenten-Upgrade

1. Laden Sie die neueste Version des Agenten herunter. Siehe [Laden Sie den Agenten herunter](#).
2. Beenden des -Agenten.

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

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

3. Aktualisieren Sie den Agenten.

```
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 des Agenten

1. Laden Sie die benötigte Agentenversion herunter. Siehe [Laden Sie den Agenten herunter](#).
2. Führen Sie ein Downgrade des Agenten durch.

```
# 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 Deinstallation des Agenten

Bei der Deinstallation des Agenten wird `/opt/aws/groundstation/etc/.json` in `/opt/aws/groundstation/etc/.json.rpmsave aws-gs-agent-config` umbenannt. Wenn Sie den Agenten erneut auf derselben Instanz installieren, werden Standardwerte für `.json` geschrieben und müssen mit den richtigen Werten aktualisiert werden, die Ihren Ressourcen entsprechen. Siehe [Agenten-Konfigurationsdatei](#).

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

AWS Ground Station Status des Agenten

Der Agentenstatus ist entweder aktiv (Agent läuft) oder inaktiv (Agent ist gestoppt).

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

Eine Beispielausgabe zeigt, dass der Agent installiert, inaktiv (gestoppt) und aktiviert (Dienst beim Booten starten) ist.

```
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 Informationen zum Agenten

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

Die Ausgabe sieht wie folgt aus:

Note

Die „Version“ kann je nach der zuletzt vom Agenten veröffentlichten Version unterschiedlich sein.

```
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
```

Den Agenten konfigurieren

Nach der Installation des Agenten müssen Sie die Agenten-Konfigurationsdatei unter aktualisieren/opt/aws/groundstation/etc/aws-gs-agent-config.json.

Agenten-Konfigurationsdatei

Beispiel

```
{
  "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
  ]
}
```

Aufschlüsselung der Felder

Funktionen

Funktionen werden als Amazon-Ressourcennamen für Dataflow-Endpunktgruppen angegeben.

Erforderlich: True

Format: Zeichenkettenarray

- Werte: Fähigkeit ARNs → Zeichenfolge

Beispiele:

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

Gerät

Dieses Feld enthält zusätzliche Felder, die zur Aufzählung des aktuellen EC2 „Geräts“ erforderlich sind.

Erforderlich: True

Format: Objekt

Mitglieder:

- privateIps
- publicIps
- agentCpuCores
- networkAdapters

privateIps

Dieses Feld wird derzeit nicht verwendet, ist aber für future Anwendungsfälle enthalten. Wenn kein Wert enthalten ist, wird standardmäßig [„127.0.0.1“] verwendet

Erforderlich: Falsch

Format: Zeichenkettenarray

- Werte: IP-Adressen → Zeichenfolge

Beispiel:

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

```
],
```

publicIps

Elastic IP (EIP) pro Dataflow-Endpunktgruppe.

Erforderlich: True

Format: Zeichenketten-Array

- Werte: IP-Adressen → Zeichenfolge

Beispiel:

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

agentCPUCores

Dies gibt an, welche virtuellen Kerne für den aws-gs-agent Prozess reserviert sind. Informationen zu [CPUKernplanung](#) den Anforderungen für die angemessene Einstellung dieses Werts finden Sie unter.

Erforderlich: True

Format: Int-Array

- Werte: Kernzahlen → int

Beispiel:

```
"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

Dies entspricht den Ethernet-Adaptern oder Schnittstellen, an ENIs die angeschlossen sind und die Daten empfangen.

Erforderlich: Falsch

Format: Zeichenketten-Array

- Werte: Namen von Ethernet-Adaptern (kann durch Ausführen gefunden werden `ifconfig`)

Beispiel:

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

EC2 Optimierung der Instanzleistung

Note

Wenn Sie Ihre AWS Ressourcen mithilfe von CloudFormation Vorlagen bereitgestellt haben, werden diese Optimierungen automatisch angewendet. Wenn Sie eine Instanz verwendet AMI oder Ihre EC2 Instanz manuell erstellt haben, müssen diese Leistungsoptimierungen angewendet werden, um eine möglichst zuverlässige Leistung zu erzielen.

Denken Sie daran, Ihre Instance neu zu starten, nachdem Sie alle Optimierungen vorgenommen haben.

Topics

- [Optimieren Sie Hardware-Interrupts und Empfangswarteschlangen — Auswirkungen auf CPU das Netzwerk](#)
- [Tune Rx Interrupt Coalescing — wirkt sich auf das Netzwerk aus](#)
- [Tune Rx Ring Buffer — wirkt sich auf das Netzwerk aus](#)
- [Tune C-State — Auswirkungen CPU CPU](#)
- [Eingangsports reservieren — wirkt sich auf das Netzwerk aus](#)
- [Neustart](#)

Optimieren Sie Hardware-Interrupts und Empfangswarteschlangen — Auswirkungen auf CPU das Netzwerk

In diesem Abschnitt wird die CPU grundlegende Verwendung von systemd SMPIRQs, Receive Packet Steering (RPS) und Receive Flow Steering (RFS) konfiguriert. RFS Eine Reihe von empfohlenen Einstellungen, die auf dem von Ihnen verwendeten Instanztyp basieren, finden Sie unter. [Anhang: Empfohlene Parameter für Interrupt/Tune RPS](#)

1. Platzieren Sie systemd-Prozesse von CPU Agentenkernen fern.
2. Leitet Hardware-Interrupt-Anfragen von den CPU Agentenkernen weg.
3. Konfigurieren Sie RPS die Konfiguration so, dass die Hardwarewarteschlange einer einzelnen Netzwerkschnittstellenkarte nicht zu einem Engpass im Netzwerkverkehr wird.

4. Konfigurieren Sie die KonfigurationRFS, um die CPU Cache-Trefferquote zu erhöhen und dadurch die Netzwerklatenz zu reduzieren.

Das von der bereitgestellte `set_irq_affinity.sh` Skript RPM konfiguriert alle oben genannten Funktionen für Sie. Zu crontab hinzufügen, damit es bei jedem Start angewendet wird:

```
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` Ersetzen Sie es durch Kerne, die für den Kernel und das Betriebssystem reserviert sind — in der Regel den ersten und den zweiten, zusammen mit Hyperthread-Kernpaaren. Dies sollte sich nicht mit den oben ausgewählten Kernen überschneiden. (Beispiel: '0,1,48,49' für eine 96-Instance mit Hyper-Threading). CPU
- `rps_core_mask` ist eine hexadezimale Bitmaske, die angibt, welche eingehenden Pakete verarbeitet werden CPUs sollen, wobei jede Ziffer für 4 steht. CPUs Sie muss außerdem alle 8 Zeichen, beginnend von rechts, durch Kommas getrennt werden. Es wird empfohlen, alles zuzulassen CPUs und das Balancing vom Caching übernehmen zu lassen.
 - Eine Liste der empfohlenen Parameter für jeden Instance-Typ finden Sie [Anhang: Empfohlene Parameter für Interrupt/Tune RPS](#) unter.
- Beispiel für eine CPU 96-Instanz:

```
echo "@reboot sudo /opt/aws/groundstation/bin/set_irq_affinity.sh '0,1,48,49'  
'ffffffff,ffffffff,ffffffff' >> /var/log/user-data.log 2>&1" >>/var/spool/cron/root
```

Tune Rx Interrupt Coalescing — wirkt sich auf das Netzwerk aus

Interrupt-Coalescing verhindert, dass das Host-System mit zu vielen Interrupts überflutet wird, und erhöht den Netzwerkdurchsatz. Bei dieser Konfiguration werden Pakete gesammelt und alle 128 Mikrosekunden ein einziger Interrupt generiert. Zu crontab hinzufügen, sodass es bei jedem Start angewendet wird:

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

- `interface` Ersetzen Sie es durch die Netzwerkschnittstelle (Ethernet-Adapter), die für den Empfang von Daten konfiguriert ist. In der Regel ist dies der Fall, `eth0` da dies die Standard-Netzwerkschnittstelle ist, die einer EC2 Instanz zugewiesen wurde.

Tune Rx Ring Buffer — wirkt sich auf das Netzwerk aus

Erhöhen Sie die Anzahl der Ringeinträge für den Rx-Ringpuffer, um Paketverluste oder -überläufe bei Burst-Verbindungen zu verhindern. Fügen Sie dem Crontab Folgendes hinzu, damit es bei jedem Start korrekt eingestellt ist:

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

- `interface` Ersetzen Sie es durch die Netzwerkschnittstelle (Ethernet-Adapter), die für den Empfang von Daten konfiguriert ist. In der Regel ist dies der Fall, `eth0` da dies die Standard-Netzwerkschnittstelle ist, die einer EC2 Instanz zugewiesen wurde.
- Wenn Sie eine `c6i.32xlarge`-Instance einrichten, muss der Befehl so geändert werden, dass der Ringpuffer auf `8192` statt auf `16384` gesetzt wird.

Tune C-State — Auswirkungen CPU CPU

Stellen Sie den CPU C-State ein, um Leerlauf zu verhindern, der zu Paketverlusten beim Start eines Kontakts führen kann. Erfordert einen Neustart der Instanz.

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

Eingangsports reservieren — wirkt sich auf das Netzwerk aus

Reservieren Sie alle Ports in Ihrem `AwsGroundStationAgentEndpoint` Eingangsadress-Portbereich, um Konflikte bei der Kernel-Nutzung zu vermeiden. Ein Konflikt bei der Portnutzung führt dazu, dass der Kontakt und die Datenübermittlung fehlschlagen.

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

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

Neustart

Nachdem alle Optimierungen erfolgreich angewendet wurden, starten Sie die Instanz neu, damit die Optimierungen wirksam werden.

```
sudo reboot
```

Anhang: Empfohlene Parameter für Interrupt/Tune RPS

In diesem Abschnitt werden die empfohlenen Parameterwerte für die Verwendung im Tuning-Abschnitt `Tune Hardware Interrupts and Receive Queues — Impacts and Network` festgelegt. CPU

Familie	Instance-Typ	<code>{interrupt_core_list}</code>	<code>{rps_core_mask}</code>
c6i	• c6i.32xlarge	• 0,1,64,65	• ffffffff, fffffff, fffffff, fffffff

Familie	Instance-Typ	\$ {interrupt_core_list}	\$ {rps_core_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, ffffffff, ffffffff • 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, ffffffff • ff, ffffffff, ffffffff
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, 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"> • 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

Familie	Instance-Typ	\$ {interrupt_core_list}	\$ {rps_core_mask}
g4dn	<ul style="list-style-type: none"> • g4dn.metal • g4dn.16xgroß • g4dn.12xgroß 	<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, ffffffff
p4d	<ul style="list-style-type: none"> • p4d.24xgroß 	<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.24xgroß 	<ul style="list-style-type: none"> • 0,1,48,49 	<ul style="list-style-type: none"> • ffffffff, ffffffff, ffffffff

Bereiten Sie sich darauf vor, einen DigiF-Kontakt aufzunehmen

1. Suchen Sie in CPU Core Planning nach den gewünschten Datenflüssen und stellen Sie eine Liste der Kerne bereit, die der Agent verwenden kann. Siehe [CPUKernplanung](#).
2. Überprüfen Sie die AWS Ground Station Agenten-Konfigurationsdatei. Siehe [AWS Ground Station Konfiguration des Agenten](#).
3. Vergewissern Sie sich, dass die erforderliche Leistungsoptimierung vorgenommen wurde. Siehe [EC2Optimierung der Instanzleistung](#).
4. Vergewissern Sie sich, dass Sie alle genannten bewährten Methoden befolgen. Siehe [Bewährte Methoden](#).
5. Vergewissern Sie sich, dass der AWS Ground Station Agent vor der geplanten Startzeit des Kontakts gestartet wurde, indem Sie:

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

6. Stellen Sie sicher, dass der AWS Ground Station Agent vor der geplanten Startzeit des Kontakts fehlerfrei ist, indem Sie:

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

Vergewissern Sie sich, dass `agentStatus` der von Ihnen `awsGroundStationAgentEndpoint` ist `ACTIVE` und der `auditResults` ist `HEALTHY`.

Bewährte Methoden

EC2Bewährte Methoden von Amazon

Halten Sie sich an EC2 die aktuellen Best Practices und stellen Sie eine ausreichende Verfügbarkeit von Datenspeichern sicher.

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

Linux-Scheduler

Der Linux-Scheduler kann Pakete auf UDP Sockets neu anordnen, wenn die entsprechenden Prozesse nicht an einen bestimmten Kern gebunden sind. Jeder Thread, der UDP Daten sendet oder empfängt, sollte sich für die Dauer der Datenübertragung an einen bestimmten Kern anheften.

AWS Ground Station verwaltete Präfixliste

Es wird empfohlen, bei der Festlegung der Netzwerkregeln für die Kommunikation über die Antenne die Präfixliste `com.amazonaws.global.groundstation AWS-managed` zu verwenden. Weitere Informationen zu [AWSverwalteten Präfixlisten finden Sie unter Arbeiten mit AWS verwalteten Präfixlisten](#).

Beschränkung auf einen einzigen Kontakt

Der AWS Ground Station Agent unterstützt mehrere Streams pro Kontakt, unterstützt jedoch jeweils nur einen Kontakt. Um Planungsprobleme zu vermeiden, sollten Sie eine Instanz nicht auf mehrere Datenfluss-Endpunktgruppen verteilen. Wenn eine einzelne Agentenkonfiguration mehreren verschiedenen zugeordnet ist DFEGARNs, kann sie nicht registriert werden.

Dienste und Prozesse werden zusammen mit dem AWS Ground Station Agenten ausgeführt

Wenn Dienste und Prozesse auf derselben EC2 Instanz wie der AWS Ground Station Agent gestartet werden, ist es wichtig, sie an diejenigen zu binden, die vCPUs nicht vom AWS Ground Station

Agenten und dem Linux-Kernel verwendet werden, da dies zu Engpässen und sogar Datenverlust bei Kontakten führen kann. Dieses Konzept der Bindung an ein bestimmtes Objekt vCPUs wird Affinität genannt.

Zu vermeidende Kerne:

- `agentCpuCores` von [Agenten-Konfigurationsdatei](#)
- `interrupt_core_list` von [Optimieren Sie Hardware-Interrupts und Empfangswarteschlangen — Auswirkungen auf CPU das Netzwerk](#).
- Standardwerte finden Sie unter [Anhang: Empfohlene Parameter für Interrupt/Tune RPS](#)

Als Beispiel mit einer **c5.24xlarge** Instanz

Wenn du angegeben hast

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

und rannte

```
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
```

vermeide dann die folgenden Kerne:

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

Affinialisierung von Diensten (systemd)

Neu eingeführte Dienste werden automatisch mit den zuvor genannten Diensten affinisiert.

`interrupt_core_list` Wenn der Anwendungsfall Ihrer gestarteten Dienste zusätzliche Kerne erfordert oder weniger ausgelastete Kerne benötigt, gehen Sie wie in diesem Abschnitt beschrieben vor.

Prüfen Sie mit dem folgenden Befehl, für welche Affinität Ihr Service derzeit konfiguriert ist:

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

Wenn Sie einen leeren Wert wie `sehenCPUAffinity=`, bedeutet das, dass wahrscheinlich die Standardkerne aus dem obigen Befehl verwendet werden `...bin/set_irq_affinity.sh <using the cores here> ...`

Um eine bestimmte Affinität zu überschreiben und festzulegen, suchen Sie den Speicherort der Servicedatei, indem Sie Folgendes ausführen:

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

Öffnen und ändern Sie die Datei (mit `vi`, usw.) und fügen Sie `CPUAffinity=<core list>` in den `[Service]` Abschnitt ein wie:

```
[Unit]
...

[Service]
...
CPUAffinity=2,3

[Install]
...
```

Speichern Sie die Datei und starten Sie den Dienst neu, um die Affinität anzuwenden mit:

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

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

Weitere Informationen finden Sie unter: [Red Hat Enterprise Linux 8 — Verwaltung, Überwachung und Aktualisierung des Kernels — Kapitel 27. CPUAffinity und NUMA Richtlinien mithilfe von Systemd konfigurieren.](#)

Affinialisierung von Prozessen (Skripten)

Es wird dringend empfohlen, neu gestartete Skripte und Prozesse manuell zu affinialisieren, da das Standardverhalten von Linux es ihnen ermöglicht, jeden Kern auf dem Computer zu verwenden.

Um Kernkonflikte bei laufenden Prozessen (wie Python, Bash-Skripten usw.) zu vermeiden, starten Sie den Prozess mit:

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

Wenn der Prozess bereits läuft, verwenden Sie Befehle wie `wiepidof`, `odertop`, `ps` um die Prozess-ID (PID) des jeweiligen Prozesses zu ermitteln. Mit dem können PID Sie die aktuelle Affinität sehen zu:

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

und kann es ändern mit:

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

Weitere Informationen zu Taskset finden Sie auf der [Manpage taskset — Linux](#)

Fehlerbehebung

Der Agent kann nicht gestartet werden

Der AWS Ground Station Agent kann aus verschiedenen Gründen nicht gestartet werden, aber das häufigste Szenario ist möglicherweise eine falsch konfigurierte Agenten-Konfigurationsdatei. Nach dem Start des Agenten (siehe [AWS Ground Station Der Agent wird gestartet](#)) erhalten Sie möglicherweise einen Status wie:

```
#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)
```

Fehlerbehebung

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

könnte zu einer Ausgabe führen von:

```
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.
```

Wenn der Agent nicht gestartet werden kann, nachdem „Config geladen“ wurde, liegt ein Problem mit der Agentenkonfiguration vor. Informationen [Agenten-Konfigurationsdatei](#) zur Überprüfung Ihrer Agentenkonfiguration finden Sie unter.

AWS Ground Station Agent-Protokolle

AWS Ground Station Der Agent schreibt Informationen über Kontaktausführungen, Fehler und den Integritätsstatus in die Protokolldateien der Instanz, auf der der Agent ausgeführt wird. Sie können die Protokolldateien anzeigen, indem Sie manuell eine Verbindung zu einer Instanz herstellen.

Sie können Agentenprotokolle an der folgenden Stelle einsehen.

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

Keine Kontakte verfügbar

Für die Planung von Kontakten ist ein funktionstüchtiger AWS Ground Station Agent erforderlich. Bitte vergewissern Sie sich, dass Ihr AWS Ground Station Agent gestartet wurde und fehlerfrei ist, indem Sie folgende Adresse abfragen AWS Ground Station API: get-dataflow-endpoint-group

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

Vergewissern Sie sich, dass agentStatus der von Ihnen awsGroundStationAgentEndpoint ist ACTIVE und der auditResults istHEALTHY.

Supportanfragen

Wenden Sie sich über den AWS Support an das Team der Ground Station.

1. Geben `contact_id` Sie alle betroffenen Kontakte an. Ohne diese Informationen kann das AWS Ground Station Team einen bestimmten Kontakt nicht untersuchen.
2. Geben Sie Einzelheiten zu allen bereits unternommenen Schritten zur Fehlerbehebung an.
3. Geben Sie in unserer Anleitung zur Fehlerbehebung alle Fehlermeldungen an, die bei der Ausführung der Befehle gefunden wurden.

Versionshinweise für Agenten

Aktuelle Agent-Version

Version 1.0.3555.0

Veröffentlichungsdatum: 27.03.2024

Datum des Endes des Support: 31.08.2024

RPMPrüfsummen:

- SHA256: 108f3aceb00e5af549839cd766c56149397e448a6e1e1429c89a9eebb6bc0fc1
- MD5: 65b72fa507fb0af32651adbb18d2e30f

Änderungen:

- Beim Start der Aufgabe wird die Agent-Metrik für die ausgewählte Version der ausführbaren Datei hinzugefügt.
- Fügen Sie Unterstützung für Konfigurationsdateien hinzu, um bestimmte ausführbare Versionen zu vermeiden, wenn andere Versionen verfügbar sind.
- Fügen Sie Netzwerk- und Routing-Diagnosen hinzu.
- Zusätzliche Sicherheitsfunktionen.
- Behebung eines Problems, bei dem einige Fehler bei der Metrikberichterstattung in die Standardausgabe oder das Journal statt in die Protokolldatei geschrieben wurden.
- Gehen Sie ordnungsgemäß mit Socket-Fehlern um, die über das Netzwerk nicht erreichbar sind.
- Messen Sie den Paketverlust und die Latenz zwischen Quell- und Zielagenten.
- Veröffentlichen Sie aws-gs-datapipe Version 2.0, um neue Protokollfunktionen zu unterstützen und Kontakte transparent auf das neue Protokoll umzustellen.

Veraltete Agent-Versionen

Version 1.0.2942.0

Veröffentlichungsdatum: 26.06.2023

Datum des Endes des Support: 31.05.2024

RPMPrüfsummen:

- SHA256: 7d94b642577504308a58bab28f938507f2591d4e1b2c7ea170b77bea97b5a9b6
- MD5: 661ff2b8f11aba5d657a6586b56e0d8f

Änderungen:

- Es wurden Fehlerprotokolle für den Fall hinzugefügt, dass der Agent auf der Festplatte aktualisiert RPM wird und der Agent neu gestartet werden muss, damit die Änderungen wirksam werden.
- Es wurde eine Überprüfung der Netzwerkoptimierung hinzugefügt, um sicherzustellen, dass die Optimierungsschritte im Benutzerhandbuch für den Agenten befolgt und korrekt angewendet werden.
- Behebung eines Fehlers, der zu falschen Warnungen in den Agent-Protokollen über die Archivierung von Protokollen führte.
- Die Erkennung von Paketverlusten wurde verbessert.
- Die Agent-Installation wurde aktualisiert, um die Installation oder das Upgrade des Agenten zu verhindern, RPM wenn der Agent bereits läuft.

Version 1.0.2716.0

Veröffentlichungsdatum: 15.03.2023

Datum des Endes des Support: 31.05.2024

RPMPrüfsummen:

- SHA256: cb05b6a77dfcd5c66d81c0072ac550affbcefefc372cc5562ee52fb220844929
- MD5: 65266490c4013b433ec39ee50008116c

Änderungen:

- Aktiviert das Hochladen von Protokollen, wenn der Agent beim Ausführen von Aufgaben ausfällt.
- Behebt einen Linux-Kompatibilitätsfehler in den bereitgestellten Netzwerk-Tuning-Skripten.

Version 1.0.2677.0

Veröffentlichungsdatum: 15.02.2023

Datum des Endes des Support: 31.05.2024

RPMPrüfsummen:

- SHA256: 77cfe94acb00af7ca637264b17c9b21bd7afdc85b99dffdd627aec9e99397489
- MD5: b8533be7644bb4d12ab84de21341adac

Änderungen:

- Erste allgemein verfügbare Agent-Version.

RPMÜberprüfung der Installation

Die neueste RPM Version, der MD5 Hash, der anhand von sha256sum validiert wurdeRPM, und der SHA256 Hash, der Sha256sum verwendet, sind unten aufgeführt. Diese Werte können zusammen verwendet werden, um die RPM Version zu validieren, die für den Bodenstationsagenten verwendet wird.

Aktuelle Agent-Version

Version 1.0.3555.0

Veröffentlichungsdatum: 27.03.2024

Datum des Endes des Support: 31.08.2024

RPMPrüfsummen:

- SHA256: 108f3aceb00e5af549839cd766c56149397e448a6e1e1429c89a9eebb6bc0fc1
- MD5: 65b72fa507fb0af32651adbb18d2e30f

Änderungen:

- Beim Start der Aufgabe wird die Agent-Metrik für die ausgewählte Version der ausführbaren Datei hinzugefügt.
- Fügen Sie Unterstützung für Konfigurationsdateien hinzu, um bestimmte ausführbare Versionen zu vermeiden, wenn andere Versionen verfügbar sind.
- Fügen Sie Netzwerk- und Routing-Diagnosen hinzu.
- Zusätzliche Sicherheitsfunktionen.
- Behebung eines Problems, bei dem einige Fehler bei der Metrikberichterstattung in die Standardausgabe oder das Journal statt in die Protokolldatei geschrieben wurden.
- Gehen Sie ordnungsgemäß mit Socket-Fehlern um, die über das Netzwerk nicht erreichbar sind.
- Messen Sie den Paketverlust und die Latenz zwischen Quell- und Zielagenten.
- Veröffentlichen Sie aws-gs-datapipe Version 2.0, um neue Protokollfunktionen zu unterstützen und Kontakte transparent auf das neue Protokoll umzustellen.

Überprüfen Sie die RPM

Folgende Tools benötigen Sie, um diese RPM Installation überprüfen zu können:

- [sha256sum](#)
- [U/min](#)

Beide Tools sind standardmäßig auf Amazon Linux 2 verfügbar. Mit diesen Tools können Sie überprüfen, ob es sich bei der von RPM Ihnen verwendeten Version um die richtige Version handelt. Laden Sie zuerst die neueste Version RPM aus dem S3-Bucket herunter (Anweisungen [Laden Sie den Agenten herunter](#) zum Herunterladen von finden Sie unter RPM). Sobald diese Datei heruntergeladen ist, müssen Sie einige Dinge überprüfen:

- Berechne die SHA256-Summe der RPM Datei. Führen Sie die folgende Aktion von der Befehlszeile der Recheninstanz aus, die Sie verwenden:

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

Nehmen Sie diesen Wert und vergleichen Sie ihn mit der obigen Tabelle. Dies zeigt, dass es sich bei der heruntergeladenen RPM Datei um eine gültige Datei zur Verwendung handelt, die AWS Ground Station an Kunden ausverkauft hat. Wenn die Hashes nicht übereinstimmen, installieren Sie die RPM nicht und löschen Sie sie aus der Recheninstanz.

- Überprüfen Sie auch den MD5 Hash der Datei, um sicherzustellen, dass die nicht kompromittiert RPM wurde. Verwenden Sie dazu das RPM Befehlszeilentool, indem Sie den folgenden Befehl ausführen:

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

Stellen Sie sicher, dass der hier aufgeführte MD5 Hash mit dem MD5 Hash der Version übereinstimmt, die in der obigen Tabelle aufgeführt ist. Sobald diese beiden Hashes anhand dieser in AWS Docs aufgeführten Tabelle validiert wurden, kann der Kunde sicher sein, dass es sich bei der RPM heruntergeladenen und installierten Version um die sichere und ungefährdete Version von handelt. RPM

Dokumentenverlauf für das AWS Ground Station Agent-Benutzerhandbuch

In der folgenden Tabelle werden die wichtigen Änderungen in den einzelnen Versionen des AWS Ground Station Agent-Benutzerhandbuchs beschrieben.

Änderung	Beschreibung	Datum
Aktualisierung der Dokumentation	In den Agentenanforderungen wurde ein Kommentar zum Beibehalten des Subnetzes und der EC2 Amazon-Instance in derselben Availability Zone hinzugefügt.	18. Juli 2024
Aktualisierung der Dokumentation	Teilen Sie den AWS Ground Station Agenten in ein eigenes Benutzerhandbuch auf. Frühere Änderungen finden Sie unter: Dokumentenverlauf für das Benutzerhandbuch zur AWS Ground Station .	18. Juli 2024

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.