



AWS Ground Station Guide de l'utilisateur de l'agent

AWS Ground Station



AWS Ground Station: AWS Ground Station Guide de l'utilisateur de l'agent

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Présentation de	1
Qu'est-ce que l' AWS Ground Station agent ?	1
Caractéristiques de l' AWS Ground Station agent	2
Exigences relatives aux agents	3
Schémas VPC	4
Système d'exploitation pris en charge	5
Recevez des données via l' AWS Ground Station agent	6
Plusieurs flux de données, récepteur unique	6
Plusieurs flux de données, plusieurs récepteurs	7
Sélectionnez l'instance Amazon EC2 et réservez des cœurs de processeur pour votre architecture	9
Types d'instances Amazon EC2 pris en charge	9
Planification du cœur du processeur	11
Collecte d'informations sur l'architecture	12
Exemple d'affectation du processeur	14
Annexe : <code>lscpu -p</code> sortie (complète) pour <code>c5.24xlarge</code>	15
Installer l'agent	18
Utiliser un CloudFormation modèle	18
Étape 1 : Création de AWS ressources	18
Étape 2 : vérifier le statut de l'agent	18
Installation manuelle sur EC2	18
Étape 1 : créer des ressources AWS	18
Étape 2 : créer une instance EC2	19
Étape 3 : télécharger et installer l'agent	19
Étape 4 : Configuration de l'agent	21
Étape 5 : Appliquer le réglage des performances	21
Étape 6 : Gérer l'agent	21
Gérer l'agent	22
AWS Ground Station Configuration de l'agent	22
AWS Ground Station Démarrage de l'agent	22
AWS Ground Station Agent, arrêtez.	23
AWS Ground Station Mise à niveau des agents	23
AWS Ground Station Rétrogradation de l'agent	24
AWS Ground Station Désinstallation de l'agent	25

AWS Ground Station Statut de l'agent	25
AWS Ground Station Informations sur l'agent RPM	26
Configuration de l'agent	27
Fichier de configuration de l'agent	27
Exemple	27
Répartition des champs	27
Optimisez les performances de votre instance EC2	31
Réglez les interruptions matérielles et les files d'attente de réception, ce qui a un impact sur le processeur et le réseau	31
Tune Rx interrompt la coalescence, ce qui a un impact sur le réseau	32
Tune Rx Ring Buffer, impacte le réseau	33
Régler le C-State du processeur : impacte le processeur	33
Ports d'entrée de réserve - impact sur le réseau	34
Redémarrer	34
Annexe : Paramètres recommandés pour le interrupt/RPS réglage	34
Préparez-vous à prendre un contact DigiF	37
Bonnes pratiques	38
Bonnes pratiques relatives à Amazon EC2	38
Planificateur Linux	38
AWS Ground Station liste de préfixes gérée	38
Limitation du contact unique	38
Exécution de services et de processus en parallèle avec l' AWS Ground Station agent	38
À titre d'exemple, en utilisant une c5.24xlarge instance	39
Services d'affinisation (systemd)	39
Processus d'affinisation (scripts)	40
Résolution des problèmes	42
L'agent ne démarre pas	42
Résolution des problèmes	42
AWS Ground Station Journaux des agents	43
Aucun contact disponible	43
Obtention de support	44
Notes de mise à jour des agents	45
Dernière version de l'agent	45
Version 1.0.4382.0	45
Versions d'agent obsolètes	45
La version 1.0.3555.0	45

Version 1.0.2942.0	46
Version 1.0.2716.0	46
La version 1.0.2677.0	47
Validation d'installation RPM	48
Dernière version de l'agent	45
Version 1.0.4382.0	45
Vérifiez le RPM	48
Historique du document	50
.....	li

Présentation de

Qu'est-ce que l' AWS Ground Station agent ?

Avec l' AWS Ground Station agent, disponible sous forme de RPM, vous pouvez recevoir (en liaison descendante) des flux de données synchrones à fréquence intermédiaire numérique (DigIF), y compris à bande étroite et à bande large, lors des contacts avec AWS Ground Station. Vous pouvez sélectionner deux options pour la livraison des données :

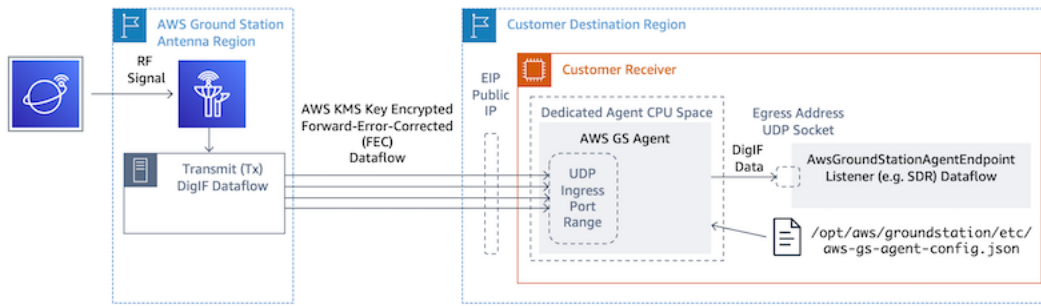
1. Livraison de données vers une instance EC2 : livraison de données vers une instance EC2 dont vous êtes le propriétaire. Vous gérez l' AWS Ground Station agent. Cette option peut vous convenir le mieux si vous avez besoin d'un traitement des données en temps quasi réel. Consultez le guide [Data Delivery to Amazon Elastic Compute Cloud](#) pour plus d'informations sur la livraison de données EC2.
2. Livraison de données vers un compartiment S3 : livraison de données vers un compartiment AWS S3 que vous possédez via un service géré par Ground Station. Consultez le AWS Ground Station guide de [démarrage](#) pour plus d'informations sur la livraison de données S3.

Les deux modes de livraison de données nécessitent que vous créiez un ensemble de ressources AWS. L'utilisation de CloudFormation pour créer vos ressources AWS est vivement recommandée afin de garantir la fiabilité, la précision et la facilité de prise en charge. Chaque contact peut uniquement transmettre des données à EC2 ou S3, mais pas aux deux simultanément.

Note

La livraison de données S3 étant un service géré par Ground Station, ce guide se concentre sur la livraison de données à votre/vos instance (s) EC2.

Le schéma suivant montre un flux de données DigIF d'une région d' AWS Ground Station antenne vers votre instance EC2 avec votre radio définie par logiciel (SDR) ou un écouteur similaire.



Caractéristiques de l' AWS Ground Station agent

L' AWS Ground Station agent reçoit les données de liaison descendante à fréquence intermédiaire numérique (DigiF) et sort les données déchiffrées qui permettent ce qui suit :

- Capacité de liaison descendante DigiF jusqu'à 400 % MHz de bande passante, y compris les flux de données à bande étroite (≤ 40 MHz) et à bande large (> 40). MHz
- Livraison de données DigiF à haut débit et à faible instabilité vers n'importe quelle adresse IP publique (AWS Elastic IP) du réseau AWS.
- Livraison de données fiable grâce à la correction d'erreur directe (FEC).
- Livraison sécurisée des données à l'aide d'une AWS KMS clé de chiffrement gérée par le client.

Exigences relatives aux agents

Note

Ce guide de AWS Ground Station l'agent part du principe que vous avez intégré Ground Station à l'aide du guide de [AWS Ground Station démarrage](#).

L'instance EC2 de l' AWS Ground Station agent récepteur nécessite un ensemble de ressources AWS dépendantes pour fournir des données DigIF de manière fiable et sécurisée à vos points de terminaison.

1. Un VPC dans lequel lancer le récepteur EC2.
2. Une clé AWS KMS pour le chiffrement/déchiffrement des données.
3. Une clé SSH ou un profil d'instance EC2 configuré pour le gestionnaire de session [SSM](#).
4. Règles du réseau ou du groupe de sécurité autorisant ce qui suit :
 1. Trafic UDP provenant des ports spécifiés dans AWS Ground Station le groupe de points de terminaison de votre flux de données. L'agent réserve une gamme de ports contigus utilisés pour fournir des données aux points de terminaison du flux de données d'entrée.
 2. Accès SSH à votre instance (Remarque : vous pouvez également utiliser AWS Session Manager pour accéder à votre instance EC2).
 3. Accès en lecture à un compartiment S3 accessible au public pour la gestion des agents.
 4. Trafic SSL sur le port 443 permettant à l'agent de communiquer avec le AWS Ground Station service.
 5. Trafic provenant de la liste `com.amazonaws.global.groundstation` des préfixes AWS Ground Station gérés.

En outre, une configuration VPC incluant un sous-réseau public est requise. Reportez-vous au [guide de l'utilisateur VPC](#) pour obtenir des informations générales sur la configuration des sous-réseaux.

Configurations compatibles :

1. Une adresse IP élastique associée à votre instance EC2 dans un sous-réseau public.

2. Une adresse IP élastique associée à une ENI dans un sous-réseau public, attachée à votre instance EC2 (dans n'importe quel sous-réseau situé dans la même zone de disponibilité que le sous-réseau public).

Vous pouvez utiliser le même groupe de sécurité que votre instance EC2 ou en spécifier un avec au moins l'ensemble minimal de règles, à savoir :

- Trafic UDP provenant des ports spécifiés dans AWS Ground Station le groupe de points de terminaison de votre flux de données.

Par exemple, des modèles de livraison de données CloudFormation EC2 avec ces ressources préconfigurées, voir [Public broadcast satellite using AWS Ground Station Agent \(wide band\)](#).

Schémas VPC

Schéma : adresse IP élastique associée à votre instance EC2 dans un sous-réseau public

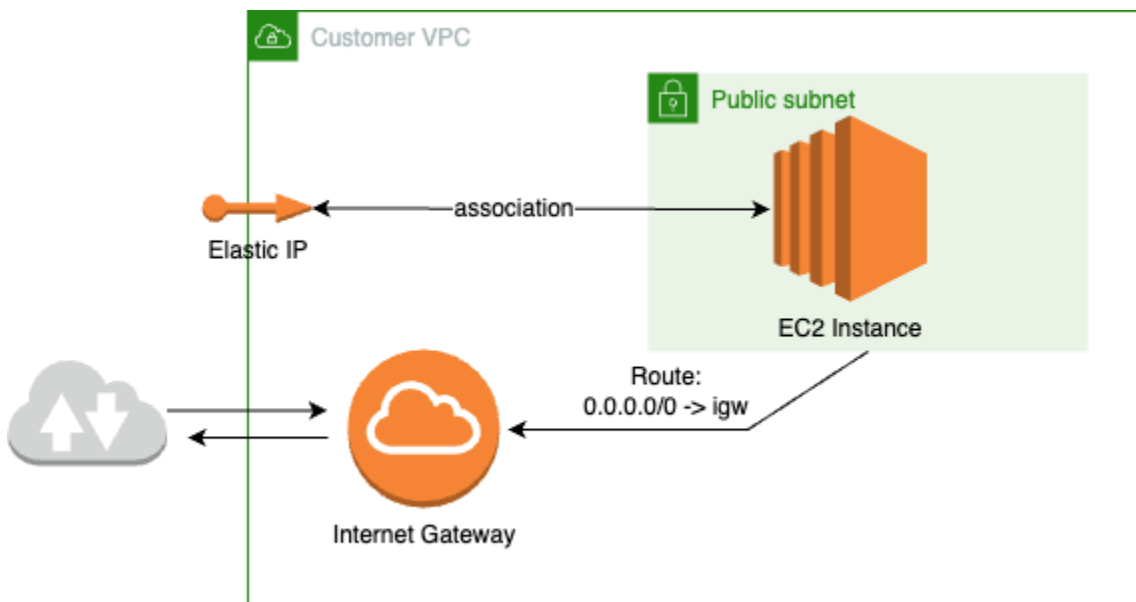
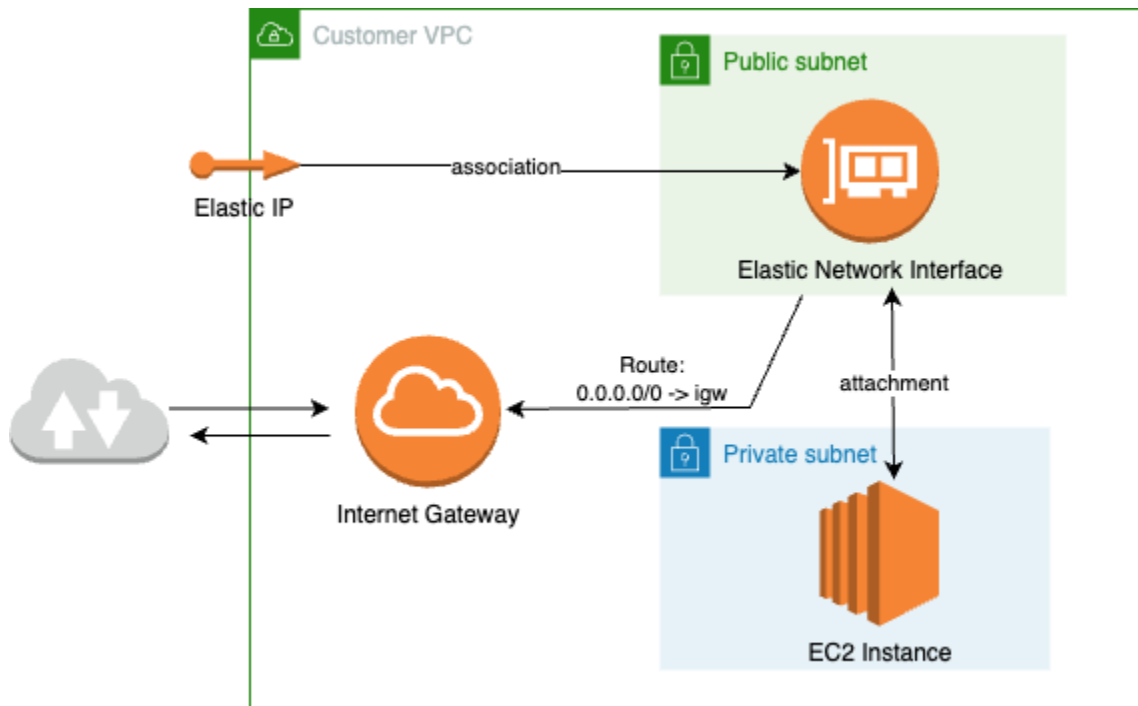


Schéma : adresse IP élastique associée à une ENI dans un sous-réseau public, attachée à votre instance EC2 dans un sous-réseau privé



Système d'exploitation pris en charge

Amazon Linux 2 avec noyau 5.10+.

Les types d'instances pris en charge sont répertoriés dans [Sélectionnez l'instance Amazon EC2 et réservez des cœurs de processeur pour votre architecture](#)

Recevez des données via l' AWS Ground Station agent

Les diagrammes ci-dessous fournissent un aperçu de la manière dont les données circulent AWS Ground Station lors des contacts à fréquence intermédiaire numérique à large bande (DigIF).

L' AWS Ground Station agent se chargera d'orchestrer les composants du plan de données pour un contact. Avant de planifier un contact, l'agent doit être correctement configuré, démarré et enregistré (l'enregistrement est automatique au démarrage de l'agent) auprès de AWS Ground Station. En outre, le logiciel de réception de données (tel qu'une radio définie par logiciel) doit être en cours d'exécution et configuré pour recevoir des données à l'adresse de [AwsGroundStationAgentEndpoints](#) sortie.

Dans les coulisses, l' AWS Ground Station agent reçoit des tâches AWS Ground Station et annule le AWS KMS chiffrement appliqué pendant le transit, avant de le transmettre à l'adresse de sortie du point de terminaison de destination que votre radio définie par logiciel (SDR) écoute. L' AWS Ground Station agent et ses composants sous-jacents respecteront les limites du processeur définies dans le fichier de configuration afin de garantir que cela n'affecte pas les performances des autres applications exécutées sur l'instance.

L' AWS Ground Station agent doit être exécuté sur l'instance de réception impliquée dans le contact. Un seul AWS Ground Station agent est capable d'orchestrer plusieurs flux de données, comme indiqué ci-dessous, si vous préférez recevoir tous les flux de données sur une seule instance de récepteur.

Plusieurs flux de données, récepteur unique

Exemple de scénario :

Vous souhaitez recevoir deux liaisons descendantes d'antennes sous forme de flux de données DigIF sur la même instance de récepteur EC2. Les deux liaisons descendantes seront 200 MHz et 100MHz.

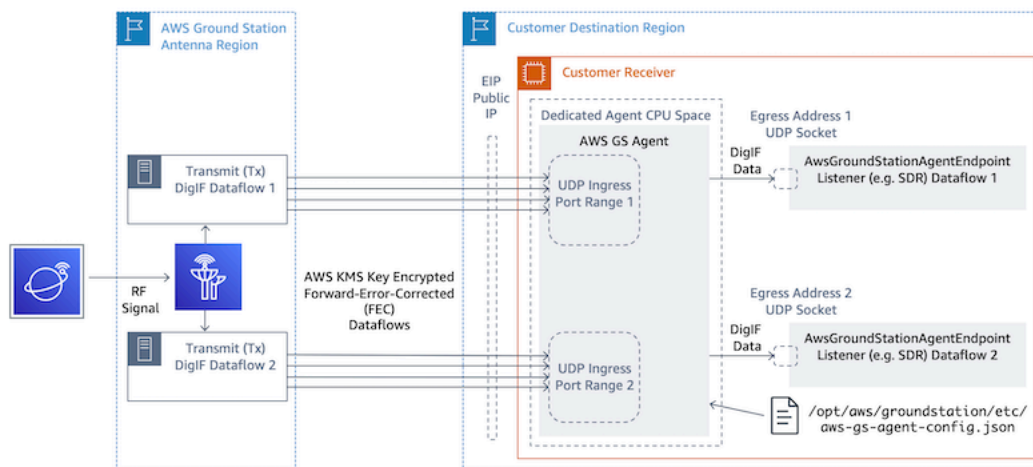
`AwsGroundStationAgentEndpoints`:

Il y aura deux `AwsGroundStationAgentEndpoint` ressources, une pour chaque flux de données. Les deux points de terminaison auront la même adresse IP publique (`ingressAddress.socketAddress.name`). Les entrées `portRange` ne doivent pas

se chevaucher, car les flux de données sont reçus sur la même instance EC2. Les deux `egressAddress.socketAddress.port` doivent être uniques.

Planification du processeur :

- 1 cœur (2 vCPU) pour exécuter l' AWS Ground Station agent unique sur l'instance.
- 6 cœurs (12 vCPU) pour recevoir DigiF Dataflow 1 (200 recherches dans le tableau). MHz [Planification du cœur du processeur](#)
- 4 cœurs (8 vCPU) pour recevoir DigiF Dataflow 2 (100 recherches dans le tableau). MHz [Planification du cœur du processeur](#)
- Espace processeur total dédié à l'agent = 11 cœurs (22 vCPU) sur le même socket.



Plusieurs flux de données, plusieurs récepteurs

Exemple de scénario :

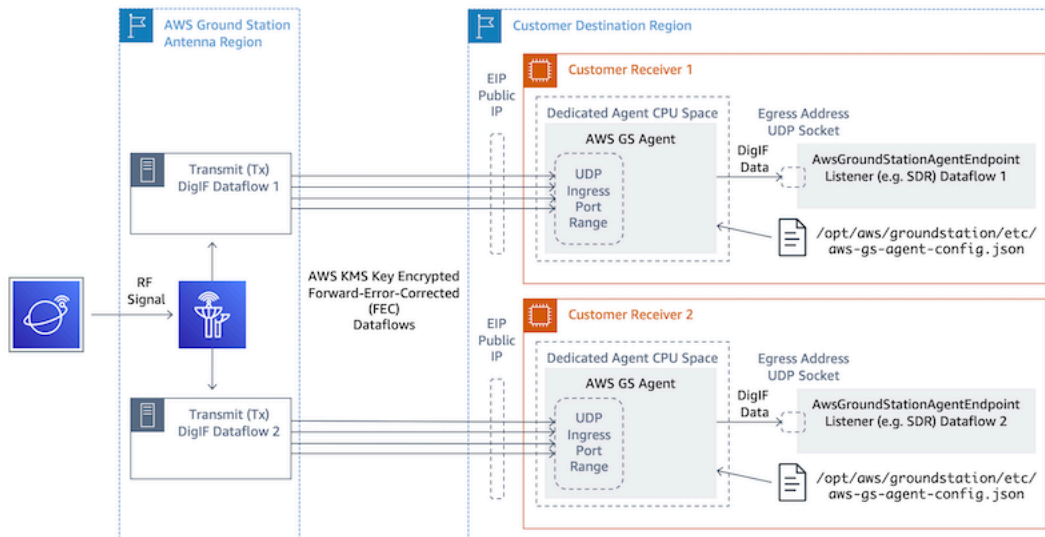
Vous souhaitez recevoir deux liaisons descendantes d'antennes sous forme de flux de données DigiF sur différentes instances du récepteur EC2. Les deux liaisons descendantes seront de 400MHz.

`AwsGroundStationAgentEndpoints`:

Il y aura deux `AwsGroundStationAgentEndpoint` ressources, une pour chaque flux de données. Les points de terminaison auront une adresse IP publique différente (`ingressAddress.socketAddress.name`). Il n'y a aucune restriction quant aux valeurs de port pour l'un `ingressAddress` ou l'autre, `egressAddress` car les flux de données sont reçus sur une infrastructure distincte et n'entreront pas en conflit les uns avec les autres.

Planification du processeur :

- Instance du récepteur 1
 - 1 cœur (2 vCPU) pour exécuter l' AWS Ground Station agent unique sur l'instance.
 - 9 cœurs (18 vCPU) pour recevoir DigiF Dataflow 1 (400 recherches dans le tableau). MHz
[Planification du cœur du processeur](#)
 - Espace processeur total dédié à l'agent = 10 cœurs (20 vCPU) sur le même socket.
- Instance du récepteur 2
 - 1 cœur (2 vCPU) pour exécuter l' AWS Ground Station agent unique sur l'instance.
 - 9 cœurs (18 vCPU) pour recevoir DigiF Dataflow 2 (400 recherches dans le tableau). MHz
[Planification du cœur du processeur](#)
 - Espace processeur total dédié à l'agent = 10 cœurs (20 vCPU) sur le même socket.



Sélectionnez l'instance Amazon EC2 et réservez des cœurs de processeur pour votre architecture

Types d'instances Amazon EC2 pris en charge

L' AWS Ground Station agent a besoin de cœurs de processeur dédiés pour fonctionner en raison des flux de travail de livraison de données gourmands en calcul. Nous prenons en charge les types d'instances suivants. Consultez [Planification du cœur du processeur](#) pour décider quel type d'instance convient le mieux à votre cas d'utilisation.

Famille d'instances	Type d'instance	Par défaut v CPUs	Cœurs d'UC par défaut	Bande passante agrégée maximale DigiF () MHz
c5	c5.12xlarge	48	24	180
	c5.18xlarge	72	36	380
	c5.24xlarge	96	48	380
c5n	c5n.18xlarge	72	36	400
	c5n.metal	72	36	400
c6i	c6i.24xlarge	96	48	400
	c6i.32xlarge	128	64	400
c7i	c7i.12xlarge	48	24	280
	c7i.24xlarge	96	48	400
p3dn	p3dn.24xlarge	96	48	400
g4dn	g4dn.12xlarge	48	24	400
	g4dn.16xlarge	64	32	400

Famille d'instances	Type d'instance	Par défaut v CPUs	Cœurs d'UC par défaut	Bande passante agrégée maximale DigiF () MHz
	g4dn.metal	96	48	400
p4d	p4d.24xlarge	96	48	400
m5	m5.8xlarge	32	16	100
	m5.12xlarge	48	24	180
	m5.24xlarge	96	48	380
m6i	m6i.32xlarge	128	64	400
r5	r5.24xlarge	96	48	380
	r5.metal	96	48	380
r5n	r5n.24xlarge	96	48	400
	r5n.metal	96	48	400
r6i	r6i.32xlarge	128	64	400

Note

La colonne de bande passante agrégée maximale DigiF indique la bande passante agrégée maximale prise en charge pour tous les flux de données DigiF combinés sur chaque type d'instance. Ces limites sont dues à la capacité du réseau EC2 allouée aux types d'instances donnés. Ces valeurs représentent des estimations prudentes et doivent être utilisées lors de la planification de vos configurations DigiF. La bande passante réelle peut varier en fonction de la charge du système et d'autres facteurs.

Planification du cœur du processeur

L' AWS Ground Station agent nécessite des cœurs de processeur dédiés qui partagent le cache L3 pour chaque flux de données. L'agent est conçu pour exploiter les paires de processeurs Hyper-threaded (HT) et nécessite que des paires HT soient réservées à son utilisation. Une paire hyper-thread est une paire de processeurs virtuels (CPUs vCPU) contenus dans un seul cœur. Les tableaux suivants fournissent un mappage du débit de données avec le nombre requis de cœurs réservés à l'agent pour un seul flux de données. Ces tables supposent Cascade Lake ou une version plus récente CPUs et sont valides pour tous les types d'instances pris en charge. Si votre bande passante se situe entre les entrées d'un tableau, sélectionnez la bande passante la plus élevée suivante.

AWS Ground Station utilise deux stratégies de diffusion différentes en fonction de la bande passante du flux de données :

- Bande étroite : bandes passantes inférieures ou égales à 40. MHz
- Bande large : bandes passantes supérieures à 40. MHz

Ces stratégies ont des exigences fondamentales différentes. Notamment, les flux de données à bande étroite peuvent nécessiter plus de cœurs que les flux de données à large bande à des bandes passantes comparables, voire supérieures. La stratégie est sélectionnée automatiquement en fonction de la AntennaDownlink bande passante configurée.

Important

Chaque flux de données nécessite son propre ensemble de cœurs dédiés. Si vous avez plusieurs flux de données, recherchez les exigences de base pour chaque flux de données indépendamment à l'aide du tableau approprié (bande étroite ou bande large) et additionnez-les. Ne partagez pas les cœurs entre les flux de données. L'agent a besoin d'un cœur réservé supplémentaire pour la gestion et la coordination. Le nombre total de cœurs requis sera donc la somme des cœurs nécessaires pour chaque flux de données plus un seul cœur supplémentaire (2 v CPUs).

Exigences relatives au noyau à bande étroite (≤ 40) MHz

AntennaDownlink Bande passante (MHz)	Débit de données DigiF VITA-49.2 attendu (Mo/ s)	Nombre de cœurs (paires de processeurs HT)	Nombre total de vCPU
≤25	≤500	3	6
30	600	4	8
35	700	4	8
40	800	4	8

Exigences de base en bande large (>40) MHz

AntennaDownlink Bande passante (MHz)	Débit de données DigiF VITA-49.2 attendu (Mo/ s)	Nombre de cœurs (paires de processeurs HT)	Nombre total de vCPU
50	1 000	3	6
100	2000	4	8
150	3000	5	10
200	4000	6	12
250	5000	6	12
300	6 000	7	14
350	7000	8	16
400	8000	9	18

Collecte d'informations sur l'architecture

`lscpu` fournit des informations sur l'architecture de votre système. La sortie de base indique quels vCPUs (étiquetés « CPU ») appartiennent à quels nœuds NUMA (et chaque nœud NUMA partage un

cache L3). Nous examinons ci-dessous une `c5.24xlarge` instance afin de recueillir les informations nécessaires à la configuration de l' AWS Ground Station agent. Cela inclut des informations utiles telles que le nombre de vCPUs, les cœurs et vCPU-to-node l'association.

```
> 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  <-----
```

Les cœurs dédiés à l' AWS Ground Station agent doivent inclure les deux v CPUs pour chaque cœur attribué. Tous les cœurs d'un flux de données doivent exister sur le même nœud NUMA. L' `-p` option de `lscpu` commande nous fournit les associations entre le cœur et le processeur nécessaires pour configurer l'agent. Les champs pertinents sont CPU (c'est ce que nous appelons le vCPU), Core et L3 (qui indique quel cache L3 est partagé par ce cœur). Notez que sur la plupart des processeurs Intel, le nœud NUMA est égal au cache L3.

Examinez le sous-ensemble suivant de la `lscpu -p` sortie pour a `c5.24xlarge` (abrégé et formaté pour plus de clarté).

```

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

```

D'après la sortie, nous pouvons voir que Core 0 inclut v CPUs 0 et 16, Core 1 inclut v CPUs 1 et 17, Core 2 inclut v CPUs 2 et 18. En d'autres termes, les paires hyperthreadées sont : 0 et 16, 1 et 17, 2 et 18.

Exemple d'affectation du processeur

À titre d'exemple, nous utiliserons une `c5.24xlarge` instance pour une liaison descendante à large bande à double polarité à 350. MHz D'après le tableau ci-dessous, [Planification du cœur du processeur](#) nous savons qu'une MHz liaison descendante 350 nécessite 8 cœurs (16 vCPUs) pour le flux de données unique. Cela signifie que cette configuration à double polarité utilisant deux flux de données nécessite un total de 16 cœurs (32 vCPUs) plus un cœur (2 vCPUs) pour l'agent.

Nous connaissons le `lscpu` résultat pour les `c5.24xlarge` inclusions NUMA `node0 CPU(s) : 0-23, 48-71` et `node1 CPU(s) : 24-47, 72-95`. Comme NUMA `node0` contient plus que ce dont nous avons besoin, nous n'assignerons qu'à partir des cœurs : 0-23 et 48-71.

Tout d'abord, nous allons sélectionner 8 cœurs pour chaque flux de données partageant un cache L3 ou un nœud NUMA. Ensuite, nous rechercherons le v correspondant CPUs (étiqueté « CPU ») dans la `lscpu -p` sortie [Annexe : lscpu -p sortie \(complète\) pour c5.24xlarge](#). Voici un exemple de processus de sélection de base :

- Réservez des cœurs de 0 à 1 pour le système d'exploitation.
- Flux 1 : sélectionnez les noyaux 2 à 9 qui correspondent aux v CPUs 2-9 et 50-57.
- Flux 2 : sélectionnez les noyaux 10-17 qui correspondent aux v CPUs 10-17 et 58-65.
- Agent core : sélectionnez le noyau 18 qui correspond aux versions CPUs 18 et 66.

Cela donne les v CPUs 2-18 et 50-66, donc la liste à fournir l'agent est. [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] Vous devez vous assurer que vos propres processus ne s'exécutent pas sur ces appareils CPUs , comme décrit dans [Exécution de services et de processus en parallèle avec l' AWS Ground Station agent](#).

Notez que les cœurs spécifiques sélectionnés dans cet exemple sont quelque peu arbitraires. D'autres ensembles de cœurs fonctionneraient tant qu'ils répondent à l'exigence selon laquelle tous partagent un cache L3 pour chaque flux de données.

Annexe : `lscpu -p` sortie (complète) pour 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
```

Installer l'agent

L' AWS Ground Station agent peut être installé de différentes manières :

1. CloudFormation modèle (recommandé).
2. Installation manuelle sur Amazon EC2.

Utiliser un CloudFormation modèle

Le CloudFormation modèle de livraison de données EC2 crée les ressources AWS requises pour fournir des données à votre instance EC2. Ce CloudFormation modèle utilise l'AMI AWS Ground Station gérée sur laquelle l' AWS Ground Station agent est préinstallé. Le script de démarrage de l'instance EC2 créée remplit ensuite le fichier de configuration de l'agent et applique le réglage des performances () [Optimisez les performances de votre instance EC2](#) nécessaire.

Étape 1 : Création de AWS ressources

Créez votre pile de ressources AWS à l'aide d'un modèle de [satellite de diffusion publique utilisant AWS Ground Station Agent \(large bande\)](#).

Étape 2 : vérifier le statut de l'agent

Par défaut, l'agent est configuré et actif (démarré). Pour vérifier l'état de l'agent, vous pouvez vous connecter à l'instance EC2 (SSH ou SSM Session Manager) et voir. [AWS Ground Station Statut de l'agent](#)

Installation manuelle sur EC2

Ground Station recommande d'utiliser des CloudFormation modèles pour mettre en service vos ressources AWS, mais dans certains cas d'utilisation, le modèle standard peut ne pas suffire. Dans de tels cas, nous vous recommandons de personnaliser le modèle en fonction de vos besoins. Si cela ne répond toujours pas à vos exigences, vous pouvez créer manuellement vos ressources AWS et installer l'agent.

Étape 1 : créer des ressources AWS

Consultez les [exemples de configurations de profil de mission](#) pour obtenir des instructions permettant de configurer manuellement les ressources AWS requises pour un contact.

La `AwsGroundStationAgentEndpointressource` définit un point de terminaison pour recevoir un flux de données DigiF via un AWS Ground Station agent et est essentielle pour établir un contact réussi. Bien que la documentation de l'API se trouve dans la [référence des API](#), cette section abordera brièvement les concepts relatifs à l' AWS Ground Station agent.

Le point de terminaison `ingressAddress` est l'endroit où l' AWS Ground Station agent recevra le trafic UDP AWS KMS crypté en provenance de l'antenne. `socketAddressname` s'agit de l'adresse IP publique de l'instance EC2 (à partir de l'EIP attachée). `portRange` doit y avoir au moins 300 ports contigus dans une plage réservée à toute autre utilisation. Pour obtenir des instructions, consultez [Ports d'entrée de réserve - impact sur le réseau](#). Ces ports doivent être configurés pour autoriser le trafic entrant UDP sur le groupe de sécurité du VPC sur lequel l'instance de récepteur est exécutée.

Le point de terminaison `egressAddress` est l'endroit où l'agent vous transmettra le flux de données DigiF. Vous devriez avoir une application (par exemple SDR) recevant les données via un socket UDP à cet emplacement.

Étape 2 : créer une instance EC2

Les éléments suivants AMIs sont pris en charge :

1. AWS Ground Station AMI, `groundstation-a12-gs-agent-ami-*` où `*` est la date de création de l'AMI, est fournie avec l'agent installé (recommandé).
2. `amzn2-ami-kernel-5.10-hvm-x86_64-gp2`.

Étape 3 : télécharger et installer l'agent


Note

Les étapes de cette section doivent être effectuées si vous n'avez pas choisi l'AMI de l' AWS Ground Station agent à l'étape précédente.

Agent de téléchargement

L' AWS Ground Station agent est disponible à partir de compartiments S3 spécifiques à la région et peut être téléchargé sur les instances EC2 prises en charge à l'aide de la ligne de commande

(CLI) AWS. Nous vous recommandons de toujours effectuer le téléchargement à partir du `latest` chemin afin de vous assurer que vous utilisez la version la plus récente de l'agent. Le chemin de téléchargement est `s3://groundstation-wb-digif-software- $\{AWS::Region\}$ /aws-groundstation-agent/latest/amazon_linux_2_x86_64/aws-groundstation-agent.rpm` où $\{AWS::Region\}$ fait référence à l'une des [régions de distribution de données et de console AWS Ground Station](#) prises en charge.

 Note

Utilisez toujours le `latest` chemin plutôt que de coder en dur un numéro de version spécifique dans vos scripts de téléchargement. L'utilisation du `latest` chemin garantit que vos instances reçoivent automatiquement les correctifs de sécurité et les corrections de bogues lors de leur mise en service.

Exemple : téléchargez la dernière version rpm de la région AWS `us-east-2` localement dans le dossier `/tmp`.

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

Si vous devez télécharger une version spécifique de l'AWS Ground Station agent à des fins de résolution des problèmes ou de restauration, vous pouvez la télécharger depuis le dossier correspondant à cette version dans le compartiment S3. Cela n'est pas recommandé pour les opérations normales.

Exemple : téléchargez la version `1.0.2716.0` du fichier rpm depuis la région AWS `us-east-2` localement dans le dossier `/tmp`.

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

Note

Si vous souhaitez vérifier que le RPM que vous avez téléchargé a été vendu par AWS Ground Station, suivez les instructions de [Validation d'installation RPM](#).

Installer l'agent

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

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

Étape 4 : Configuration de l'agent

Après avoir installé l'agent, vous devez mettre à jour le fichier de configuration de l'agent. Consultez [Configuration de l'agent](#).

Étape 5 : Appliquer le réglage des performances

AWS Ground Station AMI de l'agent : si vous avez choisi l'AMI de l' AWS Ground Station agent à l'étape précédente, appliquez les réglages de performance suivants.

- [Réglez les interruptions matérielles et les files d'attente de réception, ce qui a un impact sur le processeur et le réseau](#)
- [Ports d'entrée de réserve - impact sur le réseau](#)
- [Redémarrer](#)

Autre AMIs : si vous avez choisi une autre AMI à l'étape précédente, appliquez tous les réglages répertoriés ci-dessous [Optimisez les performances de votre instance EC2](#) et redémarrez l'instance.

Étape 6 : Gérer l'agent

Pour démarrer, arrêter et vérifier l'état de l'agent, voir [Gérer l'agent](#).

Gérer l'agent

L' AWS Ground Station agent fournit les fonctionnalités suivantes pour configurer, démarrer, arrêter, mettre à niveau, rétrograder et désinstaller l'agent à l'aide des outils de commande Linux intégrés.

Rubriques

- [AWS Ground Station Configuration de l'agent](#)
- [AWS Ground Station Démarrage de l'agent](#)
- [AWS Ground Station Agent, arrêtez.](#)
- [AWS Ground Station Mise à niveau des agents](#)
- [AWS Ground Station Rétrogradation de l'agent](#)
- [AWS Ground Station Désinstallation de l'agent](#)
- [AWS Ground Station Statut de l'agent](#)
- [AWS Ground Station Informations sur l'agent RPM](#)

AWS Ground Station Configuration de l'agent

Accédez à `/opt/aws/groundstation/etc`, qui doit contenir un seul fichier nommé `aws-gs-agent-config.json`. Consultez [Fichier de configuration de l'agent](#)

AWS Ground Station Démarrage de l'agent

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

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

Devrait produire une sortie indiquant que l'agent est actif.

```
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 Agent, arrêtez.

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

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

Devrait produire une sortie indiquant que l'agent est inactif (arrêté).

```
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 Mise à niveau des agents

1. Téléchargez la dernière version de l'agent. Consultez [Agent de téléchargement](#).
2. Arrêtez l'agent .

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

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

3. Mettez à jour l'agent.

```
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 Rétrogradation de l'agent

1. Téléchargez la version de l'agent dont vous avez besoin. Consultez [Agent de téléchargement](#).
2. Rétrogradez l'agent.

```
# 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 Désinstallation de l'agent

La désinstallation de l'agent renommera `/opt/aws/groundstation/etc/aws-gs-agent-config.json` to `/opt/aws/groundstation/etc/aws-gs-agent-config.json.rpmsave`. Si vous réinstallez l'agent sur la même instance, les valeurs par défaut du `aws-gs-agent-config` fichier `.json` seront écrites et devront être mises à jour avec les valeurs correctes correspondant à vos ressources AWS. Consultez [Fichier de configuration de l'agent](#).

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

AWS Ground Station Statut de l'agent

L'état de l'agent est actif (l'agent est en cours d'exécution) ou inactif (l'agent est arrêté).

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

Un exemple de sortie montre que l'agent est installé, inactif (arrêté) et activé (démarrage du service au démarrage).

```
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 Informations sur l'agent RPM

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

La sortie est la suivante :

Note

La « version » peut être différente en fonction de la dernière version publiée par l'agent.

```
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
```

Configuration de l'agent

Après avoir installé l'agent, vous devez mettre à jour le fichier de configuration de l'agent à l'adresse/ `opt/aws/groundstation/etc/aws-gs-agent-config.json`.

Fichier de configuration de l'agent

Exemple

```
{
  "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
  ]
}
```

Répartition des champs

caractéristiques

Les fonctionnalités sont spécifiées sous forme de noms de ressources Amazon du groupe de points de terminaison Dataflow.

Obligatoire : True

Format : tableau de chaînes

- Valeurs : capacité ARNs → Chaîne

Exemples :

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

device

Ce champ contient des champs supplémentaires nécessaires pour énumérer le « périphérique » EC2 actuel.

Obligatoire : True

Format : Objet

Membres :

- IP privées
- IP publiques
- agentCpuCores
- Adaptateurs réseau

IP privées

Ce champ n'est actuellement pas utilisé, mais il sera inclus pour les futurs cas d'utilisation. Si aucune valeur n'est incluse, la valeur par défaut sera [« 127.0.0.1 »]

Obligatoire : False

Format : tableau de chaînes

- Valeurs : Adresses IP → Chaîne

Exemple :

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

IP publiques

IP élastique (EIP) par groupe de points de terminaison de flux de données.

Obligatoire : True

Format : tableau de chaînes

- Valeurs : Adresses IP → Chaîne

Exemple :

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

agent CPUCores

Cela indique quels cœurs virtuels sont réservés au aws-gs-agent processus. Consultez [Planification du cœur du processeur](#) les exigences relatives à la définition appropriée de cette valeur.

Obligatoire : True

Format : tableau int

- Valeurs : Numéros de base → int

Exemple :

```
"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  
]
```

Adaptateurs réseau

Cela correspond aux adaptateurs Ethernet, ou aux interfaces connectées ENIs, qui recevront les données.

Obligatoire : False

Format : tableau de chaînes

- Valeurs : noms des adaptateurs Ethernet (vous pouvez les trouver en exécutant `ifconfig`)

Exemple :

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

Optimisez les performances de votre instance EC2

Note

Si vous avez provisionné vos ressources AWS à l'aide CloudFormation de modèles, ces réglages sont automatiquement appliqués. Si vous avez utilisé une AMI ou créé manuellement votre instance EC2, ces réglages de performance doivent être appliqués pour obtenir les performances les plus fiables.

N'oubliez pas de redémarrer votre instance après avoir appliqué un ou plusieurs réglages.

Rubriques

- [Réglez les interruptions matérielles et les files d'attente de réception, ce qui a un impact sur le processeur et le réseau](#)
- [Tune Rx interrompt la coalescence, ce qui a un impact sur le réseau](#)
- [Tune Rx Ring Buffer, impacte le réseau](#)
- [Réglage le C-State du processeur : impacte le processeur](#)
- [Ports d'entrée de réserve - impact sur le réseau](#)
- [Redémarrer](#)

Réglez les interruptions matérielles et les files d'attente de réception, ce qui a un impact sur le processeur et le réseau

Cette section configure l'utilisation du cœur du processeur par systemd, SMP IRQs, RPS (Receive Packet Steering) et RFS (Receive Flow Steering). Consultez [Annexe : Paramètres recommandés pour le interrupt/RPS réglage](#) un ensemble de paramètres recommandés en fonction du type d'instance que vous utilisez.

1. Éloignez les processus systemd des cœurs du processeur de l'agent.
2. Acheminez les demandes d'interruption matérielle en les éloignant des cœurs du processeur de l'agent.
3. Configurez le RPS pour éviter que la file d'attente matérielle d'une seule carte d'interface réseau ne devienne un goulot d'étranglement pour le trafic réseau.

4. Configurez RFS pour augmenter le taux de réussite du cache du processeur et réduire ainsi la latence du réseau.

Le `set_irq_affinity.sh` script fourni par le RPM configure tout ce qui précède pour vous. Ajoutez à `crontab` pour qu'il soit appliqué à chaque démarrage :

```
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` Remplacez-les par des cœurs réservés au noyau et au système d'exploitation, généralement le premier et le second, ainsi que par des paires de cœurs hyperthreadées. Cela ne doit pas se chevaucher avec les cœurs sélectionnés ci-dessus. (Par exemple : « 0,1,48,49 » pour une instance hyperthread à 96 processeurs).
- `rps_core_mask` est un masque de bits hexadécimal spécifiant celui qui CPUs doit traiter les paquets entrants, chaque chiffre représentant 4 CPUs. Il doit également être séparé par une virgule tous les 8 caractères en partant de la droite. Il est recommandé de tout autoriser CPUs et de laisser la mise en cache s'occuper de l'équilibrage.
 - Pour consulter la liste des paramètres recommandés pour chaque type d'instance, reportez-vous à [Annexe : Paramètres recommandés pour le interrupt/RPS réglage](#).
- Exemple pour une instance à 96 processeurs :

```
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 interrompt la coalescence, ce qui a un impact sur le réseau

La fusion des interruptions permet d'éviter d'inonder le système hôte d'un trop grand nombre d'interruptions et d'augmenter le débit du réseau. Avec cette configuration, les paquets sont collectés et une seule interruption est générée toutes les 128 microsecondes. Ajoutez à `crontab` pour qu'il soit appliqué à chaque démarrage :

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

- `interface` Remplacez-le par l'interface réseau (adaptateur Ethernet) configurée pour recevoir des données. En général, c'est parce `eth0` qu'il s'agit de l'interface réseau par défaut attribuée à une instance EC2.

Tune Rx Ring Buffer, impacte le réseau

Augmentez le nombre d'entrées en boucle pour la mémoire tampon Rx afin d'éviter les pertes de paquets ou les dépassements lors de connexions en rafale. Ajoutez au crontab pour qu'il soit correctement configuré à chaque démarrage :

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

- `interface` Remplacez-le par l'interface réseau (adaptateur Ethernet) configurée pour recevoir des données. En général, c'est parce `eth0` qu'il s'agit de l'interface réseau par défaut attribuée à une instance EC2.
- Si vous configurez une instance `c6i` familiale, la commande doit être modifiée pour définir le ring buffer sur `8192`, au lieu de `16384`.

Régler le C-State du processeur : impacte le processeur

Réglez l'état C du processeur pour empêcher le ralenti, ce qui peut entraîner la perte de paquets au début d'un contact. Nécessite le redémarrage de l'instance.

```
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
```

Ports d'entrée de réserve - impact sur le réseau

Réservez tous les ports de la plage `AwsGroundStationAgentEndpoint` de ports de votre adresse d'entrée pour éviter tout conflit avec l'utilisation du noyau. Un conflit d'utilisation du port entraînera un échec du contact et de la livraison des données.

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

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

Redémarrer

Une fois que tous les réglages ont été correctement appliqués, redémarrez l'instance pour qu'ils prennent effet.

```
sudo reboot
```

Annexe : Paramètres recommandés pour le interrupt/RPS réglage

Cette section détermine les valeurs de paramètres recommandées à utiliser dans la section de réglage `Tune Hardware Interrupts and Receive Queues - Impacts sur le processeur et le réseau.`

Family	Type d'instance	<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"> ffffff,ffffff,ffffff fff,ffffff

Family	Type d'instance	$\$ \{interrupt_core_list\}$	$\$ \{rps_core_mask\}$
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"> fffffff, ffffffff, ffffffff, ffffffff
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, 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"> fffffff, 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"> fffffff, ffffffff, ffffffff fffffff, ffffffff, ffffffff

Family	Type d'instance	\$ {interrupt_core_list}	\$ {rps_core_mask}
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, ffffffff, ffffffff fffffff, ffffffff, ffffffff
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"> fffffff, ffffffff, ffffffff fffffff, ffffffff ffff, ffffffff
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"> fffffff, ffffffff, ffffffff
p3dn	<ul style="list-style-type: none"> p3dn.24xlarge 	<ul style="list-style-type: none"> 0,1,48,49 	<ul style="list-style-type: none"> fffffff, ffffffff, ffffffff

Préparez-vous à prendre un contact DigiF

1. Consultez la planification des cœurs du processeur pour connaître les flux de données souhaités et fournissez une liste des cœurs que l'agent peut utiliser. Consultez [Planification du cœur du processeur](#).
2. Consultez le fichier de configuration de l' AWS Ground Station agent. Consultez [AWS Ground Station Configuration de l'agent](#).
3. Vérifiez que le réglage des performances nécessaire est appliqué. Consultez [Optimisez les performances de votre instance EC2](#).
4. Confirmez que vous suivez toutes les meilleures pratiques énoncées. Consultez [Bonnes pratiques](#).
5. Vérifiez que l' AWS Ground Station agent est démarré avant l'heure de début prévue du contact via :

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

6. Vérifiez que l' AWS Ground Station agent est en bonne santé avant l'heure prévue de début du contact en :

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

Vérifiez que le votre agentStatus awsGroundStationAgentEndpoint est ACTIF et qu'il auditResults est EN BONNE SANTÉ.

Bonnes pratiques

Bonnes pratiques relatives à Amazon EC2

Suivez les meilleures pratiques actuelles de l'EC2 et garantisiez une disponibilité suffisante du stockage des données.

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

Planificateur Linux

Le planificateur Linux peut réorganiser les paquets sur les sockets UDP si les processus correspondants ne sont pas liés à un cœur spécifique. Tout thread qui envoie ou reçoit des données UDP doit s'associer à un noyau spécifique pendant toute la durée de la transmission des données.

AWS Ground Station liste de préfixes gérée

Il est recommandé d'utiliser la liste de préfixes `com.amazonaws.global.groundstation` gérée par AWS lorsque vous spécifiez les règles du réseau pour autoriser les communications depuis l'antenne. Consultez la section [Utilisation des listes de préfixes gérées par AWS pour plus d'informations sur les listes](#) de préfixes gérées par AWS.

Limitation du contact unique

L'agent AWS Ground Station prend en charge plusieurs flux par contact, mais ne prend en charge qu'un seul contact à la fois. Pour éviter les problèmes de planification, ne partagez pas une instance entre plusieurs groupes de points de terminaison de flux de données. Si une configuration d'agent unique est associée à plusieurs DFEG différents ARNs, elle ne sera pas enregistrée.

Exécution de services et de processus en parallèle avec l' AWS Ground Station agent

Lorsque vous lancez des services et des processus sur la même instance EC2 que l' AWS Ground Station agent, il est important de les lier à v CPUs non utilisés par l' AWS Ground Station agent et le noyau Linux, car cela peut entraîner des blocages et même des pertes de données lors des contacts. Ce concept de liaison à un v spécifique CPUs est connu sous le nom d'affinité.

Noyaux à éviter :

- `agentCpuCores` à partir de [Fichier de configuration de l'agent](#)
- `interrupt_core_list` depuis [Réglez les interruptions matérielles et les files d'attente de réception, ce qui a un impact sur le processeur et le réseau.](#)
 - Les valeurs par défaut peuvent être trouvées dans [Annexe : Paramètres recommandés pour le interrupt/RPS réglage](#)

À titre d'exemple, en utilisant une **c5.24xlarge** instance

Si vous avez spécifié

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

et a couru

```
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
```

puis évitez les noyaux suivants :

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

Services d'affinisation (systemd)

Les services nouvellement lancés s'affineront automatiquement aux services `interrupt_core_list` mentionnés précédemment. Si le cas d'utilisation des services que vous avez lancés nécessite des cœurs supplémentaires ou des cœurs moins encombrés, suivez cette section.

Vérifiez l'affinité avec laquelle votre service est actuellement configuré à l'aide de la commande :

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

Si vous voyez une valeur vide telle que `CPUAffinity=`, cela signifie qu'elle utilisera probablement les cœurs par défaut de la commande ci-dessus `...bin/set_irq_affinity.sh <using the cores here> ...`

Pour remplacer et définir une affinité spécifique, recherchez l'emplacement du fichier de service en exécutant :

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

Ouvrez et modifiez le fichier (en utilisant `vim`, etc.) et placez-le `CPUAffinity=<core list>` dans la `[Service]` section comme suit :

```
[Unit]
...

[Service]
...
CPUAffinity=2,3

[Install]
...
```

Enregistrez le fichier et redémarrez le service pour appliquer l'affinité avec :

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

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

Pour plus d'informations, consultez : [Red Hat Enterprise Linux 8 - Gestion, surveillance et mise à jour du noyau - Chapitre 27. Configuration de l'affinité du processeur et des politiques NUMA à l'aide de `systemd`](#).

Processus d'affinisation (scripts)

Il est fortement recommandé d'affiniser manuellement les scripts et processus récemment lancés, car le comportement Linux par défaut leur permet d'utiliser n'importe quel cœur de la machine.

Pour éviter les conflits de base entre les processus en cours d'exécution (tels que python, scripts bash, etc.), lancez le processus avec :

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

Si le processus est déjà en cours d'exécution, utilisez des commandes telles que `pidof top`, ou `ps` pour trouver l'ID de processus (PID) du processus spécifique. Avec le PID, vous pouvez voir l'affinité actuelle avec :

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

et vous pouvez le modifier avec :

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

Pour plus d'informations sur le jeu de tâches, voir la page de manuel de [jeu de tâches - Linux](#)

Résolution des problèmes

L'agent ne démarre pas

L' AWS Ground Station agent peut ne pas démarrer pour plusieurs raisons, mais le scénario le plus courant peut être un fichier de configuration de l'agent mal configuré. Après avoir démarré l'agent (voir [AWS Ground Station Démarrage de l'agent](#)), vous pouvez obtenir un statut tel que :

```
#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)
```

Résolution des problèmes

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

peut entraîner une sortie de :

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

L'échec du démarrage de l'agent après le « Loading Config » indique un problème de configuration de l'agent. Consultez [Fichier de configuration de l'agent](#) pour vérifier la configuration de votre agent.

AWS Ground Station Journaux des agents

AWS Ground Station L'agent écrit les informations relatives à l'exécution des contacts, aux erreurs et à l'état de santé dans les fichiers journaux de l'instance qui exécute l'agent. Vous pouvez consulter les fichiers journaux en vous connectant manuellement à une instance.

Vous pouvez consulter les journaux des agents à l'emplacement suivant.

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

Aucun contact disponible

La planification des contacts nécessite un AWS Ground Station agent en bonne santé. Vérifiez que votre AWS Ground Station agent a démarré et qu'il fonctionne correctement en interrogeant l' AWS Ground Station API via `get-dataflow-endpoint-group` :

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

Vérifiez que le votre `agentStatus awsGroundStationAgentEndpoint` est **ACTIF** et qu'il `auditResults` est **EN BONNE SANTÉ**.

Obtention de support

Contactez l'équipe de Ground Station via AWS Support.

1. Indiquez `contact_id` tous les contacts concernés. L' AWS Ground Station équipe ne peut pas enquêter sur un contact spécifique sans ces informations.
2. Fournissez des détails sur toutes les mesures de dépannage déjà prises.
3. Indiquez tous les messages d'erreur détectés lors de l'exécution des commandes dans notre guide de dépannage.

Notes de mise à jour des agents

Dernière version de l'agent

Version 1.0.4382.0

Date de sortie : 18/11/2025

Sommes de contrôle du RPM :

- SHA256: 620fd307124f1276194f2faa0104fe0549427ae18e4f5655444f8c30b919c640
- MD5: 73e06dcad44adaccbe2ab005218abfc7

Changements :

- Mettez à jour le comportement du client à chaque nouvelle tentative lorsque le serveur indique une surcharge.

Versions d'agent obsolètes

La version 1.0.3555.0

Date de sortie : 27/03/2024

Sommes de contrôle du RPM :

- SHA256: 108f3aceb00e5af549839cd766c56149397e448a6e1e1429c89a9eebb6bc0fc1
- MD5: 65b72fa507fb0af32651adbb18d2e30f

Changements :

- Ajoutez la métrique de l'agent pour la version exécutable sélectionnée lors du démarrage des tâches.
- Ajoutez le support des fichiers de configuration pour éviter des versions exécutables spécifiques lorsque d'autres versions sont disponibles.
- Ajoutez des diagnostics de réseau et de routage.

- Fonctionnalités de sécurité supplémentaires.
- Correction d'un problème en raison duquel certaines erreurs de rapport de mesures étaient enregistrées dans le fichier journal stdout/journal plutôt que dans le fichier journal.
- Gérez avec élégance les erreurs de socket inaccessibles sur le réseau.
- Mesurez la perte de paquets et la latence entre les agents source et de destination.
- Publiez aws-gs-datapipe la version 2.0 pour prendre en charge les nouvelles fonctionnalités du protocole et la possibilité de mettre à niveau de manière transparente les contacts vers le nouveau protocole.

Version 1.0.2942.0

Date de sortie : 26/06/2023

Date de fin du Support : 31/05/2024

Sommes de contrôle du RPM :

- SHA256: 7d94b642577504308a58bab28f938507f2591d4e1b2c7ea170b77bea97b5a9b6
- MD5: 661ff2b8f11aba5d657a6586b56e0d8f

Changements :

- Ajout de journaux d'erreurs lorsque le RPM de l'agent est mis à jour sur le disque et que l'agent doit être redémarré pour que les modifications prennent effet.
- Ajout de la validation du réglage du réseau pour garantir que les étapes de réglage du guide de l'utilisateur de l'agent sont suivies et appliquées correctement.
- Correction d'un bogue qui provoquait des avertissements erronés dans les journaux de l'agent concernant l'archivage des journaux.
- Détection améliorée des pertes de paquets.
- Installation de l'agent mise à jour pour empêcher l'installation ou la mise à niveau du RPM si l'agent est déjà en cours d'exécution.

Version 1.0.2716.0

Date de sortie : 15/03/2023

Date de fin du Support : 31/05/2024

Sommes de contrôle du RPM :

- SHA256: cb05b6a77dfcd5c66d81c0072ac550affbcefefc372cc5562ee52fb220844929
- MD5: 65266490c4013b433ec39ee50008116c

Changements :

- Activez le téléchargement des journaux lorsque l'agent rencontre des défaillances lors de l'exécution des tâches.
- Correction d'un bogue de compatibilité avec Linux dans les scripts de réglage réseau fournis.

La version 1.0.2677.0

Date de sortie : 15/02/2023

Date de fin du Support : 31/05/2024

Sommes de contrôle du RPM :

- SHA256: 77cfe94acb00af7ca637264b17c9b21bd7afdc85b99dffdd627aec9e99397489
- MD5: b8533be7644bb4d12ab84de21341adac

Changements :

- Première version de l'agent généralement disponible.

Validation d'installation RPM

La dernière version de RPM, le MD5 hachage validé à partir de RPM et le SHA256 hachage utilisant sha256sum sont présentés ci-dessous. Ces valeurs, combinées, peuvent être utilisées pour valider la version RPM utilisée pour l'agent de la station au sol.

Dernière version de l'agent

Version 1.0.4382.0

Date de sortie : 18/11/2025

Sommes de contrôle du RPM :

- SHA256: 620fd307124f1276194f2faa0104fe0549427ae18e4f5655444f8c30b919c640
- MD5: 73e06dcad44adaccbe2ab005218abfc7

Changements :

- Mettez à jour le comportement du client à chaque nouvelle tentative lorsque le serveur indique une surcharge.

Vérifiez le RPM

Les outils dont vous aurez besoin pour vérifier cette installation RPM sont les suivants :

- [sha256sum](#)
- [tr/mn](#)

Les deux outils sont fournis par défaut sur Amazon Linux 2. Ces outils vous aideront à vérifier que le RPM que vous utilisez est la bonne version. Téléchargez d'abord le dernier RPM depuis le compartiment S3 (voir [Agent de téléchargement](#) les instructions relatives au téléchargement du RPM). Une fois ce fichier téléchargé, il y aura quelques points à vérifier :

- Calculez le sha256sum du fichier RPM. Effectuez l'action suivante depuis la ligne de commande de l'instance de calcul que vous utilisez :

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

Prenez cette valeur et comparez-la au tableau ci-dessus. Cela montre que le fichier RPM téléchargé est un fichier valide à utiliser qu'AWS Ground Station a distribué aux clients. Si les hachages ne correspondent pas, n'installez pas le RPM et supprimez-le de l'instance de calcul.

- Vérifiez également le MD5 hachage du fichier pour vous assurer que le RPM n'a pas été compromis. Pour ce faire, utilisez l'outil de ligne de commande RPM en exécutant la commande suivante :

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

Vérifiez que le MD5 hachage indiqué ici est le même que le MD5 hachage de la version figurant dans le tableau ci-dessus. Une fois que ces deux hachages ont été validés par rapport à ce tableau répertorié dans AWS Docs, le client peut être assuré que le RPM téléchargé et installé est la version sûre et non compromise du RPM.

Historique du document pour le guide de l'utilisateur de l'AWS Ground Station agent

Le tableau suivant décrit les modifications importantes apportées à chaque version du Guide de l'utilisateur de l'AWS Ground Station agent.

Modification	Description	Date
Mise à jour de documentation	Suppression de la prise en charge de l'ancienne famille d'instances : m4.	30 septembre 2024
Mise à jour de documentation	Ajout d'un commentaire sur le maintien du sous-réseau et de l'EC2 instance Amazon dans la même zone de disponibilité dans la section Exigences relatives aux agents .	18 juillet 2024
Mise à jour de documentation	Divisez l'AWS Ground Station agent en son propre guide d'utilisation. Pour les modifications antérieures, reportez-vous à : Historique des documents du guide de l'utilisateur d'AWS Ground Station .	18 juillet 2024

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.