



Guide du développeur

Amazon Data Firehose



Amazon Data Firehose: Guide du développeur

Copyright © 2024 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

.....	x
Qu'est-ce qu'Amazon Data Firehose ?	1
Découvrez les concepts clés	1
Comprendre le flux de données dans Amazon Data Firehose	2
Configuration	5
Inscrivez-vous pour AWS	5
(Facultatif) Téléchargez des bibliothèques et des outils	5
Création d'un stream Firehose	7
Configuration de la source et de la destination	7
Configuration de la transformation des enregistrements et de la conversion des formats	10
Configuration des paramètres de destination	12
Configurer les paramètres de destination pour Amazon S3	12
Configuration des paramètres de destination pour Amazon Redshift	16
Configurer les paramètres de destination pour le OpenSearch service	23
Configuration des paramètres de destination pour OpenSearch Serverless	25
Configurer les paramètres de destination pour le point de terminaison HTTP	27
Configurer les paramètres de destination pour Datadog	29
Configurer les paramètres de destination pour Honeycomb	31
Configuration des paramètres de destination pour Coralogix	33
Configuration des paramètres de destination pour Dynatrace	35
Configurer les paramètres de destination pour LogicMonitor	37
Configurer les paramètres de destination pour Logz.io	39
Configurer les paramètres de destination pour MongoDB Cloud	41
Configurer les paramètres de destination pour New Relic	43
Configurer les paramètres de destination pour Snowflake	45
Configurer les paramètres de destination pour Splunk	48
Configurer les paramètres de destination pour Splunk Observability Cloud	50
Configuration des paramètres de destination pour Sumo Logic	52
Configuration des paramètres de destination pour Elastic	53
Configuration de la sauvegarde et des paramètres avancés	55
Configuration des paramètres de sauvegarde	55
Configurer les paramètres avancés	57
Comprendre les conseils de mise en mémoire tampon	59
Testez votre stream Firehose	63

Prérequis	63
Effectuer un test à l'aide d'Amazon S3 comme destination	63
Effectuer un test à l'aide d'Amazon Redshift comme destination	64
Test en utilisant le OpenSearch service comme destination	65
Test avec Splunk comme destination	65
Envoi de données vers un stream Firehose	67
Écrire à l'aide de Kinesis Data Streams	67
Écrire à l'aide d'Amazon MSK	69
Rédaction à l'aide de l'agent Amazon Data Firehose	71
Prérequis	72
Informations d'identification	72
Fournisseurs d'informations d'identification personnalisées	73
Téléchargement et installation de l'agent	74
Configuration et démarrage de l'agent	76
Paramètres de configuration de l'agent	77
Surveillance de plusieurs répertoires de fichiers et écriture dans plusieurs flux	81
Utilisation de l'agent pour prétraiter les données	82
Interface de ligne de commande de l'agent	87
FAQ	87
Envoyer des données à l'aide du AWS SDK	89
Opérations d'écriture unique utilisant PutRecord	89
Opérations d'écriture par lots à l'aide de PutRecordBatch	90
Écrire à l'aide de CloudWatch journaux	90
Décompression des journaux CloudWatch	91
Extraction des messages après décompression des journaux CloudWatch	91
Activation et désactivation de la décompression	92
FAQ	87
Écrire à l'aide d' CloudWatch événements	96
Écriture avec AWS IoT	96
Sécurité	98
Protection des données	99
Chiffrement côté serveur avec Kinesis Data Streams comme source de données	99
Chiffrement côté serveur avec l'instruction PUT directe ou d'autres sources de données	100
Contrôle de l'accès	101
Accordez à votre application l'accès à vos ressources Amazon Data Firehose	102
Accordez à Amazon Data Firehose l'accès à votre cluster Amazon MSK privé	103

Autoriser Amazon Data Firehose à assumer un rôle IAM	103
Accorder à Amazon Data Firehose l'accès à des AWS Glue fins de conversion de formats de données	106
Accorder à Amazon Data Firehose l'accès à une destination Amazon S3	107
Accorder à Amazon Data Firehose l'accès à une destination Amazon Redshift	110
Accorder à Amazon Data Firehose l'accès à une destination de service public OpenSearch	114
Accorder à Amazon Data Firehose l'accès à une destination de OpenSearch service dans un VPC	117
Accorder à Amazon Data Firehose l'accès à une destination publique sans serveur OpenSearch	118
Accorder à Amazon Data Firehose l'accès à une destination OpenSearch sans serveur dans un VPC	121
Accorder à Amazon Data Firehose l'accès à une destination Splunk	123
Accès à Splunk dans un VPC	125
Accès à Snowflake ou au point de terminaison HTTP	126
Accorder à Amazon Data Firehose l'accès à une destination Snowflake	127
Accès à Snowflake en VPC	129
Accorder à Amazon Data Firehose l'accès à une destination de point de terminaison HTTP	132
Livraison entre comptes depuis Amazon MSK	135
Diffusion entre comptes vers une destination Amazon S3	138
Livraison entre comptes vers une destination OpenSearch de service	139
Utilisation des balises pour contrôler l'accès	141
Authentifiez-vous avec AWS Secrets Manager	143
Comprenez les secrets	144
Créer un secret	145
Utilisez le secret	145
Faites pivoter le secret	147
Gestion des rôles IAM via la console	147
Choisissez un rôle IAM existant	148
Création d'un nouveau rôle IAM depuis la console	148
Modifier le rôle IAM depuis la console	150
Surveillance	151
Validation de la conformité	152
Résilience	152

Reprise après sinistre	153
Sécurité de l'infrastructure	153
Points de terminaison d'un VPC PrivateLink	154
Bonnes pratiques de sécurité	154
Implémentation d'un accès sur la base du moindre privilège	154
Utilisation des rôles IAM	154
Implémentation d'un chiffrement côté serveur dans des ressources dépendantes	155
CloudTrail À utiliser pour surveiller les appels d'API	155
Transformation de données	156
Flux de transformation de données	156
Transformation de données et modèle d'état	156
Plans Lambda	158
Gestion des échecs de transformation des données	159
Durée d'une invocation Lambda	161
Sauvegarde de l'enregistrement source	161
Partitionnement dynamique	162
Clés de partitionnement	163
Création de clés de partitionnement avec analyse en ligne	163
Création de clés de partitionnement à l'aide d'une fonction AWS Lambda	164
Préfixe de compartiment Amazon S3 pour le partitionnement dynamique	168
Partitionnement dynamique des données agrégées	169
Ajout d'un nouveau délimiteur de ligne lors de la transmission de données à S3	170
Comment activer le partitionnement dynamique	171
Gestion des erreurs de partitionnement dynamique	171
Mise en mémoire tampon des données et partitionnement dynamique	172
Conversion du format d'enregistrement	174
Exigences pour la conversion du format d'enregistrement	174
Choix du désérialiseur JSON	175
Choix du sérialiseur	177
Conversion du format d'enregistrement d'entrée (console)	177
Conversion du format d'enregistrement d'entrée (API)	177
Gestion des erreurs de conversion de format d'enregistrement	178
Exemple de conversion du format d'enregistrement	179
Intégration au service géré pour Apache Flink	180
Livraison des données	181
Configuration du format de livraison des données	181

Comprendre la fréquence de diffusion des données	183
Gérer les défaillances de livraison des données	183
Configurer le format du nom d'objet Amazon S3	188
Configurer la rotation de l'index pour le OpenSearch service	197
Comprenez la livraison entre AWS les comptes et les régions	198
Enregistrements dupliqués	198
Suspendre et reprendre un stream Firehose	198
Comprendre comment Firehose gère les défaillances de livraison	199
Interruption d'un stream Firehose	199
Reprise d'un stream Firehose	200
Surveillance	201
Bonnes pratiques avec les alarmes CloudWatch	201
Surveillance à l'aide de CloudWatch métriques	202
Métriques de partitionnement CloudWatch dynamique	203
CloudWatch Métriques de livraison de données	204
Métriques d'ingestion de données	217
Métriques au niveau de l'API CloudWatch	224
CloudWatch Métriques de transformation des données	227
CloudWatch Métriques de décompression des journaux	228
CloudWatch Métriques de conversion de formats	228
Métriques de chiffrement côté serveur (SSE) CloudWatch	229
Dimensions pour Amazon Data Firehose	230
Mesures d'utilisation d'Amazon Data Firehose	230
Accès aux CloudWatch métriques pour Amazon Data Firehose	231
Surveillance à l'aide de CloudWatch journaux	232
Erreurs de diffusion de données	233
Accès aux CloudWatch journaux pour Amazon Data Firehose	270
Surveillance de l'état de l'agent	271
Surveillance avec CloudWatch	272
Journalisation des appels d'API Amazon Data Firehose avec AWS CloudTrail	272
Informations sur Amazon Data Firehose dans CloudTrail	273
Exemple : entrées dans le fichier journal Amazon Data Firehose	274
Préfixes Amazon S3 personnalisés	280
L'espace de noms timestamp	280
L'espace de noms firehose	281
Espaces de noms partitionKeyFromLambda et partitionKeyFromQuery	282

Règles sémantiques	283
Exemples de préfixe	284
Utilisation d'Amazon Data Firehose avec AWS PrivateLink	286
Points de terminaison VPC d'interface (AWS PrivateLink pour Amazon Data Firehose)	286
Utilisation des points de terminaison VPC d'interface (AWS PrivateLink pour Amazon Data Firehose)	286
Disponibilité	290
Marquer vos streams Firehose	292
Principes de base des identifications	292
Suivi des coûts à l'aide du balisage	293
Restrictions liées aux balises	294
Marquage de flux Firehose à l'aide de l'API Amazon Data Firehose	294
Tutoriel : Ingérer les journaux de flux VPC dans Splunk à l'aide d'Amazon Data Firehose	296
Résolution des problèmes	297
Problèmes courants	297
Résolutions des problèmes liés à Amazon S3	298
Dépannage d'Amazon Redshift	299
Résolution des problèmes liés à Amazon OpenSearch Service	300
Dépannage de Splunk	301
Résolution des problèmes liés à Snowflake	303
La création du stream Firehose échoue	303
Résolution des problèmes d'accessibilité des terminaux Firehose	305
Dépannage des points de terminaison HTTP	306
CloudWatch Journaux	306
Dépannage de MSK comme source	310
Échec de la création de tuyaux	310
Tuyau suspendu	310
Tuyau contre-pressurisé	311
Actualité des données incorrecte	311
Problèmes de connexion au cluster MSK	311
Mesure de fraîcheur des données croissante ou non émise	314
La conversion du format d'enregistrement vers Apache Parquet échoue	316
Quota	317
Annexe : spécifications des demandes et réponses de livraison de points de terminaison HTTP ...	321
Format des demandes	321
Format de la réponse	325

Exemples	328
Historique du document	329
Glossaire AWS	333

Amazon Data Firehose était auparavant connu sous le nom d'Amazon Kinesis Data Firehose

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.

Qu'est-ce qu'Amazon Data Firehose ?

Amazon Data Firehose est un service entièrement géré qui fournit des [données de streaming](#) en temps réel vers des destinations telles qu'Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon Service, OpenSearch Amazon Serverless, Splunk et tout point de terminaison HTTP personnalisé ou appartenant à des fournisseurs de services tiers compatibles, notamment Datadog, LogicMonitor Dynatrace, MongoDB, New Relic, Corco alogix et Elastic. OpenSearch Avec Amazon Data Firehose, vous n'avez pas besoin d'écrire d'applications ni de gérer des ressources. Vous configurez vos producteurs de données pour qu'ils envoient des données à Amazon Data Firehose, qui les envoie automatiquement à la destination que vous avez spécifiée. Vous pouvez également configurer Amazon Data Firehose pour transformer vos données avant de les diffuser.

Pour plus d'informations sur les solutions AWS Big Data, voir [Big Data on AWS](#). Pour en savoir plus sur les solutions de données de diffusion AWS , consultez [Qu'est-ce que les données de streaming ?](#)

Note

Notez la dernière [solution de données de AWS streaming pour Amazon MSK](#) qui fournit des AWS CloudFormation modèles dans lesquels les données circulent entre les producteurs, le stockage en streaming, les consommateurs et les destinations.

Découvrez les concepts clés

Lorsque vous débutez avec Amazon Data Firehose, vous pouvez tirer parti de la compréhension des concepts suivants :

Stream Firehose

L'entité sous-jacente d'Amazon Data Firehose. Vous utilisez Amazon Data Firehose en créant un flux Firehose, puis en lui envoyant des données. Pour en savoir plus, veuillez consulter [Création d'un stream Firehose](#) et [Envoyer des données vers un flux Firehose](#).

record

Les données présentant un intérêt que votre producteur de données envoie à un flux Firehose. Un enregistrement peut atteindre 1000 Ko.

application producteur

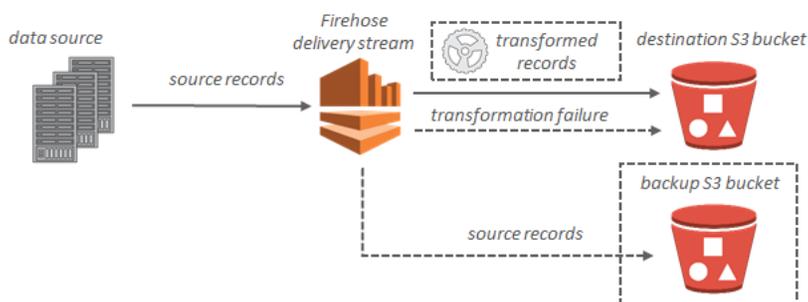
Les producteurs envoient leurs disques aux streams Firehose. Par exemple, un serveur Web qui envoie des données de journal à un flux Firehose est un producteur de données. Vous pouvez également configurer votre flux Firehose pour lire automatiquement les données d'un flux de données Kinesis existant et les charger dans des destinations. Pour plus d'informations, consultez [Envoyer des données vers un flux Firehose](#).

taille du tampon et intervalle de temps tampon

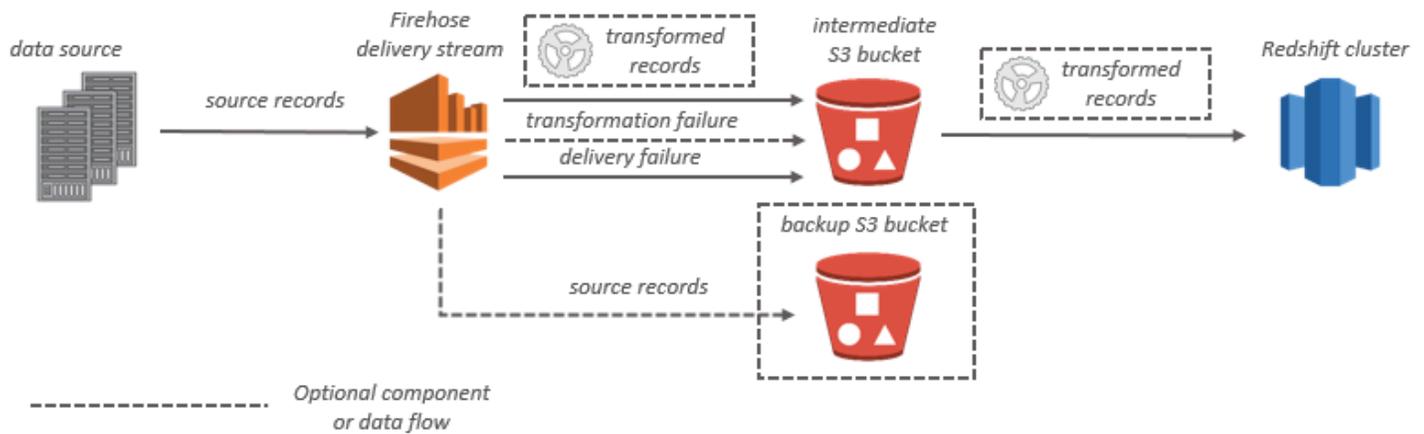
Amazon Data Firehose met en mémoire tampon les données de streaming entrantes jusqu'à une certaine taille ou pendant une certaine période avant de les diffuser vers les destinations. Buffer Size est en Mo et Buffer Interval en secondes.

Comprendre le flux de données dans Amazon Data Firehose

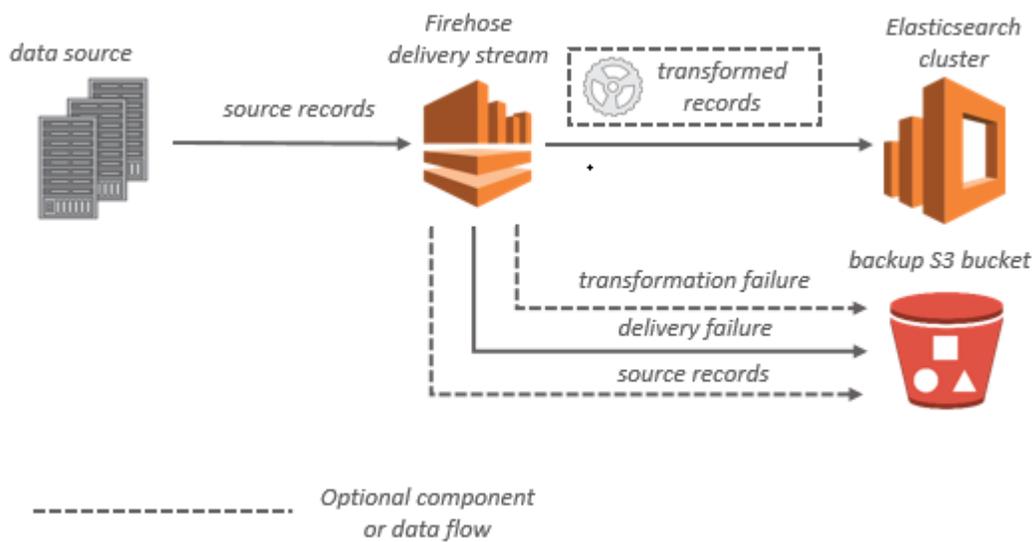
Pour les destinations Amazon S3, les données de streaming sont délivrées à votre compartiment S3. Si la transformation de données est activée, vous pouvez éventuellement sauvegarder les données source dans un autre compartiment Amazon S3.



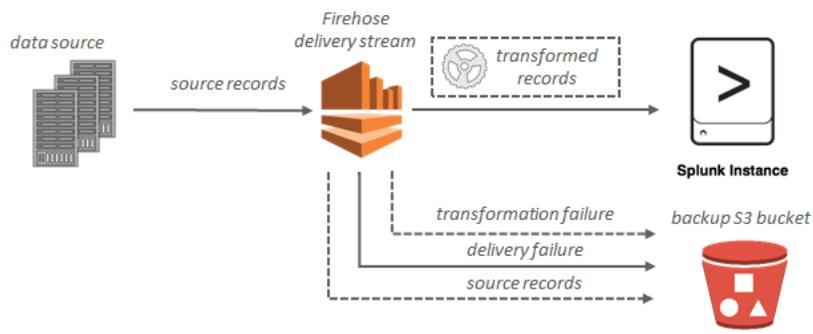
Pour les destinations Amazon Redshift, les données de streaming sont d'abord délivrées à votre compartiment S3. Amazon Data Firehose émet ensuite une commande Amazon COPY Redshift pour charger les données de votre compartiment S3 vers votre cluster Amazon Redshift. Si la transformation de données est activée, vous pouvez éventuellement sauvegarder les données source dans un autre compartiment Amazon S3.



Pour les destinations de OpenSearch service, les données de streaming sont transmises à votre cluster de OpenSearch services et peuvent éventuellement être sauvegardées simultanément dans votre compartiment S3.



Pour les destinations Splunk, les données de streaming sont remises à Splunk et peuvent éventuellement être sauvegardées dans votre compartiment S3 simultanément.



Configuration d'Amazon Data Firehose

Avant d'utiliser Amazon Data Firehose pour la première fois, effectuez les tâches suivantes.

Tâches

- [Inscrivez-vous pour AWS](#)
- [\(Facultatif\) Téléchargez des bibliothèques et des outils](#)

Inscrivez-vous pour AWS

Lorsque vous vous inscrivez à Amazon Web Services (AWS), votre AWS compte est automatiquement inscrit à tous les services AWS, y compris Amazon Data Firehose. Seuls les services que vous utilisez vous sont facturés.

Si vous avez déjà un AWS compte, passez à la tâche suivante. Si vous n'avez pas de compte AWS, observez la procédure suivante pour en créer un.

Pour créer un AWS compte

1. Ouvrez <https://portal.aws.amazon.com/billing/signup>.
2. Suivez les instructions en ligne.

Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique en matière de sécurité consiste à attribuer un accès administratif à un utilisateur et à n'utiliser que l'utilisateur root pour effectuer [les tâches nécessitant un accès utilisateur root](#).

(Facultatif) Téléchargez des bibliothèques et des outils

Les bibliothèques et outils suivants vous aideront à travailler avec Amazon Data Firehose par programmation et à partir de la ligne de commande :

- Les opérations d'[API Firehose constituent l'ensemble d'opérations](#) de base pris en charge par Amazon Data Firehose.
- Les AWS kits SDK pour [Go](#), [Java](#), [.NET](#), [Node.js](#), [Python](#) et [Ruby](#) incluent le support et des exemples d'Amazon Data Firehose.

Si votre version AWS SDK for Java ne contient pas d'exemples pour Amazon Data Firehose, vous pouvez également télécharger le dernier AWS SDK depuis. [GitHub](#)

- est [AWS Command Line Interface](#) compatible avec Amazon Data Firehose. AWS CLI Cela vous permet de contrôler plusieurs AWS services à partir de la ligne de commande et de les automatiser par le biais de scripts.

Création d'un stream Firehose

Vous pouvez utiliser le AWS Management Console ou un AWS SDK pour créer un flux Firehose vers la destination de votre choix.

Vous pouvez mettre à jour la configuration de votre stream Firehose à tout moment après sa création, à l'aide de la console Amazon Data Firehose ou. [UpdateDestination](#) Votre flux Firehose reste en l'Active état pendant la mise à jour de votre configuration, et vous pouvez continuer à envoyer des données. La configuration mise à jour prend généralement effet au bout de quelques minutes. Le numéro de version d'un stream Firehose est augmenté d'une valeur égale à une 1 fois que vous avez mis à jour la configuration. Il est pris en compte dans le nom de l'objet Amazon S3 délivré. Pour plus d'informations, consultez [Configurer le format du nom d'objet Amazon S3](#).

Les rubriques suivantes décrivent comment créer un stream Firehose.

Rubriques

- [Configuration de la source et de la destination](#)
- [Configuration de la transformation des enregistrements et de la conversion des formats](#)
- [Configuration des paramètres de destination](#)
- [Configuration de la sauvegarde et des paramètres avancés](#)
- [Comprendre les conseils de mise en mémoire tampon](#)

Configuration de la source et de la destination

1. Connectez-vous à la console Amazon Data Firehose AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/firehose>
2. Choisissez Create Firehose stream.
3. Saisissez des valeurs pour les champs suivants :

Source

- Direct PUT : choisissez cette option pour créer un flux Firehose dans lequel les applications du producteur écrivent directement. Actuellement, les AWS services, agents et services open source suivants sont intégrés à Direct PUT dans Amazon Data Firehose :
 - AWS SDK

- AWS Lambda
 - AWS CloudWatch Journaux
 - AWS CloudWatch Évènements
 - AWS Streams métriques dans le cloud
 - AWS Internet des objets
 - AWS Eventbridge
 - Amazon Simple Email Service
 - Amazon SNS
 - AWS Journaux ACL Web WAF
 - Amazon API Gateway – Journaux d'accès
 - Amazon Pinpoint
 - Journaux du broker Amazon MSK
 - Journaux de requête Amazon Route 53 Resolver
 - AWS Journaux des alertes du Network Firewall
 - AWS Journaux de flux de Network Firewall
 - Amazon Elasticache Redis SLOWLOG
 - Kinesis Agent (Linux)
 - Kinesis Tap (Windows)
 - Fluentbit
 - Fluentd
 - Apache Nifi
 - Snowflake
- Flux Kinesis : choisissez cette option pour configurer un flux Firehose qui utilise un flux de données Kinesis comme source de données. Vous pouvez ensuite utiliser Amazon Data Firehose pour lire facilement les données d'un flux de données Kinesis existant et les charger dans des destinations. Pour plus d'informations sur l'utilisation de Kinesis Data Streams comme source de données, [consultez Writing to Amazon Data Firehose Using Kinesis Data Streams](#).
 - Amazon MSK : choisissez cette option pour configurer un flux Firehose qui utilise Amazon MSK comme source de données. Vous pouvez ensuite utiliser Firehose pour lire facilement les données d'un cluster Amazon MSK existant et les charger dans des compartiments

S3 spécifiés. Pour plus d'informations sur l'utilisation d'Amazon MSK comme source de données, consultez [Writing to Amazon Data Firehose Using Amazon MSK](#).

Destination du stream Firehose

La destination de votre stream Firehose. Amazon Data Firehose peut envoyer des enregistrements de données vers différentes destinations, notamment Amazon Simple Storage Service (Amazon S3), Amazon Redshift, OpenSearch Amazon Service et tout point de terminaison HTTP appartenant à vous ou à l'un de vos fournisseurs de services tiers. Les destinations prises en charge sont les suivantes :

- Amazon OpenSearch Service
- Amazon OpenSearch sans serveur
- Amazon Redshift
- Amazon S3
- Coralogix
- Datadog
- Dynatrace
- Elasticité
- Point de terminaison HTTP
- Honeycomb
- Logic Monitor
- Logz.io
- MongoDB Cloud
- New Relic
- Splunk
- Splunk Observability Cloud
- Sumo Logic
- Snowflake

Nom du stream Firehose

Le nom de votre stream Firehose.

Configuration de la transformation des enregistrements et de la conversion des formats

Configurez Amazon Data Firehose pour transformer et convertir vos données d'enregistrement.

- Si vous choisissez Amazon MSK comme source pour votre stream Firehose.
 1. Dans la section Transformer les enregistrements source avec AWS Lambda, fournissez des valeurs pour le champ suivant :

Transformation de données

Pour créer un flux Firehose qui ne transforme pas les données entrantes, ne cochez pas la case Activer la transformation des données.

Pour spécifier une fonction Lambda que Firehose doit invoquer et utiliser pour transformer les données entrantes avant de les transmettre, cochez la case Activer la transformation des données. Vous pouvez configurer une nouvelle fonction Lambda à l'aide de vos plans Lambda ou en choisir une existante. Votre fonction Lambda doit contenir le modèle d'état requis par Firehose. Pour plus d'informations, consultez [Transformation des données avec Amazon Data Firehose](#).

2. Dans la section Conversion du format d'enregistrement, saisissez des valeurs pour le champ suivant :

Conversion du format d'enregistrement

Pour créer un flux Firehose qui ne convertit pas le format des enregistrements de données entrants, choisissez Disabled.

Pour convertir le format des enregistrements entrants, sélectionnez Enabled (Activé), puis indiquez le format de sortie voulu. Vous devez spécifier une AWS Glue table contenant le schéma que vous souhaitez que Firehose utilise pour convertir votre format d'enregistrement. Pour plus d'informations, consultez [Conversion du format d'enregistrement](#).

Pour un exemple de configuration de la conversion de format d'enregistrement avec AWS CloudFormation, voir [AWS: : KinesisFirehose : : DeliveryStream](#).

- Si vous choisissez Managed Service for Apache Flink ou Direct PUT comme source pour votre flux Firehose, dans la section Paramètres de la source :

1. Sous Transformer les enregistrements, sélectionnez l'une des options suivantes :
 - a. Si votre destination est Amazon S3 ou Splunk, dans la section Décompresser les enregistrements source Amazon CloudWatch Logs, sélectionnez Activer la décompression.
 - b. Dans la section Transformer les enregistrements source avec AWS Lambda, fournissez des valeurs pour le champ suivant :

Transformation de données

Pour créer un flux Firehose qui ne transforme pas les données entrantes, ne cochez pas la case Activer la transformation des données.

Pour spécifier une fonction Lambda à invoquer par Amazon Data Firehose et à utiliser pour transformer les données entrantes avant de les transmettre, cochez la case Activer la transformation des données. Vous pouvez configurer une nouvelle fonction Lambda à l'aide de vos plans Lambda ou en choisir une existante. Votre fonction Lambda doit contenir le modèle de statut requis par Amazon Data Firehose. Pour plus d'informations, consultez [Transformation des données avec Amazon Data Firehose](#).

2. Dans la section Conversion du format d'enregistrement, saisissez des valeurs pour le champ suivant :

Conversion du format d'enregistrement

Pour créer un flux Firehose qui ne convertit pas le format des enregistrements de données entrants, choisissez Disabled.

Pour convertir le format des enregistrements entrants, sélectionnez Enabled (Activé), puis indiquez le format de sortie voulu. Vous devez spécifier une AWS Glue table contenant le schéma que vous souhaitez qu'Amazon Data Firehose utilise pour convertir votre format d'enregistrement. Pour plus d'informations, consultez [Conversion du format d'enregistrement](#).

Pour un exemple de configuration de la conversion de format d'enregistrement avec AWS CloudFormation, voir [AWS: : KinesisFirehose : : DeliveryStream](#).

Configuration des paramètres de destination

Cette rubrique décrit les paramètres de destination de votre stream Firehose en fonction de la destination que vous sélectionnez. Pour plus d'informations sur les conseils de mise en mémoire tampon, consultez [Comprendre les conseils de mise en mémoire tampon](#).

Rubriques

- [Configurer les paramètres de destination pour Amazon S3](#)
- [Configuration des paramètres de destination pour Amazon Redshift](#)
- [Configurer les paramètres de destination pour le OpenSearch service](#)
- [Configuration des paramètres de destination pour OpenSearch Serverless](#)
- [Configurer les paramètres de destination pour le point de terminaison HTTP](#)
- [Configurer les paramètres de destination pour Datadog](#)
- [Configurer les paramètres de destination pour Honeycomb](#)
- [Configuration des paramètres de destination pour Coralogix](#)
- [Configuration des paramètres de destination pour Dynatrace](#)
- [Configurer les paramètres de destination pour LogicMonitor](#)
- [Configurer les paramètres de destination pour Logz.io](#)
- [Configurer les paramètres de destination pour MongoDB Cloud](#)
- [Configurer les paramètres de destination pour New Relic](#)
- [Configurer les paramètres de destination pour Snowflake](#)
- [Configurer les paramètres de destination pour Splunk](#)
- [Configurer les paramètres de destination pour Splunk Observability Cloud](#)
- [Configuration des paramètres de destination pour Sumo Logic](#)
- [Configuration des paramètres de destination pour Elastic](#)

Configurer les paramètres de destination pour Amazon S3

Vous devez spécifier les paramètres suivants afin d'utiliser Amazon S3 comme destination pour votre flux Firehose.

- Saisissez des valeurs pour les champs suivants.

Compartiment S3

Parmi les compartiments S3 que vous possédez, choisissez celui dans lequel les données de streaming doivent être livrées. Vous pouvez créer un compartiment S3 ou en choisir un existant.

Nouveau délimiteur de ligne

Vous pouvez configurer votre flux Firehose pour ajouter un nouveau délimiteur de ligne entre les enregistrements des objets envoyés à Amazon S3. Pour ce faire, choisissez **Activé**. Pour ne pas ajouter de nouveau délimiteur de ligne entre les enregistrements des objets diffusés à Amazon S3, choisissez **Désactivé**. Si vous envisagez d'utiliser Athena pour interroger des objets S3 avec des enregistrements agrégés, activez cette option.

Partitionnement dynamique

Choisissez **Activé** pour activer et configurer le partitionnement dynamique.

Déagrégation de plusieurs enregistrements

Il s'agit du processus qui consiste à analyser les enregistrements du flux Firehose et à les séparer en fonction du JSON valide ou du nouveau délimiteur de ligne spécifié.

Si vous regroupez plusieurs événements, journaux ou enregistrements en un seul PutRecord appel d' PutRecordBatch API, vous pouvez toujours activer et configurer le partitionnement dynamique. Avec les données agrégées, lorsque vous activez le partitionnement dynamique, Amazon Data Firehose analyse les enregistrements et recherche plusieurs objets JSON valides dans chaque appel d'API. Lorsque le flux Firehose est configuré avec Kinesis Data Stream comme source, vous pouvez également utiliser l'agrégation intégrée dans la Kinesis Producer Library (KPL). La fonctionnalité de partition des données est exécutée après la désagrégation des données. Par conséquent, chaque enregistrement de chaque appel d'API peut être transmis à différents préfixes Amazon S3. Vous pouvez également tirer parti de l'intégration de la fonction Lambda pour effectuer toute autre désagrégation ou toute autre transformation avant la fonctionnalité de partitionnement des données.

Important

Si vos données sont agrégées, le partitionnement dynamique ne peut être appliqué qu'après la désagrégation des données. Ainsi, si vous activez le partitionnement

dynamique de vos données agrégées, vous devez sélectionner **Activé** pour activer la désagrégation multi-enregistrements.

Firehose Stream exécute les étapes de traitement suivantes dans l'ordre suivant : désagrégation KPL (protobuf), désagrégation JSON ou délimiteur, traitement Lambda, partitionnement des données, conversion des formats de données et livraison Amazon S3.

Type de désagrégation à enregistrements multiples

Si vous avez activé la désagrégation multi-enregistrements, vous devez spécifier la méthode que Firehose utilisera pour désagréger vos données. Utilisez le menu déroulant pour choisir JSON ou Délimité.

Analyse en ligne

Il s'agit de l'un des mécanismes pris en charge pour partitionner dynamiquement vos données destinées à Amazon S3. Pour utiliser l'analyse en ligne pour le partitionnement dynamique pour vos données, vous devez spécifier les paramètres d'enregistrement de données à utiliser comme clés de partitionnement et fournir une valeur pour chaque clé de partitionnement spécifiée. Choisissez **Activé** pour activer et configurer le l'analyse en ligne.

Important

Si vous avez spécifié une fonction AWS Lambda dans les étapes ci-dessus pour transformer vos enregistrements source, vous pouvez utiliser cette fonction pour partitionner dynamiquement vos données liées à S3 et vous pouvez toujours créer vos clés de partitionnement grâce à l'analyse en ligne. Avec le partitionnement dynamique, vous pouvez utiliser l'analyse en ligne ou votre fonction AWS Lambda pour créer vos clés de partitionnement. Vous pouvez également utiliser à la fois l'analyse en ligne et votre fonction AWS Lambda pour créer vos clés de partitionnement.

Clés de partitionnement dynamique

Vous pouvez utiliser les champs **Clé** et **Valeur** pour spécifier les paramètres d'enregistrement de données à utiliser comme clés de partitionnement dynamique et les requêtes JQ pour générer des valeurs de clé de partitionnement dynamique. Firehose ne prend en charge que

jq 1.6. Vous pouvez spécifier jusqu'à 50 clés de partitionnement dynamique. Vous devez entrer des expressions jq valides pour les valeurs de vos clés de partitionnement dynamique afin de configurer correctement le partitionnement dynamique pour votre flux Firehose.

Préfixe du compartiment S3

Lorsque vous activez et configurez le partitionnement dynamique, vous devez spécifier les préfixes de compartiment S3 auxquels Amazon Data Firehose doit fournir les données partitionnées.

Pour que le partitionnement dynamique soit correctement configuré, le nombre de préfixes du compartiment S3 doit être identique au nombre de clés de partitionnement spécifiées.

Vous pouvez partitionner vos données sources à l'aide de l'analyse en ligne ou de la fonction Lambda AWS que vous avez spécifiée. Si vous avez spécifié une fonction AWS Lambda pour créer des clés de partitionnement pour vos données sources, vous devez saisir manuellement les valeurs du préfixe du compartiment S3 en utilisant le format suivant : « Lambda:KeyID ». partitionKeyFrom Si vous utilisez l'analyse en ligne pour spécifier les clés de partitionnement de vos données sources, vous pouvez soit saisir manuellement les valeurs d'aperçu du compartiment S3 au format suivant : « partitionKeyFrom query:keyID », soit choisir le bouton Appliquer les clés de partitionnement dynamique pour utiliser vos paires clé/valeur de partitionnement dynamique afin de générer automatiquement les préfixes de vos compartiments S3. Lorsque vous partitionnez vos données à l'aide de l'analyse en ligne ou de AWS Lambda, vous pouvez également utiliser les formes d'expression suivantes dans le préfixe de votre compartiment S3 : ! {namespace:value}, où l'espace de noms peut être Query ou Lambda. partitionKeyFrom partitionKeyFrom

Bucket S3 et fuseau horaire du préfixe de sortie d'erreur S3

Choisissez le fuseau horaire que vous souhaitez utiliser pour la date et l'heure dans les [préfixes personnalisés pour les objets Amazon Simple Storage Service](#). Par défaut, Firehose ajoute un préfixe horaire en UTC. Vous pouvez modifier le fuseau horaire utilisé dans les préfixes S3 si vous souhaitez utiliser un autre fuseau horaire.

Conseils de mise en mémoire tampon

Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Compression S3

Choisissez la compression de données GZIP, Snappy, Zip ou Snappy compatible avec Hadoop, ou aucune compression de données. La compression Snappy, Zip et compatible avec Hadoop n'est pas disponible pour les streams Firehose avec Amazon Redshift comme destination.

Format d'extension de fichier S3 (facultatif)

Spécifiez un format d'extension de fichier pour les objets livrés au compartiment de destination Amazon S3. Si vous activez cette fonctionnalité, l'extension de fichier spécifiée remplacera les extensions de fichier par défaut ajoutées par les fonctionnalités de conversion de format de données ou de compression S3 telles que `.parquet` ou `.gz`. Assurez-vous d'avoir configuré la bonne extension de fichier lorsque vous utilisez cette fonctionnalité avec la conversion de format de données ou la compression S3. L'extension de fichier doit commencer par un point (.) et peut contenir les caractères autorisés : 0-9a-z ! -_.*' (). L'extension de fichier ne peut pas dépasser 128 caractères.

Chiffrement S3

Firehose prend en charge le chiffrement côté serveur Amazon S3 avec AWS Key Management Service (SSE-KMS) pour chiffrer les données livrées dans Amazon S3. Vous pouvez choisir d'utiliser le type de chiffrement par défaut spécifié dans le compartiment S3 de destination ou de chiffrer avec une clé de la liste des AWS KMS clés que vous possédez. Si vous chiffrez les données à l'aide de AWS KMS clés, vous pouvez utiliser la clé AWS gérée par défaut (`aws/s3`) ou une clé gérée par le client. Pour plus d'informations, voir [Protection des données à l'aide du chiffrement côté serveur avec des clés AWS gérées par KMS \(SSE-KMS\)](#).

Configuration des paramètres de destination pour Amazon Redshift

Cette section décrit les paramètres d'utilisation d'Amazon Redshift comme destination de votre stream Firehose.

Choisissez l'une des procédures suivantes selon que vous disposez d'un cluster Amazon Redshift provisionné ou d'un groupe de travail Amazon Redshift sans serveur.

- [Cluster provisionné Amazon Redshift](#)
- [Configuration des paramètres de destination pour le groupe de travail Amazon Redshift Serverless](#)

Cluster provisionné Amazon Redshift

Cette section décrit les paramètres d'utilisation du cluster provisionné Amazon Redshift comme destination de votre flux Firehose.

- Saisissez des valeurs pour les champs suivants :

Cluster

Cluster Amazon Redshift dans lequel vos données de compartiment S3 sont copiées. Configurez le cluster Amazon Redshift pour qu'il soit accessible au public et débloquent les adresses IP Amazon Data Firehose. Pour plus d'informations, consultez [Accorder à Amazon Data Firehose l'accès à une destination Amazon Redshift](#).

Authentification

Vous pouvez choisir de saisir directement le nom d'utilisateur/mot de passe ou de récupérer le code secret AWS Secrets Manager pour accéder au cluster Amazon Redshift.

- Nom utilisateur

Spécifiez un utilisateur Amazon Redshift autorisé à accéder au cluster Amazon Redshift. Cet utilisateur doit avoir l'autorisation INSERT d'Amazon Redshift pour copier des données du compartiment S3 vers le cluster Amazon Redshift.

- Mot de passe :

Spécifiez le mot de passe de l'utilisateur autorisé à accéder au cluster.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant les informations d'identification du cluster Amazon Redshift. Si votre code secret ne figure pas dans la liste déroulante, créez-en un AWS Secrets Manager pour vos informations d'identification Amazon Redshift. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Database (Base de données)

La base de données Amazon Redshift dans laquelle les données sont copiées.

Tableau

~~La table Amazon Redshift dans laquelle les données sont copiées.~~

Colonnes

(Facultatif) Colonnes spécifiques de la table dans laquelle les données sont copiées. Utilisez cette option si le nombre de colonnes défini dans vos objets Amazon S3 est inférieur au nombre de colonnes de la table Amazon Redshift.

Destination S3 intermédiaire

Firehose envoie d'abord vos données dans votre compartiment S3, puis émet une commande Amazon COPY Redshift pour charger les données dans votre cluster Amazon Redshift. Spécifiez un compartiment S3 que vous possédez dans lequel les données de diffusion en continu doivent être diffusées. Créez un nouveau compartiment S3 ou choisissez un compartiment existant dont vous êtes propriétaire.

Firehose ne supprime pas les données de votre compartiment S3 après les avoir chargées dans votre cluster Amazon Redshift. Vous pouvez gérer les données de votre compartiment S3 à l'aide d'une configuration du cycle de vie. Pour plus d'informations, consultez [Gestion du cycle de vie des objets](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

Préfixe du compartiment S3 intermédiaire

(Facultatif) Pour utiliser le préfixe par défaut des objets Amazon S3, laissez cette option vide. Firehose utilise automatiquement un préfixe au format « YYYY/MM/dd/HH » UTC pour les objets Amazon S3 livrés. Vous pouvez ajouter le début de ce préfixe. Pour plus d'informations, consultez [Configurer le format du nom d'objet Amazon S3](#).

COPY options (Options COPY)

Paramètres que vous pouvez spécifier dans la commande COPY d'Amazon Redshift. Ils peuvent être nécessaires pour votre configuration. Par exemple, « GZIP » est obligatoire si la compression des données Amazon S3 est activée. « REGION » est obligatoire si votre compartiment S3 ne se trouve pas dans la même AWS région que votre cluster Amazon Redshift. Pour plus d'informations, veuillez consulter la rubrique [COPY](#) dans le Guide du développeur de la base de données Amazon Redshift.

COPY command (Commande COPY)

La commande COPY Amazon Redshift. Pour plus d'informations, veuillez consulter la rubrique [COPY](#) dans le Guide du développeur de la base de données Amazon Redshift.

Retry duration (Durée de la nouvelle tentative)

Durée (0 à 7 200 secondes) pendant laquelle Firehose doit réessayer si les données de votre cluster COPY Amazon Redshift échouent. Firehose réessaie toutes les 5 minutes jusqu'à la fin de la période d'essai. Si vous définissez la durée de la nouvelle tentative sur 0 (zéro) seconde, Firehose ne réessaie pas en cas d'échec d'une commande. COPY

Conseils de mise en mémoire tampon

Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Compression S3

Choisissez la compression de données GZIP, Snappy, Zip ou Snappy compatible avec Hadoop, ou aucune compression de données. La compression Snappy, Zip et compatible avec Hadoop n'est pas disponible pour les streams Firehose avec Amazon Redshift comme destination.

Format d'extension de fichier S3 (facultatif)

Format d'extension de fichier S3 (facultatif) : spécifiez un format d'extension de fichier pour les objets livrés au compartiment de destination Amazon S3. Si vous activez cette fonctionnalité, l'extension de fichier spécifiée remplacera les extensions de fichier par défaut ajoutées par les fonctionnalités de conversion de format de données ou de compression S3 telles que `.parquet` ou `.gz`. Assurez-vous d'avoir configuré la bonne extension de fichier lorsque vous utilisez cette fonctionnalité avec la conversion de format de données ou la compression S3. L'extension de fichier doit commencer par un point (.) et peut contenir les caractères autorisés : `0-9a-z ! - _ . * ' ()`. L'extension de fichier ne peut pas dépasser 128 caractères.

Chiffrement S3

Firehose prend en charge le chiffrement côté serveur Amazon S3 avec AWS Key Management Service (SSE-KMS) pour chiffrer les données livrées dans Amazon S3. Vous pouvez choisir d'utiliser le type de chiffrement par défaut spécifié dans le compartiment S3 de destination ou de chiffrer avec une clé de la liste des AWS KMS clés que vous possédez. Si vous chiffrez les données à l'aide de AWS KMS clés, vous pouvez utiliser la clé AWS gérée par défaut (`aws/s3`) ou une clé gérée par le client. Pour plus d'informations, voir [Protection](#)

[des données à l'aide du chiffrement côté serveur avec des clés AWS gérées par KMS \(SSE-KMS\).](#)

Configuration des paramètres de destination pour le groupe de travail Amazon Redshift Serverless

Cette section décrit les paramètres d'utilisation du groupe de travail Amazon Redshift Serverless comme destination de votre stream Firehose.

- Saisissez des valeurs pour les champs suivants :

Nom du groupe de travail

Le groupe de travail Amazon Redshift sans serveur dans lequel vos données de compartiment S3 sont copiées. Configurez le groupe de travail Amazon Redshift Serverless pour qu'il soit accessible au public et débloquez les adresses IP Firehose. Pour de plus amples informations, consultez la section de l'instance [Se connecter à Amazon Redshift sans serveur](#) lorsqu'il est publiquement accessible dans [Connexion à Amazon Redshift sans serveur](#) et également [Accorder à Amazon Data Firehose l'accès à une destination Amazon Redshift](#).

Authentification

Vous pouvez choisir de saisir directement le nom d'utilisateur/mot de passe ou de récupérer le code secret pour accéder AWS Secrets Manager au groupe de travail Amazon Redshift Serverless.

- Nom utilisateur

Spécifiez un utilisateur Amazon Redshift autorisé à accéder au groupe de travail Amazon Redshift Serverless. Cet utilisateur doit avoir l'autorisation INSERT d'Amazon Redshift pour copier des données du compartiment S3 vers le groupe de travail Amazon Redshift sans serveur.

- Mot de passe :

Spécifiez le mot de passe de l'utilisateur autorisé à accéder au groupe de travail Amazon Redshift Serverless.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant les informations d'identification du groupe de travail Amazon Redshift Serverless. Si votre code secret ne figure pas dans la liste déroulante, créez-en un AWS Secrets Manager pour vos informations d'identification Amazon Redshift. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Database (Base de données)

La base de données Amazon Redshift dans laquelle les données sont copiées.

Tableau

La table Amazon Redshift dans laquelle les données sont copiées.

Colonnes

(Facultatif) Colonnes spécifiques de la table dans laquelle les données sont copiées. Utilisez cette option si le nombre de colonnes défini dans vos objets Amazon S3 est inférieur au nombre de colonnes de la table Amazon Redshift.

Destination S3 intermédiaire

Amazon Data Firehose envoie d'abord vos données dans votre compartiment S3, puis émet une commande Amazon COPY Redshift pour charger les données dans votre groupe de travail Amazon Redshift Serverless. Spécifiez un compartiment S3 que vous possédez dans lequel les données de diffusion en continu doivent être diffusées. Créez un nouveau compartiment S3 ou choisissez un compartiment existant dont vous êtes propriétaire.

Firehose ne supprime pas les données de votre compartiment S3 après les avoir chargées dans votre groupe de travail Amazon Redshift Serverless. Vous pouvez gérer les données de votre compartiment S3 à l'aide d'une configuration du cycle de vie. Pour plus d'informations, consultez [Gestion du cycle de vie des objets](#) dans le Guide de l'utilisateur Amazon Simple Storage Service.

Préfixe du compartiment S3 intermédiaire

(Facultatif) Pour utiliser le préfixe par défaut des objets Amazon S3, laissez cette option vide. Firehose utilise automatiquement un préfixe au format « YYYY/MM/dd/HH » UTC pour les objets Amazon S3 livrés. Vous pouvez ajouter le début de ce préfixe. Pour plus d'informations, consultez [Configurer le format du nom d'objet Amazon S3](#).

COPY options (Options COPY)

Paramètres que vous pouvez spécifier dans la commande COPY d'Amazon Redshift. Ils peuvent être nécessaires pour votre configuration. Par exemple, « GZIP » est obligatoire si la compression des données Amazon S3 est activée. « REGION » est obligatoire si votre compartiment S3 ne se trouve pas dans la même AWS région que votre groupe de travail Amazon Redshift Serverless. Pour plus d'informations, veuillez consulter la rubrique [COPY](#) dans le Guide du développeur de la base de données Amazon Redshift.

COPY command (Commande COPY)

La commande COPY Amazon Redshift. Pour plus d'informations, veuillez consulter la rubrique [COPY](#) dans le Guide du développeur de la base de données Amazon Redshift.

Retry duration (Durée de la nouvelle tentative)

Durée (0 à 7 200 secondes) pendant laquelle Firehose doit réessayer si les données de votre groupe de travail COPY Amazon Redshift Serverless échouent. Firehose réessaie toutes les 5 minutes jusqu'à la fin de la période d'essai. Si vous définissez la durée de la nouvelle tentative sur 0 (zéro) seconde, Firehose ne réessaie pas en cas d'échec d'une commande COPY.

Conseils de mise en mémoire tampon

Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Compression S3

Choisissez la compression de données GZIP, Snappy, Zip ou Snappy compatible avec Hadoop, ou aucune compression de données. La compression Snappy, Zip et compatible avec Hadoop n'est pas disponible pour les streams Firehose avec Amazon Redshift comme destination.

Format d'extension de fichier S3 (facultatif)

Format d'extension de fichier S3 (facultatif) : spécifiez un format d'extension de fichier pour les objets livrés au compartiment de destination Amazon S3. Si vous activez cette fonctionnalité, l'extension de fichier spécifiée remplacera les extensions de fichier par défaut ajoutées par les fonctionnalités de conversion de format de données ou de compression S3 telles que .parquet ou .gz. Assurez-vous d'avoir configuré la bonne extension de fichier

lorsque vous utilisez cette fonctionnalité avec la conversion de format de données ou la compression S3. L'extension de fichier doit commencer par un point (.) et peut contenir les caractères autorisés : 0-9a-z ! -_.*' (). L'extension de fichier ne peut pas dépasser 128 caractères.

Chiffrement S3

Firehose prend en charge le chiffrement côté serveur Amazon S3 avec AWS Key Management Service (SSE-KMS) pour chiffrer les données livrées dans Amazon S3. Vous pouvez choisir d'utiliser le type de chiffrement par défaut spécifié dans le compartiment S3 de destination ou de chiffrer avec une clé de la liste des AWS KMS clés que vous possédez. Si vous chiffrez les données à l'aide de AWS KMS clés, vous pouvez utiliser la clé AWS gérée par défaut (aws/s3) ou une clé gérée par le client. Pour plus d'informations, voir [Protection des données à l'aide du chiffrement côté serveur avec des clés AWS gérées par KMS \(SSE-KMS\)](#).

Configurer les paramètres de destination pour le OpenSearch service

Cette section décrit les options d'utilisation OpenSearch du Service pour votre destination.

- Saisissez des valeurs pour les champs suivants :

OpenSearch Domaine de service

Le domaine de OpenSearch service vers lequel vos données sont livrées.

Index

Le nom OpenSearch de l'index de service à utiliser lors de l'indexation des données dans votre cluster OpenSearch de services.

Index rotation

Choisissez si et à quelle fréquence l'index OpenSearch de service doit être pivoté. Si la rotation de l'index est activée, Amazon Data Firehose ajoute l'horodatage correspondant au nom d'index spécifié et effectue une rotation. Pour plus d'informations, consultez [Configurer la rotation de l'index pour le OpenSearch service](#).

Type

Le nom du type de OpenSearch service à utiliser lors de l'indexation des données dans votre cluster OpenSearch de services. Pour Elasticsearch 7.x et OpenSearch 1.x, il ne peut y avoir qu'un seul type par index. Si vous essayez de spécifier un nouveau type pour un index existant qui possède déjà un autre type, Firehose renvoie une erreur lors de l'exécution.

Pour Elasticsearch 7.x, laissez ce champ vide.

Retry duration (Durée de la nouvelle tentative)

Durée pendant laquelle Firehose doit réessayer en cas d'échec d'une demande d'index. OpenSearch Dans ce cas, Firehose réessaie toutes les 5 minutes jusqu'à expiration du délai de réessai. Pour la durée de la nouvelle tentative, vous pouvez définir une valeur comprise entre 0 et 7 200 secondes.

Une fois le délai de nouvelle tentative expiré, Firehose fournit les données à Dead Letter Queue (DLQ), un compartiment d'erreur S3 configuré. Pour les données transmises à DLQ, vous devez rediriger les données du compartiment d'erreur S3 configuré vers OpenSearch leur destination.

Si vous souhaitez empêcher Firehose Stream de fournir des données à DLQ en raison d'une interruption de service ou de la maintenance de OpenSearch clusters, vous pouvez configurer la durée des nouvelles tentatives à une valeur plus élevée en secondes. [Vous pouvez augmenter la durée de la nouvelle tentative ci-dessus à 7200 secondes en contactant le AWS support.](#)

Type de DocumentID

Indique la méthode de configuration de l'ID du document. Les méthodes prises en charge sont l'ID de document généré par Firehose et l'ID de document généré par le OpenSearch Service. L'ID de document généré par Firehose est l'option par défaut lorsque la valeur de l'ID de document n'est pas définie. OpenSearch L'ID de document généré par le service est l'option recommandée car elle prend en charge les opérations nécessitant beaucoup d'écriture, notamment l'analyse des journaux et l'observabilité, consommant ainsi moins de ressources CPU dans le domaine des OpenSearch services et améliorant ainsi les performances.

Connectivité du VPC de destination

Si votre domaine OpenSearch de service se trouve dans un VPC privé, utilisez cette section pour spécifier ce VPC. Spécifiez également les sous-réseaux et sous-groupes que vous souhaitez qu'Amazon Data Firehose utilise lorsqu'il envoie des données à votre domaine de service. OpenSearch Vous pouvez utiliser les mêmes groupes de sécurité que ceux utilisés par le domaine de OpenSearch service. Si vous spécifiez différents groupes de sécurité, assurez-vous qu'ils autorisent le trafic HTTPS sortant vers le groupe de sécurité du domaine de OpenSearch service. Assurez-vous également que le groupe de sécurité du domaine de OpenSearch service autorise le trafic HTTPS provenant des groupes de sécurité que vous avez spécifiés lors de la configuration de votre flux Firehose. Si vous utilisez le même groupe de sécurité pour votre stream Firehose et pour le domaine de OpenSearch service, assurez-vous que la règle entrante du groupe de sécurité autorise le trafic HTTPS. Pour plus d'informations sur les règles des groupes de sécurité, consultez [Règles des groupes de sécurité](#) dans la documentation Amazon VPC.

Important

Lorsque vous spécifiez des sous-réseaux pour fournir des données à la destination dans un VPC privé, assurez-vous de disposer d'un nombre suffisant d'adresses IP libres dans les sous-réseaux sélectionnés. Si aucune adresse IP gratuite n'est disponible dans un sous-réseau spécifié, Firehose ne peut pas créer ou ajouter d'ENI pour la livraison des données dans le VPC privé, et la livraison sera dégradée ou échouera.

Indices de mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configuration des paramètres de destination pour OpenSearch Serverless

Cette section décrit les options permettant d'utiliser OpenSearch Serverless pour votre destination.

- Saisissez des valeurs pour les champs suivants :

OpenSearch Collecte sans serveur

Point de terminaison d'un groupe d'index OpenSearch sans serveur auquel vos données sont livrées.

Index

Le nom de l'index OpenSearch Serverless à utiliser lors de l'indexation des données dans votre collection OpenSearch Serverless.

Connectivité du VPC de destination

Si votre collection OpenSearch Serverless se trouve dans un VPC privé, utilisez cette section pour spécifier ce VPC. Spécifiez également les sous-réseaux et sous-groupes que vous souhaitez qu'Amazon Data Firehose utilise lorsqu'il envoie des données à votre collection Serverless. OpenSearch

Important

Lorsque vous spécifiez des sous-réseaux pour fournir des données à la destination dans un VPC privé, assurez-vous de disposer d'un nombre suffisant d'adresses IP libres dans les sous-réseaux sélectionnés. Si aucune adresse IP gratuite n'est disponible dans un sous-réseau spécifié, Firehose ne peut pas créer ou ajouter d'ENI pour la livraison des données dans le VPC privé, et la livraison sera dégradée ou échouera.

Retry duration (Durée de la nouvelle tentative)

Durée pendant laquelle Firehose doit réessayer en cas d'échec d'une demande d'index adressée à OpenSearch Serverless. Dans ce cas, Firehose réessaie toutes les 5 minutes jusqu'à expiration du délai de réessai. Pour la durée de la nouvelle tentative, vous pouvez définir une valeur comprise entre 0 et 7 200 secondes.

Une fois le délai de nouvelle tentative expiré, Firehose fournit les données à Dead Letter Queue (DLQ), un compartiment d'erreur S3 configuré. Pour les données transmises à DLQ, vous devez rediriger les données du compartiment d'erreur S3 configuré vers une destination OpenSearch sans serveur.

Si vous souhaitez empêcher Firehose Stream de fournir des données à DLQ en raison d'une interruption de service ou de la maintenance de clusters OpenSearch sans serveur, vous pouvez configurer la durée des nouvelles tentatives à une valeur plus élevée en secondes. [Vous pouvez augmenter la durée de la nouvelle tentative ci-dessus à 7200 secondes en contactant le AWS support.](#)

Indices de mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configurer les paramètres de destination pour le point de terminaison HTTP

Cette section décrit les options dont vous disposez pour utiliser le point de terminaison HTTP comme destination.

Important

Si vous choisissez un point de terminaison HTTP comme destination, consultez et suivez les instructions figurant dans [Annexe : spécifications des demandes et réponses de livraison de points de terminaison HTTP](#).

- Fournissez des valeurs pour les champs suivants :

Nom du point de terminaison HTTP : facultatif

Spécifiez un nom convivial pour le point de terminaison HTTP. Par exemple, My HTTP Endpoint Destination.

URL du point de terminaison HTTP

Spécifiez l'URL du point de terminaison HTTP au format suivant : `https://xyz.httpendpoint.com`. L'URL doit être une URL HTTPS.

Authentification

Vous pouvez choisir de saisir directement la clé d'accès ou de récupérer le secret AWS Secrets Manager pour accéder au point de terminaison HTTP.

- (Facultatif) Clé d'accès

Contactez le propriétaire du terminal si vous avez besoin d'obtenir la clé d'accès pour permettre la livraison de données à son point de terminaison depuis Firehose.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant la clé d'accès pour le point de terminaison HTTP. Si votre code secret ne figure pas dans la liste déroulante, créez-en un AWS Secrets Manager pour la clé d'accès. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP sélectionné.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Important

Pour les destinations du point de terminaison HTTP, si 413 codes de réponse proviennent du point de terminaison de destination dans CloudWatch Logs, réduisez la taille de l'indicateur de mise en mémoire tampon sur votre flux Firehose et réessayez.

Configurer les paramètres de destination pour Datadog

Cette section décrit les options dont vous disposez pour utiliser Datadog comme destination. Pour plus d'informations sur Datadog, consultez https://docs.datadoghq.com/integrations/amazon_web_services/.

- Entrez des valeurs pour les champs suivants.

URL du point de terminaison HTTP

Choisissez l'endroit où vous souhaitez envoyer les données à partir de l'une des options suivantes du menu déroulant.

- Journaux Datadog – US1
- Logs Datadog - US3
- Journaux Datadog – US5
- Logs Datadog - AP1

- Journaux Datadog – EU
- Journaux Datadog – GOV
- Métriques Datadog – US
- Métriques Datadog - US5
- Métriques Datadog - AP1
- Métriques Datadog – EU
- Configurations Datadog - US1
- Configurations Datadog - US3
- Configurations Datadog - US5
- Configurations Datadog - AP1
- Configurations Datadog - UE
- Configurations Datadog - US GOV

Authentification

Vous pouvez choisir de saisir directement la clé d'API ou de récupérer le code secret AWS Secrets Manager pour accéder à Datadog.

- Clé API

Contactez Datadog pour obtenir la clé d'API dont vous avez besoin pour permettre la livraison de données à ce point de terminaison depuis Firehose.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant la clé d'API de Datadog. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP sélectionné.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configurer les paramètres de destination pour Honeycomb

Cette section décrit les options dont vous disposez pour utiliser Honeycomb comme destination. Pour plus d'informations sur Honeycomb, consultez <https://docs.honeycomb.io/getting-data-in-aws-cloudwatch-metrics/metrics/>.

- Fournissez des valeurs pour les champs suivants :

Point de terminaison Kinesis Honeycomb

Spécifiez l'URL du point de terminaison HTTP au format suivant : `https://api.honeycomb.io/1/kinesis_events/{{dataset}}`

Authentification

Vous pouvez choisir de saisir directement la clé API ou de récupérer le secret AWS Secrets Manager pour accéder à Honeycomb.

- Clé API

Contactez Honeycomb pour obtenir la clé d'API dont vous avez besoin pour permettre la livraison de données à ce point de terminaison depuis Firehose.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant la clé d'API pour Honeycomb. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP pour activer le codage du contenu de votre demande. Il s'agit là de l'option recommandée pour la destination Honeycomb.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP sélectionné.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère

qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configuration des paramètres de destination pour Coralogix

Cette section décrit les options dont vous disposez pour utiliser Coralogix comme destination. Pour plus d'informations sur Coralogix, consultez <https://coralogix.com/integrations/aws-firehose>.

- Fournissez des valeurs pour les champs suivants :

URL du point de terminaison HTTP

Choisissez l'URL du point de terminaison HTTP parmi les options suivantes du menu déroulant :

- Coralogix – États-Unis

- Coralogix – SINGAPOUR
- Coralogix – IRLANDE
- Coralogix – INDE
- Coralogix – STOCKHOLM

Authentification

Vous pouvez choisir de saisir directement la clé privée ou de récupérer le secret AWS Secrets Manager pour accéder à Coralogix.

- Clé privée

Contactez Coralogix pour obtenir la clé privée dont vous avez besoin pour permettre la livraison de données à ce point de terminaison depuis Firehose.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant la clé privée de Coralogix. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP pour activer le codage du contenu de votre demande. Il s'agit là de l'option recommandée pour la destination Coralogix.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP sélectionné.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

- `applicationName` : l'environnement dans lequel vous exécutez Data Firehose
- `subsystemName` : le nom de l'intégration Data Firehose
- `ComputerName` : le nom du stream Firehose utilisé

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie en fonction du fournisseur de services.

Configuration des paramètres de destination pour Dynatrace

Cette section décrit les options dont vous disposez pour utiliser Dynatrace comme destination. Pour plus d'informations, consultez <https://www.dynatrace.com/support/help/technology-support/cloud-platforms/amazon-web-services/cloudwatch-metric-streams/integrations/>.

- Choisissez des options pour utiliser Dynatrace comme destination pour votre stream Firehose.

Type d'ingestion

Choisissez si vous souhaitez fournir des métriques ou des journaux (par défaut) dans Dynatrace pour une analyse et un traitement plus approfondis.

URL du point de terminaison HTTP

Choisissez l'URL du point de terminaison HTTP (Dynatrace US, Dynatrace EU ou Dynatrace Global) dans le menu déroulant.

Authentification

Vous pouvez choisir de saisir directement le jeton d'API ou de récupérer le secret AWS Secrets Manager pour accéder à Dynatrace.

- Jeton d'API

Générez le jeton d'API Dynatrace dont vous avez besoin pour permettre la livraison de données à ce point de terminaison depuis Firehose. Pour plus d'informations, consultez [l'API Dynatrace - Tokens](#) et authentification.

- Secret

Sélectionnez un secret AWS Secrets Manager qui contient le jeton d'API pour Dynatrace. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

URL de l'API

Fournissez l'URL de l'API de votre environnement Dynatrace.

Encodage de contenu

Choisissez si vous souhaitez activer le codage du contenu pour compresser le corps de la demande. Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Lorsque cette option est activée, le contenu est compressé au format GZIP.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP sélectionné.

Après avoir envoyé des données, Firehose attend d'abord un accusé de réception du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Ensuite, Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois que Firehose envoie des données au point de terminaison HTTP, que ce soit lors de la tentative initiale ou après une nouvelle tentative, il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne voulez pas que Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. Les indications relatives à la mémoire tampon incluent la taille de la mémoire tampon et l'intervalle de vos flux. La taille de mémoire tampon recommandée pour la destination varie en fonction du fournisseur de services.

Configurer les paramètres de destination pour LogicMonitor

Cette section décrit les options à utiliser LogicMonitor pour votre destination. Pour plus d'informations, consultez <https://www.logicmonitor.com>.

- Fournissez des valeurs pour les champs suivants :

URL du point de terminaison HTTP

Spécifiez l'URL du point de terminaison HTTP au format suivant.

```
https://ACCOUNT.logicmonitor.com
```

Authentification

Vous pouvez choisir de saisir directement la clé API ou de récupérer le secret AWS Secrets Manager pour y accéder LogicMonitor.

- Clé API

Contactez-nous LogicMonitor pour obtenir la clé d'API dont vous avez besoin pour activer la livraison de données à ce point de terminaison depuis Firehose.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant la clé d'API pour LogicMonitor. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP sélectionné.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configurer les paramètres de destination pour Logz.io

Cette section décrit les options dont vous disposez pour utiliser Logz.io comme destination. Pour plus d'informations, consultez <https://logz.io/>.

Note

Dans la région Europe (Milan), Logz.io n'est pas pris en charge en tant que destination Amazon Data Firehose.

- Fournissez des valeurs pour les champs suivants :

URL du point de terminaison HTTP

Spécifiez l'URL du point de terminaison HTTP au format suivant. L'URL doit être une HTTPS URL.

```
https://listener-aws-metrics-stream-<region>.logz.io/
```

Par exemple

```
https://listener-aws-metrics-stream-us.logz.io/
```

Authentification

Vous pouvez choisir de saisir directement le code d'expédition ou de récupérer le code secret AWS Secrets Manager pour accéder à Logz.io.

- Jeton d'expédition

Contactez Logz.io pour obtenir le jeton d'expédition dont vous avez besoin pour activer la livraison de données à ce point de terminaison depuis Firehose.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant le jeton d'expédition pour Logz.io. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données à Logz.io.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configurer les paramètres de destination pour MongoDB Cloud

Cette section décrit les options dont vous disposez pour utiliser MongoDB Cloud comme destination. Pour plus d'informations, consultez <https://www.mongodb.com>.

- Fournissez des valeurs pour les champs suivants :

URL du webhook MongoDB Realm

Spécifiez l'URL du point de terminaison HTTP au format suivant.

```
https://webhooks.mongodb-realm.com
```

L'URL doit être une HTTPS URL.

Authentification

Vous pouvez choisir de saisir directement la clé d'API ou de récupérer le secret AWS Secrets Manager pour accéder à MongoDB Cloud.

- Clé API

Contactez MongoDB Cloud pour obtenir la clé d'API dont vous avez besoin pour permettre la livraison de données à ce point de terminaison depuis Firehose.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant la clé d'API pour MongoDB Cloud. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au fournisseur tiers sélectionné.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de

réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Configurer les paramètres de destination pour New Relic

Cette section décrit les options dont vous disposez pour utiliser New Relic comme destination. Pour plus d'informations, consultez <https://newrelic.com>.

- Fournissez des valeurs pour les champs suivants :

URL du point de terminaison HTTP

Choisissez l'URL du point de terminaison HTTP parmi les options suivantes de la liste déroulante.

- Journaux New Relic – US
- Métriques New Relic – US
- Métriques New Relic – EU

Authentification

Vous pouvez choisir de saisir directement la clé API ou de récupérer le secret AWS Secrets Manager pour accéder à New Relic.

- Clé API

Entrez votre clé de licence, qui est une chaîne hexadécimale de 40 caractères, dans les paramètres de votre compte New Relic One. Vous avez besoin de cette clé d'API pour permettre la livraison de données à ce point de terminaison depuis Firehose.

- Secret

Sélectionnez un secret AWS Secrets Manager contenant la clé d'API de New Relic. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP New Relic.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configurer les paramètres de destination pour Snowflake

Cette section décrit les options d'utilisation de Snowflake pour votre destination.

Note

L'intégration de Firehose à Snowflake est disponible dans l'est des États-Unis (Virginie du Nord), dans l'ouest des États-Unis (Oregon), en Europe (Irlande), dans l'est des États-Unis (Ohio), en Asie-Pacifique (Tokyo), en Europe (Francfort), en Asie-Pacifique (Singapour), en Asie-Pacifique (Séoul) et en Asie-Pacifique (Sydney). Régions AWS

Réglages de connexion

- Fournissez des valeurs pour les champs suivants :

URL du compte Snowflake

Spécifiez l'URL d'un compte Snowflake. Par exemple : `xy12345.us-east-1.aws.snowflakecomputing.com`. Reportez-vous à la [documentation de Snowflake](#) pour savoir comment déterminer l'URL de votre compte. Notez que vous ne devez pas spécifier le numéro de port, alors que le protocole (`https://`) est facultatif.

Authentification

Vous pouvez choisir de saisir le nom d'utilisateur, la clé privée et le mot de passe manuellement ou de récupérer le code secret pour accéder à AWS Secrets Manager Snowflake.

- Login utilisateur

Spécifiez l'utilisateur Snowflake à utiliser pour le chargement des données. Assurez-vous que l'utilisateur a accès pour insérer des données dans la table Snowflake.

- Clé privée

Spécifiez la clé privée de l'utilisateur utilisée pour l'authentification avec Snowflake. Assurez-vous que le PKCS8 format de la clé privée est correct. N'incluez pas l'en-tête et le pied de page PEM dans cette clé. Si la clé est divisée sur plusieurs lignes, supprimez les sauts de ligne.

- Passphrase (Phrase secrète)

Spécifiez le mot de passe pour déchiffrer la clé privée cryptée. Vous pouvez laisser ce champ vide si la clé privée n'est pas cryptée. Pour plus d'informations, consultez la section [Utilisation de l'authentification par paire de clés et de la rotation des clés](#).

- Secret

Sélectionnez un secret AWS Secrets Manager contenant les informations d'identification de Snowflake. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Configuration des rôles

Utiliser le rôle Snowflake par défaut — Si cette option est sélectionnée, Firehose ne transmettra aucun rôle à Snowflake. Le rôle par défaut est supposé charger les données. Assurez-vous que le rôle par défaut est autorisé à insérer des données dans la table Snowflake.

Utiliser un rôle Snowflake personnalisé — Entrez un rôle Snowflake autre que celui par défaut à assumer par Firehose lors du chargement des données dans la table Snowflake.

Connectivité Snowflake

~~Les options sont privées ou publiques.~~

ID VPCE privé (facultatif)

L'identifiant VPCE permettant à Firehose de se connecter en privé à Snowflake. Le format de l'identifiant est `com.amazonaws.vpce.[région].vpce-svc-[identifiant]`. Pour plus d'informations, voir [AWS PrivateLink & Snowflake](#).

Note

Assurez-vous que votre réseau Snowflake autorise l'accès à Firehose. Pour obtenir la liste des identifiants VPCE que vous pouvez utiliser, reportez-vous au [Accès à Snowflake en VPC](#).

Configuration de base de données

- Vous devez spécifier les paramètres suivants afin d'utiliser Snowflake comme destination pour votre stream Firehose.
 - Base de données Snowflake — Toutes les données de Snowflake sont conservées dans des bases de données.
 - Schéma Snowflake : chaque base de données comprend un ou plusieurs schémas, qui sont des regroupements logiques d'objets de base de données, tels que des tables et des vues
 - Table Snowflake — Toutes les données de Snowflake sont stockées dans des tables de base de données, structurées logiquement sous forme de collections de colonnes et de lignes.

Options de chargement des données pour votre table Snowflake

- Utiliser les clés JSON comme noms de colonnes
- Utiliser les colonnes VARIANT
 - Nom de la colonne de contenu — Spécifiez un nom de colonne dans le tableau où les données brutes doivent être chargées.
 - Nom de colonne de métadonnées (facultatif) — Spécifiez un nom de colonne dans le tableau où les informations de métadonnées doivent être chargées.

Retry duration (Durée de la nouvelle tentative)

Durée (0 à 7 200 secondes) nécessaire à Firehose pour réessayer en cas d'échec de l'ouverture du canal ou de la livraison à Snowflake en raison de problèmes de service Snowflake. Firehose réessaie avec un recul exponentiel jusqu'à la fin de la durée de la nouvelle tentative. Si vous définissez la durée de la nouvelle tentative sur 0 (zéro) seconde, Firehose ne réessaie pas en cas d'échec de Snowflake et achemine les données vers le compartiment d'erreur Amazon S3.

Configurer les paramètres de destination pour Splunk

Cette section décrit les options permettant d'utiliser Splunk comme destination.

Note

Firehose fournit des données aux clusters Splunk configurés avec un Classic Load Balancer ou un Application Load Balancer.

- Fournissez des valeurs pour les champs suivants :

Splunk cluster endpoint (Point de terminaison de cluster Splunk)

Pour déterminer le point de terminaison, consultez la section [Configurer Amazon Data Firehose pour envoyer des données à la plateforme Splunk](#) dans la documentation Splunk.

Splunk endpoint type (Type de point de terminaison Splunk)

Choisissez `Raw` endpoint dans la plupart des cas. Choisissez `Event` endpoint si vous avez prétraité vos données en utilisant AWS Lambda pour envoyer des données à différents index par type d'événement. Pour plus d'informations sur le point de terminaison à utiliser, consultez la section [Configurer Amazon Data Firehose pour envoyer des données à la plateforme Splunk](#) dans la documentation Splunk.

Authentification

Vous pouvez choisir de saisir directement le jeton d'authentification ou de récupérer le secret AWS Secrets Manager pour accéder à Splunk.

- Jeton d'authentification

Pour configurer un point de terminaison Splunk capable de recevoir des données d'Amazon Data Firehose, [consultez la section Présentation de l'installation et de la configuration du module complémentaire Splunk pour Amazon Data Firehose](#) dans la

documentation Splunk. Enregistrez le jeton que vous recevez de Splunk lorsque vous configurez le point de terminaison pour ce stream Firehose et ajoutez-le ici.

- Secret

Sélectionnez un secret AWS Secrets Manager qui contient le jeton d'authentification pour Splunk. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

HEC acknowledgement timeout (Expiration de l'accusé de réception HEC)

Spécifiez la durée pendant laquelle Amazon Data Firehose attend l'accusé de réception de l'index par Splunk. Si Splunk n'envoie pas d'accusé de réception avant l'expiration du délai imparti, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données. Amazon Data Firehose réessaie ensuite ou sauvegarde les données dans votre compartiment Amazon S3, en fonction de la valeur de durée des tentatives que vous avez définie.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données à Splunk.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de Splunk. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données à Splunk (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de Splunk.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie en fonction du fournisseur de services.

Configurer les paramètres de destination pour Splunk Observability Cloud

Cette section décrit les options dont vous disposez pour utiliser Splunk Observability Cloud comme destination. Pour plus d'informations, consultez <https://docs.splunk.com/observability/en/gdi/get-data-in/connect/aws/aws-apiconfig.html#connect-to-aws-using-the-splunk-observability-cloud-api>.

- Fournissez des valeurs pour les champs suivants :

URL du point de terminaison de l'ingestion dans le Cloud

Vous pouvez trouver l'URL d'ingestion de données en temps réel de votre Splunk Observability Cloud dans Profil > Organisations > Point de terminaison d'ingestion de données en temps réel dans la console Splunk Observability.

Authentification

Vous pouvez choisir de saisir directement le jeton d'accès ou de récupérer le secret AWS Secrets Manager pour accéder à Splunk Observability Cloud.

- Jeton d'accès

Copiez votre jeton d'accès Splunk Observability avec le champ d'autorisation INGEST depuis Access Tokens sous Paramètres de la console Splunk Observability.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant le jeton d'accès à Splunk Observability Cloud. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données au point de terminaison HTTP sélectionné.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination varie d'un fournisseur de services à l'autre.

Configuration des paramètres de destination pour Sumo Logic

Cette section décrit les options dont vous disposez pour utiliser Sumo Logic comme destination. Pour plus d'informations, consultez <https://www.sumologic.com>.

- Fournissez des valeurs pour les champs suivants :

URL du point de terminaison HTTP

Spécifiez l'URL du point de terminaison HTTP au format suivant : `https://deployment.name.sumologic.net/receiver/v1/kinesis/dataType/access token`. L'URL doit être une URL HTTPS.

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données à Sumo Logic.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille de mémoire tampon recommandée pour la destination Elastic varie d'un fournisseur de services à l'autre.

Configuration des paramètres de destination pour Elastic

Cette section décrit les options dont vous disposez pour utiliser Elastic comme destination.

- Fournissez des valeurs pour les champs suivants :

URL du point de terminaison Elastic

Spécifiez l'URL du point de terminaison HTTP au format suivant : `https://<cluster-id>.es.<region>.aws.elastic-cloud.com`. L'URL doit être une URL HTTPS.

Authentification

Vous pouvez choisir de saisir directement la clé d'API ou de récupérer le code secret AWS Secrets Manager pour accéder à Elastic.

- Clé API

Contactez Elastic pour obtenir la clé d'API dont vous avez besoin pour permettre à Firehose de fournir des données à leur service.

- Secret

Sélectionnez un code secret AWS Secrets Manager contenant la clé d'API pour Elastic. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultez [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#).

Encodage de contenu

Amazon Data Firehose utilise le codage du contenu pour compresser le corps d'une demande avant de l'envoyer à la destination. Choisissez GZIP (sélectionné par défaut) ou Désactivé pour activer/désactiver le codage du contenu de votre demande.

Retry duration (Durée de la nouvelle tentative)

Spécifiez la durée pendant laquelle Amazon Data Firehose tente à nouveau d'envoyer des données à Elastic.

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du point de terminaison HTTP. Si une erreur se produit ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données au point de terminaison HTTP (tentative initiale ou nouvelle tentative), il redémarre le compteur de délais d'accusé de réception et attend un accusé de réception de la part du point de terminaison HTTP.

Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose détermine s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Si vous ne souhaitez pas qu'Amazon Data Firehose essaie à nouveau d'envoyer des données, définissez cette valeur sur 0.

Paramètres – facultatif

Amazon Data Firehose inclut ces paires clé-valeur dans chaque appel HTTP. Ces paramètres peuvent vous aider à identifier et organiser vos destinations.

Conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données entrantes avant de les livrer à la destination spécifiée. La taille recommandée de la mémoire tampon pour la destination Elastic est de 1 Mio.

Configuration de la sauvegarde et des paramètres avancés

Cette rubrique décrit comment configurer la sauvegarde et les paramètres avancés de votre stream Firehose.

Configuration des paramètres de sauvegarde

Amazon Data Firehose utilise Amazon S3 pour sauvegarder toutes les données ou uniquement les données échouées qu'il tente de livrer à la destination que vous avez choisie.

Important

- Les paramètres de sauvegarde ne sont pris en charge que si la source de votre flux Firehose est Direct PUT ou Kinesis Data Streams.
- La fonctionnalité de mise en mémoire tampon zéro n'est disponible que pour les destinations de l'application et n'est pas disponible pour la destination de sauvegarde Amazon S3.

Vous pouvez spécifier les paramètres de sauvegarde S3 pour votre flux Firehose si vous avez fait l'un des choix suivants :

- Si vous définissez Amazon S3 comme destination pour votre flux Firehose et que vous choisissiez de spécifier une fonction AWS Lambda pour transformer les enregistrements de données ou si vous choisissiez de convertir les formats d'enregistrement de données pour votre flux Firehose.

- Si vous définissez Amazon Redshift comme destination pour votre flux Firehose et que vous choisissez de spécifier une fonction AWS Lambda pour transformer les enregistrements de données.
- Si vous définissez l'un des services suivants comme destination pour votre flux Firehose : Amazon OpenSearch Service, Datadog, Dynatrace, HTTP Endpoint, LogicMonitor MongoDB Cloud, New Relic, Splunk ou Sumo Logic.

Voici les paramètres de sauvegarde de votre stream Firehose.

- Sauvegarde des enregistrements source dans Amazon S3 : si S3 ou Amazon Redshift est la destination que vous avez sélectionnée, ce paramètre indique si vous souhaitez activer la sauvegarde des données source ou la désactiver. Si un autre service pris en charge (autre que S3 ou Amazon Redshift) est défini comme destination sélectionnée, ce paramètre indique si vous souhaitez sauvegarder toutes vos données sources ou uniquement les données ayant échoué.
- Compartiment de sauvegarde S3 : il s'agit du compartiment S3 dans lequel Amazon Data Firehose sauvegarde vos données.
- Préfixe du compartiment de sauvegarde S3 : il s'agit du préfixe dans lequel Amazon Data Firehose sauvegarde vos données.
- Préfixe de sortie d'erreur du compartiment de sauvegarde S3 : toutes les données ayant échoué sont sauvegardées dans ce préfixe de sortie d'erreur de compartiment S3.
- Conseils de mise en mémoire tampon, compression et chiffrement pour la sauvegarde : Amazon Data Firehose utilise Amazon S3 pour sauvegarder toutes les données ou uniquement les données qu'il tente de livrer à la destination de votre choix, ou uniquement celles qui ont échoué. Amazon Data Firehose met en mémoire tampon les données entrantes avant de les transmettre (en les sauvegardant) à Amazon S3. Vous pouvez choisir une taille de tampon de 1 à 128 MiBs et un intervalle de mémoire tampon de 60 à 900 secondes. La première condition qui est satisfaite déclenche la livraison des données à Amazon S3. Si vous activez la transformation des données, l'intervalle de mémoire tampon s'applique entre le moment où les données transformées sont reçues par Amazon Data Firehose et leur livraison à Amazon S3. Si la livraison des données vers la destination prend du retard par rapport à l'écriture des données dans le flux Firehose, Amazon Data Firehose augmente la taille de la mémoire tampon de manière dynamique pour rattraper le retard. Cette action permet de veiller à ce que toutes les données soient livrées à la destination.
- Compression S3 : choisissez la compression de données Snappy, Zip ou GZIP compatible avec Hadoop, ou aucune compression de données. Compatible avec Snappy, Zip et Hadoop La

compression Snappy n'est pas disponible pour le flux Firehose avec Amazon Redshift comme destination.

- Format d'extension de fichier S3 (facultatif) : spécifiez un format d'extension de fichier pour les objets livrés au compartiment de destination Amazon S3. Si vous activez cette fonctionnalité, l'extension de fichier spécifiée remplacera les extensions de fichier par défaut ajoutées par les fonctionnalités de conversion de format de données ou de compression S3 telles que `.parquet` ou `.gz`. Assurez-vous d'avoir configuré la bonne extension de fichier lorsque vous utilisez cette fonctionnalité avec la conversion de format de données ou la compression S3. L'extension de fichier doit commencer par un point (.) et peut contenir les caractères autorisés : 0-9a-z ! -_.*' (). L'extension de fichier ne peut pas dépasser 128 caractères.
- Firehose prend en charge le chiffrement côté serveur Amazon S3 avec AWS Key Management Service (SSE-KMS) pour chiffrer les données livrées dans Amazon S3. Vous pouvez choisir d'utiliser le type de chiffrement par défaut spécifié dans le compartiment S3 de destination ou de chiffrer avec une clé de la liste des AWS KMS clés que vous possédez. Si vous chiffrez les données à l'aide de AWS KMS clés, vous pouvez utiliser la clé AWS gérée par défaut (`aws/s3`) ou une clé gérée par le client. Pour plus d'informations, voir [Protection des données à l'aide du chiffrement côté serveur avec des clés AWS gérées par KMS \(SSE-KMS\)](#).

Configurer les paramètres avancés

La section suivante contient des détails sur les paramètres avancés de votre stream Firehose.

- Chiffrement côté serveur : Amazon Data Firehose prend en charge le chiffrement côté serveur Amazon S3 avec le service de gestion des AWS clés (AWS KMS) pour chiffrer les données livrées dans Amazon S3. Pour plus d'informations, voir [Protection des données à l'aide du chiffrement côté serveur avec des clés AWS gérées par KMS \(SSE-KMS\)](#).
- Journalisation des erreurs : Amazon Data Firehose enregistre les erreurs liées au traitement et à la livraison. En outre, lorsque la transformation des données est activée, elle peut enregistrer les appels Lambda et envoyer des erreurs de livraison de données à Logs. CloudWatch Pour plus d'informations, consultez la section [Surveillance d'Amazon Data Firehose à l'aide CloudWatch](#) de journaux.

Important

Bien que facultative, il est fortement recommandé d'activer la journalisation des erreurs d'Amazon Data Firehose lors de la création du stream Firehose. Cette pratique garantit que

vous pouvez accéder aux détails de l'erreur en cas de traitement de l'enregistrement ou d'échec de la diffusion.

- **Autorisations** : Amazon Data Firehose utilise des rôles IAM pour toutes les autorisations dont le stream Firehose a besoin. Vous pouvez choisir de créer un nouveau rôle dans lequel les autorisations requises sont attribuées automatiquement, ou de choisir un rôle existant créé pour Amazon Data Firehose. Le rôle est utilisé pour accorder à Firehose l'accès à divers services, notamment votre compartiment S3, votre clé AWS KMS (si le chiffrement des données est activé) et la fonction Lambda (si la transformation des données est activée). La console peut créer un rôle avec des espaces réservés. Pour plus d'informations, consultez la page [Qu'est-ce qu'IAM?](#).
- **Balises** - Vous pouvez ajouter des balises pour organiser vos AWS ressources, suivre les coûts et contrôler l'accès.

Si vous spécifiez des balises dans `CreateDeliveryStreamAction`, Amazon Data Firehose octroie une autorisation supplémentaire à `firehose:TagDeliveryStreamAction` afin de vérifier si les utilisateurs sont autorisés à créer des balises. Si vous ne fournissez pas cette autorisation, les demandes de création de nouveaux flux Firehose avec des balises de ressources IAM échoueront avec l'erreur suivante `AccessDeniedException`.

`AccessDeniedException`

```
User: arn:aws:sts::x:assumed-role/x/x is not authorized to perform:
  firehose:TagDeliveryStream on resource: arn:aws:firehose:us-east-1:x:deliverystream/
x with an explicit deny in an identity-based policy.
```

L'exemple suivant illustre une politique qui permet aux utilisateurs de créer un flux Firehose et d'appliquer des balises.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "firehose:CreateDeliveryStream",
      "Resource": "*",
    }
  ],
  {
    "Effect": "Allow",
    "Action": "firehose:TagDeliveryStream",
```

```
        "Resource": "*",
      }
    ]
  }
```

Une fois que vous avez choisi votre sauvegarde et vos paramètres avancés, passez en revue vos choix, puis choisissez **Create Firehose stream**.

Le nouveau stream Firehose passe quelques instants à l'état de création avant d'être disponible. Une fois que votre stream Firehose est en état actif, vous pouvez commencer à lui envoyer des données depuis votre producteur.

Comprendre les conseils de mise en mémoire tampon

Amazon Data Firehose met en mémoire tampon les données de streaming entrantes en mémoire jusqu'à une certaine taille (taille de mise en mémoire tampon) et pendant une certaine période (intervalle de mise en mémoire tampon) avant de les transmettre aux destinations spécifiées. Vous pouvez utiliser des conseils de mise en mémoire tampon lorsque vous souhaitez fournir des fichiers de taille optimale à Amazon S3 et améliorer les performances des applications de traitement des données ou pour ajuster le taux de livraison de Firehose en fonction de la vitesse de destination.

Vous pouvez configurer la taille de la mémoire tampon et l'intervalle de mise en mémoire tampon lors de la création de nouveaux flux Firehose ou mettre à jour la taille de la mémoire tampon et l'intervalle de mise en mémoire tampon sur vos flux Firehose existants. La taille de la mémoire tampon est mesurée en Mo et l'intervalle de mise en mémoire tampon est mesuré en secondes. Toutefois, si vous spécifiez une valeur pour l'un d'eux, vous devez également fournir une valeur pour l'autre. La première condition de mémoire tampon satisfaite déclenche la livraison des données par Firehose. Si vous ne configurez pas les valeurs de mise en mémoire tampon, les valeurs par défaut sont utilisées.

Vous pouvez configurer les indices de mise en mémoire tampon de Firehose via le ou AWS Management Console les SDK AWS Command Line Interface. AWS Pour les flux existants, vous pouvez reconfigurer les indications de mise en mémoire tampon avec une valeur adaptée à vos cas d'utilisation à l'aide de l'option **Modifier** de la console ou de l'API. [UpdateDestination](#) Pour les nouveaux flux, vous pouvez configurer des indices de mise en mémoire tampon dans le cadre de la création de nouveaux flux à l'aide de la console ou de l'[CreateDeliveryStream](#) API. Pour ajuster la taille de la mémoire tampon, définissez `SizeInMBs` et `IntervalInSeconds` dans le

DestinationConfiguration paramètre spécifique à la destination de l'[UpdateDestinationAPI](#) [CreateDeliveryStreamor](#).

Note

- Pour réduire les temps de latence des cas d'utilisation en temps réel, vous pouvez utiliser un indice d'intervalle de mise en mémoire tampon nul. Lorsque vous configurez l'intervalle de mise en mémoire tampon à zéro seconde, Firehose ne met pas les données en mémoire tampon et les fournit en quelques secondes. Avant de remplacer les indications de mise en mémoire tampon par une valeur inférieure, consultez le fournisseur pour connaître les indications de mise en mémoire tampon recommandées pour Firehose pour leurs destinations.
- La fonctionnalité de mise en mémoire tampon zéro n'est disponible que pour les destinations de l'application et n'est pas disponible pour la destination de sauvegarde Amazon S3.

Note

Firehose utilise le téléchargement en plusieurs parties pour la destination S3 lorsque vous configurez un intervalle de temps de mémoire tampon inférieur à 60 secondes afin de réduire les latences. En raison du chargement en plusieurs parties pour la destination S3, vous constaterez une certaine augmentation des coûts de PUT l'API S3 si vous choisissez un intervalle de temps de mémoire tampon inférieur à 60 secondes.

Pour les plages d'indices de mise en mémoire tampon et les valeurs par défaut spécifiques à la destination, consultez le tableau suivant :

Destination	Taille de la mémoire tampon en Mo (valeur par défaut entre parenthèses)	Intervalle de mise en mémoire tampon en secondes (par défaut entre parenthèses)
S3	1-128 (5)	0-900 (300)
Redshift	1-128 (5)	0-900 (300)
OpenSearch Sans serveur	1 à 100 (5)	0-900 (300)
OpenSearch	1 à 100 (5)	0-900 (300)
Splunk	1-5 (5)	0 à 60 ans (60)
Datadog	1-4 (4)	0 à 900 (60)
Coralogix	1-64 (6)	0 à 900 (60)
Dynatrace	1-64 (5)	0 à 900 (60)
Elasticité	1	0 à 900 (60)
Honeycomb	1-64 (15)	0 à 900 (60)
Point de terminaison HTTP	1-64 (5)	0 à 900 (60)
LogicMonitor	1-64 (5)	0 à 900 (60)
Logzio	1-64 (5)	0 à 900 (60)
MongoDB	1 à 16 ans (5)	0 à 900 (60)
Nouvelle relique	1-64 (5)	0 à 900 (60)
SumoLogic	1-64 (1)	0 à 900 (60)

Destination	Taille de la mémoire tampon en Mo (valeur par défaut entre parenthèses)	Intervalle de mise en mémoire tampon en secondes (par défaut entre parenthèses)
Splunk Observability Cloud	1-64 (1)	0 à 900 (60)

Testez le flux Firehose avec des exemples de données

Vous pouvez utiliser le AWS Management Console pour ingérer des données boursières simulées. La console exécute un script dans votre navigateur pour ajouter des exemples d'enregistrements à votre stream Firehose. Cela vous permet de tester la configuration de votre flux Firehose sans avoir à générer vos propres données de test.

Voici un exemple de simulation de données :

```
{"TICKER_SYMBOL":"QXZ", "SECTOR":"HEALTHCARE", "CHANGE":-0.05, "PRICE":84.51}
```

Notez que les frais standard d'Amazon Data Firehose s'appliquent lorsque votre flux Firehose transmet les données, mais qu'ils sont gratuits lorsque les données sont générées. Pour ne pas entraîner ces frais, vous pouvez arrêter à tout moment l'exemple de flux sur la console.

Table des matières

- [Prérequis](#)
- [Effectuer un test à l'aide d'Amazon S3 comme destination](#)
- [Effectuer un test à l'aide d'Amazon Redshift comme destination](#)
- [Test en utilisant le OpenSearch service comme destination](#)
- [Test avec Splunk comme destination](#)

Prérequis

Avant de commencer, créez un stream Firehose. Pour plus d'informations, consultez [Création d'un stream Firehose](#).

Effectuer un test à l'aide d'Amazon S3 comme destination

Utilisez la procédure suivante pour tester votre flux Firehose en utilisant Amazon Simple Storage Service (Amazon S3) comme destination.

Pour tester un flux Firehose à l'aide d'Amazon S3

1. [Ouvrez la console Firehose à l'adresse https://console.aws.amazon.com/firehose/](https://console.aws.amazon.com/firehose/).

2. Choisissez un stream Firehose actif. Le flux Firehose doit être actif pour que vous puissiez commencer à envoyer des données.
3. Sous Test with demo data, choisissez Start sending demo data pour générer des exemples de données de symbole boursier.
4. Suivez les instructions à l'écran pour vérifier que les données sont remises à votre compartiment S3. Notez que les nouveaux objets peuvent prendre quelques minutes à s'afficher dans votre compartiment, selon la configuration de la mise en mémoire tampon de votre compartiment.
5. Une fois le test terminé, choisissez Stop sending demo data pour arrêter les frais d'utilisation.

Effectuer un test à l'aide d'Amazon Redshift comme destination

Suivez la procédure suivante pour tester votre flux Firehose en utilisant Amazon Redshift comme destination.

Pour tester un flux Firehose à l'aide d'Amazon Redshift

1. Votre stream Firehose s'attend à ce qu'une table soit présente dans votre cluster Amazon Redshift. [Connexion à un cluster Amazon Redshift à l'aide des outils clients SQL](#) et exécutez l'instruction suivante pour créer une table qui accepte les exemples de données.

```
create table firehose_test_table
(
  TICKER_SYMBOL varchar(4),
  SECTOR varchar(16),
  CHANGE float,
  PRICE float
);
```

2. [Ouvrez la console Firehose à l'adresse https://console.aws.amazon.com/firehose/](https://console.aws.amazon.com/firehose/).
3. Choisissez un stream Firehose actif. Le flux Firehose doit être actif pour que vous puissiez commencer à envoyer des données.
4. Modifiez les détails de destination de votre flux Firehose afin qu'il pointe vers la table nouvellement créée `firehose_test_table`.
5. Sous Test with demo data, choisissez Start sending demo data pour générer des exemples de données de symbole boursier.

6. Suivez les instructions à l'écran pour vérifier que les données sont remises à votre table. Notez que les nouvelles lignes peuvent prendre quelques minutes à s'afficher dans votre table, selon la configuration de la mise en mémoire tampon.
7. Une fois le test terminé, choisissez Stop sending demo data pour arrêter les frais d'utilisation.
8. Modifiez les détails de destination de votre flux Firehose pour qu'il pointe vers une autre table.
9. (Facultatif) Supprimez la table `firehose_test_table`.

Test en utilisant le OpenSearch service comme destination

Utilisez la procédure suivante pour tester votre flux Firehose en utilisant Amazon OpenSearch Service comme destination.

Pour tester un stream Firehose à l'aide du service OpenSearch

1. [Ouvrez la console Firehose à l'adresse `https://console.aws.amazon.com/firehose/`.](https://console.aws.amazon.com/firehose/)
2. Choisissez un stream Firehose actif. Le flux Firehose doit être actif pour que vous puissiez commencer à envoyer des données.
3. Sous Test with demo data, choisissez Start sending demo data pour générer des exemples de données de symbole boursier.
4. Suivez les instructions affichées à l'écran pour vérifier que les données sont transmises à votre domaine OpenSearch de service. Pour plus d'informations, consultez la section [Rechercher des documents dans un domaine OpenSearch de service](#) dans le manuel Amazon OpenSearch Service Developer Guide.
5. Une fois le test terminé, choisissez Stop sending demo data pour arrêter les frais d'utilisation.

Test avec Splunk comme destination

Utilisez la procédure suivante pour tester votre stream Firehose en utilisant Splunk comme destination.

Pour tester un stream Firehose à l'aide de Splunk

1. [Ouvrez la console Firehose à l'adresse `https://console.aws.amazon.com/firehose/`.](https://console.aws.amazon.com/firehose/)
2. Choisissez un stream Firehose actif. Le flux Firehose doit être actif pour que vous puissiez commencer à envoyer des données.

3. Sous Test with demo data, choisissez Start sending demo data pour générer des exemples de données de symbole boursier.
4. Vérifiez si les données sont bien remises à votre index Splunk. Voici des exemples de termes de recherche dans Splunk : `sourcetype="aws:firehose:json"` et `index="name-of-your-splunk-index"`. Pour savoir comment rechercher des événements dans Splunk, consultez le manuel [Search Manual](#) dans la documentation Splunk.

Si les données de test n'apparaissent pas dans votre index Splunk, vérifiez si votre compartiment Amazon S3 contient des événements en échec. Consultez également [Données non diffusées à Splunk](#).

5. Une fois le test terminé, choisissez Stop sending demo data pour cesser de supporter des coûts d'utilisation.

Envoyer des données vers un flux Firehose

Vous pouvez envoyer des données vers votre flux Firehose à partir de sources telles que Kinesis Data Stream, Amazon MSK, l'agent Kinesis ou l'API Amazon Data Firehose à l'aide du SDK. AWS Vous pouvez également utiliser Amazon CloudWatch Logs, CloudWatch Events ou AWS IoT comme source de données. Si vous utilisez Amazon Data Firehose pour la première fois, prenez le temps de vous familiariser avec les concepts et la terminologie présentés dans. [Qu'est-ce qu'Amazon Data Firehose ?](#)

Note

Certains AWS services peuvent uniquement envoyer des messages et des événements à un stream Firehose situé dans la même région. Si votre flux Firehose n'apparaît pas en option lorsque vous configurez une cible pour Amazon CloudWatch Logs, CloudWatch Events AWS IoT, ou lorsque vous vérifiez que votre flux Firehose se trouve dans la même région que vos autres services.

Rubriques

- [Écrire sur Amazon Data Firehose à l'aide de Kinesis Data Streams](#)
- [Écrire sur Amazon Data Firehose à l'aide d'Amazon MSK](#)
- [Écrire sur Amazon Data Firehose à l'aide de Kinesis Agent](#)
- [Écrire sur Amazon Data Firehose avec le SDK AWS](#)
- [Écrire sur Amazon Data Firehose à l'aide de journaux CloudWatch](#)
- [Écrire sur Amazon Data Firehose à l'aide d'événements CloudWatch](#)
- [Écrire sur Amazon Data Firehose à l'aide de AWS IoT](#)

Écrire sur Amazon Data Firehose à l'aide de Kinesis Data Streams

Vous pouvez configurer Amazon Kinesis Data Streams pour envoyer des informations à un flux Firehose.

⚠ Important

Si vous utilisez la bibliothèque producteur Kinesis (KPL) pour écrire des données dans un flux de données Kinesis, vous pouvez utiliser un regroupement pour combiner ces enregistrements Kinesis. Si vous utilisez ensuite ce flux de données comme source pour votre flux Firehose, Amazon Data Firehose désagrège les enregistrements avant de les envoyer à destination. Si vous configurez votre flux Firehose pour transformer les données, Amazon Data Firehose désagrège les enregistrements avant de les transmettre. AWS Lambda Pour de plus amples informations, veuillez consulter [Developing Amazon Kinesis Data Streams Producers Using the Kinesis Producer Library](#) and [Aggregation](#).

1. [Connectez-vous à la console Amazon Data Firehose AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/firehose/.](https://console.aws.amazon.com/firehose/)
2. Choisissez Create Firehose stream. Sur la page Nom et source, saisissez des valeurs dans les champs suivants :

Nom du stream Firehose

Le nom de votre stream Firehose.

Source

Choisissez Kinesis stream pour configurer un flux Firehose qui utilise un flux de données Kinesis comme source de données. Vous pouvez ensuite utiliser Amazon Data Firehose pour lire facilement les données d'un flux de données existant et les charger dans des destinations.

Pour utiliser un flux de données Kinesis en tant que source, choisissez un flux existant dans la liste Flux Kinesis ou choisissez Créer un nouveau pour créer un nouveau flux de données. Une fois le nouveau flux créé, choisissez Actualiser pour mettre à jour la liste Flux Kinesis. Si vous avez un grand nombre de flux, filtrez la liste avec l'option Filter by name (Filtrer par nom).

📘 Note

Lorsque vous configurez un flux de données Kinesis comme source d'un flux Firehose, Amazon Data PutRecord Firehose et ses opérations sont désactivés.

PutRecordBatch Dans ce cas, pour ajouter des données à votre flux Firehose, utilisez les Kinesis Data Streams et les opérations. PutRecord PutRecords

Amazon Data Firehose commence à lire les données à partir de la LATEST position de votre flux Kinesis. Pour plus d'informations sur les positions de Kinesis Data Streams, [GetShardIterator](#) consultez.

Amazon Data Firehose lance l'[GetRecords](#) opération Kinesis Data Streams une fois par seconde pour chaque partition. Toutefois, lorsque la sauvegarde complète est activée, Firehose lance l'[GetRecords](#) opération Kinesis Data Streams deux fois par seconde pour chaque partition, une pour la destination de diffusion principale et une autre pour une sauvegarde complète.

Plusieurs flux Firehose peuvent être lus à partir du même flux Kinesis. D'autres applications Kinesis (de type consommateur) peuvent également lire des données à partir du même flux. Chaque appel provenant d'un stream Firehose ou d'une autre application grand public est pris en compte dans la limite de limitation globale de la partition. Pour éviter les restrictions, planifiez attentivement vos applications. Pour plus d'informations sur les limites de Kinesis Data Streams, consultez [Amazon Kinesis Streams Limits](#).

3. Choisissez Next (Suivant) pour passer à la page [Configuration de la transformation des enregistrements et de la conversion des formats](#).

Écrire sur Amazon Data Firehose à l'aide d'Amazon MSK

Vous pouvez configurer Amazon MSK pour envoyer des informations à un flux Firehose.

1. [Connectez-vous à la console Amazon Data Firehose AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/firehose/.](https://console.aws.amazon.com/firehose/)
2. Choisissez Create Firehose stream.

Dans la section de la page Choisir une source et une destination, saisissez des valeurs dans les champs suivants :

Source

Choisissez Amazon MSK pour configurer un flux Firehose qui utilise Amazon MSK comme source de données. Vous pouvez choisir entre des clusters provisionnés par MSK et des

clusters sans serveur MSK. Vous pouvez ensuite utiliser Amazon Data Firehose pour lire facilement les données d'un cluster et d'un sujet Amazon MSK spécifiques et les charger dans la destination S3 spécifiée.

Destination (Destination)

Choisissez Amazon S3 comme destination pour votre flux Firehose.

Dans la section de la page Paramètres source, saisissez des valeurs dans les champs suivants :

Connectivité du cluster Amazon MSK

Choisissez l'option Brokers d'amorçage privés (recommandé) ou Brokers d'amorçage publics en fonction de la configuration de votre cluster. Les brokers d'amorçage sont ce que le client Apache Kafka utilise comme point de départ pour se connecter au cluster. Les courtiers bootstrap publics sont destinés à un accès public depuis l'extérieur AWS, tandis que les courtiers bootstrap privés sont destinés à un accès depuis l'intérieur. AWS Pour plus d'informations sur Amazon MSK, consultez [Amazon Managed Streaming for Apache Kafka](#).

Pour se connecter à un cluster Amazon MSK provisionné ou sans serveur via des brokers d'amorçage privés, le cluster doit répondre à toutes les exigences suivantes.

- Le cluster doit être actif.
- Le cluster doit avoir IAM comme l'une de ses méthodes de contrôle d'accès.
- La connectivité privée multi-VPC doit être activée pour la méthode de contrôle d'accès IAM.
- Vous devez ajouter à ce cluster une politique basée sur les ressources qui accorde au principal du service Amazon Data Firehose l'autorisation d'appeler l'API Amazon MSK. `CreateVpcConnection`

Pour se connecter à un cluster Amazon MSK provisionné via des brokers d'amorçage publics, le cluster doit répondre à toutes les exigences suivantes.

- Le cluster doit être actif.
- Le cluster doit avoir IAM comme l'une de ses méthodes de contrôle d'accès.
- Le cluster doit être accessible au public.

Cluster Amazon MSK

Pour le même scénario de compte, spécifiez l'ARN du cluster Amazon MSK à partir duquel votre flux Firehose lira les données.

Pour un scénario à plusieurs comptes, consultez [Livraison entre comptes depuis Amazon MSK](#).

Rubrique

Spécifiez le sujet Apache Kafka à partir duquel vous souhaitez que votre flux Firehose ingère des données. Une fois le stream Firehose créé, vous ne pouvez pas mettre à jour cette rubrique.

Dans la section Nom du flux Firehose de la page, indiquez les valeurs des champs suivants :

Nom du stream Firehose

Spécifiez le nom de votre stream Firehose.

3. Vous pouvez ensuite effectuer l'étape facultative de configuration de la transformation des enregistrements et de la conversion des formats d'enregistrement. Pour plus d'informations, voir [Configuration de la transformation des enregistrements et de la conversion des formats](#).

Écrire sur Amazon Data Firehose à l'aide de Kinesis Agent

L'agent Amazon Kinesis est une application logicielle Java autonome qui sert d'implémentation de référence pour montrer comment collecter et envoyer des données à Firehose. L'agent surveille en permanence un ensemble de fichiers et envoie de nouvelles données à votre flux Firehose. L'agent montre comment vous pouvez gérer la rotation des fichiers, les points de contrôle et réessayer en cas d'échec. Il montre comment vous pouvez fournir vos données de manière fiable, rapide et simple. Il montre également comment vous pouvez émettre CloudWatch des métriques pour mieux surveiller et résoudre les problèmes liés au processus de streaming. Pour en savoir plus, [awslabs/ amazon-kinesis-agent](#).

Par défaut, les enregistrements sont analysés à partir de chaque fichier sur la base du caractère de saut de ligne ('\n'). Toutefois, l'agent peut également être configuré pour analyser les enregistrements de plusieurs lignes (consultez [Paramètres de configuration de l'agent](#)).

Cet agent peut être installé dans des environnements basés sur des serveurs Linux tels que des serveurs Web, serveurs de journaux ou encore serveurs de base de données. Après avoir installé l'agent, configurez-le en spécifiant les fichiers à surveiller et le flux Firehose pour les données. Une fois l'agent configuré, il collecte de manière durable les données des fichiers et les envoie de manière fiable au flux Firehose.

Rubriques

- [Prérequis](#)
- [Informations d'identification](#)
- [Fournisseurs d'informations d'identification personnalisées](#)
- [Téléchargement et installation de l'agent](#)
- [Configuration et démarrage de l'agent](#)
- [Paramètres de configuration de l'agent](#)
- [Surveillance de plusieurs répertoires de fichiers et écriture dans plusieurs flux](#)
- [Utilisation de l'agent pour prétraiter les données](#)
- [Interface de ligne de commande de l'agent](#)
- [FAQ](#)

Prérequis

- Votre système d'exploitation doit être Amazon Linux ou Red Hat Enterprise Linux version 7 ou ultérieure.
- La version 2.0.0 ou ultérieure de l'agent s'exécute à l'aide de la version 1.8 ou ultérieure de JRE. La version 1.1.x de l'agent s'exécute à l'aide de la version 1.7 ou ultérieure de JRE.
- Si vous utilisez Amazon EC2 pour exécuter votre agent, lancez votre instance EC2.
- Le rôle ou les AWS informations d'identification IAM que vous spécifiez doivent être autorisés à effectuer l'opération Amazon Data [PutRecordBatch](#) Firehose pour que l'agent envoie des données à votre flux Firehose. Si vous activez la CloudWatch surveillance pour l'agent, l'autorisation d'effectuer l'opération CloudWatch [PutMetricData](#) est également nécessaire. Pour plus d'informations, consultez [Contrôler l'accès avec Amazon Data Firehose Surveillance de l'état de Kinesis Agent](#), et [Authentification et contrôle d'accès pour Amazon CloudWatch](#).

Informations d'identification

Gérez vos AWS informations d'identification à l'aide de l'une des méthodes suivantes :

- Créez un fournisseur d'informations d'identification personnalisé. Pour plus de détails, consultez [the section called “Fournisseurs d'informations d'identification personnalisées”](#).
- Spécifiez un rôle IAM lorsque vous lancez votre instance EC2.

- Spécifiez les AWS informations d'identification lorsque vous configurez l'agent (voir les entrées pour `awsAccessKeyId` et `awsSecretAccessKey` dans le tableau de configuration ci-dessous [the section called "Paramètres de configuration de l'agent"](#)).
- Modifiez `/etc/sysconfig/aws-kinesis-agent` pour spécifier votre AWS région et vos clés AWS d'accès.
- Si votre instance EC2 se trouve sur un autre AWS compte, créez un rôle IAM pour accéder au service Amazon Data Firehose. [Spécifiez ce rôle lorsque vous configurez l'agent \(voir AssumeRoLearn et IdassumeRoleExternal\)](#). Utilisez l'une des méthodes précédentes pour spécifier les AWS informations d'identification d'un utilisateur de l'autre compte autorisé à assumer ce rôle.

Fournisseurs d'informations d'identification personnalisées

Vous pouvez créer un fournisseur d'informations d'identification personnalisées et donner son nom de classe et son chemin d'accès jar à l'agent Kinesis dans les paramètres de configuration suivants : `userDefinedCredentialsProvider.classname` et `userDefinedCredentialsProvider.location`. Pour obtenir la description de ces deux paramètres de configuration, veuillez consulter [the section called "Paramètres de configuration de l'agent"](#).

Pour créer un fournisseur d'informations d'identification personnalisé, définissez une classe qui implémente l'interface `AWS CredentialsProvider`, comme celle de l'exemple suivant.

```
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;

public class YourClassName implements AWSCredentialsProvider {
    public YourClassName() {
    }

    public AWSCredentials getCredentials() {
        return new BasicAWSCredentials("key1", "key2");
    }

    public void refresh() {
    }
}
```

Votre classe doit avoir un constructeur qui ne prend aucun argument.

AWS invoque régulièrement la méthode d'actualisation pour obtenir des informations d'identification mises à jour. Si vous souhaitez que votre fournisseur d'informations d'identification fournisse différentes informations d'identification tout au long de sa durée de vie, incluez le code pour actualiser les informations d'identification dans cette méthode. Vous pouvez également laisser cette méthode vide si vous voulez un fournisseur d'informations d'identification qui vend des informations d'identification statiques (non modifiées).

Téléchargement et installation de l'agent

Commencez par vous connecter à votre instance. Pour plus d'informations, consultez [Connect to Your Instance](#) dans le guide de l'utilisateur Amazon EC2. Si vous rencontrez des difficultés pour vous connecter, consultez la section [Résolution des problèmes liés à la connexion à votre instance](#) dans le guide de l'utilisateur Amazon EC2.

Ensuite, installez l'agent à l'aide d'une des méthodes suivantes.

- Pour configurer l'agent à partir des référentiels Amazon Linux

Cette méthode fonctionne uniquement pour les instances Amazon Linux. Utilisez la commande suivante :

```
sudo yum install -y aws-kinesis-agent
```

La version 2.0.0 de l'agent ou une version ultérieure est installée sur les ordinateurs dotés du système d'exploitation Amazon Linux 2 (AL2). Cette version de l'agent nécessite Java version 1.8 ou ultérieure. Si la version Java requise n'est pas encore présente, le processus d'installation de l'agent l'installe. Pour en savoir plus sur Amazon Linux 2, consultez <https://aws.amazon.com/amazon-linux-2/>.

- Pour configurer l'agent à partir d'un référentiel Amazon S3

Cette méthode fonctionne pour Red Hat Enterprise Linux, ainsi que pour les instances Amazon Linux 2, car elle installe l'agent à partir du référentiel accessible au public. Utilisez la commande suivante pour télécharger et installer la dernière version de la version 2.x.x de l'agent :

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-latest.amzn2.noarch.rpm
```

Pour installer une version spécifique de l'agent, spécifiez le numéro de version dans la commande. Par exemple, la commande suivante installe la version 2.0.1 de l'agent.

```
sudo yum install -y https://streaming-data-agent.s3.amazonaws.com/aws-kinesis-agent-2.0.1-1.amzn1.noarch.rpm
```

Si vous utilisez Java 1.7 et que vous ne souhaitez pas le mettre à niveau, vous pouvez télécharger la version 1.x.x de l'agent, qui est compatible avec Java 1.7. Par exemple, pour télécharger la version 1.1.6 de l'agent, vous pouvez utiliser la commande suivante :

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-1.1.6-1.amzn1.noarch.rpm
```

La dernière version 1.x.x de l'agent peut être téléchargée à l'aide de la commande suivante :

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-latest.amzn1.noarch.rpm
```

- Pour configurer l'agent à partir du GitHub dépôt
 1. Tout d'abord, assurez-vous que la version de Java requise est installée, en fonction de la version de l'agent.
 2. Téléchargez l'agent depuis le dépôt [awslabs/ amazon-kinesis-agent](#) GitHub .
 3. Installez l'agent en accédant au répertoire de téléchargement et en exécutant la commande suivante :

```
sudo ./setup --install
```

- Configuration de l'agent dans un conteneur Docker

L'agent Kinesis peut également être exécuté dans un conteneur via la base de conteneurs [amazonlinux](#). Utilisez le Dockerfile suivant, puis exécutez `docker build`.

```
FROM amazonlinux

RUN yum install -y aws-kinesis-agent which findutils
COPY agent.json /etc/aws-kinesis/agent.json

CMD ["start-aws-kinesis-agent"]
```

Configuration et démarrage de l'agent

Pour configurer et démarrer l'agent

1. Ouvrez le fichier de configuration et modifiez-le (en tant que super-utilisateur si vous utilisez les autorisations d'accès fichier par défaut) : `/etc/aws-kinesis/agent.json`

Dans ce fichier de configuration, spécifiez les fichiers ("`filePattern`") à partir desquels l'agent collecte les données, ainsi que le nom du flux Firehose ("`deliveryStream`") auquel l'agent envoie les données. Le nom de fichier est un modèle et l'agent reconnaît les rotations de fichier. Vous pouvez effectuer une rotation de fichier ou créer de nouveaux fichiers pas plus d'une fois par seconde. L'agent utilise l'horodatage de création des fichiers pour déterminer les fichiers à suivre et à intégrer dans votre flux Firehose. La création de fichiers ou la rotation de fichiers plus d'une fois par seconde ne permet pas à l'agent de différencier correctement les fichiers.

```
{
  "flows": [
    {
      "filePattern": "/tmp/app.log*",
      "deliveryStream": "yourdeliverystream"
    }
  ]
}
```

La AWS région par défaut est `us-east-1`. Si vous utilisez une autre région, ajoutez le paramètre `firehose.endpoint` au fichier de configuration, en spécifiant le point de terminaison de votre région. Pour plus d'informations, consultez [Paramètres de configuration de l'agent](#).

2. Démarrez l'agent manuellement :

```
sudo service aws-kinesis-agent start
```

3. (Facultatif) Configurez l'agent pour qu'il soit lancé au démarrage du système :

```
sudo chkconfig aws-kinesis-agent on
```

L'agent fonctionne maintenant en arrière-plan en tant que service système. Il surveille en permanence les fichiers spécifiés et envoie des données au flux Firehose spécifié. L'activité de l'agent est enregistrée dans `/var/log/aws-kinesis-agent/aws-kinesis-agent.log`.

Paramètres de configuration de l'agent

L'agent prend en charge les deux paramètres de configuration obligatoires `filePattern` et `deliveryStream`, plus les paramètres de configuration facultatifs des fonctionnalités supplémentaires. Vous pouvez spécifier aussi bien des paramètres de configuration obligatoires que facultatifs dans `/etc/aws-kinesis/agent.json`.

Chaque fois que vous modifiez le fichier de configuration, vous devez arrêter et démarrer l'agent en utilisant les commandes suivantes :

```
sudo service aws-kinesis-agent stop
sudo service aws-kinesis-agent start
```

Vous pouvez également utiliser la commande suivante :

```
sudo service aws-kinesis-agent restart
```

Les paramètres de configuration générale sont indiqués ci-après.

Paramètre de configuration	Description
<code>assumeRoleARN</code>	L'ARN (Amazon Resource Name) du rôle que l'utilisateur doit assumer. Pour plus d'informations, consultez la section Accès délégué entre AWS comptes à l'aide de rôles IAM dans le guide de l'utilisateur IAM.

Paramètre de configuration	Description
<code>assumeRoleExternalId</code>	Identifiant facultatif qui détermine qui peut assumer le rôle. Pour en savoir plus, consultez la rubrique Procédure d'utilisation d'un ID externe dans le Guide de l'utilisateur IAM.
<code>awsAccessKeyId</code>	AWS ID de clé d'accès qui remplace les informations d'identification par défaut. Ce paramètre est prioritaire sur tous les autres fournisseurs d'informations d'identification.
<code>awsSecretAccessKey</code>	AWS clé secrète qui remplace les informations d'identification par défaut. Ce paramètre est prioritaire sur tous les autres fournisseurs d'informations d'identification.
<code>cloudwatch.emitMetrics</code>	Permet à l'agent d'émettre des métriques CloudWatch s'il est défini (true). Valeur par défaut : true
<code>cloudwatch.endpoint</code>	Le point de terminaison régional pour CloudWatch. Par défaut : <code>monitoring.us-east-1.amazonaws.com</code>
<code>firehose.endpoint</code>	Point de terminaison régional pour Amazon Data Firehose. Par défaut : <code>firehose.us-east-1.amazonaws.com</code>
<code>sts.endpoint</code>	Point de terminaison régional pour le service AWS de jetons de sécurité. Par défaut : <code>https://sts.amazonaws.com</code>
<code>userDefinedCredentialsProvider.classname</code>	Si vous définissez un fournisseur d'informations d'identification personnalisées, indiquez son nom de classe complet à l'aide de ce paramètre. N'ajoutez pas <code>.class</code> à la fin du nom de la classe.

Paramètre de configuration	Description
<code>userDefinedCredentialsProvider.location</code>	Si vous définissez un fournisseur d'informations d'identification personnalisées, utilisez ce paramètre pour spécifier le chemin absolu du fichier jar contenant le fournisseur d'informations d'identification personnalisées. L'agent recherche également le fichier jar à l'emplacement suivant : <code>/usr/share/aws-kinesis-agent/lib/</code> .

Les paramètres de configuration de flux sont indiqués ci-après.

Paramètre de configuration	Description
<code>aggregateRecordSizeBytes</code>	Pour que l'agent agrège les enregistrements puis les place dans le flux Firehose en une seule opération, spécifiez ce paramètre. Réglez-le à la taille que vous souhaitez attribuer à l'enregistrement agrégé avant que l'agent ne le place dans le flux Firehose. Par défaut : 0 (pas de regroupement)
<code>dataProcessingOptions</code>	La liste des options de traitement appliquées à chaque enregistrement analysé avant son envoi au flux Firehose. Les options de traitement sont exécutées dans l'ordre spécifié. Pour plus d'informations, consultez Utilisation de l'agent pour prétraiter les données .
<code>deliveryStream</code>	[Obligatoire] Nom du stream Firehose.
<code>filePattern</code>	[Obligatoire] Modèle global des fichiers qui doivent être surveillés par l'agent. N'importe quel fichier qui correspond à ce modèle est collecté automatiquement par l'agent et surveillé. Pour tous les fichiers correspondant à ce modèle, accordez l'autorisation de lecture à <code>aws-kinesis-agent-user</code> . Pour l'annuaire contenant les fichiers, accordez les autorisations de lecture et d'exécution à <code>aws-kinesis-agent-user</code> .

Paramètre de configuration	Description
	<div style="border: 1px solid #f08080; padding: 10px; background-color: #fff9f9;"> <p> Important</p> <p>L'agent récupère tous les fichiers correspondant à ce modèle. Pour vous assurer que l'agent ne récupère pas d'autres enregistrements que ceux prévus à cet effet, choisissez soigneusement ce modèle.</p> </div>
<code>initialPosition</code>	<p>Position initiale à partir de laquelle le fichier a commencé à être analysé. Les valeurs valides sont <code>START_OF_FILE</code> et <code>END_OF_FILE</code> .</p> <p>Par défaut : <code>END_OF_FILE</code></p>
<code>maxBufferAgeMillis</code>	<p>Durée maximale, en millisecondes, pendant laquelle l'agent met les données en mémoire tampon avant de les envoyer au flux Firehose.</p> <p>Plage de valeurs : 1 000 à 900 000 (1 seconde à 15 minutes)</p> <p>Par défaut : 60 000 (1 minute)</p>
<code>maxBufferSizeBytes</code>	<p>Taille maximale, en octets, pour laquelle l'agent met en mémoire tampon les données avant de les envoyer au flux Firehose.</p> <p>Plage de valeurs : 1 à 4 194 304 (4 Mo)</p> <p>Par défaut : 4 194 304 (4 Mo)</p>
<code>maxBufferSizeRecords</code>	<p>Le nombre maximum d'enregistrements pour lesquels l'agent met les données en mémoire tampon avant de les envoyer au flux Firehose.</p> <p>Plage de valeurs : 1 à 500</p> <p>Par défaut : 500</p>

Paramètre de configuration	Description
<code>minTimeBetweenFilePollsMillis</code>	<p>Fréquence, en millisecondes, à laquelle l'agent interroge et analyse les fichiers surveillés pour rechercher les nouvelles données.</p> <p>Plage de valeurs : 1 ou plus</p> <p>Par défaut : 100</p>
<code>multilineStartPattern</code>	<p>Modèle d'identification du début d'un enregistrement. Un enregistrement se compose d'une ligne qui correspond au modèle et de lignes suivantes qui ne correspondent pas au modèle. Les valeurs valides sont les expressions régulières. Par défaut, chaque nouvelle ligne comprise dans les fichiers journaux est analysée comme étant un enregistrement.</p>
<code>skipHeaderLines</code>	<p>Nombre de lignes que l'agent doit ignorer lors de l'analyse au début des fichiers surveillés.</p> <p>Plage de valeurs : 0 ou plus</p> <p>Par défaut : 0 (zéro)</p>
<code>truncatedRecord Terminator</code>	<p>Chaîne utilisée par l'agent pour tronquer un enregistrement analysé lorsque la taille de l'enregistrement dépasse la limite de taille d'enregistrement d'Amazon Data Firehose. (1,000 Ko)</p> <p>Par défaut : <code>'\n'</code> (nouvelle ligne)</p>

Surveillance de plusieurs répertoires de fichiers et écriture dans plusieurs flux

En spécifiant plusieurs paramètres de configuration de flux, vous pouvez configurer l'agent pour surveiller plusieurs répertoires de fichiers et envoyer des données dans plusieurs flux. Dans l'exemple de configuration suivant, l'agent surveille deux répertoires de fichiers et envoie des données à un flux de données Kinesis et à un flux Firehose respectivement. Vous pouvez spécifier différents points de terminaison pour Kinesis Data Streams et Amazon Data Firehose afin que votre flux de données et votre flux Firehose n'aient pas besoin de se trouver dans la même région.

```
{
  "cloudwatch.emitMetrics": true,
  "kinesis.endpoint": "https://your/kinesis/endpoint",
  "firehose.endpoint": "https://your/firehose/endpoint",
  "flows": [
    {
      "filePattern": "/tmp/app1.log*",
      "kinesisStream": "yourkinesisstream"
    },
    {
      "filePattern": "/tmp/app2.log*",
      "deliveryStream": "yourfirehosedeliverystream"
    }
  ]
}
```

Pour plus d'informations sur l'utilisation de l'agent avec Amazon Kinesis Data Streams, consultez [Writing to Amazon Kinesis Data Streams with Kinesis Agent](#).

Utilisation de l'agent pour prétraiter les données

L'agent peut prétraiter les enregistrements analysés à partir des fichiers surveillés avant de les envoyer à votre flux Firehose. Vous pouvez activer cette fonctionnalité en ajoutant le paramètre de configuration `dataProcessingOptions` à votre flux de fichiers. Une ou plusieurs options de traitement peuvent être ajoutées. Elles sont exécutées dans l'ordre spécifié.

L'agent prend en charge les options de traitement suivantes. Comme l'agent est open source, vous pouvez continuer à développer et étendre ses options de traitement. Vous pouvez télécharger l'agent depuis [Kinesis Agent](#).

Options de traitement

SINGLELINE

Convertit un enregistrement de plusieurs lignes en un enregistrement d'une seule ligne en supprimant les caractères de saut de ligne, les espaces de début et les espaces de fin.

```
{
  "optionName": "SINGLELINE"
}
```

CSVTOJSON

Convertit un enregistrement du format séparé par délimiteur au format JSON.

```
{
  "optionName": "CSVTOJSON",
  "customFieldNames": [ "field1", "field2", ... ],
  "delimiter": "yourdelimiter"
}
```

customFieldNames

[Obligatoire] Noms de domaine utilisés comme clés dans chaque paire clé-valeur JSON.

Par exemple, si vous spécifiez ["f1", "f2"], l'enregistrement « v1, v2 » est converti en { "f1": "v1", "f2": "v2" }.

delimiter

Chaîne utilisée comme délimiteur dans l'enregistrement. La valeur par défaut est une virgule (,).

LOGTOJSON

Convertit un enregistrement du format de journal au format JSON. Les formats de journal pris en charge sont Apache Common Log, Apache Combined Log, Apache Error Log et RFC3164 Syslog.

```
{
  "optionName": "LOGTOJSON",
  "logFormat": "logformat",
  "matchPattern": "yourregexpattern",
  "customFieldNames": [ "field1", "field2", ... ]
}
```

logFormat

[Obligatoire] Format d'entrée de journal. Les valeurs admises sont les suivantes :

- COMMONAPACHELOG : le format de journal courant d'Apache. Chaque entrée de journal a le schéma suivant par défaut : « %{host} %{ident} %{authuser} [%{datetime}] \"%{request}\" %{response} %{bytes} ».
- COMBINEDAPACHELOG : le format de journal combiné d'Apache. Chaque entrée de journal a le schéma suivant par défaut : « %{host} %{ident} %{authuser} [%{datetime}] \"%{request}\" %{response} %{bytes} %{referrer} %{agent} ».

- **APACHEERRORLOG** : le format de journal d'erreurs d'Apache. Chaque entrée de journal a le schéma suivant par défaut : « `[%{timestamp}] [%{module}:%{severity}] [pid %{processid}:tid %{threadid}] [client: %{client}] %{message}` ».
- **SYSLOG** : le format Syslog RFC3164. Chaque entrée de journal a le schéma suivant par défaut : « `%{timestamp} %{hostname} %{program}[%{processid}]: %{message}` ».

matchPattern

Remplace le modèle par défaut pour le format de journal spécifié. Utilisez ce paramètre pour extraire les valeurs des entrées du journal si elles utilisent un format personnalisé. Si vous spécifiez `matchPattern`, vous devez également spécifier `customFieldNames`.

customFieldNames

Noms de champ personnalisés utilisés comme clés dans chaque paire clé-valeur JSON. Vous pouvez utiliser ce paramètre pour définir les noms de champ pour les valeurs extraites de `matchPattern`, ou remplacer les noms de champ par défaut des formats de journalisation prédéfinis.

Exemple : Configuration LOGTOJSON

Voici un exemple de configuration LOGTOJSON pour une entrée au format Journal courant Apache convertie au format JSON :

```
{
  "optionName": "LOGTOJSON",
  "logFormat": "COMMONAPACHELOG"
}
```

Avant la conversion :

```
64.242.88.10 - - [07/Mar/2004:16:10:02 -0800] "GET /mailman/listinfo/hsdivision
HTTP/1.1" 200 6291
```

Après la conversion :

```
{"host":"64.242.88.10","ident":null,"authuser":null,"datetime":"07/
Mar/2004:16:10:02 -0800","request":"GET /mailman/listinfo/hsdivision
HTTP/1.1","response":"200","bytes":"6291"}
```

Exemple : Configuration LOGTOJSON avec champs personnalisés

Voici un autre exemple de configuration LOGTOJSON :

```
{
  "optionName": "LOGTOJSON",
  "logFormat": "COMMONAPACHELOG",
  "customFieldNames": ["f1", "f2", "f3", "f4", "f5", "f6", "f7"]
}
```

Avec ce paramètre de configuration, la même entrée au format Journal courant Apache de journal issue de l'exemple précédent est convertie au format JSON comme suit :

```
{"f1":"64.242.88.10","f2":null,"f3":null,"f4":"07/Mar/2004:16:10:02 -0800","f5":"GET /
mailman/listinfo/hsdivision HTTP/1.1","f6":"200","f7":"6291"}
```

Exemple : Conversion d'une entrée au format Journal courant Apache

La configuration de flux suivante convertit une entrée au format Journal courant Apache en un enregistrement d'une seule ligne au format JSON :

```
{
  "flows": [
    {
      "filePattern": "/tmp/app.log*",
      "deliveryStream": "my-delivery-stream",
      "dataProcessingOptions": [
        {
          "optionName": "LOGTOJSON",
          "logFormat": "COMMONAPACHELOG"
        }
      ]
    }
  ]
}
```

Exemple : Conversion des enregistrements de plusieurs lignes

La configuration de flux suivante analyse les enregistrements de plusieurs lignes dont la première ligne commence par « [SEQUENCE= ». Chaque enregistrement est d'abord converti en un enregistrement d'une seule ligne. Les valeurs sont ensuite extraites de l'enregistrement sur la base

d'un séparateur tabulation. Les valeurs extraites sont mappées à des valeurs `customFieldNames` spécifiées pour former un enregistrement d'une seule ligne au format JSON.

```
{
  "flows": [
    {
      "filePattern": "/tmp/app.log*",
      "deliveryStream": "my-delivery-stream",
      "multilineStartPattern": "\\[[SEQUENCE=",
      "dataProcessingOptions": [
        {
          "optionName": "SINGLELINE"
        },
        {
          "optionName": "CSVTOJSON",
          "customFieldNames": [ "field1", "field2", "field3" ],
          "delimiter": "\\t"
        }
      ]
    }
  ]
}
```

Exemple : Configuration LOGTOJSON avec modèle de correspondance

Voici un exemple de configuration LOGTOJSON pour une entrée au format Journal courant Apache convertie au format JSON, avec le dernier champ (octets) omis :

```
{
  "optionName": "LOGTOJSON",
  "logFormat": "COMMONAPACHELOG",
  "matchPattern": "^(\\d\\.\\d\\.\\d\\.\\d) (\\S+) (\\S+) \\[[([\\w:/]+\\s[+-]\\d{4})\\] \\\"(.+?)\\\" (\\d{3})\"",
  "customFieldNames": ["host", "ident", "authuser", "datetime", "request", "response"]
}
```

Avant la conversion :

```
123.45.67.89 - - [27/Oct/2000:09:27:09 -0400] "GET /java/javaResources.html HTTP/1.0"
200
```

Après la conversion :

```
{"host":"123.45.67.89","ident":null,"authuser":null,"datetime":"27/Oct/2000:09:27:09-0400","request":"GET /java/javaResources.html HTTP/1.0","response":"200"}
```

Interface de ligne de commande de l'agent

Lancer automatiquement l'agent au démarrage du système :

```
sudo chkconfig aws-kinesis-agent on
```

Vérifier le statut de l'agent :

```
sudo service aws-kinesis-agent status
```

Arrêter l'agent :

```
sudo service aws-kinesis-agent stop
```

Lire le fichier journal de l'agent à partir de cet emplacement :

```
/var/log/aws-kinesis-agent/aws-kinesis-agent.log
```

Désinstaller l'agent :

```
sudo yum remove aws-kinesis-agent
```

FAQ

Existe-t-il un Kinesis Agent pour Windows ?

[Kinesis Agent pour Windows](#) est un logiciel différent de Kinesis Agent pour les plateformes Linux.

Pourquoi Kinesis Agent ralentit-il ou **RecordSendErrors** augmente-t-il ?

Cela est généralement dû à une limitation de Kinesis. Vérifiez la `WriteProvisionedThroughputExceeded` métrique pour Kinesis Data Streams ou `ThrottledRecords` la métrique pour les flux Firehose. Toute augmentation de ces métriques

par rapport à 0 indique que les limites de flux doivent être augmentées. [Pour plus d'informations, consultez les sections Limites de Kinesis Data Stream et Firehose Streams.](#)

Une fois que vous avez exclu la limitation, vérifiez si Kinesis Agent est configuré pour suivre un grand nombre de petits fichiers. Il y a un délai lorsque Kinesis Agent suit un nouveau fichier, c'est pourquoi Kinesis Agent doit suivre un petit nombre de fichiers plus volumineux. Essayez de regrouper vos fichiers journaux dans des fichiers plus volumineux.

Pourquoi est-ce que je reçois des exceptions **java.lang.OutOfMemoryError** ?

Kinesis Agent ne dispose pas de suffisamment de mémoire pour gérer sa charge de travail actuelle. Essayez d'augmenter `JAVA_START_HEAP` et `JAVA_MAX_HEAP` dans `/usr/bin/start-aws-kinesis-agent` et de redémarrer l'agent.

Pourquoi est-ce que je reçois des exceptions **IllegalStateException : connection pool shut down** ?

Kinesis Agent ne dispose pas de suffisamment de connexions pour gérer sa charge de travail actuelle. Essayez d'augmenter `maxConnections` et `maxSendingThreads` dans les paramètres de configuration de votre agent général à `/etc/aws-kinesis/agent.json`. La valeur par défaut de ces champs est 12 fois supérieure au nombre de processeurs d'exécution disponibles. Consultez [AgentConfiguration.java](#) pour en savoir plus sur les paramètres de configuration avancés des agents.

Comment puis-je déboguer un autre problème avec Kinesis Agent ?

Les journaux de niveau `DEBUG` peuvent être activés dans `/etc/aws-kinesis/log4j.xml`.

Comment dois-je configurer Kinesis Agent ?

Plus la taille de `maxBufferSizeBytes` est petite, plus Kinesis Agent envoie fréquemment des données. Cela peut être une bonne chose, car cela réduit le délai de livraison des enregistrements, mais cela augmente également le nombre de requêtes par seconde adressées à Kinesis.

Pourquoi Kinesis Agent envoie-t-il des enregistrements dupliqués ?

Cela se produit en raison d'une mauvaise configuration dans le suivi des fichiers. Assurez-vous que chaque `fileFlow's filePattern` ne correspond qu'à un seul fichier. Cela peut également se produire si le mode `logrotate` utilisé est en mode `copytruncate`. Essayez de remplacer le mode par défaut ou de créer le mode pour éviter les doublons. Pour plus d'informations sur la gestion des

enregistrements dupliqués, consultez la rubrique [Gestion des enregistrements dupliqués](#) (français non garanti).

Écrire sur Amazon Data Firehose avec le SDK AWS

[Vous pouvez utiliser l'API Amazon Data Firehose pour envoyer des données vers un flux Firehose à l'aide du SDK pour AWS Java, .NET, Node.js, Python ou Ruby.](#) Si vous utilisez Amazon Data Firehose pour la première fois, prenez le temps de vous familiariser avec les concepts et la terminologie présentés dans [Qu'est-ce qu'Amazon Data Firehose ?](#) Pour plus d'informations, consultez [Start Developing with Amazon Web Services](#).

Ces exemples ne représentent pas du code prêt à la production, car ils ne recherchent pas toutes les exceptions possibles ou ne tiennent pas compte de toutes les considérations possibles en matière de sécurité ou de performances.

L'API Amazon Data Firehose propose deux opérations pour envoyer des données à votre flux Firehose : et. [PutRecordPutRecordBatch](#) `PutRecord()` envoie un enregistrement de données au cours d'un appel et `PutRecordBatch()` peut envoyer plusieurs enregistrements de données au cours d'un appel.

Rubriques

- [Opérations d'écriture unique utilisant PutRecord](#)
- [Opérations d'écriture par lots à l'aide de PutRecordBatch](#)

Opérations d'écriture unique utilisant PutRecord

L'ajout de données nécessite uniquement le nom du flux Firehose et un tampon d'octets (≤ 1000 Ko). Dans la mesure où Amazon Data Firehose regroupe plusieurs enregistrements avant de charger le fichier dans Amazon S3, vous souhaitez peut-être ajouter un séparateur d'enregistrements. Pour placer les données un enregistrement à la fois dans un flux Firehose, utilisez le code suivant :

```
PutRecordRequest putRecordRequest = new PutRecordRequest();
putRecordRequest.setDeliveryStreamName(deliveryStreamName);

String data = line + "\n";

Record record = new Record().withData(ByteBuffer.wrap(data.getBytes()));
putRecordRequest.setRecord(record);
```

```
// Put record into the DeliveryStream
firehoseClient.putRecord(putRecordRequest);
```

Pour plus de contexte de code, consultez l'exemple de code inclus dans le AWS SDK. Pour plus d'informations sur la syntaxe des demandes et des réponses, consultez la rubrique correspondante dans [Firehose API Operations](#).

Opérations d'écriture par lots à l'aide de PutRecordBatch

La saisie de données nécessite uniquement le nom du flux Firehose et une liste d'enregistrements. Dans la mesure où Amazon Data Firehose regroupe plusieurs enregistrements avant de charger le fichier dans Amazon S3, vous souhaitez peut-être ajouter un séparateur d'enregistrements. Pour placer des enregistrements de données par lots dans un flux Firehose, utilisez le code suivant :

```
PutRecordBatchRequest putRecordBatchRequest = new PutRecordBatchRequest();
putRecordBatchRequest.setDeliveryStreamName(deliveryStreamName);
putRecordBatchRequest.setRecords(recordList);

// Put Record Batch records. Max No.Of Records we can put in a
// single put record batch request is 500
firehoseClient.putRecordBatch(putRecordBatchRequest);

recordList.clear();
```

Pour plus de contexte de code, consultez l'exemple de code inclus dans le AWS SDK. Pour plus d'informations sur la syntaxe des demandes et des réponses, consultez la rubrique correspondante dans [Firehose API Operations](#).

Écrire sur Amazon Data Firehose à l'aide de journaux CloudWatch

CloudWatch Les événements des journaux peuvent être envoyés à Firehose à l'aide de filtres d'abonnement CloudWatch. Pour plus d'informations, consultez la section [Filtres d'abonnement avec Amazon Data Firehose](#).

CloudWatch Les événements des journaux sont envoyés à Firehose au format gzip compressé. Si vous souhaitez transmettre des événements de journal décompressés aux destinations de Firehose, vous pouvez utiliser la fonction de décompression de Firehose pour décompresser automatiquement les journaux. CloudWatch

⚠ Important

À l'heure actuelle, Firehose ne prend pas en charge la livraison de CloudWatch journaux vers la destination Amazon OpenSearch Service, car Amazon CloudWatch combine plusieurs événements de journal dans un seul enregistrement Firehose et Amazon OpenSearch Service ne peut pas accepter plusieurs événements de journal dans un seul enregistrement. Vous pouvez également envisager d'[utiliser un filtre d'abonnement pour Amazon OpenSearch Service in CloudWatch Logs](#).

Décompression des journaux CloudWatch

[Si vous utilisez Firehose pour envoyer des CloudWatch journaux et que vous souhaitez transmettre des données décompressées à votre destination de flux Firehose, utilisez la conversion du format de données Firehose \(Parquet, ORC\) ou le partitionnement dynamique.](#) Vous devez activer la décompression pour votre flux Firehose.

Vous pouvez activer la décompression à l'aide du AWS Command Line Interface ou AWS Management Console des AWS SDK.

ℹ Note

Si vous activez la fonction de décompression sur un flux, utilisez ce flux exclusivement pour les filtres d'abonnement CloudWatch Logs, et non pour Vended Logs. Si vous activez la fonction de décompression sur un flux utilisé pour ingérer à la fois les CloudWatch journaux et les journaux vendus, l'ingestion des journaux vendus dans Firehose échoue. Cette fonctionnalité de décompression ne concerne que les CloudWatch journaux.

Extraction des messages après décompression des journaux CloudWatch

Lorsque vous activez la décompression, vous pouvez également activer l'extraction des messages. Lors de l'extraction de messages, Firehose filtre toutes les métadonnées, telles que le propriétaire, le groupe de journaux, le flux de journaux, etc., des enregistrements décompressés des CloudWatch journaux et ne diffuse que le contenu contenu dans les champs de message. Si vous transmettez des données vers une destination Splunk, vous devez activer l'extraction des messages pour que Splunk puisse analyser les données. Vous trouverez ci-dessous des exemples de sorties après décompression avec et sans extraction de message.

Figure 1 : Exemple de sortie après décompression sans extraction de message :

```
{
  "owner": "111111111111",
  "logGroup": "CloudTrail/logs",
  "logStream": "111111111111_CloudTrail/logs_us-east-1",
  "subscriptionFilters": [
    "Destination"
  ],
  "messageType": "DATA_MESSAGE",
  "logEvents": [
    {
      "id": "31953106606966983378809025079804211143289615424298221568",
      "timestamp": 1432826855000,
      "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root1\"}}"
    },
    {
      "id": "31953106606966983378809025079804211143289615424298221569",
      "timestamp": 1432826855000,
      "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root2\"}}"
    },
    {
      "id": "31953106606966983378809025079804211143289615424298221570",
      "timestamp": 1432826855000,
      "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root3\"}}"
    }
  ]
}
```

Figure 2 : Exemple de sortie après décompression avec extraction du message :

```
{"eventVersion":"1.03","userIdentity":{"type":"Root1"}}
{"eventVersion":"1.03","userIdentity":{"type":"Root2"}}
{"eventVersion":"1.03","userIdentity":{"type":"Root3"}}
```

Activation et désactivation de la décompression

Vous pouvez activer et désactiver la décompression à l'aide du AWS Command Line Interface ou AWS Management Console des AWS SDK.

Activation de la décompression sur un nouveau flux de données à l'aide du AWS Management Console

Pour activer la décompression sur un nouveau flux de données à l'aide du AWS Management Console

1. [Connectez-vous à la console Kinesis AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/kinesis](https://console.aws.amazon.com/kinesis).
2. Choisissez Amazon Data Firehose dans le volet de navigation.
3. Choisissez Create Firehose stream.
4. Sous Choisir la source et la destination

Source

La source de votre stream Firehose. Choisissez l'une des sources suivantes :

- Direct PUT — Choisissez cette option pour créer un flux Firehose dans lequel les applications du producteur écrivent directement. Pour obtenir la liste des AWS services, agents et services open source intégrés à Direct PUT dans Firehose, consultez [cette section](#).
- Flux Kinesis : choisissez cette option pour configurer un flux Firehose qui utilise un flux de données Kinesis comme source de données. Vous pouvez ensuite utiliser Firehose pour lire facilement les données d'un flux de données Kinesis existant et les charger dans des destinations. Pour plus d'informations, consultez [Écrire dans Firehose à l'aide de Kinesis Data Streams](#)

Destination (Destination)

La destination de votre stream Firehose. Sélectionnez l'une des méthodes suivantes :

- Amazon S3
 - Splunk
5. Sous Nom du flux Firehose, entrez le nom de votre flux.
 6. (Facultatif) Sous Transformer les enregistrements :
 - Dans la section Décompresser les enregistrements source depuis Amazon CloudWatch Logs, sélectionnez Activer la décompression.
 - Si vous souhaitez utiliser l'extraction des messages après la décompression, choisissez Activer l'extraction des messages.

Activation de la décompression sur un flux de données existant à l'aide du AWS Management Console

Si vous disposez d'un flux Firehose doté d'une fonction Lambda pour effectuer la décompression, vous pouvez le remplacer par la fonction de décompression Firehose. Avant de continuer, vérifiez le code de votre fonction Lambda pour vérifier qu'il effectue uniquement la décompression ou l'extraction des messages. La sortie de votre fonction Lambda doit ressembler aux exemples présentés dans la figure 1 ou la figure 2 de la section précédente. Si le résultat est similaire, vous pouvez remplacer la fonction Lambda en procédant comme suit.

1. [Remplacez votre fonction Lambda actuelle par ce plan](#). La nouvelle fonction Blueprint Lambda détecte automatiquement si les données entrantes sont compressées ou décompressées. Il n'effectue la décompression que si ses données d'entrée sont compressées.
2. Activez la décompression à l'aide de l'option Firehose intégrée pour la décompression.
3. Activez CloudWatch les métriques pour votre stream Firehose si ce n'est pas déjà fait. Surveillez la métrique CloudWatchProcessorLambda _ IncomingCompressedData et attendez qu'elle passe à zéro. Cela confirme que toutes les données d'entrée envoyées à votre fonction Lambda sont décompressées et que la fonction Lambda n'est plus requise.
4. Supprimez la transformation de données Lambda car vous n'en avez plus besoin pour décompresser votre flux.

Désactivation de la décompression à l'aide du AWS Management Console

Pour désactiver la décompression d'un flux de données à l'aide du AWS Management Console

1. [Connectez-vous à la console Kinesis AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/kinesis>](https://console.aws.amazon.com/kinesis).
2. Choisissez Amazon Data Firehose dans le volet de navigation.
3. Choisissez le stream Firehose que vous souhaitez modifier.
4. Sur la page des détails du stream Firehose, choisissez l'onglet Configuration.
5. Dans la section Transformer et convertir des enregistrements, choisissez Modifier.
6. Sous Décompresser les enregistrements source depuis Amazon CloudWatch Logs, décochez Activer la décompression, puis sélectionnez Enregistrer les modifications.

FAQ

Qu'advient-il des données sources en cas d'erreur lors de la décompression ?

Si Amazon Data Firehose ne parvient pas à décompresser l'enregistrement, celui-ci est livré tel quel (au format compressé) au compartiment S3 d'erreur que vous avez spécifié lors de la création du stream Firehose. Outre l'enregistrement, l'objet livré inclut également un code d'erreur et un message d'erreur et ces objets seront livrés à un préfixe de compartiment S3 appelé `decompression-failed`. Firehose continuera à traiter les autres enregistrements après l'échec de la décompression d'un enregistrement.

Qu'advient-il des données sources en cas d'erreur dans le pipeline de traitement après une décompression réussie ?

Si Amazon Data Firehose échoue dans les étapes de traitement après la décompression, telles que le partitionnement dynamique et la conversion du format de données, l'enregistrement est envoyé au format compressé dans le compartiment S3 d'erreur que vous avez spécifié lors de la création du stream Firehose. Outre l'enregistrement, l'objet livré inclut également un code d'erreur et un message d'erreur.

Comment êtes-vous informé en cas d'erreur ou d'exception ?

En cas d'erreur ou d'exception lors de la décompression, si vous configurez les CloudWatch journaux, Firehose enregistrera les messages CloudWatch d'erreur dans les journaux. En outre, Firehose envoie des métriques à des CloudWatch métriques que vous pouvez surveiller. Vous pouvez également éventuellement créer des alarmes en fonction des métriques émises par Firehose.

Que se passe-t-il lorsque **put** les opérations ne proviennent pas CloudWatch des journaux ?

Lorsque le client `puts` ne provient pas de CloudWatch Logs, le message d'erreur suivant est renvoyé :

```
Put to Firehose failed for AccountId: <accountID>, FirehoseName: <firehosename> because the request is not originating from allowed source types.
```

Quelles mesures Firehose émet-elle pour la fonction de décompression ?

Firehose émet des métriques pour la décompression de chaque enregistrement. Vous devez sélectionner la période (1 min), la statistique (somme), la plage de dates pour obtenir le nombre

d'échecs ou de `DecompressedRecords` réussites ou d'`DecompressedBytes` échecs ou de réussites. Pour plus d'informations, voir [CloudWatch Métriques de décompression des journaux](#).

Écrire sur Amazon Data Firehose à l'aide d'événements CloudWatch

Vous pouvez configurer Amazon CloudWatch pour envoyer des événements à un flux Firehose en ajoutant une cible à une règle d' CloudWatch événements.

Pour créer une cible pour une règle d' CloudWatch événements qui envoie des événements à un flux Firehose existant

1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).
2. Choisissez Créer une règle.
3. Sur la page Étape 1 : Créer une règle, pour Targets, choisissez Ajouter une cible, puis Firehose stream.
4. Choisissez un stream Firehose existant.

Pour plus d'informations sur la création de règles relatives aux CloudWatch événements, consultez [Getting Started with Amazon CloudWatch Events](#).

Écrire sur Amazon Data Firehose à l'aide de AWS IoT

Vous pouvez configurer AWS IoT pour envoyer des informations à un flux Firehose en ajoutant une action.

Pour créer une action qui envoie des événements à un stream Firehose existant

1. Lorsque vous créez une règle dans la console AWS IoT, sur la page Créer une règle, sous Définissez une ou plusieurs actions, choisissez Ajouter action.
2. Choisissez Envoyer des messages à un flux Amazon Kinesis Firehose.
3. Choisissez Configurer une action.
4. Pour le nom du stream, choisissez un stream Firehose existant.
5. Dans le champ Séparateur, choisissez un caractère de séparation à insérer entre les enregistrements.

6. Dans le champ Nom du rôle IAM, sélectionnez un rôle IAM existant ou choisissez Créer un rôle.
7. Choisissez Add action.

Pour plus d'informations sur la création de règles AWS IoT, consultez [AWSDidacticiels sur les règles IoT](#).

Sécurité dans Amazon Data Firehose

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficierez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. L'efficacité de notre sécurité est régulièrement testée et vérifiée par des auditeurs tiers dans le cadre des [programmes de conformité AWS](#). Pour en savoir plus sur les programmes de conformité qui s'appliquent à Data Firehose, consultez la section [AWS Services concernés par programme de conformité](#).
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre organisation ainsi que les lois et réglementations applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de Data Firehose. Les rubriques suivantes expliquent comment configurer Data Firehose pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui peuvent vous aider à surveiller et à sécuriser vos ressources Data Firehose.

Rubriques

- [Protection des données dans Amazon Data Firehose](#)
- [Contrôler l'accès avec Amazon Data Firehose](#)
- [Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose](#)
- [Gérez les rôles IAM via la console Amazon Data Firehose](#)
- [Surveillance d'Amazon Data Firehose](#)
- [Validation de conformité pour Amazon Data Firehose](#)
- [Résilience dans Amazon Data Firehose](#)
- [Sécurité de l'infrastructure dans Amazon Data Firehose](#)

- [Bonnes pratiques de sécurité pour Amazon Data Firehose](#)

Protection des données dans Amazon Data Firehose

Amazon Data Firehose chiffre toutes les données en transit à l'aide du protocole TLS. En outre, pour les données stockées dans un espace de stockage intermédiaire pendant le traitement, Amazon Data Firehose chiffre les données à l'aide de la vérification par somme de contrôle [AWS Key Management Service](#) et vérifie leur intégrité.

Si vous avez des données sensibles, vous pouvez activer le chiffrement des données côté serveur lorsque vous utilisez Amazon Data Firehose. La méthode utilisée dépend de la source de vos données.

Note

Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

Chiffrement côté serveur avec Kinesis Data Streams comme source de données

Lorsque vous envoyez des données de vos producteurs de données vers votre flux de données, Kinesis Data Streams chiffre vos données à l'aide AWS Key Management Service d'une clé AWS KMS() avant de les stocker au repos. Lorsque votre flux Firehose lit les données de votre flux de données, Kinesis Data Streams déchiffre d'abord les données, puis les envoie à Amazon Data Firehose. Amazon Data Firehose met en mémoire tampon les données en mémoire en fonction des indications de mise en mémoire tampon que vous spécifiez. Il diffuse ensuite à vos destinations sans avoir à stocker les données non chiffrées au repos.

Pour plus d'informations sur l'activation du chiffrement côté serveur pour Kinesis Data Streams, consultez [Using Server-Side Encryption](#) dans le Guide du développeur Amazon Kinesis Data Streams.

Chiffrement côté serveur avec l'instruction PUT directe ou d'autres sources de données

Si vous envoyez des données vers votre flux Firehose en utilisant [PutRecord](#) ou [PutRecordBatch](#), ou si vous envoyez les données en utilisant AWS IoT Amazon CloudWatch Logs ou CloudWatch Events, vous pouvez activer le chiffrement côté serveur à l'aide de cette opération. [StartDeliveryStreamEncryption](#)

Pour arrêter server-side-encryption, utilisez l'[StopDeliveryStreamEncryption](#) opération.

Vous pouvez également activer SSE lorsque vous créez le flux Firehose. Pour ce faire, spécifiez à [DeliveryStreamEncryptionConfigurationInput](#) quel moment vous invoquez [CreateDeliveryStream](#).

Lorsque le CMK est de type `CUSTOMER_MANAGED_CMK`, si le service Amazon Data Firehose n'est pas en mesure de déchiffrer les enregistrements à cause de `KMSNotFoundException`, `KMSInvalidStateException`, `KMSDisabledException` ou `KMSAccessDeniedException`, le service attend jusqu'à 24 heures (période de rétention) pour que vous résolviez le problème. Si le problème persiste au-delà de la période de rétention, le service ignore les enregistrements qui ont dépassé la période de rétention et n'ont pas pu être déchiffrés, puis supprime les données. Amazon Data Firehose fournit les quatre CloudWatch indicateurs suivants que vous pouvez utiliser pour suivre les quatre AWS KMS exceptions :

- `KMSKeyAccessDenied`
- `KMSKeyDisabled`
- `KMSKeyInvalidState`
- `KMSKeyNotFound`

Pour plus d'informations sur ces quatre métriques, consultez [the section called “Surveillance à l'aide de CloudWatch métriques”](#).

Important

Pour chiffrer votre flux Firehose, utilisez des CMK symétriques. Amazon Data Firehose ne prend pas en charge les CMK asymétriques. Pour plus d'informations sur les CMK symétriques et asymétriques, voir [À propos des CMK symétriques et asymétriques dans le guide du développeur](#). AWS Key Management Service

Note

Lorsque vous utilisez une [clé gérée par le client](#) (CUSTOMER_MANAGED_CMK) pour activer le chiffrement côté serveur (SSE) pour votre flux Firehose, le service Firehose définit un contexte de chiffrement chaque fois qu'il utilise votre clé. Étant donné que ce contexte de chiffrement représente un cas d'utilisation d'une clé appartenant à votre AWS compte, il est enregistré dans les journaux d' AWS CloudTrail événements de votre AWS compte. Ce contexte de chiffrement est généré par le système par le service Firehose. Votre application ne doit émettre aucune hypothèse quant au format ou au contenu du contexte de chiffrement défini par le service Firehose.

Contrôler l'accès avec Amazon Data Firehose

Les sections suivantes expliquent comment contrôler l'accès à et depuis vos ressources Amazon Data Firehose. Les informations qu'ils couvrent incluent la manière d'accorder l'accès à votre application afin qu'elle puisse envoyer des données à votre flux Firehose. Ils décrivent également comment vous pouvez accorder à Amazon Data Firehose l'accès à votre bucket Amazon Simple Storage Service (Amazon S3), à votre cluster Amazon Redshift ou à votre cluster Amazon OpenSearch Service, ainsi que les autorisations d'accès dont vous avez besoin si vous utilisez Datadog, Dynatrace, LogicMonitor MongoDB, New Relic, Splunk ou Sumo Logic comme destination. Enfin, vous trouverez dans cette rubrique des conseils sur la façon de configurer Amazon Data Firehose afin qu'il puisse envoyer des données vers une destination appartenant à un autre AWS compte. La technologie permettant de gérer toutes ces formes d'accès est AWS Identity and Access Management (IAM). Pour plus d'informations sur IAM, consultez [En quoi consiste IAM ?](#).

Table des matières

- [Accordez à votre application l'accès à vos ressources Amazon Data Firehose](#)
- [Accordez à Amazon Data Firehose l'accès à votre cluster Amazon MSK privé](#)
- [Autoriser Amazon Data Firehose à assumer un rôle IAM](#)
- [Accorder à Amazon Data Firehose l'accès à des AWS Glue fins de conversion de formats de données](#)
- [Accorder à Amazon Data Firehose l'accès à une destination Amazon S3](#)
- [Accorder à Amazon Data Firehose l'accès à une destination Amazon Redshift](#)
- [Accorder à Amazon Data Firehose l'accès à une destination de service public OpenSearch](#)

- [Accorder à Amazon Data Firehose l'accès à une destination de OpenSearch service dans un VPC](#)
- [Accorder à Amazon Data Firehose l'accès à une destination publique sans serveur OpenSearch](#)
- [Accorder à Amazon Data Firehose l'accès à une destination OpenSearch sans serveur dans un VPC](#)
- [Accorder à Amazon Data Firehose l'accès à une destination Splunk](#)
- [Accès à Splunk dans un VPC](#)
- [Accès à Snowflake ou au point de terminaison HTTP](#)
- [Accorder à Amazon Data Firehose l'accès à une destination Snowflake](#)
- [Accès à Snowflake en VPC](#)
- [Accorder à Amazon Data Firehose l'accès à une destination de point de terminaison HTTP](#)
- [Livraison entre comptes depuis Amazon MSK](#)
- [Diffusion entre comptes vers une destination Amazon S3](#)
- [Livraison entre comptes vers une destination OpenSearch de service](#)
- [Utilisation des balises pour contrôler l'accès](#)

Accordez à votre application l'accès à vos ressources Amazon Data Firehose

Pour permettre à votre application d'accéder à votre stream Firehose, utilisez une politique similaire à cet exemple. Vous pouvez ajuster des opérations d'API individuelles auxquelles vous accordez l'accès en modifiant la section `Action` ou accorder l'accès à toutes les opérations avec `"firehose:*"`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "firehose:DeleteDeliveryStream",
        "firehose:PutRecord",
        "firehose:PutRecordBatch",
        "firehose:UpdateDestination"
      ],
      "Resource": [
```

```

        "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"
    ]
}

```

Accordez à Amazon Data Firehose l'accès à votre cluster Amazon MSK privé

Si la source de votre flux Firehose est un cluster Amazon MSK privé, utilisez une politique similaire à cet exemple.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "Service": [
          "firehose.amazonaws.com"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "kafka:CreateVpcConnection"
      ],
      "Resource": "cluster-arn"
    }
  ]
}

```

Autoriser Amazon Data Firehose à assumer un rôle IAM

Cette section décrit les autorisations et les politiques qui accordent à Amazon Data Firehose l'accès pour ingérer, traiter et transmettre des données de la source à la destination.

Note

Si vous utilisez la console pour créer un stream Firehose et que vous choisissez l'option de création d'un nouveau rôle, associez AWS la politique de confiance requise au rôle. Si

vous souhaitez qu'Amazon Data Firehose utilise un rôle IAM existant ou si vous créez vous-même un rôle, associez les politiques de confiance suivantes à ce rôle afin qu'Amazon Data Firehose puisse l'assumer. Modifiez les politiques pour remplacer *account-id par votre identifiant* de AWS compte. Pour de plus amples informations sur la modification de la relation d'approbation d'un rôle, veuillez consulter [Modification d'un rôle](#).

Amazon Data Firehose utilise un rôle IAM pour toutes les autorisations dont le flux Firehose a besoin pour traiter et diffuser des données. Assurez-vous que les politiques de confiance suivantes sont associées à ce rôle afin qu'Amazon Data Firehose puisse l'assumer.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": "firehose.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "sts:ExternalId": "account-id"
      }
    }
  }]
}
```

Cette politique utilise la clé de contexte de `sts:ExternalId` condition pour garantir que seule l'activité Amazon Data Firehose provenant de votre AWS compte peut assumer ce rôle IAM. Pour de plus amples informations sur la prévention de l'utilisation non autorisée des rôles IAM, consultez [Le problème de l'adjoint confus](#) dans le Guide de l'utilisateur IAM.

Si vous choisissez Amazon MSK comme source pour votre flux Firehose, vous devez spécifier un autre rôle IAM qui accorde à Amazon Data Firehose l'autorisation d'ingérer les données sources du cluster Amazon MSK spécifié. Assurez-vous que les politiques de confiance suivantes sont associées à ce rôle afin qu'Amazon Data Firehose puisse l'assumer.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Principal": {
      "Service": [
        "firehose.amazonaws.com"
      ]
    },
    "Effect": "Allow",
    "Action": "sts:AssumeRole"
  }
]
```

Assurez-vous que ce rôle qui accorde à Amazon Data Firehose l'autorisation d'ingérer les données sources du cluster Amazon MSK spécifié accorde les autorisations suivantes :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "kafka:GetBootstrapBrokers",
      "kafka:DescribeCluster",
      "kafka:DescribeClusterV2",
      "kafka-cluster:Connect"
    ],
    "Resource": "CLUSTER-ARN"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kafka-cluster:DescribeTopic",
      "kafka-cluster:DescribeTopicDynamicConfiguration",
      "kafka-cluster:ReadData"
    ],
    "Resource": "TOPIC-ARN"
  }
]
```

Accorder à Amazon Data Firehose l'accès à des AWS Glue fins de conversion de formats de données

Si votre flux Firehose effectue une conversion de format de données, Amazon Data Firehose fait référence aux définitions de table stockées dans AWS Glue. Pour accorder à Amazon Data Firehose l'accès nécessaire à AWS Glue, ajoutez la déclaration suivante à votre politique. Pour plus d'informations sur la manière de trouver l'ARN de la table, consultez la section [Specifying AWS Glue Resource ARNs](#).

```
[{
  "Effect": "Allow",
  "Action": [
    "glue:GetTable",
    "glue:GetTableVersion",
    "glue:GetTableVersions"
  ],
  "Resource": "table-arn"
}, {
  "Sid": "GetSchemaVersion",
  "Effect": "Allow",
  "Action": [
    "glue:GetSchemaVersion"
  ],
  "Resource": ["*"]
}]
```

La politique recommandée pour obtenir des schémas à partir du registre des schémas ne comporte aucune restriction de ressources. Pour plus d'informations, consultez les [exemples IAM pour les désérialiseurs](#) dans le Guide du développeur. AWS Glue

Note

Actuellement, AWS Glue n'est pas pris en charge dans les régions d'Israël (Tel Aviv), de l'Asie-Pacifique (Jakarta) ou du Moyen-Orient (Émirats arabes unis). Si vous travaillez avec Amazon Data Firehose dans la région Asie-Pacifique (Jakarta) ou au Moyen-Orient (Émirats arabes unis), assurez-vous d'autoriser votre Amazon Data Firehose à accéder à l'une des régions actuellement prises en charge par AWS Glue. L'interopérabilité entre les régions entre Data Firehose AWS Glue est prise en charge. Pour plus d'informations sur

les régions AWS Glue prises en charge, consultez <https://docs.aws.amazon.com/general/latest/gr/glue.html>

Accorder à Amazon Data Firehose l'accès à une destination Amazon S3

Lorsque vous utilisez une destination Amazon S3, Amazon Data Firehose fournit des données à votre compartiment S3 et peut éventuellement utiliser une AWS KMS clé que vous possédez pour le chiffrement des données. Si la journalisation des erreurs est activée, Amazon Data Firehose envoie également des erreurs de livraison de données à votre groupe de CloudWatch journaux et à vos flux. Vous devez disposer d'un rôle IAM lors de la création d'un stream Firehose. Amazon Data Firehose assume le rôle IAM et accède au bucket, à la clé, au groupe de CloudWatch journaux et aux flux spécifiés.

Utilisez la politique d'accès suivante pour permettre à Amazon Data Firehose d'accéder à votre compartiment et AWS KMS à votre clé S3. Si vous n'êtes pas propriétaire du compartiment S3, ajoutez `s3:PutObjectACL` à la liste des actions Amazon S3. Cela donne au propriétaire du bucket un accès complet aux objets fournis par Amazon Data Firehose.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
```

```

        "kinesis:DescribeStream",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
},
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "arn:aws:kms:region:account-id:key/key-id"
    ],
    "Condition": {
        "StringEquals": {
            "kms:ViaService": "s3.region.amazonaws.com"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "logs:PutLogEvents"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-
stream-name"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction",
        "lambda:GetFunctionConfiguration"
    ],
    "Resource": [

```

```

        "arn:aws:lambda:region:account-id:function:function-name:function-
version"
    ]
}
]
}

```

La politique ci-dessus contient également une déclaration qui autorise l'accès à Amazon Kinesis Data Streams. Si vous n'utilisez pas Kinesis Data Streams comme source de données, vous pouvez supprimer cette déclaration. Si vous utilisez Amazon MSK comme source, vous pouvez remplacer cette déclaration par la suivante :

```

{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka:GetBootstrapBrokers",
    "kafka:DescribeCluster",
    "kafka:DescribeClusterV2",
    "kafka-cluster:Connect"
  ],
  "Resource": "arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:cluster/
{{mskClusterName}}/{{clusterUUID}}"
},
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:ReadData"
  ],
  "Resource": "arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:topic/
{{mskClusterName}}/{{clusterUUID}}/{{mskTopicName}}"
},
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka-cluster:DescribeGroup"
  ],
  "Resource": "arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:group/
{{mskClusterName}}/{{clusterUUID}}/*"
}

```

```
}
```

Pour plus d'informations sur l'autorisation d'autres AWS services d'accéder à vos AWS ressources, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

Pour savoir comment accorder à Amazon Data Firehose l'accès à une destination Amazon S3 via un autre compte, consultez [the section called "Diffusion entre comptes vers une destination Amazon S3"](#)

Accorder à Amazon Data Firehose l'accès à une destination Amazon Redshift

Reportez-vous à ce qui suit lorsque vous accordez l'accès à Amazon Data Firehose lorsque vous utilisez une destination Amazon Redshift.

Rubriques

- [Rôle IAM et stratégie d'accès](#)
- [Accès VPC à un cluster provisionné Amazon Redshift ou à un groupe de travail Amazon Redshift sans serveur](#)

Rôle IAM et stratégie d'accès

Lorsque vous utilisez une destination Amazon Redshift, Amazon Data Firehose fournit des données à votre compartiment S3 en tant qu'emplacement intermédiaire. Il peut éventuellement utiliser une AWS KMS clé que vous possédez pour le chiffrement des données. Amazon Data Firehose charge ensuite les données depuis le compartiment S3 vers votre cluster provisionné Amazon Redshift ou votre groupe de travail Amazon Redshift Serverless. Si la journalisation des erreurs est activée, Amazon Data Firehose envoie également des erreurs de livraison de données à votre groupe de CloudWatch journaux et à vos flux. Amazon Data Firehose utilise le nom d'utilisateur et le mot de passe Amazon Redshift spécifiés pour accéder à votre cluster provisionné ou à votre groupe de travail Amazon Redshift Serverless, et utilise un rôle IAM pour accéder au bucket, à la clé, au groupe de journaux et aux flux spécifiés. CloudWatch Vous devez disposer d'un rôle IAM lors de la création d'un stream Firehose.

Utilisez la politique d'accès suivante pour permettre à Amazon Data Firehose d'accéder à votre compartiment et AWS KMS à votre clé S3. Si vous ne possédez pas le compartiment S3, ajoutez-le `s3:PutObjectACL` à la liste des actions Amazon S3, qui accordent au propriétaire du compartiment

un accès complet aux objets fournis par Amazon Data Firehose. Cette politique contient également une déclaration qui autorise l'accès à Amazon Kinesis Data Streams. Si vous n'utilisez pas Kinesis Data Streams comme source de données, vous pouvez supprimer cette déclaration.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:region:account-id:key/key-id"
      ],
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.region.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
        }
      }
    }
  ]
}
```

```

    "Effect": "Allow",
    "Action": [
        "kinesis:DescribeStream",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
},
{
    "Effect": "Allow",
    "Action": [
        "logs:PutLogEvents"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-stream-name"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction",
        "lambda:GetFunctionConfiguration"
    ],
    "Resource": [
        "arn:aws:lambda:region:account-id:function:function-name:function-version"
    ]
}
]
}

```

Pour plus d'informations sur l'autorisation d'autres AWS services d'accéder à vos AWS ressources, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

Accès VPC à un cluster provisionné Amazon Redshift ou à un groupe de travail Amazon Redshift sans serveur

Si votre cluster provisionné Amazon Redshift ou votre groupe de travail Amazon Redshift sans serveur se trouve dans un cloud privé virtuel (VPC), il doit être accessible publiquement avec

une adresse IP publique. Accordez également à Amazon Data Firehose l'accès à votre cluster provisionné Amazon Redshift ou à votre groupe de travail Amazon Redshift Serverless en débloquant les adresses IP Amazon Data Firehose. Amazon Data Firehose utilise actuellement un bloc CIDR pour chaque région disponible :

- 13.58.135.96/27 pour USA Est (Ohio)
- 52.70.63.192/27 pour USA Est (Virginie du Nord)
- 13.57.135.192/27 pour USA Ouest (Californie du Nord)
- 52.89.255.224/27 pour USA Ouest (Oregon)
- 18.253.138.96/27 pour AWS GovCloud (USA Est)
- 52.61.204.160/27 pour AWS GovCloud (US-West)
- 35.183.92.128/27 pour Canada (Centre)
- 40.176.98.192/27 pour Canada West (Calgary)
- 18.162.221.32/27 pour Asie-Pacifique (Hong Kong)
- 13.232.67.32/27 pour Asie-Pacifique (Mumbai)
- 18.60.192.128/27 pour Asie-Pacifique (Hyderabad)
- 13.209.1.64/27 pour Asie-Pacifique (Séoul)
- 13.228.64.192/27 pour Asie-Pacifique (Singapour)
- 13.210.67.224/27 pour Asie-Pacifique (Sydney)
- 108.136.221.64/27 pour Asie-Pacifique (Jakarta)
- 13.113.196.224/27 pour Asie-Pacifique (Tokyo)
- 13.208.177.192/27 pour Asie-Pacifique (Osaka)
- 52.81.151.32/27 pour la Chine (Beijing)
- 161.189.23.64/27 pour Chine (Ningxia)
- 16.62.183.32/27 pour Europe (Zurich)
- 35.158.127.160/27 pour Europe (Francfort)
- 52.19.239.192/27 pour Europe (Irlande)
- 18.130.1.96/27 pour Europe (Londres)
- 35.180.1.96/27 pour Europe (Paris)
- 13.53.63.224/27 pour Europe (Stockholm)
- 15.185.91.0/27 pour Moyen-Orient (Bahreïn)

- 18.228.1.128/27 pour Amérique du Sud (São Paulo)
- 15.161.135.128/27 pour Europe (Milan)
- 13.244.121.224/27 pour Afrique (Le Cap)
- 3.28.159.32/27 pour Moyen-Orient (EAU)
- 51.16.102.0/27 pour Israël (Tel Aviv)
- 16.50.161.128/27 pour Asie-Pacifique (Melbourne)

Pour en savoir plus sur la façon de débloquent les adresses IP, consultez l'étape [Authorize Access to the Cluster](#) dans le Guide de démarrage Amazon Redshift.

Accorder à Amazon Data Firehose l'accès à une destination de service public OpenSearch

Lorsque vous utilisez une destination de OpenSearch service, Amazon Data Firehose fournit des données à votre cluster de OpenSearch services et sauvegarde simultanément tous les documents défaillants dans votre compartiment S3. Si la journalisation des erreurs est activée, Amazon Data Firehose envoie également des erreurs de livraison de données à votre groupe de CloudWatch journaux et à vos flux. Amazon Data Firehose utilise un rôle IAM pour accéder au domaine de OpenSearch service, au compartiment S3, à la AWS KMS clé, au groupe de CloudWatch journaux et aux flux spécifiés. Vous devez disposer d'un rôle IAM lors de la création d'un stream Firehose.

Utilisez la politique d'accès suivante pour permettre à Amazon Data Firehose d'accéder à votre compartiment S3, à votre domaine de OpenSearch service et AWS KMS à votre clé. Si vous ne possédez pas le compartiment S3, ajoutez-le `s3:PutObjectACL` à la liste des actions Amazon S3, ce qui accorde au propriétaire du compartiment un accès complet aux objets fournis par Amazon Data Firehose. Cette politique contient également une déclaration qui autorise l'accès à Amazon Kinesis Data Streams. Si vous n'utilisez pas Kinesis Data Streams comme source de données, vous pouvez supprimer cette déclaration.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
```

```

        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "arn:aws:kms:region:account-id:key/key-id"
    ],
    "Condition": {
        "StringEquals": {
            "kms:ViaService": "s3.region.amazonaws.com"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "es:DescribeDomain",
        "es:DescribeDomains",
        "es:DescribeDomainConfig",
        "es:ESHttpPost",
        "es:ESHttpPut"
    ],
    "Resource": [
        "arn:aws:es:region:account-id:domain/domain-name",
        "arn:aws:es:region:account-id:domain/domain-name/*"
    ]
},
{

```

```

    "Effect": "Allow",
    "Action": [
      "es:ESHttpGet"
    ],
    "Resource": [
      "arn:aws:es:region:account-id:domain/domain-name/_all/_settings",
      "arn:aws:es:region:account-id:domain/domain-name/_cluster/stats",
      "arn:aws:es:region:account-id:domain/domain-name/index-name*/
_mapping/type-name",
      "arn:aws:es:region:account-id:domain/domain-name/_nodes",
      "arn:aws:es:region:account-id:domain/domain-name/_nodes/stats",
      "arn:aws:es:region:account-id:domain/domain-name/_nodes/*/stats",
      "arn:aws:es:region:account-id:domain/domain-name/_stats",
      "arn:aws:es:region:account-id:domain/domain-name/index-name*/_stats",
      "arn:aws:es:region:account-id:domain/domain-name/"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:GetShardIterator",
      "kinesis:GetRecords",
      "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents"
    ],
    "Resource": [
      "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-
stream-name"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "lambda:InvokeFunction",
      "lambda:GetFunctionConfiguration"
    ],
    "Resource": [

```

```
        "arn:aws:lambda:region:account-id:function:function-name:function-  
version"  
      ]  
    }  
  ]  
}
```

Pour plus d'informations sur l'autorisation d'autres AWS services d'accéder à vos AWS ressources, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

Pour savoir comment accorder à Amazon Data Firehose l'accès à un cluster de OpenSearch services dans un autre compte, consultez [the section called "Livraison entre comptes vers une destination OpenSearch de service"](#)

Accorder à Amazon Data Firehose l'accès à une destination de OpenSearch service dans un VPC

Si votre domaine de OpenSearch service se trouve dans un VPC, assurez-vous d'accorder à Amazon Data Firehose les autorisations décrites dans la section précédente. En outre, vous devez accorder à Amazon Data Firehose les autorisations suivantes pour lui permettre d'accéder au VPC de votre domaine OpenSearch de service.

- `ec2:DescribeVpcs`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`
- `ec2>DeleteNetworkInterface`

Important

Ne révoquez pas ces autorisations après avoir créé le stream Firehose. Si vous révoquez ces autorisations, votre flux Firehose sera dégradé ou cessera de fournir des données à

OpenSearch votre domaine de service chaque fois que le service tentera d'interroger ou de mettre à jour les ENI.

Important

Lorsque vous spécifiez des sous-réseaux pour fournir des données à la destination dans un VPC privé, assurez-vous de disposer d'un nombre suffisant d'adresses IP libres dans les sous-réseaux sélectionnés. Si aucune adresse IP gratuite n'est disponible dans un sous-réseau spécifié, Firehose ne peut pas créer ou ajouter d'ENI pour la livraison des données dans le VPC privé, et la livraison sera dégradée ou échouera.

Lorsque vous créez ou mettez à jour votre flux Firehose, vous spécifiez un groupe de sécurité que Firehose doit utiliser lorsqu'il envoie des données à votre domaine de service. OpenSearch Vous pouvez utiliser le même groupe de sécurité que celui utilisé par le domaine de OpenSearch service ou un autre. Si vous spécifiez un autre groupe de sécurité, assurez-vous qu'il autorise le trafic HTTPS sortant vers le groupe de sécurité du domaine de OpenSearch service. Assurez-vous également que le groupe de sécurité du domaine de OpenSearch service autorise le trafic HTTPS provenant du groupe de sécurité que vous avez spécifié lors de la configuration de votre flux Firehose. Si vous utilisez le même groupe de sécurité pour votre stream Firehose et pour le domaine de OpenSearch service, assurez-vous que la règle d'entrée du groupe de sécurité autorise le trafic HTTPS. Pour plus d'informations sur les règles des groupes de sécurité, consultez [Règles des groupes de sécurité](#) dans la documentation Amazon VPC.

Accorder à Amazon Data Firehose l'accès à une destination publique sans serveur OpenSearch

Lorsque vous utilisez une destination OpenSearch sans serveur, Amazon Data Firehose fournit des données à OpenSearch votre collection sans serveur et sauvegarde simultanément les documents défaillants ou tous les documents dans votre compartiment S3. Si la journalisation des erreurs est activée, Amazon Data Firehose envoie également des erreurs de livraison de données à votre groupe de CloudWatch journaux et à vos flux. Amazon Data Firehose utilise un rôle IAM pour accéder à la collection OpenSearch Serverless, au compartiment S3, à la AWS KMS clé, au groupe de CloudWatch journaux et aux flux spécifiés. Vous devez disposer d'un rôle IAM lors de la création d'un stream Firehose.

Utilisez la politique d'accès suivante pour permettre à Amazon Data Firehose d'accéder à votre compartiment S3, à votre domaine OpenSearch Serverless et à votre clé AWS KMS. Si vous ne possédez pas le compartiment S3, ajoutez-le `s3:PutObject` à la liste des actions Amazon S3, ce qui accorde au propriétaire du compartiment un accès complet aux objets fournis par Amazon Data Firehose. Cette politique contient également une déclaration qui autorise l'accès à Amazon Kinesis Data Streams. Si vous n'utilisez pas Kinesis Data Streams comme source de données, vous pouvez supprimer cette déclaration.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:region:account-id:key/key-id"
      ],
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.region.amazonaws.com"
        },
        "StringLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/prefix*"
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:GetShardIterator",
      "kinesis:GetRecords",
      "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents"
    ],
    "Resource": [
      "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-stream-name"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "lambda:InvokeFunction",
      "lambda:GetFunctionConfiguration"
    ],
    "Resource": [
      "arn:aws:lambda:region:account-id:function:function-name:function-version"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "aoss:APIAccessAll",
    "Resource": "arn:aws:aoss:region:account-id:collection/collection-id"
  }
]
}

```

Outre la politique ci-dessus, vous devez également configurer Amazon Data Firehose pour que les autorisations minimales suivantes soient attribuées dans une politique d'accès aux données :

```
[
  {
    "Rules":[
      {
        "ResourceType":"index",
        "Resource":[
          "index/target-collection/target-index"
        ],
        "Permission":[
          "aoss:WriteDocument",
          "aoss:UpdateIndex",
          "aoss>CreateIndex"
        ]
      }
    ],
    "Principal":[
      "arn:aws:sts::account-id:assumed-role/firehose-delivery-role-name/*"
    ]
  }
]
```

Pour plus d'informations sur l'autorisation d'autres AWS services d'accéder à vos AWS ressources, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

Accorder à Amazon Data Firehose l'accès à une destination OpenSearch sans serveur dans un VPC

Si votre collection OpenSearch Serverless se trouve dans un VPC, assurez-vous d'accorder à Amazon Data Firehose les autorisations décrites dans la section précédente. En outre, vous devez accorder à Amazon Data Firehose les autorisations suivantes pour lui permettre d'accéder au VPC de votre collection OpenSearch Serverless.

- `ec2:DescribeVpcs`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeSubnets`

- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`
- `ec2>DeleteNetworkInterface`

 Important

Ne révoquez pas ces autorisations après avoir créé le stream Firehose. Si vous révoquez ces autorisations, votre flux Firehose sera dégradé ou cessera de fournir des données à OpenSearch votre domaine de service chaque fois que le service tentera d'interroger ou de mettre à jour les ENI.

 Important

Lorsque vous spécifiez des sous-réseaux pour fournir des données à la destination dans un VPC privé, assurez-vous de disposer d'un nombre suffisant d'adresses IP libres dans les sous-réseaux sélectionnés. Si aucune adresse IP gratuite n'est disponible dans un sous-réseau spécifié, Firehose ne peut pas créer ou ajouter d'ENI pour la livraison des données dans le VPC privé, et la livraison sera dégradée ou échouera.

Lorsque vous créez ou mettez à jour votre flux Firehose, vous spécifiez un groupe de sécurité que Firehose doit utiliser lorsqu'il envoie des données à votre collection Serverless. OpenSearch Vous pouvez utiliser le même groupe de sécurité que celui utilisé par la collection OpenSearch Serverless ou un autre. Si vous spécifiez un autre groupe de sécurité, assurez-vous qu'il autorise le trafic HTTPS sortant vers le groupe de sécurité de la collection OpenSearch Serverless. Assurez-vous également que le groupe de sécurité de la collection OpenSearch Serverless autorise le trafic HTTPS provenant du groupe de sécurité que vous avez spécifié lors de la configuration de votre stream Firehose. Si vous utilisez le même groupe de sécurité pour votre flux Firehose et pour la collection OpenSearch Serverless, assurez-vous que la règle entrante du groupe de sécurité autorise le trafic HTTPS. Pour plus d'informations sur les règles des groupes de sécurité, consultez [Règles des groupes de sécurité](#) dans la documentation Amazon VPC.

Accorder à Amazon Data Firehose l'accès à une destination Splunk

Lorsque vous utilisez une destination Splunk, Amazon Data Firehose fournit des données à votre point de terminaison Splunk HTTP Event Collector (HEC). Il sauvegarde également ces données dans le compartiment Amazon S3 que vous spécifiez, et vous pouvez éventuellement utiliser une AWS KMS clé que vous possédez pour le chiffrement côté serveur Amazon S3. Si la journalisation des erreurs est activée, Firehose envoie des erreurs de livraison de données à vos flux de CloudWatch journaux. Vous pouvez également l'utiliser AWS Lambda pour la transformation des données.

Si vous utilisez un équilibreur de AWS charge, assurez-vous qu'il s'agit d'un Classic Load Balancer ou d'un Application Load Balancer. Activez également les sessions persistantes basées sur la durée avec l'expiration des cookies désactivée pour Classic Load Balancer et l'expiration fixée au maximum (7 jours) pour Application Load Balancer. [Pour plus d'informations sur la procédure à suivre, consultez la section *Stickiness de session basée sur la durée pour un Classic Load Balancer ou un Application Load Balancer*.](#)

Vous devez avoir un rôle IAM lorsque vous créez un stream Firehose. Firehose assume le rôle IAM et accède au bucket, à la clé, au groupe de CloudWatch journaux et aux flux spécifiés.

Utilisez la politique d'accès suivante pour permettre à Amazon Data Firehose d'accéder à votre compartiment S3. Si vous ne possédez pas le compartiment S3, ajoutez-le `s3:PutObjectACL` à la liste des actions Amazon S3, qui accordent au propriétaire du compartiment un accès complet aux objets fournis par Amazon Data Firehose. Cette politique accorde également à Amazon Data Firehose l'accès à la journalisation CloudWatch des erreurs et à la transformation AWS Lambda des données. La politique contient également une déclaration qui autorise l'accès à Amazon Kinesis Data Streams. Si vous n'utilisez pas Kinesis Data Streams comme source de données, vous pouvez supprimer cette déclaration. Amazon Data Firehose n'utilise pas IAM pour accéder à Splunk. Pour accéder à Splunk, il utilise votre jeton HEC.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
```

```

        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "arn:aws:kms:region:account-id:key/key-id"
    ],
    "Condition": {
        "StringEquals": {
            "kms:ViaService": "s3.region.amazonaws.com"
        },
        "StringLike": {
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "kinesis:DescribeStream",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
},
{
    "Effect": "Allow",
    "Action": [
        "logs:PutLogEvents"
    ],
    "Resource": [

```

```
        "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "lambda:InvokeFunction",
        "lambda:GetFunctionConfiguration"
    ],
    "Resource": [
        "arn:aws:lambda:region:account-id:function:function-name:function-
version"
    ]
}
]
```

Pour plus d'informations sur l'autorisation d'autres AWS services d'accéder à vos AWS ressources, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

Accès à Splunk dans un VPC

Si votre plateforme Splunk se trouve dans un VPC, il doit être accessible publiquement avec une adresse IP publique. Accordez également à Amazon Data Firehose l'accès à votre plateforme Splunk en débloquant les adresses IP Amazon Data Firehose. Amazon Data Firehose utilise actuellement les blocs CIDR suivants.

- 18.216.68.160/27, 18.216.170.64/27, 18.216.170.96/27 pour USA Est (Ohio)
- 34.238.188.128/26, 34.238.188.192/26, 34.238.195.0/26 pour USA Est (Virginie du Nord)
- 13.57.180.0/26 pour USA Ouest (Californie du Nord)
- 34.216.24.32/27, 34.216.24.192/27, 34.216.24.224/27 pour USA Ouest (Oregon)
- 18.253.138.192/26 pour AWS GovCloud (USA Est)
- 52.61.204.192/26 pour AWS GovCloud (US-West)
- 18.162.221.64/26 pour Asie-Pacifique (Hong Kong)
- 13.232.67.64/26 pour Asie-Pacifique (Mumbai)
- 13.209.71.0/26 pour Asie-Pacifique (Séoul)

- 13.229.187.128/26 pour Asie-Pacifique (Singapour)
- 13.211.12.0/26 pour Asie-Pacifique (Sydney)
- 13.230.21.0/27, 13.230.21.32/27 pour Asie-Pacifique (Tokyo)
- 51.16.102.64/26 pour Israël (Tel Aviv)
- 35.183.92.64/26 pour Canada (Centre)
- 40.176.98.128/26 pour Canada West (Calgary)
- 18.194.95.192/27, 18.194.95.224/27, 18.195.48.0/27 pour Europe (Francfort)
- 34.241.197.32/27, 34.241.197.64/27, 34.241.197.96/27 pour Europe (Irlande)
- 18.130.91.0/26 pour Europe (Londres)
- 35.180.112.0/26 pour Europe (Paris)
- 13.53.191.0/26 pour Europe (Stockholm)
- 15.185.91.64/26 pour Moyen-Orient (Bahreïn)
- 18.228.1.192/26 pour Amérique du Sud (São Paulo)
- 15.161.135.192/26 pour Europe (Milan)
- 13.244.165.128/26 pour Afrique (Le Cap)
- 13.208.217.0/26 pour Asie-Pacifique (Osaka)
- 52.81.151.64/26 pour la Chine (Beijing)
- 161.189.23.128/26 pour Chine (Ningxia)
- 108.136.221.128/26 pour Asie-Pacifique (Jakarta)
- 3.28.159.64/26 pour Moyen-Orient (EAU)
- 51.16.102.64/26 pour Israël (Tel Aviv)
- 16.62.183.64/26 pour Europe (Zurich)
- 18.60.192.192/26 pour Asie-Pacifique (Hyderabad)
- 16.50.161.192/26 pour Asie-Pacifique (Melbourne)

Accès à Snowflake ou au point de terminaison HTTP

Il n'existe aucun sous-ensemble de [plages d'adresses AWS IP](#) spécifique à Amazon Data Firehose lorsque la destination est un point de terminaison HTTP ou des clusters publics Snowflake.

Pour ajouter Firehose à une liste d'autorisation pour les clusters Snowflake publics ou à vos points de terminaison HTTP ou HTTPS publics, ajoutez toutes les [plages d'adresses AWS IP](#) actuelles à vos règles d'entrée.

Note

Les notifications ne proviennent pas toujours d'adresses IP situées dans la même AWS région que le sujet qui leur est associé. Vous devez inclure la plage d'adresses AWS IP pour toutes les régions.

Accorder à Amazon Data Firehose l'accès à une destination Snowflake

Lorsque vous utilisez Snowflake comme destination, Firehose fournit des données à un compte Snowflake en utilisant l'URL de votre compte Snowflake. Il sauvegarde également les données d'erreur dans le compartiment Amazon Simple Storage Service que vous spécifiez, et vous pouvez éventuellement utiliser une AWS Key Management Service clé que vous possédez pour le chiffrement côté serveur Amazon S3. Si la journalisation des erreurs est activée, Firehose envoie des erreurs de livraison de données à vos flux de CloudWatch journaux.

Vous devez avoir un rôle IAM avant de créer un stream Firehose. Firehose assume le rôle IAM et accède au bucket, à la clé, au groupe de CloudWatch journaux et aux flux spécifiés. Utilisez la politique d'accès suivante pour permettre à Firehose d'accéder à votre compartiment S3. Si vous ne possédez pas le compartiment S3, ajoutez-le `s3:PutObjectAc1` à la liste des actions Amazon Simple Storage Service, qui accordent au propriétaire du compartiment un accès complet aux objets fournis par Firehose. Cette politique accorde également à Firehose l'accès à la journalisation CloudWatch des erreurs. La politique contient également une déclaration qui autorise l'accès à Amazon Kinesis Data Streams. Si vous n'utilisez pas Kinesis Data Streams comme source de données, vous pouvez supprimer cette déclaration. Firehose n'utilise pas IAM pour accéder à Snowflake. Pour accéder à Snowflake, il utilise l'URL de votre compte Snowflake et l'identifiant PrivateLink Vpce dans le cas d'un cluster privé.

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Effect": "Allow",
```

```

    "Action": [
      "s3:AbortMultipartUpload",
      "s3:GetBucketLocation",
      "s3:GetObject",
      "s3:ListBucket",
      "s3:ListBucketMultipartUploads",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource": [
      "arn:aws:kms:region:account-id:key/key-id"
    ],
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.region.amazonaws.com"
      },
      "StringLike": {
        "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/prefix*"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:GetShardIterator",
      "kinesis:GetRecords",
      "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
  },
  {
    "Effect": "Allow",
    "Action": [

```

```

        "logs:PutLogEvents"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:*"
    ]
}
]
}

```

Pour plus d'informations sur l'autorisation d'autres AWS services d'accéder à vos AWS ressources, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

Accès à Snowflake en VPC

Si votre cluster Snowflake est activé par les liens privés, Firehose utilise des points de terminaison VPC pour fournir des données à votre cluster privé sans passer par Internet public. Pour cela, créez des règles réseau Snowflake pour autoriser l'entrée à partir des éléments suivants `AwsVpceIds` pour le cluster dans lequel se trouve Région AWS votre cluster. Pour plus d'informations, consultez la section [Création d'une règle réseau](#) dans le guide de l'utilisateur de Snowflake.

Identifiants de point de terminaison VPC à utiliser en fonction des régions dans lesquelles se trouve votre cluster

Région AWS	VPCE IDs
USA Est (Ohio)	vpce-0d96cafcd96a50aeb
	vpce-0cec34343d48f537b
USA Est (Virginie du Nord)	vpce-0b4d7e8478e141ba8
	vpce-0b75cd681fb507352
	vpce-01c03e63820ec00d8
	vpce-0c2cfc51dc2882422
	vpce-06ca862f019e4e056
	vpce-020cda0cfa63f8d1c

Région AWS	VPCE IDs
	vpce-0b80504a1a783cd70
	vpce-0289b9ff0b5259a96
	vpce-0d7add8628bd69a12
	vpce-02bfb5966cc59b2af
	vpce-09e707674af878bf2
	vpce-049b52e96cc1a2165
	vpce-0bb6c7b7a8a86cdbb
	vpce-03b22d599f51e80f3
	vpce-01d60dc60fc106fe1
	vpce-0186d20a4b24ecbef
	vpce-0533906401a36e416
	vpce-05111fb13d396710e
	vpce-0694613f4fbd6f514
	vpce-09b21cb25fe4cc4f4
	vpce-06029c3550e4d2399
	vpce-00961862a21b033da
	vpce-01620b9ae33273587
	vpce-078cf4ec226880ac9
	vpce-0d711bf076ce56381
	vpce-066b7e13cbfca6f6e
	vpce-0674541252d9ccc26

Région AWS	VPCE IDs
	<ul style="list-style-type: none">vpce-03540b88dedb4b000vpce-0b1828e79ad394b95vpce-0dc0e6f001fb1a60dvpce-0d8f82e71a244098avpce-00e374d9e3f1af5cevpce-0c1e3d6631ddb442f
USA Ouest (Oregon)	<ul style="list-style-type: none">vpce-0f60f72da4cd1e4e7vpce-0c60d21eb8b1669fdvpce-01c4e3e29afdafbefvpce-0cc6bf2a88da139devpce-0797e08e169e50662vpce-033cbe480381b5c0evpce-00debbdd8f9eb10a5vpce-08ec2f386c809e889vpce-0856d14310857b545
Europe (Francfort)	<ul style="list-style-type: none">vpce-068dbb7d71c9460fbvpce-0a7a7f095942d4ec9
Europe (Irlande)	<ul style="list-style-type: none">vpce-06857e59c005a6276vpce-04390f4f8778b75f2vpce-011fd2b1f0aa172fd

Région AWS	VPCE IDs
Asie-Pacifique (Tokyo)	vpce-06369e5258144e68a
	vpce-0f2363cdb8926fbe8
Asie-Pacifique (Singapour)	vpce-049cd46cce7a12d52
	vpce-0e8965a1a4bdb8941
Asie-Pacifique (Séoul)	vpce-0aa444d9001e1faa1
	vpce-04a49d4dcfd02b884
Asie-Pacifique (Sydney)	vpce-048a60a182c52be63
	vpce-03c19949787fd1859

Accorder à Amazon Data Firehose l'accès à une destination de point de terminaison HTTP

Vous pouvez utiliser Amazon Data Firehose pour transmettre des données à n'importe quelle destination de point de terminaison HTTP. Amazon Data Firehose sauvegarde également ces données dans le compartiment Amazon S3 que vous spécifiez, et vous pouvez éventuellement utiliser une AWS KMS clé que vous possédez pour le chiffrement côté serveur Amazon S3. Si la journalisation des erreurs est activée, Amazon Data Firehose envoie les erreurs de livraison de données à vos flux de CloudWatch journaux. Vous pouvez également l'utiliser AWS Lambda pour la transformation des données.

Vous devez disposer d'un rôle IAM lors de la création d'un stream Firehose. Amazon Data Firehose assume le rôle IAM et accède au bucket, à la clé, au groupe de CloudWatch journaux et aux flux spécifiés.

Utilisez la politique d'accès suivante pour permettre à Amazon Data Firehose d'accéder au compartiment S3 que vous avez spécifié pour la sauvegarde des données. Si vous ne possédez pas le compartiment S3, ajoutez-le `s3:PutObjectACL` à la liste des actions Amazon S3, qui accordent au propriétaire du compartiment un accès complet aux objets fournis par Amazon Data Firehose. Cette politique accorde également à Amazon Data Firehose l'accès à la journalisation CloudWatch des erreurs et à la transformation AWS Lambda des données. La politique contient également une

déclaration qui autorise l'accès à Amazon Kinesis Data Streams. Si vous n'utilisez pas Kinesis Data Streams comme source de données, vous pouvez supprimer cette déclaration.

Important

Amazon Data Firehose n'utilise pas IAM pour accéder aux destinations de point de terminaison HTTP détenues par des fournisseurs de services tiers pris en charge, notamment Datadog, Dynatrace, LogicMonitor MongoDB, New Relic, Splunk ou Sumo Logic. Pour accéder à une destination de point de terminaison HTTP spécifiée appartenant à un fournisseur de services tiers pris en charge, contactez ce fournisseur de services pour obtenir la clé d'API ou la clé d'accès requise pour permettre la livraison de données à ce service depuis Amazon Data Firehose.

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:region:account-id:key/key-id"
      ]
    }
  ]
}
```

```

    ],
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.region.amazonaws.com"
      },
      "StringLike": {
        "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis:DescribeStream",
      "kinesis:GetShardIterator",
      "kinesis:GetRecords",
      "kinesis:ListShards"
    ],
    "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents"
    ],
    "Resource": [
      "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "lambda:InvokeFunction",
      "lambda:GetFunctionConfiguration"
    ],
    "Resource": [
      "arn:aws:lambda:region:account-id:function:function-name:function-
version"
    ]
  }
]
}

```

Pour plus d'informations sur l'autorisation d'autres AWS services d'accéder à vos AWS ressources, consultez la section [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

⚠ Important

Actuellement, Amazon Data Firehose ne prend PAS en charge la livraison de données aux points de terminaison HTTP dans un VPC.

Livraison entre comptes depuis Amazon MSK

Lorsque vous créez un stream Firehose à partir de votre compte Firehose (par exemple, le compte B) et que votre source est un cluster MSK d'un autre AWS compte (compte A), les configurations suivantes doivent être en place.

Compte A :

1. Dans la console Amazon MSK, choisissez le cluster provisionné, puis choisissez Propriétés.
2. Sous Paramètres réseau, choisissez Modifier et activez la Connectivité multi-VPC.
3. Sous Paramètres de sécurité, choisissez Modifier la politique de cluster.
 - a. Si aucune politique n'est déjà configurée pour le cluster, cochez Inclure le principal de service Firehose et Activer la diffusion S3 entre comptes Firehose. Cela AWS Management Console générera automatiquement une politique avec les autorisations appropriées.
 - b. Si le cluster dispose déjà d'une politique configurée, ajoutez les autorisations suivantes à la politique existante :

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::arn:role/mskaasTestDeliveryRole"
  },
  "Action": [
    "kafka:GetBootstrapBrokers",
    "kafka:DescribeCluster",
    "kafka:DescribeClusterV2",
    "kafka-cluster:Connect"
  ],
}
```

```

    "Resource": "arn:aws:kafka:us-east-1:arn:cluster/D0-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20" // ARN of the
cluster
  },
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::arn:role/mskaasTestDeliveryRole"
    },
    "Action": [
      "kafka-cluster:DescribeTopic",
      "kafka-cluster:DescribeTopicDynamicConfiguration",
      "kafka-cluster:ReadData"
    ],
    "Resource": "arn:aws:kafka:us-east-1:arn:topic/D0-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20/*" //topic of the
cluster
  },
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::233450236687:role/mskaasTestDeliveryRole"
    },
    "Action": "kafka-cluster:DescribeGroup",
    "Resource": "arn:aws:kafka:us-east-1:arn:group/D0-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20/*" //topic of
the cluster
  },
}

```

4. Sous Principal AWS , saisissez l'ID du principal du compte B.
5. Sous Sujet, spécifiez le sujet Apache Kafka à partir duquel vous souhaitez que votre flux Firehose ingère des données. Une fois le stream Firehose créé, vous ne pouvez pas mettre à jour cette rubrique.
6. Choisissez Enregistrer les modifications

Compte B :

1. Dans la console Firehose, choisissez Create Firehose stream using Account B.
2. Sous Source, choisissez Amazon Managed Streaming for Apache Kafka.

3. Sous Paramètres source, pour le cluster Amazon Managed Streaming for Apache Kafka, saisissez l'ARN du cluster Amazon MSK dans le compte A.
4. Sous Sujet, spécifiez le sujet Apache Kafka à partir duquel vous souhaitez que votre flux Firehose ingère des données. Une fois le stream Firehose créé, vous ne pouvez pas mettre à jour cette rubrique.
5. Dans Nom du flux de diffusion, spécifiez le nom de votre flux Firehose.

Dans le compte B, lorsque vous créez votre flux Firehose, vous devez disposer d'un rôle IAM (créé par défaut lorsque vous utilisez le AWS Management Console) qui accorde au flux Firehose un accès en « lecture » au cluster Amazon MSK multicomptes pour le sujet configuré.

Voici ce qui est configuré par AWS Management Console :

```
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka:GetBootstrapBrokers",
    "kafka:DescribeCluster",
    "kafka:DescribeClusterV2",
    "kafka-cluster:Connect"
  ],
  "Resource": "arn:aws:kafka:us-east-1:arn:cluster/D0-NOT-TOUCH-mskaas-provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20/*" //topic of the cluster
},
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
    "kafka-cluster:DescribeTopic",
    "kafka-cluster:DescribeTopicDynamicConfiguration",
    "kafka-cluster:ReadData"
  ],
  "Resource": "arn:aws:kafka:us-east-1:arn:topic/D0-NOT-TOUCH-mskaas-provisioned-privateLink/xxxxxxxx-2f3a-462a-ba09-xxxxxxxx-20/mskaas_test_topic" //topic of the cluster
},
{
  "Sid": "",
  "Effect": "Allow",
  "Action": [
```

```
    "kafka-cluster:DescribeGroup"  
  ],  
  "Resource": "arn:aws:kafka:us-east-1:arn:group/D0-NOT-TOUCH-mskaas-provisioned-  
privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*" //topic of the cluster  
  },  
}
```

Vous pouvez ensuite effectuer l'étape facultative de configuration de la transformation des enregistrements et de la conversion des formats d'enregistrement. Pour plus d'informations, consultez [Configuration de la transformation des enregistrements et de la conversion des formats](#).

Diffusion entre comptes vers une destination Amazon S3

Vous pouvez utiliser les API AWS CLI ou les API Amazon Data Firehose pour créer un flux Firehose sur un AWS compte avec une destination Amazon S3 sur un autre compte. La procédure suivante montre un exemple de configuration d'un flux Firehose appartenant au compte A pour fournir des données à un compartiment Amazon S3 appartenant au compte B.

1. Créez un rôle IAM sous le compte A en suivant les étapes décrites dans [Accorder à Firehose l'accès à une destination Amazon S3](#).

Note

Le compartiment Amazon S3 spécifié dans la stratégie d'accès appartient ici au compte B. Assurez-vous de les ajouter `s3:PutObjectAcl` à la liste des actions Amazon S3 dans la politique d'accès, qui accorde au compte B un accès complet aux objets fournis par Amazon Data Firehose. Cette autorisation est requise pour la diffusion entre comptes. Amazon Data Firehose définit l'en-tête « `x-amz-acl` » de la demande sur « `bucket-owner-full-control` ».

2. Pour autoriser l'accès à partir du rôle IAM précédemment configuré, créez une politique de compartiment S3 dans le compte B. Le code suivant est un exemple de politique de compartiment. Pour plus d'informations, consultez [Utilisation de stratégies de compartiment et de stratégies utilisateur](#).

```
{  
  
  "Version": "2012-10-17",  
  "Id": "PolicyID",  
  "Statement": [  

```

```

    {
      "Sid": "StmtID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::accountA-id:role/iam-role-name"
      },
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3::bucket-name",
        "arn:aws:s3::bucket-name/*"
      ]
    }
  ]
}

```

3. Créez un stream Firehose sous le compte A en utilisant le rôle IAM que vous avez créé à l'étape 1.

Livraison entre comptes vers une destination OpenSearch de service

Vous pouvez utiliser les API AWS CLI ou les API Amazon Data Firehose pour créer un flux Firehose sur un AWS compte avec une destination de OpenSearch service sur un autre compte. La procédure suivante montre un exemple de la façon dont vous pouvez créer un flux Firehose sous le compte A et le configurer pour fournir des données à une destination de OpenSearch service appartenant au compte B.

1. Créez un rôle IAM sous le compte A en suivant les étapes décrites dans [the section called "Accorder à Amazon Data Firehose l'accès à une destination de service public OpenSearch"](#).
2. Pour autoriser l'accès depuis le rôle IAM que vous avez créé à l'étape précédente, créez une politique de OpenSearch service sous le compte B. Le JSON suivant est un exemple.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::Account-A-ID:role/firehose_delivery_role "
    },
    "Action": "es:ESHttpGet",
    "Resource": [
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_all/_settings",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_cluster/stats",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/roletest*/_mapping/roletest",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_nodes",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_nodes/stats",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_nodes/*/stats",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_stats",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/roletest*/_stats",
      "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/"
    ]
  }
]
}

```

3. Créez un stream Firehose sous le compte A en utilisant le rôle IAM que vous avez créé à l'étape 1. Lorsque vous créez le flux Firehose, utilisez les API Amazon Data Firehose AWS CLI ou les API Amazon Data Firehose et spécifiez le `ClusterEndpoint` champ au lieu de celui de `Service.DomainARN` OpenSearch

Note

Pour créer un stream Firehose sur un AWS compte avec une destination de OpenSearch service sur un autre compte, vous devez utiliser les API Amazon Data Firehose AWS CLI ou les API Amazon Data Firehose. Vous ne pouvez pas utiliser le AWS Management Console pour créer ce type de configuration entre comptes.

Utilisation des balises pour contrôler l'accès

Vous pouvez utiliser l'élément facultatif (ou le Condition bloc) d'une politique IAM pour affiner l'accès aux opérations Amazon Data Firehose en fonction des clés et des valeurs des balises. Les sous-sections suivantes décrivent comment procéder pour les différentes opérations Amazon Data Firehose. Pour en savoir plus sur l'utilisation de l'élément Condition et sur les opérateurs que vous pouvez utiliser à l'intérieur, consultez la section [Éléments de stratégie JSON IAM : Condition](#).

CreateDeliveryStream

Pour l'opération CreateDeliveryStream, utilisez la clé de condition aws:RequestTag. Dans l'exemple suivant, MyKey et MyValue représentent la clé et sa valeur correspondante pour une balise. Pour plus d'informations, consultez [Principes de base des identifications](#).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "firehose:CreateDeliveryStream",
      "firehose:TagDeliveryStream"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/MyKey": "MyValue"
      }
    }
  }]
}
```

TagDeliveryStream

Pour l'opération TagDeliveryStream, utilisez la clé de condition aws:TagKeys. Dans l'exemple suivant, MyKey est un exemple de clé de balise.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
    "Action": "firehose:TagDeliveryStream",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:TagKeys": "MyKey"
      }
    }
  ]
}
```

UntagDeliveryStream

Pour l'opération `UntagDeliveryStream`, utilisez la clé de condition `aws:TagKeys`. Dans l'exemple suivant, `MyKey` est un exemple de clé de balise.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "firehose:UntagDeliveryStream",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": "MyKey"
        }
      }
    }
  ]
}
```

ListDeliveryStreams

Vous ne pouvez pas utiliser le contrôle d'accès basé sur les balises avec `ListDeliveryStreams`.

Autres opérations d'Amazon Data Firehose

Pour toutes les opérations Amazon Data Firehose autres que `CreateDeliveryStream`, et `TagDeliveryStream` `UntagDeliveryStream` `ListDeliveryStreams`, utilisez la clé de `aws:RequestTag` condition. Dans l'exemple suivant, `MyKey` et `MyValue` représentent la clé et sa valeur correspondante pour une balise.

ListDeliveryStreams, utilisez la clé de `firehose:ResourceTag` condition pour contrôler l'accès en fonction des balises de ce stream Firehose.

Dans l'exemple suivant, MyKey et MyValue représentent la clé et sa valeur correspondante pour une balise. La politique ne s'appliquerait qu'aux streams Data Firehose ayant une balise nommée MyKey avec une valeur de MyValue. Pour plus d'informations sur le contrôle de l'accès en fonction des balises de ressources, consultez la section [Contrôle de l'accès aux AWS ressources à l'aide de balises](#) dans le guide de l'utilisateur IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "firehose:DescribeDeliveryStream",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "firehose:ResourceTag/MyKey": "MyValue"
        }
      }
    }
  ]
}
```

Authentifiez-vous AWS Secrets Manager dans Amazon Data Firehose

Amazon Data Firehose s'intègre AWS Secrets Manager pour fournir un accès sécurisé à vos secrets et automatiser la rotation des identifiants. Cette intégration permet à Firehose de récupérer un secret depuis Secrets Manager au moment de l'exécution pour se connecter aux destinations de streaming mentionnées précédemment et diffuser vos flux de données. Ainsi, vos secrets ne sont pas visibles en texte brut pendant le flux de travail de création de flux, que ce soit dans les paramètres de l'API AWS Management Console ou dans les paramètres de l'API. Il fournit une pratique sécurisée pour gérer vos secrets et vous soulage des activités complexes de gestion des informations d'identification, telles que la configuration de fonctions Lambda personnalisées pour gérer les rotations de mots de passe.

Pour plus d'informations, consultez le [Guide de l'utilisateur AWS Secrets Manager](#).

Comprenez les secrets

Un secret peut être un mot de passe, un ensemble d'informations d'identification telles qu'un nom d'utilisateur et un mot de passe, un jeton OAuth ou d'autres informations secrètes que vous stockez sous une forme chiffrée dans Secrets Manager.

Pour chaque destination, vous devez spécifier la paire clé-valeur secrète au format JSON correct, comme indiqué dans la section suivante. Amazon Data Firehose ne parviendra pas à se connecter à votre destination si le format JSON de votre secret ne correspond pas à la destination.

Format du secret pour le cluster Amazon Redshift Provisioned et le groupe de travail Amazon Redshift Serverless

```
{
  "username": "<username>",
  "password": "<password>"
}
```

Format du secret pour Splunk

```
{
  "hec_token": "<hec token>"
}
```

Format du secret pour Snowflake

```
{
  "user": "<user>",
  "private_key": "<private_key>",
  "key_passphrase": "<passphrase>" // optional
}
```

Format du secret pour le point de terminaison HTTP, Coralogix, Datadog, Dynatrace, Elastic, Honeycomb, Logz.io, MongoDB Cloud et New Relic LogicMonitor

```
{
  "api_key": "<apikey>"
}
```

Créer un secret

Pour créer un secret, suivez les étapes décrites dans [Créer un AWS Secrets Manager secret](#) dans le Guide de AWS Secrets Manager l'utilisateur.

Utilisez le secret

Nous vous recommandons de stocker vos informations d'identification ou vos clés AWS Secrets Manager pour vous connecter à des destinations de streaming telles qu'Amazon Redshift, le point de terminaison HTTP, Snowflake, Splunk, Coralogix, Datadog, Dynatrace, Elastic, Honeycomb, Logz.io, MongoDB Cloud et New Relic. LogicMonitor

Vous pouvez configurer l'authentification avec Secrets Manager pour ces destinations via la console AWS de gestion au moment de la création du stream Firehose. Pour plus d'informations, consultez [Configuration des paramètres de destination](#). Vous pouvez également utiliser les opérations [CreateDeliveryStream](#) et [UpdateDestination](#) API pour configurer l'authentification avec Secrets Manager.

Firehose met en cache les secrets avec un cryptage et les utilise pour chaque connexion aux destinations. Il actualise le cache toutes les 10 minutes pour s'assurer que les dernières informations d'identification sont utilisées.

Vous pouvez choisir de désactiver la fonctionnalité de récupération des secrets depuis Secrets Manager à tout moment pendant le cycle de vie du flux. Si vous ne souhaitez pas utiliser Secrets Manager pour récupérer des secrets, vous pouvez utiliser le nom d'utilisateur/mot de passe ou la clé API à la place.

Note

Bien que cette fonctionnalité soit gratuite dans Firehose, l'accès et la maintenance de Secrets Manager vous sont facturés. Pour plus d'informations, consultez la page de [AWS Secrets Manager](#) tarification.

Accordez l'accès à Firehose pour récupérer le secret

Pour que Firehose puisse récupérer un secret AWS Secrets Manager, vous devez fournir à Firehose les autorisations requises pour accéder au secret et à la clé qui chiffre votre secret.

Lors AWS Secrets Manager de l'utilisation pour stocker et récupérer des secrets, il existe différentes options de configuration en fonction de l'endroit où le secret est stocké et de la manière dont il est crypté.

- Si le secret est stocké dans le même AWS compte que votre rôle IAM et qu'il est chiffré avec la clé AWS gérée par défaut (`aws/secretsmanager`), le rôle IAM assumé par Firehose n'a besoin que d'une `secretsmanager:GetSecretValue` autorisation sur le secret.

```
// secret role policy
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "Secret ARN"
    }
  ]
}
```

Pour plus d'informations sur les politiques IAM, consultez les [exemples de politiques d'autorisation pour AWS Secrets Manager](#).

- Si le secret est stocké dans le même compte que le rôle mais chiffré à l'aide d'une [clé gérée par le client](#) (CMK), le rôle a besoin à la fois d'`kms:Decrypt` autorisations `secretsmanager:GetSecretValue` et d'autorisations. La politique CMK doit également permettre au rôle IAM de fonctionner. `kms:Decrypt`

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "Secret ARN"
  },
  {
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "KMSKeyARN"
  }
]
```

```
}
```

- Si le secret est stocké dans un AWS compte différent de votre rôle et qu'il est chiffré avec la clé AWS gérée par défaut, cette configuration n'est pas possible car Secrets Manager n'autorise pas l'accès entre comptes lorsque le secret est chiffré avec une clé AWS gérée.
- Si le secret est stocké dans un autre compte et crypté avec une clé CMK, le rôle IAM a besoin d'une `secretsmanager:GetSecretValue` autorisation sur le secret et d'une `kms:Decrypt` autorisation sur la clé CMK. La politique de ressources du secret et la politique CMK de l'autre compte doivent également accorder au rôle IAM les autorisations nécessaires. Pour plus d'informations, consultez la section [Accès entre comptes](#).

Faites pivoter le secret

La rotation se produit lorsque vous mettez régulièrement à jour un secret. Vous pouvez configurer AWS Secrets Manager pour faire pivoter automatiquement le secret selon un calendrier que vous spécifiez. De cette façon, vous pouvez remplacer les secrets à long terme par des secrets à court terme. Cela permet de réduire le risque de compromission. Pour plus d'informations, voir [Rotation AWS Secrets Manager des secrets](#) dans le guide de AWS Secrets Manager l'utilisateur.

Gérez les rôles IAM via la console Amazon Data Firehose

Amazon Data Firehose est un service entièrement géré qui fournit des données de streaming en temps réel aux destinations. Vous pouvez également configurer Firehose pour transformer et convertir le format de vos données avant leur livraison. Pour utiliser ces fonctionnalités, vous devez d'abord fournir des rôles IAM pour accorder des autorisations à Firehose lorsque vous créez ou modifiez un flux Firehose. Firehose utilise ce rôle IAM pour toutes les autorisations dont le stream Firehose a besoin.

Par exemple, imaginez un scénario dans lequel vous créez un flux Firehose qui fournit des données à Amazon S3, et ce flux Firehose contient des enregistrements source Transform avec fonctionnalité activée. AWS Lambda Dans ce cas, vous devez fournir des rôles IAM pour accorder à Firehose l'autorisation d'accéder au compartiment S3 et d'invoquer la fonction Lambda, comme indiqué ci-dessous.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "lambdaProcessing",
```

```
    "Effect": "Allow",
    "Action": ["lambda:InvokeFunction", "lambda:GetFunctionConfiguration"],
    "Resource": "arn:aws:lambda:us-east-1:<account id>:function:<lambda function
name>:<lambda function version>"
  }, {
    "Sid": "s3Permissions",
    "Effect": "Allow",
    "Action": ["s3:AbortMultipartUpload", "s3:GetBucketLocation", "s3:GetObject",
"s3:ListBucket", "s3:ListBucketMultipartUploads", "s3:PutObject"],
    "Resource": ["arn:aws:s3:::<bucket name>", "arn:aws:s3:::<bucket name>/*"]
  }]
}
```

La console Firehose vous permet de choisir la manière dont vous souhaitez fournir ces rôles. Vous pouvez choisir l'une des options suivantes.

- [Choisissez un rôle IAM existant](#)
- [Création d'un nouveau rôle IAM depuis la console](#)

Choisissez un rôle IAM existant

Vous pouvez choisir parmi un rôle IAM existant. Avec cette option, assurez-vous que le rôle IAM que vous choisissez dispose d'une politique de confiance appropriée et que les autorisations requises pour votre source et votre destination sont requises. Pour plus d'informations, consultez [Contrôler l'accès avec Amazon Data Firehose](#).

Création d'un nouveau rôle IAM depuis la console

Vous pouvez également utiliser la console Firehose pour créer un nouveau rôle en votre nom.

Lorsque Firehose crée un rôle IAM en votre nom, celui-ci inclut automatiquement toutes les politiques d'autorisation et de confiance qui accordent les autorisations requises en fonction de la configuration du flux Firehose.

Par exemple, si vous n'avez pas activé la AWS Lambda fonctionnalité Transform source records with, la console génère l'instruction suivante dans la politique d'autorisation.

```
{
  "Sid": "lambdaProcessing",
  "Effect": "Allow",
```

```
"Action": [
  "lambda:InvokeFunction",
  "lambda:GetFunctionConfiguration"
],
"Resource": "arn:aws:lambda:us-east-1:<account id>:function:
%FIREHOSE_POLICY_TEMPLATE_PLACEHOLDER%"
}
```

Note

Il est prudent d'ignorer les déclarations de politique qui s'y %FIREHOSE_POLICY_TEMPLATE_PLACEHOLDER% trouvent, car elles n'accordent aucune autorisation sur aucune ressource.

La console de création et de modification des flux de travail Firehose Stream crée également une politique de confiance et l'associe au rôle IAM. La politique de confiance permet à Firehose d'assumer le rôle IAM. Voici un exemple de politique de confiance.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "firehoseAssume",
    "Effect": "Allow",
    "Principal": {
      "Service": "firehose.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }]
}
```

Important

- Vous devez éviter d'utiliser le même rôle IAM géré par la console pour plusieurs flux Firehose. Dans le cas contraire, le rôle IAM pourrait devenir trop permissif ou entraîner des erreurs.
- Pour utiliser différentes déclarations de politique dans une politique d'autorisation à partir d'un rôle IAM géré par la console, vous pouvez créer votre propre rôle IAM et copier les déclarations de politique dans une politique d'autorisation attachée au nouveau rôle. Pour

associer le rôle au flux Firehose, sélectionnez l'option Choisir un rôle IAM existant dans l'accès au service.

- La console gère tout rôle IAM dont l'ARN contient la chaîne service-role. Lorsque vous choisissez l'option de rôle IAM existant, assurez-vous de sélectionner un rôle IAM sans la chaîne service-role dans son ARN afin que la console n'y apporte aucune modification.

Étapes pour créer un rôle IAM depuis la console

1. [Ouvrez la console Firehose à l'adresse https://console.aws.amazon.com/firehose/](https://console.aws.amazon.com/firehose/).
2. Choisissez Create Firehose stream.
3. Choisissez une source et une destination. Pour plus d'informations, consultez [Création d'un stream Firehose](#).
4. Choisissez les paramètres de destination. Pour plus d'informations, consultez [Configuration des paramètres de destination](#).
5. Sous [Paramètres avancés](#), pour Accès au service, choisissez Créer ou mettre à jour le rôle IAM.

Note

Il s'agit d'une option par défaut. Pour utiliser un rôle existant, sélectionnez l'option Choisir un rôle IAM existant. La console Firehose n'apportera aucune modification à votre propre rôle.

6. Choisissez Create Firehose stream.

Modifier le rôle IAM depuis la console

Lorsque vous modifiez un stream Firehose, Firehose met à jour la politique d'autorisation correspondante en conséquence pour refléter les modifications de configuration et d'autorisation.

Par exemple, lorsque vous modifiez le flux Firehose et que vous activez la AWS Lambda fonctionnalité Transform source records with using the dernière version de Lambda function `asexampleLambdaFunction`, vous obtenez la déclaration de politique suivante dans la politique d'autorisation.

```
{
  "Sid": "lambdaProcessing",
```

```
"Effect": "Allow",
"Action": [
  "lambda:InvokeFunction",
  "lambda:GetFunctionConfiguration"
],
"Resource": "arn:aws:lambda:us-east-1:<account id>:function:exampleLambdaFunction:
$LATEST"
}
```

Important

Un rôle IAM géré par la console est conçu pour être autonome. Nous vous déconseillons de modifier la politique d'autorisation ou la politique de confiance en dehors de la console.

Modifier le rôle IAM depuis la console

1. [Ouvrez la console Firehose à l'adresse https://console.aws.amazon.com/firehose/](https://console.aws.amazon.com/firehose/).
2. Choisissez les flux Firehose et le nom du flux Firehose que vous souhaitez mettre à jour.
3. Dans l'onglet Configuration, dans la section Accès au serveur, choisissez Modifier.
4. Mettez à jour l'option de rôle IAM.

Note

Par défaut, la console met toujours à jour un rôle IAM avec le pattern service-role dans son ARN. Lorsque vous choisissez l'option de rôle IAM existant, assurez-vous de sélectionner un rôle IAM sans la chaîne service-role dans son ARN afin que la console n'y apporte aucune modification.

5. Sélectionnez Enregistrer les modifications.

Surveillance d'Amazon Data Firehose

Amazon Data Firehose fournit des fonctionnalités de surveillance pour vos flux Firehose. Pour plus d'informations, consultez [Surveillance](#).

Validation de conformité pour Amazon Data Firehose

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon Data Firehose dans le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir [AWS Services concernés par programme de conformité](#). Pour obtenir des informations générales, consultez [Programmes de conformité AWS](#).

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#).

Lorsque vous utilisez Data Firehose, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. Si votre utilisation de Data Firehose est soumise au respect de normes telles que HIPAA, PCI ou FedRAMP, fournit des ressources pour vous aider à : AWS

- [Guides de démarrage rapide sur la sécurité et la conformité](#) : ces guides de déploiement abordent les considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- Livre blanc [sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc](#) décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- [AWS Ressources relatives à la conformité](#) — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [AWS Config](#) — Ce AWS service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#) — Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience dans Amazon Data Firehose

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones

de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS mondiale](#).

Outre l'infrastructure AWS mondiale, Data Firehose propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

Reprise après sinistre

Amazon Data Firehose fonctionne en mode sans serveur et prend en charge les dégradations de l'hôte, la disponibilité des zones de disponibilité et les autres problèmes liés à l'infrastructure en effectuant une migration automatique. Dans ce cas, Amazon Data Firehose veille à ce que le flux Firehose soit migré sans perte de données.

Sécurité de l'infrastructure dans Amazon Data Firehose

En tant que service géré, Amazon Data Firehose est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Firehose via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Note

Pour les requêtes HTTPS sortantes, Amazon Data Firehose utilise une bibliothèque HTTP qui sélectionne automatiquement la version de protocole TLS la plus élevée prise en charge côté destination.

Points de terminaison d'un VPC PrivateLink

Amazon Data Firehose fournit un support pour les points de terminaison VPC (). PrivateLink Pour plus d'informations, consultez [Utilisation d'Amazon Data Firehose avec AWS PrivateLink](#).

Bonnes pratiques de sécurité pour Amazon Data Firehose

Amazon Data Firehose propose un certain nombre de fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

Implémentation d'un accès sur la base du moindre privilège

Lorsque vous accordez des autorisations, vous décidez qui obtient quelles autorisations à quelles ressources Amazon Data Firehose. Vous activez des actions spécifiques que vous souhaitez autoriser sur ces ressources. Par conséquent, vous devez accorder uniquement les autorisations qui sont requises pour exécuter une tâche. L'implémentation d'un accès sur la base du moindre privilège est fondamentale pour réduire les risques en matière de sécurité et l'impact que pourraient avoir des d'erreurs ou des actes de malveillance.

Utilisation des rôles IAM

Les applications productrices et clientes doivent disposer d'informations d'identification valides pour accéder aux flux Firehose, et votre flux Firehose doit disposer d'informations d'identification valides pour accéder aux destinations. Vous ne devez pas stocker les AWS informations d'identification directement dans une application cliente ou dans un compartiment Amazon S3. Il s'agit d'autorisations à long terme qui ne font pas automatiquement l'objet d'une rotation et qui pourraient avoir un impact commercial important si elles étaient compromises.

Vous devez plutôt utiliser un rôle IAM pour gérer les informations d'identification temporaires permettant à vos applications productrices et clientes d'accéder aux flux Firehose. Lorsque vous utilisez un rôle, vous n'avez pas à utiliser d'informations d'identification à long terme (par exemple, un nom d'utilisateur et un mot de passe ou des clés d'accès) pour accéder à d'autres ressources.

Pour plus d'informations, consultez les rubriques suivantes dans le Guide de l'utilisateur IAM :

- [Rôles IAM](#)
- [Scénarios courants pour les rôles : utilisateurs, applications et services.](#)

Implémentation d'un chiffrement côté serveur dans des ressources dépendantes

Les données au repos et les données en transit peuvent être chiffrées dans Amazon Data Firehose. Pour plus d'informations, consultez la section [Protection des données dans Amazon Amazon Data Firehose](#).

CloudTrail À utiliser pour surveiller les appels d'API

Amazon Data Firehose est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un AWS service dans Amazon Data Firehose.

À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Data Firehose, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour plus d'informations, consultez [the section called “Journalisation des appels d'API Amazon Data Firehose avec AWS CloudTrail”](#).

Transformation des données avec Amazon Data Firehose

Amazon Data Firehose peut appeler votre fonction Lambda pour transformer les données sources entrantes et transmettre les données transformées aux destinations. Vous pouvez activer la transformation des données Amazon Data Firehose lorsque vous créez votre flux Firehose.

Flux de transformation de données

Lorsque vous activez la transformation des données Firehose, Firehose met en mémoire tampon les données entrantes. L'indicateur de taille de la mémoire tampon est compris entre 0,2 Mo et 3 Mo. L'indicateur de taille de mémoire tampon Lambda par défaut est de 1 Mo pour toutes les destinations, à l'exception de Splunk et Snowflake. Pour Splunk et Snowflake, l'indicateur de mise en mémoire tampon par défaut est de 256 Ko. L'indicateur de l'intervalle de mise en mémoire tampon Lambda est compris entre 0 et 900 secondes. L'indicateur d'intervalle de mise en mémoire tampon Lambda par défaut est de soixante secondes pour toutes les destinations sauf Snowflake. Pour Snowflake, l'intervalle d'indication de mise en mémoire tampon par défaut est de 30 secondes. Pour ajuster la taille de la mémoire tampon, définissez le [ProcessingConfiguration](#) paramètre de l'[UpdateDestinationAPI CreateDeliveryStream](#) avec l'[ProcessorParameter](#) appel `BufferSizeInMBs` and `IntervalInSeconds`. Firehose invoque ensuite la fonction Lambda spécifiée de manière asynchrone avec chaque lot mis en mémoire tampon en utilisant le mode d'appel synchrone. AWS Lambda Les données transformées sont envoyées de Lambda à Firehose. Firehose l'envoie ensuite à la destination lorsque la taille de mémoire tampon ou l'intervalle de mise en mémoire tampon de destination spécifiés sont atteints, selon la première éventualité.

Important

Le mode d'invocation synchrone Lambda a une taille de charge utile limitée à 6 Mo pour la demande et la réponse. Assurez-vous que votre taille de mise en mémoire tampon pour l'envoi de la demande à la fonction est inférieure ou égale à 6 Mo. Assurez-vous également que la réponse que votre fonction renvoie ne dépasse pas 6 Mo.

Transformation de données et modèle d'état

Tous les enregistrements transformés depuis Lambda doivent contenir les paramètres suivants, sinon Amazon Data Firehose les rejette et considère cela comme un échec de transformation des données.

Pour Kinesis Data Streams et Direct PUT :

recordId

L'ID d'enregistrement est transmis d'Amazon Data Firehose à Lambda lors de l'appel.

L'enregistrement transformé doit comporter le même ID d'enregistrement. La moindre incohérence entre l'ID de l'enregistrement initial et l'ID de l'enregistrement transformé est traitée comme un échec de la transformation des données.

result

État de la transformation de données de l'enregistrement. Les valeurs possibles sont : `Ok` (l'enregistrement a été transformé), `Dropped` (l'enregistrement a été abandonné de manière intentionnelle par votre logique de traitement) et `ProcessingFailed` (impossible de transformer l'enregistrement). Si un enregistrement a le statut `Ok` ou `Dropped`, Amazon Data Firehose considère qu'il a été traité avec succès. Dans le cas contraire, Amazon Data Firehose considère qu'il n'a pas été traité correctement.

data

Charge utile des données transformées, d'après l'encodage en base64.

Voici un exemple de résultat Lambda :

```
{
  "recordId": "<recordId from the Lambda input>",
  "result": "Ok",
  "data": "<Base64 encoded Transformed data>"
}
```

Pour Amazon MSK

recordId

L'ID d'enregistrement est transmis de Firehose à Lambda lors de l'invocation. L'enregistrement transformé doit comporter le même ID d'enregistrement. La moindre incohérence entre l'ID de l'enregistrement initial et l'ID de l'enregistrement transformé est traitée comme un échec de la transformation des données.

result

État de la transformation de données de l'enregistrement. Les valeurs possibles sont : `Ok` (l'enregistrement a été transformé), `Dropped` (l'enregistrement a été abandonné de manière intentionnelle par votre logique de traitement) et `ProcessingFailed` (impossible de transformer l'enregistrement). Si un enregistrement a le statut `Ok` ou `Dropped`, Firehose considère qu'il a été traité avec succès. Dans le cas contraire, Firehose considère qu'il n'a pas été traité correctement.

KafkaRecordValue

Charge utile des données transformées, d'après l'encodage en base64.

Voici un exemple de résultat Lambda :

```
{
  "recordId": "<recordId from the Lambda input>",
  "result": "Ok",
  "kafkaRecordValue": "<Base64 encoded Transformed data>"
}
```

Plans Lambda

Ces plans montrent comment créer et utiliser des fonctions AWS Lambda pour transformer les données de vos flux de données Amazon Data Firehose.

Pour voir les plans disponibles dans la console AWS Lambda

1. Connectez-vous à la AWS Lambda console AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/).
2. Choisissez Create function (Créer une fonction), puis Use a blueprint (Utiliser un plan).
3. Dans le champ Blueprints, recherchez le mot clé `firehose` pour trouver les plans Amazon Data Firehose Lambda.

Liste des plans :

- Traiter les enregistrements envoyés au flux Amazon Data Firehose (Node.js, Python)

Ce plan montre un exemple de base de la manière de traiter les données de votre flux de données Firehose à l'aide AWS de Lambda.

Date de la dernière version : novembre 2016.

Notes de publication : aucune.

- CloudWatch Journaux de processus envoyés à Firehose

Ce plan est obsolète. Pour plus d'informations sur le traitement CloudWatch des journaux envoyés à Firehose, consultez [Writing to Firehose](#) Using Logs. CloudWatch

- Convertir les enregistrements de flux Amazon Data Firehose au format syslog en JSON (Node.js)

Ce plan montre comment convertir les enregistrements d'entrée au format Syslog RFC3164 en JSON.

Date de la dernière version : novembre 2016.

Notes de publication : aucune.

Pour voir les plans disponibles dans le AWS Serverless Application Repository

1. Accédez à [AWS Serverless Application Repository](#).
2. Choisissez Parcourir toutes les applications.
3. Dans le champ Applications recherchez le mot-clé firehose.

Vous pouvez également créer une fonction Lambda sans utiliser de plan. Voir [Commencer à utiliser AWS Lambda](#).

Gestion des échecs de transformation des données

Si l'appel de votre fonction Lambda échoue en raison d'un délai d'attente réseau ou parce que vous avez atteint la limite d'invocation Lambda, Amazon Data Firehose réessaie l'appel trois fois par défaut. Si l'invocation échoue, Amazon Data Firehose ignore alors ce lot d'enregistrements. Les enregistrements ignorés sont traités comme des enregistrements comme n'ayant pas été correctement traités. Vous pouvez spécifier ou remplacer les options de nouvelle tentative à l'aide de l'API [CreateDeliveryStreamorUpdateDestination](#). Pour ce type d'échec, vous pouvez enregistrer les erreurs d'appel dans Amazon CloudWatch Logs. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de journaux CloudWatch](#).

Si le statut de la transformation des données d'un enregistrement est indiqué `ProcessingFailed`, Amazon Data Firehose considère l'enregistrement comme n'ayant pas été traité correctement. Pour ce type de panne, vous pouvez envoyer des journaux d'erreurs vers Amazon CloudWatch Logs à partir de votre fonction Lambda. Pour plus d'informations, consultez la section [Accès à Amazon CloudWatch Logs AWS Lambda](#) dans le manuel du AWS Lambda développeur.

En cas d'échec de la transformation des données, les enregistrements n'ayant pas été correctement traités sont diffusés au compartiment S3 du dossier `processing-failed`. Les enregistrements sont au format suivant :

```
{
  "attemptsMade": "count",
  "arrivalTimestamp": "timestamp",
  "errorCode": "code",
  "errorMessage": "message",
  "attemptEndingTimestamp": "timestamp",
  "rawData": "data",
  "lambdaArn": "arn"
}
```

`attemptsMade`

Nombre de tentatives de demandes d'invocation.

`arrivalTimestamp`

Heure à laquelle l'enregistrement a été reçu par Amazon Data Firehose.

`errorCode`

Code d'erreur HTTP renvoyé par Lambda.

`errorMessage`

Message d'erreur renvoyé par Lambda.

`attemptEndingTimestamp`

Heure à laquelle Amazon Data Firehose a cessé de tenter d'appeler Lambda.

`rawData`

Données d'enregistrement encodées en base64.

LambdaArn

L'Amazon Resource Name (ARN) de la fonction Lambda.

Durée d'une invocation Lambda

Amazon Data Firehose prend en charge une durée d'invocation Lambda allant jusqu'à 5 minutes. Si l'exécution de votre fonction Lambda prend plus de 5 minutes, le message d'erreur suivant s'affiche : Firehose a rencontré des erreurs de temporisation lors de l'appel à Lambda. AWS Le délai maximum de fonctionnement pris en charge est de cinq minutes.

Pour plus d'informations sur ce que fait Amazon Data Firehose en cas d'erreur de ce type, consultez [the section called “Gestion des échecs de transformation des données”](#)

Sauvegarde de l'enregistrement source

Amazon Data Firehose peut sauvegarder simultanément tous les enregistrements non transformés dans votre compartiment S3 tout en livrant les enregistrements transformés à destination. Vous pouvez activer la sauvegarde des enregistrements source lorsque vous créez ou mettez à jour votre stream Firehose. Vous ne pouvez pas désactiver la sauvegarde de l'enregistrement source après l'avoir activée.

Partitionnement dynamique dans Amazon Data Firehose

Le partitionnement dynamique vous permet de partitionner en continu les données de streaming dans Firehose en utilisant des clés contenues dans les données (par exemple `customer_id`, `transaction_id` ou), puis de transmettre les données regroupées par ces clés dans les préfixes Amazon Simple Storage Service (Amazon S3) correspondants. Cela facilite l'exécution d'analyses performantes et rentables sur les données de streaming dans Amazon S3 à l'aide de divers services tels qu'Amazon Athena, Amazon EMR, Amazon Redshift Spectrum et Amazon. QuickSight En outre, AWS Glue peut effectuer des tâches d'extraction, de transformation et de chargement (ETL) plus sophistiquées une fois que les données de streaming partitionnées dynamiquement ont été transmises à Amazon S3, dans les cas d'utilisation nécessitant un traitement supplémentaire.

Le partitionnement de vos données permet de minimiser la quantité de données analysées, d'optimiser les performances et de réduire les coûts de vos requêtes analytiques sur Amazon S3. Cela augmente également l'accès détaillé à vos données. Les flux Firehose sont traditionnellement utilisés pour capturer et charger des données dans Amazon S3. Pour partitionner un ensemble de données de streaming en vue d'une analyse basée sur Amazon S3, vous devez exécuter des applications de partitionnement entre les compartiments Amazon S3 avant de mettre les données à disposition pour analyse, ce qui peut s'avérer compliqué ou coûteux.

Grâce au partitionnement dynamique, Firehose regroupe en permanence les données en transit à l'aide de clés de données définies dynamiquement ou statiquement, et fournit les données aux préfixes Amazon S3 individuels par clé. Cela se réduit time-to-insight de quelques minutes ou heures. Il permet également de réduire les coûts et de simplifier les architectures.

Rubriques

- [Clés de partitionnement](#)
- [Préfixe de compartiment Amazon S3 pour le partitionnement dynamique](#)
- [Partitionnement dynamique des données agrégées](#)
- [Ajout d'un nouveau délimiteur de ligne lors de la transmission de données à S3](#)
- [Comment activer le partitionnement dynamique](#)
- [Gestion des erreurs de partitionnement dynamique](#)
- [Mise en mémoire tampon des données et partitionnement dynamique](#)

Clés de partitionnement

Avec le partitionnement dynamique, vous créez des jeux de données ciblés à partir des données S3 de streaming en partitionnant les données en fonction des clés de partitionnement. Les clés de partitionnement vous permettent de filtrer vos données de streaming en fonction de valeurs spécifiques. Par exemple, si vous devez filtrer vos données en fonction de l'ID du client et du pays, vous pouvez spécifier le champ de données de `customer_id` comme une clé de partitionnement et le champ de données de `country` comme une autre clé de partitionnement. Ensuite, vous spécifiez les expressions (en utilisant les formats pris en charge) pour définir les préfixes de compartiment S3 auxquels les enregistrements de données partitionnés dynamiquement doivent être livrés.

Les méthodes prises en charge pour créer des clés de partitionnement sont les suivantes :

- **Analyse en ligne** : cette méthode utilise le mécanisme de support intégré de Firehose, un [analyseur jq](#), pour extraire les clés de partitionnement à partir d'enregistrements de données au format JSON. Actuellement, nous ne prenons en charge que `jq 1.6` la version.
- **AWS Fonction Lambda** : cette méthode utilise une fonction AWS Lambda spécifiée pour extraire et renvoyer les champs de données nécessaires au partitionnement.

Important

Lorsque vous activez le partitionnement dynamique, vous devez configurer au moins l'une de ces méthodes pour partitionner vos données. Vous pouvez configurer l'une de ces méthodes pour spécifier vos clés de partitionnement ou les deux en même temps.

Création de clés de partitionnement avec analyse en ligne

Pour configurer l'analyse en ligne comme méthode de partitionnement dynamique pour vos données de streaming, vous devez choisir les paramètres d'enregistrement de données à utiliser comme clés de partitionnement et fournir une valeur pour chaque clé de partitionnement spécifiée.

L'exemple d'enregistrement de données suivant montre comment vous pouvez définir des clés de partitionnement pour celui-ci à l'aide de l'analyse en ligne. Notez que les données doivent être codées au format Base64. Vous pouvez également vous référer à [l'exemple de la CLI](#).

```
{  
  "type": {
```

```
"device": "mobile",
"event": "user_clicked_submit_button"
},
"customer_id": "1234567890",
"event_timestamp": 1565382027, #epoch timestamp
"region": "sample_region"
}
```

Par exemple, vous pouvez choisir de partitionner vos données en fonction du paramètre `customer_id` ou du paramètre `event_timestamp`. Cela signifie que vous souhaitez que la valeur du paramètre `customer_id` ou du paramètre `event_timestamp` de chaque enregistrement soit utilisée pour déterminer le préfixe S3 auquel l'enregistrement doit être livré. Vous pouvez également choisir un paramètre imbriqué, comme `device` avec une expression `.type.device`. Votre logique de partitionnement dynamique peut dépendre de plusieurs paramètres.

Après avoir sélectionné les paramètres de données pour vos clés de partitionnement, vous mappez chaque paramètre à une expression jq valide. Le tableau suivant montre un tel mappage de paramètres avec des expressions jq :

Paramètre	expression jq
<code>customer_id</code>	<code>.customer_id</code>
<code>device</code>	<code>.type.device</code>
<code>year</code>	<code>.event_timestamp strftime("%Y")</code>
<code>month</code>	<code>.event_timestamp strftime("%m")</code>
<code>day</code>	<code>.event_timestamp strftime("%d")</code>
<code>hour</code>	<code>.event_timestamp strftime("%H")</code>

Au moment de l'exécution, Firehose utilise la colonne de droite ci-dessus pour évaluer les paramètres en fonction des données de chaque enregistrement.

Création de clés de partitionnement à l'aide d'une fonction AWS Lambda

Pour les enregistrements de données compressés ou chiffrés, ou pour les données dans un format de fichier autre que JSON, vous pouvez utiliser la fonction AWS Lambda intégrée avec votre propre

code personnalisé pour décompresser, déchiffrer ou transformer les enregistrements afin d'extraire et de renvoyer les champs de données nécessaires au partitionnement. Il s'agit d'une extension de la fonction Lambda de transformation existante qui est disponible aujourd'hui avec Firehose. Vous pouvez transformer, analyser et renvoyer les champs de données que vous pouvez ensuite utiliser pour le partitionnement dynamique à l'aide de la même fonction Lambda.

Voici un exemple de fonction Lambda de traitement de flux Firehose en Python qui rejoue chaque enregistrement lu de l'entrée à la sortie et extrait les clés de partitionnement des enregistrements.

```
from __future__ import print_function
import base64
import json
import datetime

# Signature for all Lambda functions that user must implement
def lambda_handler(firehose_records_input, context):
    print("Received records for processing from DeliveryStream: " +
          firehose_records_input['deliveryStreamArn']
          + ", Region: " + firehose_records_input['region']
          + ", and InvocationId: " + firehose_records_input['invocationId'])

    # Create return value.
    firehose_records_output = {'records': []}

    # Create result object.
    # Go through records and process them

    for firehose_record_input in firehose_records_input['records']:
        # Get user payload
        payload = base64.b64decode(firehose_record_input['data'])
        json_value = json.loads(payload)

        print("Record that was received")
        print(json_value)
        print("\n")
        # Create output Firehose record and add modified payload and record ID to it.
        firehose_record_output = {}
        event_timestamp = datetime.datetime.fromtimestamp(json_value['eventTimestamp'])
        partition_keys = {"customerId": json_value['customerId'],
                          "year": event_timestamp.strftime('%Y'),
                          "month": event_timestamp.strftime('%m'),
                          "date": event_timestamp.strftime('%d'),
```

```

        "hour": event_timestamp.strftime('%H'),
        "minute": event_timestamp.strftime('%M')
    }

    # Create output Firehose record and add modified payload and record ID to it.
    firehose_record_output = {'recordId': firehose_record_input['recordId'],
                              'data': firehose_record_input['data'],
                              'result': 'Ok',
                              'metadata': { 'partitionKeys': partition_keys }}

    # Must set proper record ID
    # Add the record to the list of output records.

    firehose_records_output['records'].append(firehose_record_output)

# At the end return processed records
return firehose_records_output

```

Voici un exemple de fonction Lambda de traitement de flux Firehose dans Go qui rejoue chaque enregistrement lu de l'entrée vers la sortie et extrait les clés de partitionnement des enregistrements.

```

package main

import (
    "fmt"
    "encoding/json"
    "time"
    "strconv"

    "github.com/aws/aws-lambda-go/events"
    "github.com/aws/aws-lambda-go/lambda"
)

type DataFirehoseEventRecordData struct {
    CustomerId string `json:"customerId"`
}

func handleRequest(evnt events.DataFirehoseEvent) (events.DataFirehoseResponse, error) {
    {
        fmt.Printf("InvocationID: %s\n", evnt.InvocationID)
        fmt.Printf("DeliveryStreamArn: %s\n", evnt.DeliveryStreamArn)
    }
}

```

```
fmt.Printf("Region: %s\n", evnt.Region)

var response events.DataFirehoseResponse

for _, record := range evnt.Records {
    fmt.Printf("RecordID: %s\n", record.RecordID)
    fmt.Printf("ApproximateArrivalTimestamp: %s\n", record.ApproximateArrivalTimestamp)

    var transformedRecord events.DataFirehoseResponseRecord
    transformedRecord.RecordID = record.RecordID
    transformedRecord.Result = events.DataFirehoseTransformedStateOk
    transformedRecord.Data = record.Data

    var metaData events.DataFirehoseResponseRecordMetadata
    var recordData DataFirehoseEventRecordData
    partitionKeys := make(map[string]string)

    currentTime := time.Now()
    json.Unmarshal(record.Data, &recordData)
    partitionKeys["customerId"] = recordData.CustomerId
    partitionKeys["year"] = strconv.Itoa(currentTime.Year())
    partitionKeys["month"] = strconv.Itoa(int(currentTime.Month()))
    partitionKeys["date"] = strconv.Itoa(currentTime.Day())
    partitionKeys["hour"] = strconv.Itoa(currentTime.Hour())
    partitionKeys["minute"] = strconv.Itoa(currentTime.Minute())
    metaData.PartitionKeys = partitionKeys
    transformedRecord.Metadata = metaData

    response.Records = append(response.Records, transformedRecord)
}

return response, nil
}

func main() {
    lambda.Start(handleRequest)
}
```

Préfixe de compartiment Amazon S3 pour le partitionnement dynamique

Lorsque vous créez un flux Firehose qui utilise Amazon S3 comme destination, vous devez spécifier un compartiment Amazon S3 dans lequel Firehose doit livrer vos données. Les préfixes de compartiment Amazon S3 sont utilisés pour organiser les données que vous stockez dans des compartiments Amazon S3. Un préfixe de compartiment Amazon S3 est similaire à un répertoire qui vous permet de regrouper des objets similaires.

Avec le partitionnement dynamique, vos données partitionnées sont transmises dans les préfixes Amazon S3 spécifiés. Si vous n'activez pas le partitionnement dynamique, il est facultatif de spécifier un préfixe de compartiment S3 pour votre flux Firehose. Toutefois, si vous choisissez d'activer le partitionnement dynamique, vous devez spécifier les préfixes de compartiment S3 auxquels Firehose fournit les données partitionnées.

Dans chaque flux Firehose dans lequel vous activez le partitionnement dynamique, la valeur du préfixe du compartiment S3 est constituée d'expressions basées sur les clés de partitionnement spécifiées pour ce flux Firehose. En reprenant l'exemple d'enregistrement de données ci-dessus, vous pouvez créer la valeur de préfixe S3 suivante, qui se compose d'expressions basées sur les clés de partitionnement définies ci-dessus :

```
"ExtendedS3DestinationConfiguration": {
  "BucketARN": "arn:aws:s3:::my-logs-prod",
  "Prefix": "customer_id={!partitionKeyFromQuery:customer_id}/
    device={!partitionKeyFromQuery:device}/
    year={!partitionKeyFromQuery:year}/
    month={!partitionKeyFromQuery:month}/
    day={!partitionKeyFromQuery:day}/
    hour={!partitionKeyFromQuery:hour}/"
}
```

Firehose évalue l'expression ci-dessus au moment de l'exécution. Il regroupe en un seul jeu de données les enregistrements qui correspondent à la même expression de préfixe S3 évaluée. Firehose fournit ensuite chaque ensemble de données au préfixe S3 évalué. La fréquence de transmission de l'ensemble de données à S3 est déterminée par le paramètre de la mémoire tampon de flux Firehose. Par conséquent, l'enregistrement de cet exemple est transmis à la clé d'objet S3 suivante :

```
s3://my-logs-prod/customer_id=1234567890/device=mobile/year=2019/month=08/day=09/
hour=20/my-delivery-stream-2019-08-09-23-55-09-a9fa96af-e4e4-409f-bac3-1f804714faaa
```

Pour le partitionnement dynamique, vous devez utiliser le format d'expression suivant dans le préfixe de votre compartiment S3 : `!{namespace:value}`, où l'espace de noms peut être `partitionKeyFromQuery` ou `partitionKeyFromLambda`, ou les deux. Si vous utilisez l'analyse en ligne pour créer les clés de partitionnement de vos données sources, vous devez spécifier une valeur de préfixe de compartiment S3 qui consiste en des expressions spécifiées dans le format suivant : `"partitionKeyFromQuery:keyID"`. Si vous utilisez une fonction AWS Lambda pour créer les clés de partitionnement de vos données sources, vous devez spécifier une valeur de préfixe de compartiment S3 qui consiste en des expressions spécifiées dans le format suivant : `"partitionKeyFromLambda:keyID"`.

Note

Vous pouvez également spécifier la valeur du préfixe du compartiment S3 en utilisant le format de style hive, par exemple `customer_id= ! {partitionKeyFromRequête:Customer_ID}`.

Pour plus d'informations, consultez la section « Choisissez Amazon S3 pour votre destination » dans [Création d'un flux Amazon Firehose](#) et [préfixes personnalisés pour les objets Amazon S3](#).

Partitionnement dynamique des données agrégées

Vous pouvez appliquer un partitionnement dynamique aux données agrégées (par exemple, plusieurs événements, journaux ou enregistrements agrégés en un seul appel d'API `PutRecord` ou `PutRecordBatch`), mais ces données doivent d'abord être désagrégées. Vous pouvez désagréger vos données en activant la désagrégation multi-enregistrements, qui consiste à analyser les enregistrements du flux Firehose et à les séparer.

La désagrégation multi-enregistrements peut être de l'un ou l'autre JSON type, ce qui signifie que la séparation des enregistrements est basée sur des objets JSON consécutifs. La désagrégation peut également être du type `Delimited`, ce qui signifie que la séparation des enregistrements est effectuée sur la base d'un délimiteur personnalisé spécifié. Ce délimiteur personnalisé doit être une chaîne encodée en base 64. Par exemple, si vous souhaitez utiliser la chaîne suivante comme

séparateur personnalisé####, vous devez la spécifier au format codé en base 64, qui la traduit en `IyMjIw==`

Note

Lorsque vous désagrégez des enregistrements JSON, assurez-vous que votre entrée est toujours présentée dans le format JSON pris en charge. Les objets JSON doivent se trouver sur une seule ligne sans délimiteur ou être délimités par de nouvelles lignes (JSONL) uniquement. Un tableau d'objets JSON n'est pas une entrée valide.

Voici des exemples de saisie correcte : `{"a":1}{ "a":2}` and `{"a":1}\n{"a":2}`

Voici un exemple de saisie incorrecte : `[{"a":1}, {"a":2}]`

Avec les données agrégées, lorsque vous activez le partitionnement dynamique, Firehose analyse les enregistrements et recherche des objets JSON valides ou des enregistrements délimités dans chaque appel d'API en fonction du type de désagrégation multi-enregistrements spécifié.

Important

Si vos données sont agrégées, le partitionnement dynamique ne peut être appliqué que si vos données sont d'abord désagrégées.

Important

Lorsque vous utilisez la fonctionnalité de transformation des données dans Firehose, la désagrégation sera appliquée avant la transformation des données. Les données entrant dans Firehose seront traitées dans l'ordre suivant : Désagrégation → Transformation des données via Lambda → Clés de partitionnement.

Ajout d'un nouveau délimiteur de ligne lors de la transmission de données à S3

Vous pouvez activer `New Line Delimiter` pour ajouter un nouveau délimiteur de ligne entre les enregistrements des objets livrés à Amazon S3. Cela peut être utile pour analyser des objets dans Amazon S3. Cela est également particulièrement utile lorsque le partitionnement dynamique

est appliqué à des données agrégées, car la désagrégation multi-enregistrements (qui doit être appliquée aux données agrégées avant de pouvoir être partitionnées dynamiquement) supprime de nouvelles lignes des enregistrements dans le cadre du processus d'analyse.

Comment activer le partitionnement dynamique

Vous pouvez configurer le partitionnement dynamique pour vos flux Firehose via la console de gestion Amazon Data Firehose, la CLI ou les API.

Important

Vous ne pouvez activer le partitionnement dynamique que lorsque vous créez un nouveau flux Firehose. Vous ne pouvez pas activer le partitionnement dynamique pour un flux Firehose existant pour lequel le partitionnement dynamique n'est pas déjà activé.

Pour savoir comment activer et configurer le partitionnement dynamique via la console de gestion Firehose lors de la création d'un nouveau flux Firehose, consultez [Création d'un flux Amazon Firehose](#). Lorsque vous aurez à spécifier la destination de votre flux Firehose, assurez-vous de suivre les étapes décrites dans la section [Choisissez Amazon S3 pour votre destination](#), car actuellement, le partitionnement dynamique n'est pris en charge que pour les flux Firehose utilisant Amazon S3 comme destination.

Une fois le partitionnement dynamique activé sur un flux Firehose actif, vous pouvez mettre à jour la configuration en ajoutant de nouvelles clés de partitionnement, en supprimant ou en mettant à jour les clés de partitionnement existantes et les expressions du préfixe S3. Une fois mis à jour, Firehose commence à utiliser les nouvelles clés et les nouvelles expressions du préfixe S3.

Important

Une fois que vous avez activé le partitionnement dynamique sur un flux Firehose, il ne peut pas être désactivé sur ce flux Firehose.

Gestion des erreurs de partitionnement dynamique

Si Amazon Data Firehose n'est pas en mesure d'analyser les enregistrements de données de votre flux Firehose ou s'il ne parvient pas à extraire les clés de partitionnement spécifiées, ou à évaluer les

expressions incluses dans la valeur du préfixe S3, ces enregistrements de données sont transmis au préfixe du compartiment d'erreur S3 que vous devez spécifier lorsque vous créez le flux Firehose dans lequel vous activez le partitionnement dynamique. Le préfixe du compartiment d'erreur S3 contient tous les enregistrements que Firehose n'est pas en mesure de fournir à la destination S3 spécifiée. Ces enregistrements sont organisés en fonction du type d'erreur. Avec l'enregistrement, l'objet livré comprend également des informations sur l'erreur afin d'aider à la comprendre et à la résoudre.

Vous devez spécifier un préfixe de compartiment d'erreur S3 pour un flux Firehose si vous souhaitez activer le partitionnement dynamique pour ce flux Firehose. Si vous ne souhaitez pas activer le partitionnement dynamique pour un flux Firehose, il est facultatif de spécifier un préfixe de compartiment d'erreur S3.

Mise en mémoire tampon des données et partitionnement dynamique

Amazon Data Firehose met en mémoire tampon les données de streaming entrantes jusqu'à une certaine taille et pendant une certaine période avant de les diffuser vers les destinations spécifiées. Vous pouvez configurer la taille de la mémoire tampon et l'intervalle de la mémoire tampon lors de la création de nouveaux flux Firehose ou mettre à jour la taille de la mémoire tampon et l'intervalle de mémoire tampon sur vos flux Firehose existants. La taille de la mémoire tampon est mesurée en Mo et l'intervalle de mémoire tampon est mesuré en secondes.

Lorsque le partitionnement dynamique est activé, Firehose met en mémoire tampon en interne les enregistrements appartenant à une partition donnée en fonction de l'indice de mise en mémoire tampon configuré (taille et durée) avant de les transférer dans votre compartiment Amazon S3. Afin de fournir des objets de taille maximale, Firehose utilise une mise en mémoire tampon en plusieurs étapes en interne. Par conséquent, end-to-end le délai d'un lot d'enregistrements peut être 1,5 fois supérieur au délai d'indication de mise en mémoire tampon configuré. Cela affecte la fraîcheur des données d'un flux Firehose.

Le nombre de partitions actives est le nombre total de partitions actives dans le tampon de diffusion. Par exemple, si la requête de partitionnement dynamique crée trois partitions par seconde et que vous avez une configuration d'indice de mémoire tampon déclenchant la livraison toutes les 60 secondes, vous aurez en moyenne 180 partitions actives. Si Firehose ne parvient pas à livrer les données d'une partition à une destination, cette partition est considérée comme active dans le tampon de livraison jusqu'à ce qu'elle puisse être livrée.

Une nouvelle partition est créée lorsqu'un préfixe S3 est évalué à une nouvelle valeur sur la base des champs de données de l'enregistrement et des expressions du préfixe S3. Une nouvelle mémoire tampon est créée pour chaque partition active. Chaque enregistrement suivant avec le même préfixe S3 évalué est envoyé dans cette mémoire tampon.

Une fois que le tampon atteint la limite de taille de la mémoire tampon ou l'intervalle de temps de la mémoire tampon, Firehose crée un objet avec les données de la mémoire tampon et les transmet au préfixe Amazon S3 spécifié. Une fois l'objet livré, la mémoire tampon de cette partition et la partition elle-même sont supprimées et supprimées du nombre de partitions actives.

Firehose fournit chaque donnée de la mémoire tampon sous la forme d'un objet unique une fois que la taille ou l'intervalle de la mémoire tampon sont atteints pour chaque partition séparément. Une fois que le nombre de partitions actives atteint la limite de 500 par flux Firehose, le reste des enregistrements du flux Firehose est transmis au préfixe de compartiment d'erreur S3 spécifié (`activePartitionExceeded`). Vous pouvez utiliser le [formulaire Amazon Data Firehose Limits](#) pour demander une augmentation de ce quota jusqu'à 5 000 partitions actives par flux Firehose donné. Si vous avez besoin de plus de partitions, vous pouvez créer davantage de flux Firehose et répartir les partitions actives entre eux.

Conversion de votre format d'enregistrement d'entrée dans Firehose

Amazon Data Firehose peut convertir le format de vos données d'entrée de JSON vers [Apache Parquet](#) ou [Apache ORC](#) avant de les stocker dans Amazon S3. Parquet et ORC sont des formats de données en colonnes qui gagnent de l'espace et permettent des requêtes plus rapides que les formats orientés lignes comme JSON. Si vous souhaitez convertir un format d'entrée autre que JSON, tel que des valeurs séparées par des virgules (CSV) ou du texte structuré, vous pouvez AWS Lambda d'abord le transformer en JSON. Pour plus d'informations, consultez [Transformation de données](#).

Rubriques

- [Exigences pour la conversion du format d'enregistrement](#)
- [Choix du désérialiseur JSON](#)
- [Choix du sérialiseur](#)
- [Conversion du format d'enregistrement d'entrée \(console\)](#)
- [Conversion du format d'enregistrement d'entrée \(API\)](#)
- [Gestion des erreurs de conversion de format d'enregistrement](#)
- [Exemple de conversion du format d'enregistrement](#)

Exigences pour la conversion du format d'enregistrement

Amazon Data Firehose a besoin des trois éléments suivants pour convertir le format de vos données d'enregistrement :

- Un désérialiseur pour lire le JSON de vos données d'entrée — [Vous pouvez choisir l'un des deux types de désérialiseurs : Apache Hive JSON ou OpenX JSON. SerDe SerDe](#)

Note

Lorsque vous combinez plusieurs documents JSON dans le même enregistrement, assurez-vous que votre entrée est toujours présentée dans le format JSON pris en charge. Un tableau de documents JSON n'est pas une entrée valide. Par exemple, voici la bonne entrée : `{"a":1}{ "a":2}`

Et voici la saisie incorrecte : [{"a":1}, {"a":2}]

- Un schéma pour déterminer comment interpréter les données : Utilisez [AWS Glue](#) pour créer un schéma dans AWS Glue Data Catalog. Amazon Data Firehose fait ensuite référence à ce schéma et l'utilise pour interpréter vos données d'entrée. Vous pouvez utiliser le même schéma pour configurer Amazon Data Firehose et votre logiciel d'analyse. Pour plus d'informations, consultez la section [Remplissage du catalogue de données AWS Glue](#) dans le manuel du AWS Glue développeur.

Note

Le schéma créé dans le catalogue de AWS Glue données doit correspondre à la structure des données d'entrée. Sinon, les données converties ne contiendront pas d'attributs non spécifiés dans le schéma. Si vous utilisez du JSON imbriqué, utilisez un type STRUCT dans le schéma qui reflète la structure de vos données JSON. Consultez [cet exemple](#) pour savoir comment gérer le JSON imbriqué avec un type STRUCT.

- Un sérialiseur pour convertir les données au format de stockage en colonnes cible (Parquet ou ORC) — [Vous pouvez choisir l'un des deux types de sérialiseurs : ORC ou Parquet. SerDe SerDe](#)

Important

Si vous activez la conversion du format d'enregistrement, vous ne pouvez pas définir votre destination Amazon Data Firehose comme Amazon OpenSearch Service, Amazon Redshift ou Splunk. Lorsque la conversion de format est activée, Amazon S3 est la seule destination que vous pouvez utiliser pour votre flux Firehose.

Vous pouvez convertir le format de vos données même si vous agrégez vos enregistrements avant de les envoyer à Amazon Data Firehose.

Choix du désérialiseur JSON

Choisissez le JSON [OpenX SerDe si votre fichier JSON](#) d'entrée contient des horodatages dans les formats suivants :

- `aaaa-MM-jj'T'HH:mm:ss[.S]'Z'`, où la fraction peut avoir jusqu'à 9 chiffres : Par exemple `2017-02-07T15:13:01.39256Z`.
- `aaaa-[M]M-[d]j HH:mm:ss[.S]`, où la fraction peut avoir jusqu'à 9 chiffres : Par exemple `2017-02-07 15:13:01.14`.
- Heure Unix en secondes : Par exemple `1518033528`.
- Heure Unix en millisecondes : Par exemple `1518033528123`.
- Heure Unix en secondes à virgule flottante : Par exemple `1518033528.123`.

Le JSON OpenX SerDe peut convertir des points (.) en traits de soulignement (_). Il peut également convertir les clés JSON en minuscules avant de les désérialiser. [Pour plus d'informations sur les options disponibles avec ce désérialiseur via Amazon Data Firehose, consultez OpenX. JsonSerDe](#)

Si vous ne savez pas quel désérialiseur choisir, utilisez le JSON OpenX SerDe, sauf si vous avez des horodatages qu'il ne prend pas en charge.

Si vous avez des horodatages dans des formats autres que ceux listés précédemment, utilisez le [JSON SerDe d'Apache Hive](#). Lorsque vous choisissez ce désérialiseur, vous pouvez spécifier les formats d'horodatage à utiliser. Pour ce faire, suivez la syntaxe du modèle des chaînes de format `DateTimeFormat` Joda Time. Pour plus d'informations, consultez la section [Classe DateTimeFormat](#).

Vous pouvez également utiliser la valeur spéciale `millis` pour analyser les horodatages Unix en millisecondes. Si vous ne spécifiez aucun format, Amazon Data Firehose l'utilise `java.sql.Timestamp::valueOf` par défaut.

Le Hive JSON SerDe n'autorise pas les opérations suivantes :

- Des points (.) dans les noms de colonnes.
- Les champs dont le type est `uniontype`.
- Les champs qui ont des types numériques dans le schéma, mais qui sont des chaînes dans le code JSON. Par exemple, si le schéma est (un int) et que le JSON l'est `{"a": "123"}`, le Hive génère SerDe une erreur.

The Hive SerDe ne convertit pas le JSON imbriqué en chaînes. Par exemple, si vous avez `{"a": {"inner": 1}}`, il ne traite pas `{"inner": 1}` comme une chaîne.

Choix du sérialiseur

Le sérialiseur que vous choisissez dépend de vos besoins métier. [Pour en savoir plus sur les deux options du sérialiseur, consultez ORC SerDe et Parquet. SerDe](#)

Conversion du format d'enregistrement d'entrée (console)

Vous pouvez activer la conversion des formats de données sur la console lorsque vous créez ou mettez à jour un flux Firehose. Lorsque la conversion des formats de données est activée, Amazon S3 est la seule destination que vous pouvez configurer pour le flux Firehose. De plus, la compression Amazon S3 est désactivée lorsque vous activez la conversion de format. Toutefois, la compression Snappy se fait automatiquement dans le cadre du processus de conversion. Le format de cadrage pour Snappy utilisé par Amazon Data Firehose dans ce cas est compatible avec Hadoop. Cela signifie que vous pouvez utiliser les résultats de la compression Snappy et exécuter des requêtes sur ces données dans Athena. [Pour le format de cadrage Snappy sur lequel repose Hadoop, consultez `.java.BlockCompressorStream`](#)

Pour activer la conversion des formats de données pour un flux Firehose

1. [Connectez-vous à la AWS Management Console console Amazon Data Firehose et ouvrez-la à l'adresse `https://console.aws.amazon.com/firehose/`.](#)
2. Choisissez un stream Firehose à mettre à jour ou créez un nouveau stream Firehose en suivant les étapes décrites dans [Création d'un stream Firehose](#)
3. Sous Convert record format (Convertir le format d'enregistrement), définissez Record format conversion (Conversion du format d'enregistrement) sur Enabled (Activé).
4. Choisissez le format de sortie souhaité. Pour plus d'informations sur les deux options, consultez [Apache Parquet](#) et [Apache ORC](#).
5. Choisissez une AWS Glue table pour spécifier un schéma pour vos enregistrements sources. Définissez la région, la base de données, la table et la version de la table.

Conversion du format d'enregistrement d'entrée (API)

[Si vous souhaitez qu'Amazon Data Firehose convertisse le format de vos données d'entrée de JSON en Parquet ou ORC, spécifiez l'`DataFormatConversionConfiguration` élément facultatif dans `ExtendedS3` ou `ExtendedS3 DestinationConfiguration`. `DestinationUpdate`](#) Si vous le spécifiez [DataFormatConversionConfiguration](#), les restrictions suivantes s'appliquent :

- Dans [BufferingHints](#), vous ne pouvez pas définir une valeur inférieure `SizeInMBs` à 64 si vous activez la conversion du format d'enregistrement. De plus, lorsque la conversion de format n'est pas activée, la valeur par défaut est 5. La valeur devient 128 lorsque vous activez la conversion.
- [Vous devez définir `CompressionFormat` dans `ExtendedS3 DestinationConfiguration` ou `ExtendedS3` sur `DestinationUpdate`](#) `UNCOMPRESSED` La valeur par défaut de `CompressionFormat` est `UNCOMPRESSED`. Par conséquent, vous pouvez également le laisser non spécifié dans [DestinationConfigurationExtendedS3](#). Les données sont toujours compressées dans le cadre du processus de sérialisation, en utilisant la compression Snappy, par défaut. Le format de cadrage pour Snappy utilisé par Amazon Data Firehose dans ce cas est compatible avec Hadoop. Cela signifie que vous pouvez utiliser les résultats de la compression Snappy et exécuter des requêtes sur ces données dans Athena. [Pour le format de cadrage Snappy sur lequel repose Hadoop, consultez `.java.BlockCompressorStream`](#) Lorsque vous configurez le sérialiseur, vous pouvez choisir d'autres types de compression.

Gestion des erreurs de conversion de format d'enregistrement

Lorsqu'Amazon Data Firehose ne parvient pas à analyser ou à désérialiser un enregistrement (par exemple, lorsque les données ne correspondent pas au schéma), il l'écrit dans Amazon S3 avec un préfixe d'erreur. Si cette écriture échoue, Amazon Data Firehose réessaie définitivement, bloquant ainsi toute livraison ultérieure. Pour chaque échec d'enregistrement, Amazon Data Firehose écrit un document JSON avec le schéma suivant :

```
{
  "attemptsMade": long,
  "arrivalTimestamp": long,
  "lastErrorCode": string,
  "lastErrorMessage": string,
  "attemptEndingTimestamp": long,
  "rawData": string,
  "sequenceNumber": string,
  "subSequenceNumber": long,
  "dataCatalogTable": {
    "catalogId": string,
    "databaseName": string,
    "tableName": string,
    "region": string,
    "versionId": string,
    "catalogArn": string
  }
}
```

```
}
```

Exemple de conversion du format d'enregistrement

Pour un exemple de configuration de la conversion de format d'enregistrement avec AWS CloudFormation, voir [AWS: : DataFirehose : : DeliveryStream](#).

Utilisation du service géré Amazon pour Apache Flink

Avec le service géré Amazon pour Apache Flink vous pouvez utiliser Java, Scala ou SQL pour traiter et analyser les données de streaming. Ce service vous permet de créer et d'exécuter rapidement un code sur des sources de diffusion pour effectuer des analyses de séries temporelles, alimenter des tableaux de bord en temps réel et générer des métriques en temps réel.

Pour un exemple d'intégration avec Amazon Managed Service pour Apache Flink, consultez [Exemple : Writing to Amazon Data Firehose](#).

Dans cet exercice, vous allez créer une application Apache Flink qui possède un flux de données Kinesis comme source et un flux Firehose comme récepteur. À l'aide du récepteur, vous pouvez vérifier la sortie de l'application dans un compartiment Amazon S3.

Avant de commencer, configurez les prérequis :

- [Composants du service géré pour l'application Apache Flink](#)
- [Prérequis pour effectuer les exercices](#)

Comprendre la livraison des données Amazon Data Firehose

Une fois les données envoyées à votre flux Firehose, elles sont automatiquement transmises à la destination que vous avez choisie.

Important

Si vous utilisez la bibliothèque producteur Kinesis (KPL) pour écrire des données dans un flux de données Kinesis, vous pouvez utiliser un regroupement pour combiner ces enregistrements Kinesis. Si vous utilisez ensuite ce flux de données comme source pour votre flux Firehose, Amazon Data Firehose désagrège les enregistrements avant de les envoyer à destination. Si vous configurez votre flux Firehose pour transformer les données, Amazon Data Firehose désagrège les enregistrements avant de les transmettre. AWS Lambda Pour en savoir plus, consultez [Developing Amazon Kinesis Data Streams Producers Using the Kinesis Producer Library](#) et [Aggregation](#) dans le Guide du développeur Amazon Kinesis Data Streams.

Rubriques

- [Configuration du format de livraison des données](#)
- [Comprendre la fréquence de diffusion des données](#)
- [Gérer les défaillances de livraison des données](#)
- [Configurer le format du nom d'objet Amazon S3](#)
- [Configurer la rotation de l'index pour le OpenSearch service](#)
- [Comprenez la livraison entre AWS les comptes et les régions](#)
- [Enregistrements dupliqués](#)
- [Suspendre et reprendre un stream Firehose](#)

Configuration du format de livraison des données

Pour la livraison de données à Amazon Simple Storage Service (Amazon S3), Firehose concatène plusieurs enregistrements entrants en fonction de la configuration de mise en mémoire tampon de

vos flux Firehose. Il diffuse ensuite les enregistrements vers Amazon S3 en tant qu'objet Amazon S3. Par défaut, Firehose concatène les données sans aucun délimiteur. [Si vous souhaitez avoir de nouveaux délimiteurs de ligne entre les enregistrements, vous pouvez ajouter de nouveaux délimiteurs de ligne en activant cette fonctionnalité dans la configuration de la console Firehose ou dans le paramètre API.](#)

Pour la livraison de données à Amazon Redshift, Firehose envoie d'abord les données entrantes à votre compartiment S3 dans le format décrit précédemment. Firehose émet ensuite une commande Amazon COPY Redshift pour charger les données de votre compartiment S3 vers votre cluster provisionné Amazon Redshift ou votre groupe de travail Amazon Redshift Serverless. Assurez-vous qu'une fois qu'Amazon Data Firehose a concaténé plusieurs enregistrements entrants dans un objet Amazon S3, celui-ci peut être copié sur votre cluster provisionné Amazon Redshift ou votre groupe de travail Amazon Redshift Serverless. Pour plus d'informations, consultez [Paramètres du format de données de la commande COPY Amazon Redshift.](#)

Pour la livraison de données vers OpenSearch Service et OpenSearch Serverless, Amazon Data Firehose met en mémoire tampon les enregistrements entrants en fonction de la configuration de mise en mémoire tampon de votre flux Firehose. Il génère ensuite une demande OpenSearch groupée Service ou OpenSearch Serverless pour indexer plusieurs enregistrements dans votre cluster de OpenSearch services ou votre collection OpenSearch Serverless. Assurez-vous que votre enregistrement est codé en UTF-8 et aplati en un objet JSON d'une seule ligne avant de l'envoyer à Amazon Data Firehose. En outre, l'`rest.action.multi.allow_explicit_indexoption` de votre cluster de OpenSearch services doit être définie sur `true` (par défaut) pour prendre en charge les demandes groupées avec un index explicite défini par enregistrement. Pour plus d'informations, consultez les [options avancées de configuration du OpenSearch service](#) dans le manuel Amazon OpenSearch Service Developer Guide.

Pour la livraison des données à Splunk, Amazon Data Firehose concatène les octets que vous envoyez. Si vous voulez des délimiteurs dans vos données, tels qu'un caractère de nouvelle ligne, vous devez les insérer par vous-même. Vérifiez que Splunk est configuré pour analyser ces délimiteurs.

Lorsque vous transmettez des données à un point de terminaison HTTP appartenant à un fournisseur de services tiers pris en charge, vous pouvez utiliser le service Amazon Lambda intégré pour créer une fonction permettant de transformer le ou les enregistrements entrants au format correspondant à celui attendu par l'intégration du fournisseur de services. Contactez le fournisseur de services tiers dont vous avez choisi le point de terminaison HTTP pour votre destination pour en savoir plus sur son format d'enregistrement accepté.

Pour la transmission des données à Snowflake, Amazon Data Firehose met en mémoire tampon les données en interne pendant une seconde et utilise les opérations de l'API de streaming Snowflake pour insérer des données dans Snowflake. Par défaut, les enregistrements que vous insérez sont vidés et validés dans la table Snowflake toutes les secondes. Une fois que vous avez effectué l'appel d'insertion, Firehose émet une CloudWatch métrique qui mesure le temps nécessaire pour que les données soient validées dans Snowflake. Firehose ne prend actuellement en charge qu'un seul élément JSON comme charge utile d'enregistrement et ne prend pas en charge les tableaux JSON. Assurez-vous que votre charge utile d'entrée est un objet JSON valide et qu'elle est bien formée sans guillemets, guillemets ou caractères d'échappement supplémentaires.

Comprendre la fréquence de diffusion des données

Chaque destination Firehose possède sa propre fréquence de livraison des données. Pour plus d'informations, consultez [Comprendre les conseils de mise en mémoire tampon](#).

Gérer les défaillances de livraison des données

Chaque destination Amazon Data Firehose dispose de sa propre gestion des défaillances de livraison de données.

Amazon S3

La diffusion de données à votre compartiment S3 peut échouer pour plusieurs raisons. Par exemple, il se peut que le compartiment n'existe plus, que le rôle IAM assumé par Amazon Data Firehose n'ait pas accès au compartiment, que le réseau soit défaillant ou que des événements similaires se produisent. Dans ces conditions, Amazon Data Firehose continue de réessayer pendant 24 heures au maximum jusqu'à ce que la livraison aboutisse. La durée maximale de stockage des données d'Amazon Data Firehose est de 24 heures. En cas de défaillance de leur diffusion pendant plus de 24 heures, vos données sont perdues.

Amazon Redshift

Pour une destination Amazon Redshift, vous pouvez spécifier une durée de nouvelle tentative (0 à 7 200 secondes) lors de la création d'un flux Firehose.

La diffusion des données à votre cluster Amazon Redshift ou à votre groupe de travail Amazon Redshift sans serveur peut échouer pour plusieurs raisons. Par exemple, vous pouvez avoir une configuration de cluster incorrecte pour votre flux Firehose, un cluster ou un groupe de

travail en cours de maintenance, ou une panne réseau. Dans ces conditions, Amazon Data Firehose réessaie pendant la durée spécifiée et ignore ce lot spécifique d'objets Amazon S3. Les informations concernant les documents ignorés sont transmises à votre compartiment S3 en tant que fichier manifeste dans le dossier `errors/` que vous pouvez utiliser pour le renvoi manuel. Pour plus d'informations sur la manière de COPIER manuellement des données à l'aide de fichiers manifestes, consultez [Utilisation d'un manifeste pour spécifier les fichiers de données](#).

Amazon OpenSearch Service et OpenSearch Serverless

Pour la destination OpenSearch Service et OpenSearch Serverless, vous pouvez spécifier une durée de nouvelle tentative (0 à 7 200 secondes) lors de la création du stream Firehose.

La livraison des données à votre cluster OpenSearch de services ou à votre collecte OpenSearch sans serveur peut échouer pour plusieurs raisons. Par exemple, vous pouvez avoir une configuration de cluster de OpenSearch services ou de collecte OpenSearch sans serveur incorrecte pour votre flux Firehose, OpenSearch un cluster de services OpenSearch ou une collection sans serveur en cours de maintenance, une panne réseau ou des événements similaires. Dans ces conditions, Amazon Data Firehose réessaie pendant la durée spécifiée, puis ignore cette demande d'index en particulier. Les documents ignorés sont transmis à votre compartiment S3 dans le dossier `AmazonOpenSearchService_failed/` que vous pouvez utiliser pour le renvoi manuel.

Pour le OpenSearch service, chaque document possède le format JSON suivant :

```
{
  "attemptsMade": "(number of index requests attempted)",
  "arrivalTimestamp": "(the time when the document was received by Firehose)",
  "errorCode": "(http error code returned by OpenSearch Service)",
  "errorMessage": "(error message returned by OpenSearch Service)",
  "attemptEndingTimestamp": "(the time when Firehose stopped attempting index request)",
  "esDocumentId": "(intended OpenSearch Service document ID)",
  "esIndexName": "(intended OpenSearch Service index name)",
  "esTypeName": "(intended OpenSearch Service type name)",
  "rawData": "(base64-encoded document data)"
}
```

Pour OpenSearch Serverless, chaque document possède le format JSON suivant :

```
{
```

```
"attemptsMade": "(number of index requests attempted)",
"arrivalTimestamp": "(the time when the document was received by Firehose)",
"errorCode": "(http error code returned by OpenSearch Serverless)",
"errorMessage": "(error message returned by OpenSearch Serverless)",
"attemptEndingTimestamp": "(the time when Firehose stopped attempting index
request)",
"osDocumentId": "(intended OpenSearch Serverless document ID)",
"osIndexName": "(intended OpenSearch Serverless index name)",
"rawData": "(base64-encoded document data)"
}
```

Splunk

Lorsqu'Amazon Data Firehose envoie des données à Splunk, il attend un accusé de réception de Splunk. En cas d'erreur ou si l'accusé de réception n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données à Splunk, qu'il s'agisse d'une première tentative ou d'une nouvelle tentative, il redémarre le compteur de délais d'accusé de réception. Ensuite, il attend un accusé de réception de la part de Splunk. Même si la durée de la nouvelle tentative expire, Amazon Data Firehose attend toujours l'accusé de réception jusqu'à ce qu'il le reçoive ou que le délai d'expiration de l'accusé de réception soit atteint. Si l'accusé de réception expire, Amazon Data Firehose vérifie s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive un accusé de réception ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Le défaut de réception d'un accusé de réception n'est pas le seul type d'erreur de remise de données possible. Pour en savoir plus sur les autres types d'erreurs de remise de données, consultez [Erreurs de livraison de données Splunk](#). Les erreurs de remise de données déclenchent la logique de relance si votre durée des nouvelles tentatives est supérieure à 0.

Voici un exemple d'enregistrement d'erreur.

```
{
  "attemptsMade": 0,
```

```
"arrivalTimestamp": 1506035354675,
"errorCode": "Splunk.AckTimeout",
"errorMessage": "Did not receive an acknowledgement from HEC before the HEC
acknowledgement timeout expired. Despite the acknowledgement timeout, it's possible
the data was indexed successfully in Splunk. Amazon Data Firehose backs up in
Amazon S3 data for which the acknowledgement timeout expired.",
"attemptEndingTimestamp": 13626284715507,
"rawData":
"MiAyNTE2MjAyNzIyMDkgZW5pLTA1ZjMyMmQ1IDIxOC45Mi4xODguMjE0IDE3Mi4xNi4xLjE2NyAyNTIzMyAxNDMzID
"EventId": "49577193928114147339600778471082492393164139877200035842.0"
}
```

Destination du point de terminaison HTTP

Lorsqu'Amazon Data Firehose envoie des données vers une destination de point de terminaison HTTP, il attend une réponse de cette destination. En cas d'erreur ou si la réponse n'arrive pas dans le délai imparti, Amazon Data Firehose lance le compteur de durée des nouvelles tentatives. Il effectue de nouvelles tentatives jusqu'à ce que la durée des nouvelles tentatives arrive à expiration. Après cela, Amazon Data Firehose considère qu'il s'agit d'un échec de livraison des données et sauvegarde les données dans votre compartiment Amazon S3.

Chaque fois qu'Amazon Data Firehose envoie des données vers une destination de point de terminaison HTTP, qu'il s'agisse d'une tentative initiale ou d'une nouvelle tentative, il redémarre le compteur de délais de réponse. Il attend ensuite qu'une réponse arrive de la destination du point de terminaison HTTP. Même si le délai de réessai expire, Amazon Data Firehose attend toujours la réponse jusqu'à ce qu'il la reçoive ou que le délai de réponse soit atteint. Si le délai de réponse est dépassé, Amazon Data Firehose vérifie s'il reste du temps dans le compteur de nouvelles tentatives. Si c'est le cas, il réitère les tentatives et répète la logique jusqu'à ce qu'il reçoive une réponse ou qu'il détermine que le délai imparti pour les nouvelles tentatives est arrivé à son terme.

Le défaut de réception d'une réponse n'est pas le seul type d'erreur de remise de données possible. Pour en savoir plus sur les autres types d'erreurs de remise de données, consultez [HTTP Endpoint Data Delivery Errors](#)

Voici un exemple d'enregistrement d'erreur.

```
{
  "attemptsMade":5,
  "arrivalTimestamp":1594265943615,
  "errorCode":"HttpEndpoint.DestinationException",
```

```
"errorMessage":"Received the following response from the endpoint destination.
{"requestId": "109777ac-8f9b-4082-8e8d-b4f12b5fc17b", "timestamp": 1594266081268,
"errorMessage": "Unauthorized"}",
"attemptEndingTimestamp":1594266081318,
"rawData":"c2FtcGx1IHJhdyBkYXRh",
"subsequenceNumber":0,
"dataId":"49607357361271740811418664280693044274821622880012337186.0"
}
```

Destination aux flocons de neige

Pour la destination Snowflake, lorsque vous créez un stream Firehose, vous pouvez spécifier une durée de nouvelle tentative facultative (0 à 7 200 secondes). La valeur par défaut pour la durée des nouvelles tentatives est de 60 secondes.

La livraison des données vers votre table Snowflake peut échouer pour plusieurs raisons, telles qu'une configuration de destination Snowflake incorrecte, une panne de Snowflake, une défaillance du réseau, etc. La politique de nouvelle tentative ne s'applique pas aux erreurs non récupérables. Par exemple, si Snowflake rejette votre charge utile JSON parce qu'une colonne supplémentaire est manquante dans le tableau, Firehose n'essaie pas de la livrer à nouveau. Il crée plutôt une sauvegarde pour tous les échecs d'insertion dus à des problèmes de charge utile JSON dans votre compartiment d'erreurs S3.

De même, si la livraison échoue en raison d'un rôle, d'une table ou d'une base de données incorrects, Firehose ne réessaie pas et écrit les données dans votre compartiment S3. La durée de la nouvelle tentative ne s'applique qu'en cas d'échec dû à un problème de service Snowflake, à des problèmes réseau transitoires, etc. Dans ces conditions, Firehose réessaie pendant la durée spécifiée avant de les transmettre à S3. Les enregistrements défectueux sont livrés dans le dossier snowflake-failed/, que vous pouvez utiliser pour le remplissage manuel.

Voici un exemple de JSON pour chaque enregistrement que vous transmettez à S3.

```
{
  "attemptsMade": 3,
  "arrivalTimestamp": 1594265943615,
  "errorCode": "Snowflake.InvalidColumns",
  "errorMessage": "Snowpipe Streaming does not support columns of type
  AUTOINCREMENT, IDENTITY, GEO, or columns with a default value or collation",
  "attemptEndingTimestamp": 1712937865543,
  "rawData": "c2FtcGx1IHJhdyBkYXRh"
}
```

Configurer le format du nom d'objet Amazon S3

Lorsque Firehose fournit des données à Amazon S3, le nom de la clé de l'objet S3 suit le format `<evaluated prefix><suffix>`, où le suffixe a le format `----- <Firehose stream name><Firehose stream version><year><month><day><hour><minute><second><uuid><file extension><Firehose stream version>` commence par 1 et augmente de 1 pour chaque changement de configuration du flux Firehose. Vous pouvez modifier les configurations de flux Firehose (par exemple, le nom du compartiment S3, les indications de mise en mémoire tampon, la compression et le chiffrement). Vous pouvez le faire à l'aide de la console Firehose ou de l'opération [UpdateDestinationAPI](#).

Pour `<evaluated prefix>`, Firehose ajoute un préfixe d'heure par défaut au format `YYYY/MM/dd/HH`. Ce préfixe crée une hiérarchie logique dans le compartiment, où chaque barre oblique (`/`) crée un niveau dans la hiérarchie. Vous pouvez modifier cette structure en spécifiant un préfixe personnalisé qui inclut les expressions évaluées lors de l'exécution. Pour plus d'informations sur la manière de spécifier un préfixe personnalisé, consultez la section [Préfixes personnalisés pour les objets Amazon Simple Storage Service](#).

Par défaut, le fuseau horaire utilisé pour le préfixe et le suffixe est en UTC, mais vous pouvez le modifier pour le fuseau horaire de votre choix. Par exemple, pour utiliser l'heure normale du Japon au lieu de l'heure UTC, vous pouvez configurer le fuseau horaire vers Asia/Tokyo dans le [paramètre AWS Management Console ou dans le paramétrage de l'API \(\) CustomTimeZone](#). La liste suivante contient les fuseaux horaires pris en charge par Firehose pour la configuration du préfixe S3.

Fuseaux horaires

Voici une liste des fuseaux horaires pris en charge par Firehose pour la configuration du préfixe S3.

Africa

```
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmera
Africa/Bangui
Africa/Banjul
Africa/Bissau
Africa/Blantyre
Africa/Bujumbura
Africa/Cairo
```

Africa/Casablanca
Africa/Conakry
Africa/Dakar
Africa/Dar_es_Salaam
Africa/Djibouti
Africa/Douala
Africa/Freetown
Africa/Gaborone
Africa/Harare
Africa/Johannesburg
Africa/Kampala
Africa/Khartoum
Africa/Kigali
Africa/Kinshasa
Africa/Lagos
Africa/Libreville
Africa/Lome
Africa/Luanda
Africa/Lubumbashi
Africa/Lusaka
Africa/Malabo
Africa/Maputo
Africa/Maseru
Africa/Mbabane
Africa/Mogadishu
Africa/Monrovia
Africa/Nairobi
Africa/Ndjamena
Africa/Niamey
Africa/Nouakchott
Africa/Ouagadougou
Africa/Porto-Novo
Africa/Sao_Tome
Africa/Timbuktu
Africa/Tripoli
Africa/Tunis
Africa/Windhoek

America

America/Adak
America/Anchorage
America/Anguilla

America/Antigua
America/Aruba
America/Asuncion
America/Barbados
America/Belize
America/Bogota
America/Buenos_Aires
America/Caracas
America/Cayenne
America/Cayman
America/Chicago
America/Costa_Rica
America/Cuiaba
America/Curacao
America/Dawson_Creek
America/Denver
America/Dominica
America/Edmonton
America/El_Salvador
America/Fortaleza
America/Godthab
America/Grand_Turk
America/Grenada
America/Guadeloupe
America/Guatemala
America/Guayaquil
America/Guyana
America/Halifax
America/Havana
America/Indianapolis
America/Jamaica
America/La_Paz
America/Lima
America/Los_Angeles
America/Managua
America/Manaus
America/Martinique
America/Mazatlan
America/Mexico_City
America/Miquelon
America/Montevideo
America/Montreal
America/Montserrat
America/Nassau

```
America/New_York  
America/Noronha  
America/Panama  
America/Paramaribo  
America/Phoenix  
America/Port_of_Spain  
America/Port-au-Prince  
America/Porto_Acre  
America/Puerto_Rico  
America/Regina  
America/Rio_Branco  
America/Santiago  
America/Santo_Domingo  
America/Sao_Paulo  
America/Scoresbysund  
America/St_Johns  
America/St_Kitts  
America/St_Lucia  
America/St_Thomas  
America/St_Vincent  
America/Tegucigalpa  
America/Thule  
America/Tijuana  
America/Tortola  
America/Vancouver  
America/Winnipeg
```

Antarctica

```
Antarctica/Casey  
Antarctica/DumontDUrville  
Antarctica/Mawson  
Antarctica/McMurdo  
Antarctica/Palmer
```

Asia

```
Asia/Aden  
Asia/Almaty  
Asia/Amman  
Asia/Anadyr  
Asia/Aqtau  
Asia/Aqtobe
```

Asia/Ashgabat
Asia/Ashkhabad
Asia/Baghdad
Asia/Bahrain
Asia/Baku
Asia/Bangkok
Asia/Beirut
Asia/Bishkek
Asia/Brunei
Asia/Calcutta
Asia/Colombo
Asia/Dacca
Asia/Damascus
Asia/Dhaka
Asia/Dubai
Asia/Dushanbe
Asia/Hong_Kong
Asia/Irkutsk
Asia/Jakarta
Asia/Jayapura
Asia/Jerusalem
Asia/Kabul
Asia/Kamchatka
Asia/Karachi
Asia/Katmandu
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuwait
Asia/Macao
Asia/Magadan
Asia/Manila
Asia/Muscat
Asia/Nicosia
Asia/Novosibirsk
Asia/Phnom_Penh
Asia/Pyongyang
Asia/Qatar
Asia/Rangoon
Asia/Riyadh
Asia/Saigon
Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Taipei

```
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Thimbu
Asia/Thimphu
Asia/Tokyo
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Vientiane
Asia/Vladivostok
Asia/Yakutsk
Asia/Yekaterinburg
Asia/Yerevan
```

Atlantic

```
Atlantic/Azores
Atlantic/Bermuda
Atlantic/Canary
Atlantic/Cape_Verde
Atlantic/Faeroe
Atlantic/Jan_Mayen
Atlantic/Reykjavik
Atlantic/South_Georgia
Atlantic/St_Helena
Atlantic/Stanley
```

Australia

```
Australia/Adelaide
Australia/Brisbane
Australia/Broken_Hill
Australia/Darwin
Australia/Hobart
Australia/Lord_Howe
Australia/Perth
Australia/Sydney
```

Europe

```
Europe/Amsterdam
```

Europe/Andorra
Europe/Athens
Europe/Belgrade
Europe/Berlin
Europe/Brussels
Europe/Bucharest
Europe/Budapest
Europe/Chisinau
Europe/Copenhagen
Europe/Dublin
Europe/Gibraltar
Europe/Helsinki
Europe/Istanbul
Europe/Kaliningrad
Europe/Kiev
Europe/Lisbon
Europe/London
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Minsk
Europe/Monaco
Europe/Moscow
Europe/Oslo
Europe/Paris
Europe/Prague
Europe/Riga
Europe/Rome
Europe/Samara
Europe/Simferopol
Europe/Sofia
Europe/Stockholm
Europe/Tallinn
Europe/Tirane
Europe/Vaduz
Europe/Vienna
Europe/Vilnius
Europe/Warsaw
Europe/Zurich

Indian

Indian/Antananarivo

```
Indian/Chagos  
Indian/Christmas  
Indian/Cocos  
Indian/Comoro  
Indian/Kerguelen  
Indian/Mahe  
Indian/Maldives  
Indian/Mauritius  
Indian/Mayotte  
Indian/Reunion
```

Pacific

```
Pacific/Apia  
Pacific/Auckland  
Pacific/Chatham  
Pacific/Easter  
Pacific/Efate  
Pacific/Enderbury  
Pacific/Fakaofu  
Pacific/Fiji  
Pacific/Funafuti  
Pacific/Galapagos  
Pacific/Gambier  
Pacific/Guadalcanal  
Pacific/Guam  
Pacific/Honolulu  
Pacific/Kiritimati  
Pacific/Kosrae  
Pacific/Majuro  
Pacific/Marquesas  
Pacific/Nauru  
Pacific/Niue  
Pacific/Norfolk  
Pacific/Noumea  
Pacific/Pago_Pago  
Pacific/Palau  
Pacific/Pitcairn  
Pacific/Ponape  
Pacific/Port_Moresby  
Pacific/Rarotonga  
Pacific/Saipan  
Pacific/Tahiti
```

```
Pacific/Tarawa
Pacific/Tongatapu
Pacific/Truk
Pacific/Wake
Pacific/Wallis
```

<file extension>Vous ne pouvez pas modifier le champ du suffixe sauf. Lorsque vous activez la conversion ou la compression des formats de données, Firehose ajoute une extension de fichier en fonction de la configuration. Le tableau suivant explique l'extension de fichier par défaut ajoutée par Firehose :

Configuration	Extension de fichier
Conversion de format de données : parquet	.parquet
Conversion de format de données : ORC	.orc
Compression : Gzip	.gz
Compression : fermeture éclair	.zip
Compression : Snappy	.snappy
Compression : Hadoop-Snappy	.hsnappy

Vous pouvez également spécifier l'extension de fichier que vous préférez dans la console ou l'API Firehose. L'extension de fichier doit commencer par un point (.) et peut contenir les caractères autorisés : 0-9a-z ! -_.*' (). L'extension de fichier ne peut pas dépasser 128 caractères.

Note

Lorsque vous spécifiez une extension de fichier, elle remplace l'extension de fichier par défaut ajoutée par Firehose [lorsque la conversion ou la compression des formats de données sont](#) activées.

Configurer la rotation de l'index pour le OpenSearch service

Pour la destination du OpenSearch service, vous pouvez spécifier une option de rotation d'index basée sur le temps parmi l'une des cinq options suivantes : NoRotationOneHour,, OneDayOneWeek, ouOneMonth.

En fonction de l'option de rotation que vous choisissez, Amazon Data Firehose ajoute une partie de l'horodatage d'arrivée UTC au nom d'index que vous avez spécifié. Il effectue une rotation de cet horodatage en conséquence. L'exemple suivant montre le nom d'index obtenu dans OpenSearch Service pour chaque option de rotation d'index, où le nom d'index spécifié est myindex et l'horodatage d'arrivée est. 2016-02-25T13:00:00Z

RotationPeriod	IndexName
NoRotation	myindex
OneHour	myindex-2016-02-25-13
OneDay	myindex-2016-02-25
OneWeek	myindex-2016-w08
OneMonth	myindex-2016-02

Note

Avec l'option OneWeek, Data Firehose crée automatiquement des index au format <ANNÉE>-w<NUMÉRO DE SEMAINE> (par exemple, 2020-w33), où le numéro de semaine est calculé en utilisant l'heure universelle coordonnées (UTC) et selon les conventions américaines suivantes :

- Une semaine commence le dimanche
- La première semaine de l'année est la première semaine de l'année qui contient un samedi.

Comprenez la livraison entre AWS les comptes et les régions

Amazon Data Firehose prend en charge la diffusion de données vers des destinations de point de terminaison HTTP pour tous les AWS comptes. Le flux Firehose et le point de terminaison HTTP que vous choisissez comme destination peuvent appartenir à des comptes différents AWS .

Amazon Data Firehose prend également en charge la livraison de données vers des destinations de point de terminaison HTTP dans toutes les régions AWS . Vous pouvez transmettre les données d'un flux Firehose d'une AWS région à un point de terminaison HTTP d'une autre AWS région. Vous pouvez également transmettre des données d'un flux Firehose vers une destination de point de terminaison HTTP située en dehors des AWS régions, par exemple vers votre propre serveur local en définissant l'URL du point de terminaison HTTP sur la destination souhaitée. Pour ces scénarios, des frais supplémentaires de transfert de données sont ajoutés à vos frais de diffusion. Pour plus d'informations, consultez la section [Transfert de données](#) sur la page « Tarification à la demande ».

Enregistrements dupliqués

Amazon Data Firehose utilise la at-least-once sémantique pour la diffusion des données. Dans certains cas, par exemple en cas d'expiration des délais de livraison des données, les nouvelles tentatives de livraison effectuées par Amazon Data Firehose peuvent entraîner des doublons si la demande de livraison de données d'origine est finalement acceptée. Cela s'applique à tous les types de destinations pris en charge par Amazon Data Firehose.

Suspendre et reprendre un stream Firehose

Une fois que vous avez configuré un flux Firehose, les données disponibles dans la source du flux sont continuellement transmises à la destination. Si la destination de votre flux est temporairement indisponible (par exemple, lors d'opérations de maintenance planifiées), vous pouvez interrompre temporairement la transmission des données et la reprendre lorsque la destination redevient disponible. Les sections suivantes montrent comment vous pouvez y parvenir :

⚠ Important

Lorsque vous utilisez l'approche décrite ci-dessous pour suspendre et reprendre un flux, après la reprise du flux, vous constaterez que peu d'enregistrements sont envoyés au compartiment d'erreur d'Amazon S3, tandis que le reste du flux continue d'être livré à destination. Il s'agit d'une limite connue de l'approche, qui se produit parce qu'un petit nombre d'enregistrements qui n'ont pas pu être livrés à destination auparavant après plusieurs tentatives sont considérés comme ayant échoué.

Comprendre comment Firehose gère les défaillances de livraison

Lorsque vous configurez un flux Firehose, pour de nombreuses destinations telles que OpenSearch Splunk et les points de terminaison HTTP, vous configurez également un compartiment S3 dans lequel les données qui ne sont pas livrées peuvent être sauvegardées. Pour plus d'informations sur la façon dont Firehose sauvegarde les données en cas d'échec de livraison, voir [Gestion des défaillances de livraison des données](#). Pour plus d'informations sur la façon d'accorder l'accès aux compartiments S3 dans lesquels les données qui ne sont pas livrées peuvent être sauvegardées, consultez [Accorder l'accès à Firehose à une destination Amazon S3](#). Lorsque Firehose (a) ne parvient pas à fournir les données à la destination du flux et (b) ne parvient pas à écrire les données dans le compartiment S3 de sauvegarde en cas d'échec des livraisons, il suspend effectivement la diffusion du flux jusqu'à ce que les données puissent être livrées à la destination ou écrites sur l'emplacement de sauvegarde S3.

Interruption d'un stream Firehose

Pour suspendre la diffusion de flux dans Firehose, supprimez d'abord les autorisations permettant à Firehose d'écrire sur l'emplacement de sauvegarde S3 en cas d'échec de diffusion. Par exemple, si vous souhaitez suspendre le stream Firehose avec une OpenSearch destination, vous pouvez le faire en mettant à jour les autorisations. Pour plus d'informations, voir [Accorder à Firehose l'accès à une destination de OpenSearch service public](#).

Supprimez l'autorisation "Effect": "Allow" pour l'action `s3:PutObject`, et ajoutez explicitement une instruction qui applique l'autorisation "Effect": "Deny" sur l'action `s3:PutObject` pour le compartiment S3 utilisé pour sauvegarder les diffusions qui ont échoué. Ensuite, désactivez la destination du flux (par exemple, désactivez le OpenSearch domaine de destination) ou supprimez les autorisations permettant à Firehose d'écrire sur la destination. Pour mettre à jour les autorisations pour d'autres destinations, consultez la section relative à votre

destination dans [Contrôler l'accès avec Amazon Data Firehose](#). Une fois ces deux actions effectuées, Firehose cessera de diffuser des streams, et vous pourrez surveiller cela à l'aide des [CloudWatch métriques de Firehose](#).

Important

Lorsque vous suspendez la diffusion d'un flux dans Firehose, vous devez vous assurer que la source du flux (par exemple, dans Kinesis Data Streams ou dans Managed Service for Kafka) est configurée pour conserver les données jusqu'à ce que la diffusion du flux reprenne et que les données soient livrées à la destination. Si la source est DirectPut, Firehose conservera les données pendant 24 heures. Une perte de données peut se produire si vous ne reprenez pas le flux et ne diffusez pas les données avant l'expiration de la période de conservation des données.

Reprise d'un stream Firehose

Pour reprendre la diffusion, annulez d'abord la modification apportée précédemment à la destination du flux en activant la destination et en vous assurant que Firehose dispose des autorisations nécessaires pour diffuser le flux vers la destination. Ensuite, annulez les modifications apportées précédemment aux autorisations appliquées au compartiment S3 pour la sauvegarde des diffusions ayant échoué. En d'autres termes, appliquez l'autorisation "Effect": "Allow" pour l'action `s3:PutObject`, et supprimez l'autorisation "Effect": "Deny" sur l'action `s3:PutObject` pour le compartiment S3 utilisé pour sauvegarder les diffusions qui ont échoué. Enfin, surveillez l'utilisation de [CloudWatch métriques pour Firehose](#) afin de confirmer que le flux est envoyé à destination. Pour consulter et résoudre les erreurs, utilisez le système de [surveillance Amazon CloudWatch Logs pour Firehose](#).

Surveillance d'Amazon Data Firehose

Vous pouvez surveiller Amazon Data Firehose à l'aide des fonctionnalités suivantes :

Rubriques

- [Bonnes pratiques avec les alarmes CloudWatch](#)
- [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#)
- [Accès aux CloudWatch métriques pour Amazon Data Firehose](#)
- [Surveillance d'Amazon Data Firehose à l'aide de journaux CloudWatch](#)
- [Accès aux CloudWatch journaux pour Amazon Data Firehose](#)
- [Surveillance de l'état de Kinesis Agent](#)
- [Journalisation des appels d'API Amazon Data Firehose avec AWS CloudTrail](#)

Bonnes pratiques avec les alarmes CloudWatch

Ajoutez des CloudWatch alarmes lorsque les mesures suivantes dépassent la limite de mise en mémoire tampon (15 minutes maximum) :

- `DeliveryToS3.DataFreshness`
- `DeliveryToSplunk.DataFreshness`
- `DeliveryToAmazonOpenSearchService.DataFreshness`
- `DeliveryToAmazonOpenSearchServerless.DataFreshness`
- `DeliveryToHttpEndpoint.DataFreshness`

Créez également des alarmes en fonction des expressions mathématiques de mesure suivantes.

- `IncomingBytes (Sum per 5 Minutes) / 300` approche un pourcentage de `BytesPerSecondLimit`.
- `IncomingRecords (Sum per 5 Minutes) / 300` approche un pourcentage de `RecordsPerSecondLimit`.
- `IncomingPutRequests (Sum per 5 Minutes) / 300` approche un pourcentage de `PutRequestsPerSecondLimit`.

Une autre métrique pour laquelle nous recommandons une alarme est `ThrottledRecords`.

Pour plus d'informations sur la résolution des problèmes lorsque les alarmes passent à l'état ALARM, consultez [Résolution des problèmes](#).

Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch

Important

Assurez-vous d'activer les alarmes sur toutes les CloudWatch métriques associées à votre destination afin d'identifier les erreurs en temps opportun.

Amazon Data Firehose s'intègre aux CloudWatch métriques Amazon afin que vous puissiez collecter, consulter et analyser les CloudWatch métriques de vos flux Firehose. Par exemple, vous pouvez surveiller les `IncomingRecords` indicateurs `IncomingBytes` et pour suivre les données ingérées dans Amazon Data Firehose par les producteurs de données.

Amazon Data Firehose collecte et publie des CloudWatch statistiques chaque minute. Toutefois, si les rafales de données entrantes ne se produisent que pendant quelques secondes, il se peut qu'elles ne soient pas entièrement capturées ou visibles dans les métriques d'une minute. Cela est dû au fait que CloudWatch les métriques sont agrégées à partir d'Amazon Data Firehose sur des intervalles d'une minute.

Les métriques collectées pour les streams Firehose sont gratuites. Pour en savoir plus sur les métriques relatives à l'agent Kinesis, consultez [Surveillance de l'état de Kinesis Agent](#).

Rubriques

- [Métriques de partitionnement CloudWatch dynamique](#)
- [CloudWatch Métriques de livraison de données](#)
- [Métriques d'ingestion de données](#)
- [Métriques au niveau de l'API CloudWatch](#)
- [CloudWatch Métriques de transformation des données](#)
- [CloudWatch Métriques de décompression des journaux](#)

- [CloudWatch Métriques de conversion de formats](#)
- [Métriques de chiffrement côté serveur \(SSE\) CloudWatch](#)
- [Dimensions pour Amazon Data Firehose](#)
- [Mesures d'utilisation d'Amazon Data Firehose](#)

Métriques de partitionnement CloudWatch dynamique

Si le [partitionnement dynamique](#) est activé, l'espace de noms AWS/Firehose inclut les métriques suivantes.

Métrique	Description
<code>ActivePartitionsLimit</code>	<p>Le nombre maximum de partitions actives qu'un stream Firehose traite avant d'envoyer des données au compartiment d'erreur.</p> <p>Unités : nombre</p>
<code>PartitionCount</code>	<p>Le nombre de partitions en cours de traitement, en d'autres termes, le nombre de partitions actives. Ce nombre varie entre un et la limite de 500 partitions (par défaut).</p> <p>Unités : nombre</p>
<code>PartitionCountExceeded</code>	<p>Cette métrique indique si vous dépassez la limite du nombre de partitions. Elle émet un ou zéro selon que la limite est dépassée ou non.</p>
<code>JQProcessing.Duration</code>	<p>Renvoie le temps nécessaire à l'exécution de l'expression JQ dans la fonction Lambda JQ.</p> <p>Unités : millisecondes</p>
<code>PerPartitionThroughput</code>	<p>Indique le débit traité par partition. Cette métrique vous permet de surveiller le débit par partition.</p> <p>Unités : StandardUnit. BytesSecond</p>

Métrique	Description
<code>DeliveryToS3.ObjectCount</code>	Indique le nombre d'objets qui sont livrés à votre compartiment S3. Unités : nombre

CloudWatch Métriques de livraison de données

L'espace de noms `AWS/Firehose` inclut les métriques au niveau des services suivantes. Si vous constatez de légères baisses de la moyenne pour `BackupToS3.Success`, `DeliveryToS3.Success`, `DeliveryToSplunk.Success`, `DeliveryToAmazonOpenSearchService.Success` ou `DeliveryToRedshift.Success`, cela n'indique pas qu'il y a une perte de données. Amazon Data Firehose essaie à nouveau de corriger les erreurs de livraison et n'avance pas tant que les enregistrements ne sont pas correctement livrés, que ce soit à la destination configurée ou au compartiment S3 de sauvegarde.

Rubriques

- [Livraison au OpenSearch service](#)
- [Livraison en mode OpenSearch Serverless](#)
- [Diffusion dans Amazon Redshift](#)
- [Diffusion dans Amazon S3](#)
- [Livraison à Snowflake](#)
- [Remise à Splunk](#)
- [Diffusion aux points de terminaison HTTP](#)

Livraison au OpenSearch service

Métrique	Description
<code>DeliveryToAmazonOpenSearchService.Bytes</code>	Le nombre d'octets indexés sur le OpenSearch service au cours de la période spécifiée. Unités : octets

Métrique	Description
<code>DeliveryToAmazonOpenSearchService.DataFreshness</code>	<p>L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tout dossier datant de plus de cet âge a été remis au OpenSearch Service.</p> <p>Unités : secondes</p>
<code>DeliveryToAmazonOpenSearchService.Records</code>	<p>Le nombre d'enregistrements indexés sur OpenSearch Service au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToAmazonOpenSearchService.Success</code>	<p>Somme des enregistrements indexés avec succès sur la somme des enregistrements qui ont été tentés.</p>
<code>DeliveryToS3.Bytes</code>	<p>Le nombre d'octets fournis à Amazon S3 au cours de la période spécifiée. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents.</p> <p>Unités : nombre</p>
<code>DeliveryToS3.DataFreshness</code>	<p>L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment S3. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents.</p> <p>Unités : secondes</p>
<code>DeliveryToS3.Records</code>	<p>Le nombre d'enregistrements fournis à Amazon S3 au cours de la période spécifiée. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents.</p> <p>Unités : nombre</p>

Métrique	Description
<code>DeliveryToS3.Success</code>	La somme des commandes put Amazon S3 réussies sur la somme de toutes les commandes put Amazon S3. Amazon Data Firehose émet toujours cette métrique, que la sauvegarde soit activée uniquement pour les documents défaillants ou pour tous les documents.
<code>DeliveryToAmazonOpenSearchService.AuthFailure</code>	Erreur d'authentification ou d'autorisation. Vérifiez la politique du cluster OS/ES et les autorisations de rôle. Zéro signifie qu'aucun problème n'est survenu. Un signifie un échec d'authentification.
<code>DeliveryToAmazonOpenSearchService.DeliveryRejected</code>	Erreur de diffusion rejetée. Vérifiez la politique du cluster OS/ES et les autorisations de rôle. Zéro indique qu'il n'y a aucun problème. Un indique qu'il y a un échec de diffusion.

Livraison en mode OpenSearch Serverless

Métrique	Description
<code>DeliveryToAmazonOpenSearchServerless.Bytes</code>	Nombre d'octets indexés sur OpenSearch Serverless au cours de la période spécifiée. Unités : octets
<code>DeliveryToAmazonOpenSearchServerless.DataFreshness</code>	L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tout enregistrement datant de plus de cet âge a été livré à OpenSearch Serverless. Unités : secondes

Métrique	Description
<code>DeliveryToAmazonOpenSearchServerless.Records</code>	<p>Nombre d'enregistrements indexés sur OpenSearch Serverless au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToAmazonOpenSearchServerless.Success</code>	<p>Somme des enregistrements indexés avec succès sur la somme des enregistrements qui ont été tentés.</p>
<code>DeliveryToS3.Bytes</code>	<p>Le nombre d'octets fournis à Amazon S3 au cours de la période spécifiée. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents.</p> <p>Unités : nombre</p>
<code>DeliveryToS3.DataFreshness</code>	<p>L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment S3. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents.</p> <p>Unités : secondes</p>
<code>DeliveryToS3.Records</code>	<p>Le nombre d'enregistrements fournis à Amazon S3 au cours de la période spécifiée. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents.</p> <p>Unités : nombre</p>
<code>DeliveryToS3.Success</code>	<p>La somme des commandes put Amazon S3 réussies sur la somme de toutes les commandes put Amazon S3. Amazon Data Firehose émet toujours cette métrique, que la sauvegarde soit activée uniquement pour les documents défaillants ou pour tous les documents.</p>

Métrique	Description
<code>DeliveryToAmazonOpenSearchServerless.AuthFailure</code>	<p>Erreur d'authentification ou d'autorisation. Vérifiez la politique du cluster OS/ES et les autorisations de rôle.</p> <p>Zéro signifie qu'aucun problème n'est survenu. Un signifie qu'il y a un échec d'authentification.</p>
<code>DeliveryToAmazonOpenSearchServerless.DeliveryRejected</code>	<p>Erreur de diffusion rejetée. Vérifiez la politique du cluster OS/ES et les autorisations de rôle.</p> <p>Zéro indique qu'il n'y a aucun problème. Un indique qu'il y a un échec de diffusion.</p>

Diffusion dans Amazon Redshift

Métrique	Description
<code>DeliveryToRedshift.Bytes</code>	<p>Le nombre d'octets copiés vers Amazon Redshift au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToRedshift.Records</code>	<p>Le nombre d'enregistrements copiés vers Amazon Redshift au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToRedshift.Success</code>	<p>La somme des commandes COPY Amazon Redshift réussies sur la somme de toutes les commandes COPY Amazon Redshift.</p>
<code>DeliveryToS3.Bytes</code>	<p>Le nombre d'octets fournis à Amazon S3 au cours de la période spécifiée.</p> <p>Unités : octets</p>
<code>DeliveryToS3.DataFreshness</code>	<p>L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans</p>

Métrique	Description
	<p>Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment S3.</p> <p>Unités : secondes</p>
<code>DeliveryToS3.Records</code>	<p>Le nombre d'enregistrements fournis à Amazon S3 au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToS3.Success</code>	<p>La somme des commandes put Amazon S3 réussies sur la somme de toutes les commandes put Amazon S3.</p>
<code>BackupToS3.Bytes</code>	<p>Le nombre d'octets fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque la sauvegarde sur Amazon S3 est activée.</p> <p>Unités : nombre</p>
<code>BackupToS3.DataFreshness</code>	<p>Âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment Amazon S3 pour être sauvegardés. Amazon Data Firehose émet cette métrique lorsque la sauvegarde sur Amazon S3 est activée.</p> <p>Unités : secondes</p>
<code>BackupToS3.Records</code>	<p>Le nombre d'enregistrements fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque la sauvegarde sur Amazon S3 est activée.</p> <p>Unités : nombre</p>

Métrique	Description
BackupToS3.Success	La somme des commandes put Amazon S3 réussies pour la sauvegarde sur la somme de toutes les commandes put Amazon S3. Amazon Data Firehose émet cette métrique lorsque la sauvegarde sur Amazon S3 est activée.

Diffusion dans Amazon S3

Les métriques du tableau suivant concernent la livraison vers Amazon S3 lorsque celui-ci est la destination principale du flux Firehose.

Métrique	Description
DeliveryToS3.Bytes	Le nombre d'octets fournis à Amazon S3 au cours de la période spécifiée. Unités : octets
DeliveryToS3.DataFreshness	L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment S3. Unités : secondes
DeliveryToS3.Records	Le nombre d'enregistrements fournis à Amazon S3 au cours de la période spécifiée. Unités : nombre
DeliveryToS3.Success	La somme des commandes put Amazon S3 réussies sur la somme de toutes les commandes put Amazon S3.
BackupToS3.Bytes	Le nombre d'octets fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque la sauvegarde

Métrique	Description
	<p>est activée (ce qui n'est possible que lorsque la transformation des données est également activée).</p> <p>Unités : nombre</p>
<code>BackupToS3.DataFreshness</code>	<p>Âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment Amazon S3 pour être sauvegardés. Amazon Data Firehose émet cette métrique lorsque la sauvegarde est activée (ce qui n'est possible que lorsque la transformation des données est également activée).</p> <p>Unités : secondes</p>
<code>BackupToS3.Records</code>	<p>Le nombre d'enregistrements fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque la sauvegarde est activée (ce qui n'est possible que lorsque la transformation des données est également activée).</p> <p>Unités : nombre</p>
<code>BackupToS3.Success</code>	<p>La somme des commandes put Amazon S3 réussies pour la sauvegarde sur la somme de toutes les commandes put Amazon S3. Amazon Data Firehose émet cette métrique lorsque la sauvegarde est activée (ce qui n'est possible que lorsque la transformation des données est également activée).</p>

Livraison à Snowflake

Métrique	Description
<code>DeliveryToSnowflake.Bytes</code>	<p>Le nombre d'octets fournis à Snowflake au cours de la période spécifiée.</p> <p>Unités : octets</p>
<code>DeliveryToSnowflake.DataFreshness</code>	<p>Âge (depuis le début de Firehose jusqu'à aujourd'hui) du plus vieux record de Firehose. Tout enregistrement datant de plus de cet âge a été livré à Snowflake. Notez que la validation des données dans Snowflake peut prendre quelques secondes une fois l'appel d'insertion Firehose réussi. Pour connaître le temps nécessaire à la validation des données dans Snowflake, reportez-vous à la <code>DeliveryToSnowflake.DataCommitLatency</code> métrique.</p> <p>Unités : secondes</p>
<code>DeliveryToSnowflake.DataCommitLatency</code>	<p>Le temps nécessaire pour que les données soient validées dans Snowflake une fois que Firehose a correctement inséré des enregistrements.</p> <p>Unités : secondes</p>
<code>DeliveryToSnowflake.Records</code>	<p>Le nombre d'enregistrements livrés à Snowflake au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToSnowflake.Success</code>	<p>La somme des appels d'insertion réussis effectués à Snowflake par rapport à la somme des appels d'insertion tentés.</p>
<code>DeliveryToS3.Bytes</code>	<p>Le nombre d'octets fournis à Amazon S3 au cours de la période spécifiée. Cette métrique n'est disponible que</p>

Métrique	Description
	<p>lorsque la livraison à Snowflake échoue et que Firehose tente de sauvegarder les données défectueuses sur S3.</p> <p>Unités : octets</p>
<code>DeliveryToS3.Records</code>	<p>Le nombre d'enregistrements fournis à Amazon S3 au cours de la période spécifiée. Cette métrique n'est disponible que lorsque la livraison à Snowflake échoue et que Firehose tente de sauvegarder les données défectueuses sur S3.</p> <p>Unités : nombre</p>
<code>DeliveryToS3.Success</code>	<p>La somme des commandes put Amazon S3 réussies sur la somme de toutes les commandes put Amazon S3. Cette métrique n'est disponible que lorsque la livraison à Snowflake échoue et que Firehose tente de sauvegarder les données défectueuses sur S3.</p>
<code>BackupToS3.DataFreshness</code>	<p>Âge (de Firehose à aujourd'hui) du plus ancien enregistrement de Firehose. Tout enregistrement antérieur à cet âge est sauvegardé dans le compartiment Amazon S3. Cette métrique est disponible lorsque le flux Firehose est configuré pour sauvegarder toutes les données.</p> <p>Unités : secondes</p>
<code>BackupToS3.Records</code>	<p>Le nombre d'enregistrements fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Cette métrique est disponible lorsque le flux Firehose est configuré pour sauvegarder toutes les données.</p> <p>Unités : nombre</p>

Métrique	Description
BackupToS3.Bytes	<p>Le nombre d'octets fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Cette métrique est disponible lorsque le flux Firehose est configuré pour sauvegarder toutes les données.</p> <p>Unités : nombre</p>
BackupToS3.Success	<p>La somme des commandes Amazon S3 put réussies pour la sauvegarde par rapport à la somme de toutes les commandes de sauvegarde Amazon S3. Firehose émet cette métrique lorsque le flux Firehose est configuré pour sauvegarder toutes les données.</p>

Remise à Splunk

Métrique	Description
DeliveryToSplunk.Bytes	<p>Nombre d'octets fournis à Splunk au cours de la période spécifiée.</p> <p>Unités : octets</p>
DeliveryToSplunk.DataAckLatency	<p>Durée approximative nécessaire pour recevoir un accusé de réception de Splunk une fois qu'Amazon Data Firehose lui a envoyé des données. L'augmentation ou la réduction de la tendance de cette métrique est plus utile que la valeur approximative absolue. L'augmentation de tendance peut indiquer une indexation et des taux d'accusé de réception des indexeurs Splunk plus lents.</p> <p>Unités : secondes</p>
DeliveryToSplunk.DataFreshness	<p>Âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis à Splunk.</p>

Métrique	Description
	Unités : secondes
<code>DeliveryToSplunk.Records</code>	Nombre d'enregistrements fournis à Splunk au cours de la période spécifiée. Unités : nombre
<code>DeliveryToSplunk.Success</code>	Somme des enregistrements indexés avec succès sur la somme des enregistrements qui ont été tentés.
<code>DeliveryToS3.Success</code>	La somme des commandes put Amazon S3 réussies sur la somme de toutes les commandes put Amazon S3. Cette métrique est émise si la sauvegarde vers Amazon S3 est activée.
<code>BackupToS3.Bytes</code>	Le nombre d'octets fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque le flux Firehose est configuré pour sauvegarder tous les documents. Unités : nombre
<code>BackupToS3.DataFreshness</code>	Âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment Amazon S3 pour être sauvegardés. Amazon Data Firehose émet cette métrique lorsque le flux Firehose est configuré pour sauvegarder tous les documents. Unités : secondes

Métrique	Description
BackupToS3.Records	<p>Le nombre d'enregistrements fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque le flux Firehose est configuré pour sauvegarder tous les documents.</p> <p>Unités : nombre</p>
BackupToS3.Success	<p>La somme des commandes put Amazon S3 réussies pour la sauvegarde sur la somme de toutes les commandes put Amazon S3. Amazon Data Firehose émet cette métrique lorsque le flux Firehose est configuré pour sauvegarder tous les documents.</p>

Diffusion aux points de terminaison HTTP

Métrique	Description
DeliveryToHttpEndpoint.Bytes	<p>Le nombre d'octets transmis avec succès au point de terminaison HTTP.</p> <p>Unités : octets</p>
DeliveryToHttpEndpoint.Records	<p>Le nombre d'enregistrements transmis avec succès au point de terminaison HTTP.</p> <p>Unités : nombres</p>
DeliveryToHttpEndpoint.DataFreshness	<p>Âge du plus ancien enregistrement dans Amazon Data Firehose.</p> <p>Unités : secondes</p>
DeliveryToHttpEndpoint.Success	<p>La somme de toutes les demandes de diffusion de données réussies au point de terminaison HTTP</p>

Métrique	Description
	Unités : nombre
<code>DeliveryToHttpEndpoint.ProcessedBytes</code>	Le nombre de tentatives de traitement d'octets, y compris les nouvelles tentatives.
<code>DeliveryToHttpEndpoint.ProcessedRecords</code>	Le nombre de tentatives d'enregistrement, y compris les nouvelles tentatives.

Métriques d'ingestion de données

Rubriques

- [Ingestion de données via Kinesis Data Streams](#)
- [Ingestion de données via Direct PUT](#)
- [Ingestion de données depuis MSK](#)

Ingestion de données via Kinesis Data Streams

Métrique	Description
<code>DataReadFromKinesisStream.Bytes</code>	<p>Lorsque la source de données est un flux de données Kinesis, cette métrique indique le nombre d'octets lus à partir de ce flux de données. Ce nombre inclut les relectures dues à des basculements.</p> <p>Unités : octets</p>
<code>DataReadFromKinesisStream.Records</code>	<p>Lorsque la source de données est un flux de données Kinesis, cette métrique indique le nombre d'enregistrements lus à partir de ce flux de données. Ce nombre inclut les relectures dues à des basculements.</p> <p>Unités : nombre</p>

Métrique	Description
ThrottledDescribeStream	<p>Nombre total de fois où l'opération DescribeStream est limitée lorsque la source de données est un flux de données Kinesis.</p> <p>Unités : nombre</p>
ThrottledGetRecords	<p>Nombre total de fois où l'opération GetRecords est limitée lorsque la source de données est un flux de données Kinesis.</p> <p>Unités : nombre</p>
ThrottledGetShardIterator	<p>Nombre total de fois où l'opération GetShardIterator est limitée lorsque la source de données est un flux de données Kinesis.</p> <p>Unités : nombre</p>

Ingestion de données via Direct PUT

Métrique	Description
BackupToS3.Bytes	<p>Le nombre d'octets fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque la transformation des données est activée pour les destinations Amazon S3 ou Amazon Redshift.</p> <p>Unités : octets</p>
BackupToS3.DataFreshness	<p>Âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment Amazon S3 pour être sauvegardés. Amazon Data Firehose émet cette métrique</p>

Métrique	Description
	<p>lorsque la transformation des données est activée pour les destinations Amazon S3 ou Amazon Redshift.</p> <p>Unités : secondes</p>
BackupToS3.Records	<p>Le nombre d'enregistrements fournis à Amazon S3 pour la sauvegarde au cours de la période spécifiée. Amazon Data Firehose émet cette métrique lorsque la transformation des données est activée pour les destinations Amazon S3 ou Amazon Redshift.</p> <p>Unités : nombre</p>
BackupToS3.Success	<p>La somme des commandes put Amazon S3 réussies pour la sauvegarde sur la somme de toutes les commandes put Amazon S3. Amazon Data Firehose émet cette métrique lorsque la transformation des données est activée pour les destinations Amazon S3 ou Amazon Redshift.</p>
BytesPerSecondLimit	<p>Le nombre maximum actuel d'octets par seconde qu'un flux Firehose peut ingérer avant d'être limité. Pour demander une augmentation de cette limite, accédez au Centre de support AWS et choisissez Créer un cas, puis choisissez Augmentation de la limite de service.</p>
DataReadFromKinesisStream.Bytes	<p>Lorsque la source de données est un flux de données Kinesis, cette métrique indique le nombre d'octets lus à partir de ce flux de données. Ce nombre inclut les relectures dues à des basculements.</p> <p>Unités : octets</p>

Métrique	Description
<code>DataReadFromKinesisStream.Records</code>	<p>Lorsque la source de données est un flux de données Kinesis, cette métrique indique le nombre d'enregistrements lus à partir de ce flux de données. Ce nombre inclut les relectures dues à des basculements.</p> <p>Unités : nombre</p>
<code>DeliveryToAmazonOpenSearchService.Bytes</code>	<p>Le nombre d'octets indexés sur le OpenSearch service au cours de la période spécifiée.</p> <p>Unités : octets</p>
<code>DeliveryToAmazonOpenSearchService.DataFreshness</code>	<p>L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tout dossier datant de plus de cet âge a été remis au OpenSearch Service.</p> <p>Unités : secondes</p>
<code>DeliveryToAmazonOpenSearchService.Records</code>	<p>Le nombre d'enregistrements indexés sur OpenSearch Service au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToAmazonOpenSearchService.Success</code>	<p>Somme des enregistrements indexés avec succès sur la somme des enregistrements qui ont été tentés.</p>
<code>DeliveryToRedshift.Bytes</code>	<p>Le nombre d'octets copiés vers Amazon Redshift au cours de la période spécifiée.</p> <p>Unités : octets</p>
<code>DeliveryToRedshift.Records</code>	<p>Le nombre d'enregistrements copiés vers Amazon Redshift au cours de la période spécifiée.</p> <p>Unités : nombre</p>

Métrique	Description
<code>DeliveryToRedshift.Success</code>	La somme des commandes COPY Amazon Redshift réussies sur la somme de toutes les commandes COPY Amazon Redshift.
<code>DeliveryToS3.Bytes</code>	Le nombre d'octets fournis à Amazon S3 au cours de la période spécifiée. Unités : octets
<code>DeliveryToS3.DataFreshness</code>	L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis au compartiment S3. Unités : secondes
<code>DeliveryToS3.Records</code>	Le nombre d'enregistrements fournis à Amazon S3 au cours de la période spécifiée. Unités : nombre
<code>DeliveryToS3.Success</code>	La somme des commandes put Amazon S3 réussies sur la somme de toutes les commandes put Amazon S3.
<code>DeliveryToSplunk.Bytes</code>	Nombre d'octets fournis à Splunk au cours de la période spécifiée. Unités : octets
<code>DeliveryToSplunk.DataAckLatency</code>	Durée approximative nécessaire pour recevoir un accusé de réception de Splunk une fois qu'Amazon Data Firehose lui a envoyé des données. L'augmentation ou la réduction de la tendance de cette métrique est plus utile que la valeur approximative absolue. L'augmentation de tendance peut indiquer une indexation et des taux d'accusé de réception des indexeurs Splunk plus lents. Unités : secondes

Métrique	Description
<code>DeliveryToSplunk.DataFreshness</code>	<p>Âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans Amazon Data Firehose. Tous les enregistrements plus anciens ayant été remis à Splunk.</p> <p>Unités : secondes</p>
<code>DeliveryToSplunk.Records</code>	<p>Nombre d'enregistrements fournis à Splunk au cours de la période spécifiée.</p> <p>Unités : nombre</p>
<code>DeliveryToSplunk.Success</code>	<p>Somme des enregistrements indexés avec succès sur la somme des enregistrements qui ont été tentés.</p>
<code>IncomingBytes</code>	<p>Le nombre d'octets ingérés avec succès dans le flux Firehose au cours de la période spécifiée. L'ingestion de données peut être limitée lorsqu'elle dépasse l'une des limites de flux de Firehose. Les données limitées ne seront pas prises en compte pour <code>IncomingBytes</code>.</p> <p>Unités : octets</p>
<code>IncomingPutRequests</code>	<p>Le nombre de <code>PutRecordBatch</code> demandes <code>PutRecord</code> et de demandes réussies sur une période donnée.</p> <p>Unités : nombre</p>
<code>IncomingRecords</code>	<p>Le nombre d'enregistrements correctement ingérés dans le flux Firehose au cours de la période spécifiée. L'ingestion de données peut être limitée lorsqu'elle dépasse l'une des limites de flux de Firehose. Les données limitées ne seront pas prises en compte pour <code>IncomingRecords</code>.</p> <p>Unités : nombre</p>

Métrique	Description
<code>KinesisMillisBehindLatest</code>	<p>Lorsque la source de données est un flux de données Kinesis, cette métrique indique le nombre de millisecondes séparant le dernier enregistrement lu de l'enregistrement le plus récent dans le flux de données Kinesis.</p> <p>Unités : millisecondes</p>
<code>RecordsPerSecondLimit</code>	<p>Le nombre maximum actuel d'enregistrements par seconde qu'un stream Firehose peut ingérer avant d'être limité.</p> <p>Unités : nombre</p>
<code>ThrottledRecords</code>	<p>Le nombre d'enregistrements qui ont été limités parce que l'ingestion de données a dépassé l'une des limites du flux Firehose.</p> <p>Unités : nombre</p>

Ingestion de données depuis MSK

Métrique	Description
<code>DataReadFromSource.Records</code>	<p>Le nombre d'enregistrements lus à partir de la source Rubrique Kafka.</p> <p>Unités : nombre</p>
<code>DataReadFromSource.Bytes</code>	<p>Le nombre d'octets lus à partir de la source Rubrique Kafka.</p> <p>Unités : octets</p>
<code>SourceThrottled.Delay</code>	<p>La durée pendant laquelle le cluster Kafka source est retardé pour renvoyer les enregistrements depuis la source Rubrique Kafka.</p>

Métrique	Description
	Unités : millisecondes
BytesPerSecondLimit	Limite actuelle du débit auquel Firehose va lire chaque partition de la source Rubrique Kafka. Unités : octets/seconde
KafkaOffsetLag	Différence entre le plus grand décalage de l'enregistrement lu par Firehose à partir de la source Rubrique Kafka et le plus grand décalage de l'enregistrement disponible à partir de la source Rubrique Kafka. Unités : nombre
FailedValidation.Records	Le nombre d'enregistrements dont la validation a échoué. Unités : nombre
FailedValidation.Bytes	Le nombre d'octets dont la validation a échoué. Unités : octets
DataReadFromSource .Backpressured	Indique qu'un flux Firehose est retardé dans la lecture des enregistrements de la partition source, soit parce que le nombre de partitions BytesPerSecondLimit par partition est dépassé, soit parce que le flux normal de diffusion est lent ou s'est arrêté Unités : booléennes

Métriques au niveau de l'API CloudWatch

L'espace de noms `AWS/Firehose` inclut les métriques suivantes au niveau des API.

Métrique	Description
DescribeDeliveryStream.Latency	Délai nécessaire par opération <code>DescribeDeliveryStream</code> , mesurée pendant la période spécifiée.

Métrique	Description
	Unités : millisecondes
DescribeDeliveryStream.Requests	Nombre total de demandes DescribeDeliveryStream . Unités : nombre
ListDeliveryStreams.Latency	Délai nécessaire par opération ListDeliveryStreams , mesurée pendant la période spécifiée. Unités : millisecondes
ListDeliveryStreams.Requests	Nombre total de demandes ListFirehose . Unités : nombre
PutRecord.Bytes	Le nombre d'octets utilisés par le flux Firehose PutRecord au cours de la période spécifiée. Unités : octets
PutRecord.Latency	Délai nécessaire par opération PutRecord , mesurée pendant la période spécifiée. Unités : millisecondes
PutRecord.Requests	Nombre total de demandes PutRecord , correspondant au nombre total d'enregistrements issus des opérations PutRecord . Unités : nombre
PutRecordBatch.Bytes	Le nombre d'octets utilisés par le flux Firehose PutRecordBatch au cours de la période spécifiée. Unités : octets

Métrique	Description
<code>PutRecordBatch.Latency</code>	Délai nécessaire par opération <code>PutRecordBatch</code> , mesurée pendant la période spécifiée. Unités : millisecondes
<code>PutRecordBatch.Records</code>	Nombre total d'enregistrements issus des opérations <code>PutRecordBatch</code> . Unités : nombre
<code>PutRecordBatch.Requests</code>	Nombre total de demandes <code>PutRecordBatch</code> . Unités : nombre
<code>PutRequestsPerSecondLimit</code>	Le nombre maximum de requêtes put par seconde qu'un stream Firehose peut traiter avant la limitation. Ce nombre inclut les <code>PutRecordBatch</code> demandes <code>PutRecord</code> et les demandes. Unités : nombre
<code>ThrottledDescribeStream</code>	Nombre total de fois où l'opération <code>DescribeStream</code> est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
<code>ThrottledGetRecords</code>	Nombre total de fois où l'opération <code>GetRecords</code> est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
<code>ThrottledGetShardIterator</code>	Nombre total de fois où l'opération <code>GetShardIterator</code> est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre

Métrique	Description
UpdateDeliveryStream.Latency	Délai nécessaire par opération UpdateDeliveryStream, mesurée pendant la période spécifiée. Unités : millisecondes
UpdateDeliveryStream.Requests	Nombre total de demandes UpdateDeliveryStream. Unités : nombre

CloudWatch Métriques de transformation des données

Si la transformation des données avec Lambda est activée, l'espace de noms AWS/Firehose inclut les métriques suivantes.

Métrique	Description
ExecuteProcessingDuration	Le temps nécessaire pour chaque appel de fonction Lambda effectué par Firehose. Unités : millisecondes
ExecuteProcessingSuccess	La somme des invocations de fonction Lambda réussies sur la somme du total des invocations de fonction Lambda.
SucceedProcessingRecords	Le nombre d'enregistrements traités avec succès au cours de la période spécifiée. Unités : nombre
SucceedProcessingBytes	Le nombre d'octets traités avec succès au cours de la période spécifiée. Unités : octets

CloudWatch Métriques de décompression des journaux

Si la décompression est activée pour la livraison CloudWatch des journaux, l'espace de noms AWS/Firehose inclut les métriques suivantes.

Métrique	Description
OutputDecompressedBytes.Success	Données décompressées réussies en octets Unités : octets
OutputDecompressedBytes.Failed	Échec de la décompression des données en octets Unités : octets
OutputDecompressedRecords.Success	Nombre d'enregistrements décompressés réussis Unités : nombre
OutputDecompressedRecords.Failed	Nombre d'enregistrements décompressés ayant échoué Unités : nombre

CloudWatch Métriques de conversion de formats

Si la conversion de format est activée, l'espace de noms AWS/Firehose inclut les métriques suivantes.

Métrique	Description
SucceedConversion.Records	Nombre d'enregistrements convertis avec succès. Unités : nombre
SucceedConversion.Bytes	Taille des enregistrements convertis avec succès. Unités : octets

Métrique	Description
FailedConversion.Records	Nombre d'enregistrements qui n'ont pas pu être convertis. Unités : nombre
FailedConversion.Bytes	Taille des enregistrements qui n'ont pas pu être convertis. Unités : octets

Métriques de chiffrement côté serveur (SSE) CloudWatch

L'espace de noms `AWS/Firehose` inclut les métriques suivantes liées à SSE.

Métrique	Description
KMSKeyAccessDenied	Le nombre de fois que le service rencontre un <code>KMSAccessDeniedException</code> pour le stream Firehose. Unités : nombre
KMSKeyDisabled	Le nombre de fois que le service rencontre un <code>KMSDisabledException</code> pour le stream Firehose. Unités : nombre
KMSKeyInvalidState	Le nombre de fois que le service rencontre un <code>KMSInvalidStateException</code> pour le stream Firehose. Unités : nombre
KMSKeyNotFound	Le nombre de fois que le service rencontre un <code>KMSNotFoundException</code> pour le stream Firehose. Unités : nombre

Dimensions pour Amazon Data Firehose

Pour filtrer les métriques par flux Firehose, utilisez la `DeliveryStreamName` dimension.

Mesures d'utilisation d'Amazon Data Firehose

Vous pouvez utiliser les statistiques CloudWatch d'utilisation pour obtenir une visibilité sur l'utilisation des ressources par votre compte. Utilisez ces indicateurs pour visualiser l'utilisation actuelle de vos services sur CloudWatch des graphiques et des tableaux de bord.

Les métriques d'utilisation des quotas de service se trouvent dans l'espace de noms `AWS/Usage` et sont collectées toutes les minutes.

Actuellement, le seul nom de métrique publié dans cet espace de noms CloudWatch est `ResourceCount`. Cette métrique est publiée avec les dimensions `Service`, `Class`, `Type` et `Resource`.

Métrique	Description
<code>ResourceCount</code>	<p>Nombre des ressources spécifiées exécutées dans votre compte. Les ressources sont définies par les dimensions associées à la métrique.</p> <p>La statistique la plus utile pour cette métrique est <code>MAXIMUM</code>, qui représente le nombre maximal de ressources utilisées pendant la période d'une minute.</p>

Les dimensions suivantes sont utilisées pour affiner les statistiques d'utilisation publiées par Amazon Data Firehose.

Dimension	Description
<code>Service</code>	Nom du AWS service contenant la ressource. Pour les métriques d'utilisation d'Amazon Data Firehose, la valeur de cette dimension est <code>Firehose</code>

Dimension	Description
Class	Classe de ressource suivie. Les métriques d'utilisation de l'API Amazon Data Firehose utilisent cette dimension avec une valeur de. None
Type	Type de ressource suivi. Actuellement, lorsque la dimension Service est Firehose, la seule valeur valide pour Type est Resource.
Resource	Le nom de la AWS ressource. Actuellement, lorsque la dimension Service est Firehose, la seule valeur valide pour Resource est DeliveryStreams .

Accès aux CloudWatch métriques pour Amazon Data Firehose

Vous pouvez surveiller les métriques d'Amazon Data Firehose à l'aide de la CloudWatch console, de la ligne de commande ou CloudWatch de l'API. Les procédures suivantes vous montrent comment accéder aux métriques à l'aide de ces différentes méthodes.

Pour accéder aux métriques à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).
2. Sélectionnez une région dans la barre de navigation.
3. Dans le panneau de navigation, sélectionnez Métriques.
4. Choisissez l'espace de nom Firehose.
5. Choisissez Firehose Stream Metrics ou Firehose Metrics.
6. Sélectionnez une métrique à ajouter au graphique.

Pour accéder aux métriques à l'aide du AWS CLI

Utilisez les [métriques et get-metric-statistics](#) les commandes de liste.

```
aws cloudwatch list-metrics --namespace "AWS/Firehose"
```

```
aws cloudwatch get-metric-statistics --namespace "AWS/Firehose" \
```

```
--metric-name DescribeDeliveryStream.Latency --statistics Average --period 3600 \  
--start-time 2017-06-01T00:00:00Z --end-time 2017-06-30T00:00:00Z
```

Surveillance d'Amazon Data Firehose à l'aide de journaux CloudWatch

Amazon Data Firehose s'intègre à Amazon CloudWatch Logs afin que vous puissiez consulter les journaux d'erreurs spécifiques en cas d'échec de l'appel Lambda pour la transformation ou la livraison des données. Vous pouvez activer la journalisation des erreurs Amazon Data Firehose lorsque vous créez votre stream Firehose.

Si vous activez la journalisation des erreurs Amazon Data Firehose dans la console Amazon Data Firehose, un groupe de journaux et les flux de journaux correspondants sont créés pour le flux Firehose en votre nom. Le format du nom du groupe de logs est `/aws/kinesisfirehose/delivery-stream-name` le suivant : où *delivery-stream-name* est le nom du flux Firehose correspondant. `DestinationDelivery` est le flux de journal créé et utilisé pour consigner les erreurs liées à la livraison vers la destination principale. Un autre flux de journal appelé `BackupDelivery` est créé uniquement si la sauvegarde S3 est activée pour la destination. Le flux de journal `BackupDelivery` est utilisé pour consigner toutes les erreurs liées à la diffusion vers la sauvegarde S3.

Par exemple, si vous créez un flux Firehose « MyStream » avec Amazon Redshift comme destination et que vous activez la journalisation des erreurs Amazon Data Firehose, les éléments suivants sont créés en votre nom : un groupe de journaux nommé `aws/kinesisfirehose/MyStream` et deux flux de journaux `DestinationDelivery` et `BackupDelivery`. Dans cet exemple, `DestinationDelivery` sera utilisé pour enregistrer toutes les erreurs liées à la diffusion vers la destination Amazon Redshift ainsi que vers la destination intermédiaire S3. `BackupDelivery`, dans le cas où la sauvegarde S3 est activée, sera utilisé pour enregistrer toute erreur liée à la diffusion vers le compartiment de sauvegarde S3.

Vous pouvez activer la journalisation des erreurs Amazon Data Firehose via l' AWS CLI API ou à l' AWS CloudFormation aide de la `CloudWatchLoggingOptions` configuration. Pour ce faire, créez un groupe de journaux et un flux de journaux à l'avance. Nous vous recommandons de réserver ce groupe de journaux et ce flux de journaux exclusivement à la journalisation des erreurs d'Amazon Data Firehose. Assurez-vous également que la politique IAM associée dispose de l'autorisation `"logs:putLogEvents"`. Pour plus d'informations, consultez [Contrôler l'accès avec Amazon Data Firehose](#).

Notez qu'Amazon Data Firehose ne garantit pas que tous les journaux d'erreurs de livraison soient envoyés à CloudWatch Logs. Dans les cas où le taux d'échec de livraison est élevé, Amazon Data Firehose échantillonne les journaux d'erreurs de livraison avant de les envoyer à CloudWatch Logs.

Des frais nominaux sont facturés pour les journaux d'erreurs envoyés à CloudWatch Logs. Pour plus d'informations, consultez [Amazon CloudWatch Pricing](#).

Table des matières

- [Erreurs de diffusion de données](#)

Erreurs de diffusion de données

Vous trouverez ci-dessous une liste des codes d'erreur et des messages relatifs à la livraison des données pour chaque destination Amazon Data Firehose. Chaque message d'erreur décrit également l'action appropriée à entreprendre afin de résoudre le problème.

Erreurs

- [Erreurs de diffusion de données Amazon S3](#)
- [Erreurs de diffusion de données Amazon Redshift](#)
- [Erreurs de livraison de données Snowflake](#)
- [Erreurs de livraison de données Splunk](#)
- [ElasticSearch Erreurs de livraison de données](#)
- [Erreurs de diffusion des données du point de terminaison HTTPS](#)
- [Erreurs de livraison des données Amazon OpenSearch Service](#)
- [Erreurs d'invocation Lambda](#)
- [Erreurs d'invocation Kinesis](#)
- [Erreurs d'invocation Kinesis DirectPut](#)
- [AWS Glue Erreurs d'invocation](#)
- [DataFormatConversion Erreurs d'invocation](#)

Erreurs de diffusion de données Amazon S3

Amazon Data Firehose peut envoyer les erreurs suivantes liées à Amazon S3 à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
<code>S3.KMS.NotFoundException</code>	« La AWS KMS clé fournie n'a pas été trouvée. Si vous utilisez ce que vous pensez être une AWS KMS clé valide avec le bon rôle, vérifiez s'il y a un problème avec le compte auquel la AWS KMS clé est associée. »
<code>S3.KMS.RequestLimitExceeded</code>	« La limite par seconde appliquée aux demandes KMS a été dépassée lors de la tentative de chiffrement d'objets S3. Augmentez-la. » Pour plus d'informations, consultez la section Limites dans le Guide du développeur AWS Key Management Service .
<code>S3.AccessDenied</code>	« Accès refusé. Assurez-vous que la politique de confiance associée au rôle IAM fourni autorise Amazon Data Firehose à assumer le rôle et que la politique d'accès autorise l'accès au compartiment S3. »
<code>S3.AccountProblem</code>	« Un problème lié à votre AWS compte empêche le bon déroulement de l'opération. Contacter AWS Support. »
<code>S3.AllAccessDisabled</code>	« L'accès au compte fourni a été désactivé. Contactez AWS le Support. »
<code>S3.InvalidPayer</code>	« L'accès au compte fourni a été désactivé. Contactez AWS le Support. »
<code>S3.NotSignedUp</code>	« Le compte n'est pas inscrit à Amazon S3. S'inscrire au compte ou utiliser un autre compte. »
<code>S3.NoSuchBucket</code>	« Le compartiment spécifié n'existe pas. Créez le compartiment ou utilisez un autre compartiment existant. »
<code>S3.MethodNotAllowed</code>	« La méthode spécifiée n'est pas autorisée sur cette ressource. Modifiez la politique du compartiment pour autoriser les opérations Amazon S3 correctes. »
<code>InternalServerError</code>	« Une erreur interne s'est produite lors de la tentative de diffusion des données. La livraison sera réessayée ; si l'erreur persiste, elle sera signalée AWS pour résolution. »

Code d'erreur	Message et informations sur les erreurs
S3.KMS.KeyDisabled	« La clé KMS fournie est désactivée. Activez la clé ou utilisez-en une autre. »
S3.KMS.InvalidStateException	« La clé KMS fournie est dans un état non valide. Veuillez utiliser une autre clé. »
KMS.InvalidStateException	« La clé KMS fournie est dans un état non valide. Veuillez utiliser une autre clé. »
KMS.DisabledException	« La clé KMS fournie est désactivée. Corrigez la clé ou utilisez-en une autre. »
S3.SlowDown	« Le taux de demandes put pour le compartiment spécifié était trop élevé. Augmentez la taille de la mémoire tampon des flux Firehose ou réduisez les demandes de vente provenant d'autres applications. »
S3.SubscriptionRequired	« L'accès a été refusé lors de l'appel de S3. Assurez-vous que le rôle IAM et la clé KMS (si fournis) transmis sont associés à un abonnement Amazon S3. »
S3.InvalidToken	« Le jeton fourni est mal formé ou n'est pas valide. Veuillez vérifier les informations d'identification fournies. »
S3.KMS.KeyNotConfigured	« La clé KMS n'est pas configurée. Configurez votre MasterKey ID KMS ou désactivez le chiffrement de votre compartiment S3. »
S3.KMS.AsymmetricCMKNotSupported	« Amazon S3 prend uniquement en charge les clés CMK symétriques. Vous ne pouvez pas utiliser une clé CMK asymétrique pour chiffrer vos données dans Amazon S3. Pour obtenir le type de votre clé CMK, utilisez l' DescribeKey opération KMS. »

Code d'erreur	Message et informations sur les erreurs
<code>S3.IllegalLocationConstraintException</code>	« Firehose utilise actuellement le point de terminaison mondial s3 pour la diffusion des données au compartiment s3 configuré. La région du compartiment s3 configuré ne prend pas en charge le point de terminaison mondial s3. Créez un stream Firehose dans la même région que le compartiment s3 ou utilisez le compartiment s3 dans la région qui prend en charge le point de terminaison global. »
<code>S3.InvalidPrefixConfigurationException</code>	« Le préfixe s3 personnalisé utilisé pour l'évaluation de l'horodatage n'est pas valide. Vérifiez que votre préfixe s3 contient des expressions valides pour la date et l'heure actuelles de l'année. »
<code>DataFormatConversion.MalformedData</code>	« Un caractère illégal a été trouvé entre les jetons. »

Erreurs de diffusion de données Amazon Redshift

Amazon Data Firehose peut envoyer les erreurs suivantes liées à Amazon Redshift à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
<code>Redshift.TableNotFound</code>	<p>« La table dans laquelle charger les données est introuvable. Vérifiez que la table spécifiée existe. »</p> <p>La table de destination d'Amazon Redshift dans laquelle les données doivent être copiées à partir de S3 n'a pas été trouvée. Notez qu'Amazon Data Firehose ne crée pas la table Amazon Redshift si elle n'existe pas.</p>
<code>Redshift.SyntaxError</code>	« La commande COPY contient une erreur de syntaxe. Relancez la commande. »

Code d'erreur	Message et informations sur les erreurs
Redshift. AuthenticationFailed	« Le nom d'utilisateur et le mot de passe d'authentification fournis ont échoué. Fournissez un nom d'utilisateur et un mot de passe valides. »
Redshift. AccessDenied	« Accès refusé. Assurez-vous que la politique de confiance associée au rôle IAM fourni autorise Amazon Data Firehose à assumer ce rôle. »
Redshift. S3BucketAccessDenied	« La commande COPY n'a pas pu accéder au compartiment S3. Vérifiez que la stratégie d'accès pour le rôle IAM fourni permet d'accéder au compartiment S3. »
Redshift. DataLoadFailed	« Échec de chargement des données dans la table. Vérifiez le contenu de la table système STL_LOAD_ERRORS pour plus de détails. »
Redshift. ColumnNotFound	« Une colonne de la commande COPY n'existe pas dans le tableau. Spécifiez un nom de colonne valide. »
Redshift. DatabaseNotFound	« La base de données spécifiée dans la configuration de destination Amazon Redshift ou l'URL JDBC est introuvable. Spécifiez un nom de base de données valide. »
Redshift. IncorrectCopyOptions	<p>« Des options de la commande COPY redondantes ou contradictoires ont été fournies. Certaines options ne sont pas compatibles dans certaines combinaisons. Vérifiez la référence de la commande COPY pour plus d'informations. »</p> <p>Pour en savoir plus, consultez Commande COPY Amazon Redshift dans le Guide du développeur de base de données Amazon Redshift.</p>
Redshift. MissingColumn	« Il y a une colonne définie dans le schéma de table comme NOT NULL sans valeur DEFAULT et pas incluse dans la liste de colonnes. Vous devez exclure cette colonne, vous assurer que les données chargées offrent toujours une valeur pour cette colonne ou ajouter une valeur par défaut pour le schéma Amazon Redshift de cette table ».

Code d'erreur	Message et informations sur les erreurs
Redshift. ConnectionFailed	« Échec de connexion au cluster Amazon Redshift spécifié. Assurez-vous que les paramètres de sécurité autorisent les connexions Amazon Data Firehose, que le cluster ou la base de données spécifié dans la configuration de destination Amazon Redshift ou dans l'URL JDBC est correct et que le cluster est disponible. »
Redshift. ColumnMismatch	« Le nombre de jsonpaths dans la commande COPY et le nombre de colonnes dans la table de destination doivent correspondre. Relancez la commande. »
Redshift. IncorrectOrMissingRegion	« Amazon Redshift a tenté d'utiliser le point de terminaison de région incorrect pour accéder au compartiment S3. Spécifiez une valeur de région correcte dans les options de commande COPY ou assurez-vous que le compartiment S3 est dans la même région que la base de données Amazon Redshift. »
Redshift. IncorrectJsonPathsFile	« Le fichier jsonpaths fourni n'est pas dans un format JSON pris en charge. Relancez la commande. »
Redshift. MissingS3File	« Un ou plusieurs fichiers S3 requis par Amazon Redshift ont été supprimés du compartiment S3. Vérifiez les stratégies de compartiment S3 pour retirer toute suppression automatique des fichiers S3. »
Redshift. InsufficientPrivilege	« L'utilisateur ne dispose pas des autorisations pour charger les données dans la table. Vérifiez les autorisations d'utilisateur Amazon Redshift pour le privilège INSERT. »
Redshift. ReadOnlyCluster	« La requête ne peut pas être exécutée, car le système est en mode de redimensionnement. Réessayez la requête plus tard. »
Redshift. DiskFull	« Les données n'ont pas pu être chargées, car le disque est plein. Augmentez la capacité du cluster Amazon Redshift ou supprimez des données inutilisées pour libérer de l'espace sur le disque. »

Code d'erreur	Message et informations sur les erreurs
<code>InternalError</code>	« Une erreur interne s'est produite lors de la tentative de diffusion des données. La livraison sera réessayée ; si l'erreur persiste, elle sera signalée AWS pour résolution. »
<code>Redshift. ArgumentN otSupported</code>	« La commande COPY contient des options non prises en charge. »
<code>Redshift. AnalyzeTa bleAccess Denied</code>	« Accès refusé. La copie de S3 vers Redshift échoue, car l'analyse de la table ne peut être effectuée que par le propriétaire de la table ou de la base de données. »
<code>Redshift. SchemaNot Found</code>	« Le schéma spécifié dans la configuration <code>DataTableName</code> de destination d'Amazon Redshift est introuvable. Spécifiez un nom de schéma valide. »
<code>Redshift. ColumnSpe cifiedMor eThanOnce</code>	« Une colonne est spécifiée plusieurs fois dans la liste des colonnes. Assurez-vous que les colonnes dupliquées sont supprimées. »
<code>Redshift. ColumnNot NullWitho utDefault</code>	« Il existe une colonne non nulle sans <code>DEFAULT</code> qui n'est pas incluse dans la liste des colonnes. Assurez-vous que ces colonnes sont incluses dans la liste des colonnes. »
<code>Redshift. Incorrect BucketRegion</code>	« Redshift a tenté d'utiliser un compartiment dans une région autre que celle dans laquelle se trouve le cluster. Veuillez spécifier un compartiment dans la même région que le cluster. »
<code>Redshift. S3SlowDown</code>	« Taux de requêtes élevé vers S3. Réduisez le taux pour éviter d'être limité. »

Code d'erreur	Message et informations sur les erreurs
Redshift. InvalidCopyOptionForJson	« Veuillez utiliser soit auto soit un chemin S3 valide pour json copyOption. »
Redshift. InvalidCopyOptionJSONPathFormat	« COPY a échoué avec une erreur \« Format JSONPath non valide. L'index du tableau est hors de portée \». Veuillez rectifier l'expression JSONPath. »
Redshift. InvalidCopyOptionRBACAc1NotAllowed	« COPY a échoué avec l'erreur \« Impossible d'utiliser le cadre RBAC acl tant que la propagation des autorisations n'est pas activée. \»
Redshift. DiskSpaceQuotaExceeded	« Transaction abandonnée en raison d'un dépassement du quota d'espace disque. Libérez de l'espace disque ou demandez une augmentation du quota pour le ou les schémas. »
Redshift. ConnectionsLimitExceeded	« Limite de connexion dépassée pour l'utilisateur. »
Redshift. SslNotSupported	« La connexion au cluster Amazon Redshift spécifié a échoué, car le serveur ne prend pas en charge le protocole SSL. Vérifiez les paramètres de votre cluster. »
Redshift. HoseNotFound	« Le tuyau a été supprimé. Veuillez vérifier le statut de votre tuyau. »
Redshift. Delimiter	« Le délimiteur copyOptions dans la copyCommand n'est pas valide. Assurez-vous qu'il s'agit d'un seul caractère. »

Code d'erreur	Message et informations sur les erreurs
Redshift. QueryCancelled	« L'utilisateur a annulé l'opération COPY. »
Redshift. CompressionMismatch	« Le tuyau est configuré avec UNCOMPRESSED, mais copyOption inclut un format de compression. »
Redshift. EncryptionCredentials	« L'option ENCRYPTED nécessite des informations d'identification au format : 'aws_iam_role=...;master_symmetric_key=...' or 'aws_access_key_id=...;aws_secret_access_key=...[;token=...];master_symmetric_key=...' »
Redshift. InvalidCopyOptions	« Options de configuration COPY non valides. »
Redshift. InvalidMessageFormat	« La commande de copie contient un caractère non valide. »
Redshift. TransactionIdLimitReached	« La limite d'identifiants de transaction est atteinte. »
Redshift. DestinationRemoved	« Vérifiez que la destination Redshift existe et qu'elle est correctement configurée dans la configuration de Firehose. »
Redshift. OutOfMemory	« Le cluster Redshift est à court de mémoire. Assurez-vous que le cluster dispose d'une capacité suffisante. »
Redshift. Cannot Fork Process	« Le cluster Redshift est à court de mémoire. Assurez-vous que le cluster dispose d'une capacité suffisante. »

Code d'erreur	Message et informations sur les erreurs
Redshift. SslFailure	« La connexion SSL s'est fermée lors de la négociation. »
Redshift.Resize	« Le cluster Redshift se redimensionne. Firehose ne sera pas en mesure de fournir des données pendant le redimensionnement du cluster. »
Redshift. ImproperQ ualifiedName	« Le nom qualifié est incorrect (trop de noms séparés par des points). »
Redshift. InvalidJs onPathFormat	« Format JSONPath non valide. »
Redshift. TooManyCo nnections Exception	« Trop de connexions à Redshift. »
Redshift. PSQLError	« PS QLError observé depuis Redshift. »
Redshift. Duplicate SecondsSp ecification	« Spécification des secondes dupliquées au format date/heure. »
Redshift. RelationC ouldNotBe Opened	« Erreur Redshift détectée, la relation n'a pas pu être ouverte. Vérifiez les journaux Redshift pour la base de données spécifiée. »
Redshift. TooManyClients	« Le système a rencontré trop d'exceptions de la part des clients de Redshift. Réexaminez les connexions maximales à la base de données si plusieurs producteurs y écrivent simultanément. »

Erreurs de livraison de données Snowflake

Firehose peut envoyer les erreurs suivantes liées à Snowflake à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
Snowflake .InvalidUrl	« Firehose ne parvient pas à se connecter à Snowflake. Assurez-vous que l'URL du compte est correctement spécifiée dans la configuration de destination de Snowflake. »
Snowflake .InvalidUser	« Firehose ne parvient pas à se connecter à Snowflake. Assurez-vous que l'utilisateur est correctement spécifié dans la configuration de destination de Snowflake. »
Snowflake .InvalidRole	« Le rôle Snowflake spécifié n'existe pas ou n'est pas autorisé. Assurez-vous que le rôle est accordé à l'utilisateur spécifié. »
Snowflake .InvalidTable	« La table fournie n'existe pas ou n'est pas autorisée »
Snowflake .InvalidSchema	« Le schéma fourni n'existe pas ou n'est pas autorisé »
Snowflake .InvalidDatabase	« La base de données fournie n'existe pas ou n'est pas autorisée »
Snowflake .InvalidPrivateKeyOrPassphrase	« La clé privée ou le mot de passe spécifié n'est pas valide. Notez que la clé privée fournie doit être une clé privée PEM RSA valide. »
Snowflake .MissingColumns	« La demande d'insertion est rejetée en raison de colonnes manquantes dans la charge utile d'entrée. Assurez-vous que des valeurs sont spécifiées pour toutes les colonnes non nullables. »
Snowflake .ExtraColumns	« La demande d'insertion est rejetée en raison de colonnes supplémentaires. Les colonnes absentes du tableau ne doivent pas être spécifiées »

Code d'erreur	Message et informations sur les erreurs
<code>Snowflake.InvalidInput</code>	« La livraison a échoué en raison d'un format de saisie non valide. Assurez-vous que la charge utile d'entrée fournie est au format JSON acceptable. »
<code>Snowflake.InvalidValue</code>	« La livraison a échoué en raison d'un type de données incorrect dans la charge utile d'entrée. Assurez-vous que les valeurs JSON spécifiées dans la charge utile d'entrée respectent le type de données déclaré dans la définition de la table Snowflake. »

Erreurs de livraison de données Splunk

Amazon Data Firehose peut envoyer les erreurs suivantes liées à Splunk à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
<code>Splunk.ProxyWithoutStickySessions</code>	« Si vous avez un proxy (ELB ou autre) entre Amazon Data Firehose et le nœud HEC, vous devez activer les sessions persistantes pour prendre en charge les HEC ACK. »
<code>Splunk.DisabledToken</code>	« Le jeton HEC est désactivé. Activez le jeton pour autoriser la livraison de données à Splunk. »
<code>Splunk.InvalidToken</code>	« Le jeton HEC n'est pas valide. Mettez à jour Amazon Data Firehose avec un jeton HEC valide. »
<code>Splunk.InvalidDataFormat</code>	« Les données ne sont pas formatées correctement. Pour savoir comment formater correctement les données pour les points de terminaison Raw ou d'événement HEC, reportez-vous à la section Données d'événement Splunk . »
<code>Splunk.InvalidIndex</code>	« Le jeton ou l'entrée HEC est configuré avec un index non valide. Vérifiez la configuration de l'index et réessayez. »
<code>Splunk.ServerError</code>	« La livraison de données à Splunk a échoué en raison d'une erreur de serveur provenant du nœud HEC. Amazon Data Firehose réessaiera

Code d'erreur	Message et informations sur les erreurs
	d'envoyer les données si la durée de la nouvelle tentative dans votre Amazon Data Firehose est supérieure à 0. Si toutes les tentatives échouent, Amazon Data Firehose sauvegarde les données sur Amazon S3. »
<code>Splunk.DisabledAck</code>	« L'accusé de réception indexeur est désactivé pour le jeton HEC. Activez l'accusé de réception indexeur et réessayez. Pour plus d'informations, consultez la section Activer l'accusé de réception indexeur . »
<code>Splunk.AckTimeout</code>	« Vous n'avez pas reçu d'accusé de réception HEC avant que le délai des accusés de réception HEC arrive à expiration. Malgré l'expiration de ce délai, il est possible que les données aient été indexées avec succès dans Splunk. Amazon Data Firehose sauvegarde dans Amazon S3 les données pour lesquelles le délai d'accusé de réception a expiré. »
<code>Splunk.MaxRetriesFailed</code>	« Échec de livraison des données à Splunk ou absence d'accusé de réception. Vérifiez l'état de votre HEC et réessayez. »
<code>Splunk.ConnectionTimeout</code>	« Expiration de la connexion à Splunk. Il s'agit peut-être d'une erreur temporaire et la requête sera renvoyée. Amazon Data Firehose sauvegarde les données sur Amazon S3 si toutes les tentatives échouent. »
<code>Splunk.InvalidEndpoint</code>	« Impossible de se connecter au point de terminaison HEC. Assurez-vous que l'URL du point de terminaison HEC est valide et accessible depuis Amazon Data Firehose. »
<code>Splunk.ConnectionClosed</code>	« Impossible d'envoyer des données à Splunk en raison d'un échec de connexion. Il s'agit peut être d'une erreur temporaire. L'augmentation de la durée des nouvelles tentatives dans votre configuration Amazon Data Firehose peut vous protéger contre de tels échecs transitoires. »
<code>Splunk.SSLUnverified</code>	« Impossible de se connecter au point de terminaison HEC. L'hôte ne correspond pas au certificat fourni par l'homologue. Vérifiez que le certificat et l'hôte sont valides . »

Code d'erreur	Message et informations sur les erreurs
<code>Splunk.SSLHandshake</code>	« Impossible de se connecter au point de terminaison HEC. Vérifiez que le certificat et l'hôte sont valides . »
<code>Splunk.URLNotFound</code>	« L'URL demandée n'a pas été trouvée sur le serveur Splunk. Vérifiez le cluster Splunk et assurez-vous qu'il est correctement configuré. »
<code>Splunk.ServerError.ContentTooLarge</code>	« La diffusion des données à Splunk a échoué en raison d'une erreur du serveur avec un statusCode : 413, message : la demande envoyée par votre client était trop importante. Consultez la documentation de Splunk pour configurer max_content_length. »
<code>Splunk.IndexerBusy</code>	« La livraison de données à Splunk a échoué en raison d'une erreur de serveur provenant du nœud HEC. Assurez-vous que le point de terminaison HEC ou l'Elastic Load Balancer sont accessibles et fonctionnent correctement. »
<code>Splunk.ConnectionRecycled</code>	« La connexion entre Firehose et Splunk a été recyclée. La diffusion sera tentée à nouveau. »
<code>Splunk.AcknowledgmentsDisabled</code>	« Impossible d'obtenir des accusés de réception sur POST. Assurez-vous que les accusés de réception sont activés sur le point de terminaison HEC. »
<code>Splunk.InvalidHecResponseCharacter</code>	« Caractères non valides trouvés dans la réponse HEC, assurez-vous de vérifier le service et la configuration HEC. »

ElasticSearch Erreurs de livraison de données

Amazon Data Firehose peut envoyer les ElasticSearch erreurs suivantes à CloudWatch Logs.

Code d'erreur	Message et informations sur les erreurs
ES.AccessDenied	« Accès refusé. Assurez-vous que le rôle IAM fourni associé à Firehose n'est pas supprimé. »
ES.ResourceNotFound	« Le domaine AWS Elasticsearch spécifié n'existe pas. »

Erreurs de diffusion des données du point de terminaison HTTPS

Amazon Data Firehose peut envoyer les erreurs suivantes liées au point de terminaison HTTP à Logs. CloudWatch Si aucune de ces erreurs ne correspond au problème que vous rencontrez, l'erreur par défaut est la suivante : « Une erreur interne s'est produite lors de la tentative de diffusion des données. La livraison sera réessayée ; si l'erreur persiste, elle sera signalée AWS pour résolution. »

Code d'erreur	Message et informations sur les erreurs
HttpEndpoint.RequestTimeout	La diffusion a expiré avant qu'une réponse ne soit reçue et sera réessayée. Si cette erreur persiste, contactez l'équipe du service AWS Firehose.
HttpEndpoint.ResponseTooLarge	« La réponse reçue du point de terminaison est trop volumineuse. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
HttpEndpoint.InvalidResponseFromDestination	« La réponse reçue du point de terminaison spécifié n'est pas valide. Contactez le propriétaire du point de terminaison pour résoudre le problème. »
HttpEndpoint.DestinationException	« La réponse suivante a été reçue de la destination du point de terminaison. »

Code d'erreur	Message et informations sur les erreurs
<code>HttpEndpoint.ConnectionFailed</code>	« Impossible de se connecter au point de terminaison de destination. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
<code>HttpEndpoint.ConnectionReset</code>	« Impossible de maintenir la connexion avec le point de terminaison. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
<code>HttpEndpoint.ConnectionReset</code>	« Difficulté à maintenir la connexion avec le point de terminaison. Veuillez contacter le propriétaire du point de terminaison. »
<code>HttpEndpoint.ResponseReasonPhraseExceededLimit</code>	« L'expression de raison de la réponse reçue du point de terminaison dépasse la limite configurée de 64 caractères. »
<code>HttpEndpoint.InvalidResponseFromDestination</code>	« La réponse reçue du point de terminaison n'est pas valide. Consultez la section Résolution des problèmes liés aux points de terminaison HTTP dans la documentation de Firehose pour plus d'informations. Raison : «
<code>HttpEndpoint.DestinationException</code>	« La livraison au point de terminaison a échoué. Consultez la section Résolution des problèmes liés aux points de terminaison HTTP dans la documentation de Firehose pour plus d'informations. Réponse reçue avec le code d'état »
<code>HttpEndpoint.InvalidStatusCode</code>	« Code d'état de la réponse reçue non valide. »
<code>HttpEndpoint.SSLHandshakeFailure</code>	« Impossible de terminer une négociation SSL avec le point de terminaison. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »

Code d'erreur	Message et informations sur les erreurs
<code>HttpEndpoint.SSLHandshakeFailure</code>	« Impossible de terminer une négociation SSL avec le point de terminaison. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
<code>HttpEndpoint.SSLFailure</code>	« Impossible de terminer la négociation TLS avec le point de terminaison. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
<code>HttpEndpoint.SSLHandshakeCertificatePathFailure</code>	« Impossible de terminer une négociation SSL avec le point de terminaison en raison d'un chemin de certification non valide. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
<code>HttpEndpoint.SSLHandshakeCertificatePathValidationFailure</code>	« Impossible de terminer une négociation SSL avec le point de terminaison en raison d'un échec de validation du chemin de certification. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
<code>HttpEndpoint.MakeRequestFailure.IllegalUriException</code>	« la HttpEndpoint demande a échoué en raison d'une saisie non valide dans l'URI. Assurez-vous que tous les caractères de l'URI d'entrée sont valides. »
<code>HttpEndpoint.MakeRequestFailure.IllegalCharacterInHeaderValue</code>	« la HttpEndpoint demande a échoué en raison d'une erreur de réponse illégale. Caractère non autorisé « \n » dans la valeur d'en-tête. »

Code d'erreur	Message et informations sur les erreurs
<code>HttpEndpoint.IllegalResponseFailure</code>	« la HttpEndpoint demande a échoué en raison d'une erreur de réponse illégale. Le message HTTP ne doit pas contenir plus d'un en-tête Content-Type. »
<code>HttpEndpoint.IllegalMessageStart</code>	« la HttpEndpoint demande a échoué en raison d'une erreur de réponse illégale. Début d'un message HTTP illégal. Consultez la section Résolution des problèmes liés aux points de terminaison HTTP dans la documentation de Firehose pour plus d'informations. »

Erreurs de livraison des données Amazon OpenSearch Service

En ce qui concerne la destination du OpenSearch service, Amazon Data Firehose envoie des erreurs aux CloudWatch journaux lorsqu'ils sont renvoyés par OpenSearch le service.

Outre les erreurs qui peuvent être renvoyées par les OpenSearch clusters, vous pouvez rencontrer les deux erreurs suivantes :

- Une erreur d'authentification/d'autorisation se produit lors de la tentative de livraison de données au cluster de OpenSearch services de destination. Cela peut se produire en raison de problèmes d'autorisation et/ou par intermittence lorsque la configuration de votre domaine du OpenSearch service cible Amazon Data Firehose est modifiée. Vérifiez la politique du cluster et les autorisations des rôles.
- Les données n'ont pas pu être livrées au cluster de OpenSearch services de destination en raison d'échecs d'authentification/d'autorisation. Cela peut se produire en raison de problèmes d'autorisation et/ou par intermittence lorsque la configuration de votre domaine du OpenSearch service cible Amazon Data Firehose est modifiée. Vérifiez la politique du cluster et les autorisations des rôles.

Code d'erreur	Message et informations sur les erreurs
<code>OS.AccessDenied</code>	« Accès refusé. Assurez-vous que la politique de confiance associée au rôle IAM fourni autorise Firehose à assumer ce rôle et que la politique d'accès autorise l'accès à l'API OpenSearch Amazon Service. »

Code d'erreur	Message et informations sur les erreurs
OS.AccessDenied	« Accès refusé. Assurez-vous que la politique de confiance associée au rôle IAM fourni autorise Firehose à assumer ce rôle et que la politique d'accès autorise l'accès à l'API OpenSearch Amazon Service. »
OS.AccessDenied	« Accès refusé. Assurez-vous que le rôle IAM fourni associé à Firehose n'est pas supprimé. »
OS.AccessDenied	« Accès refusé. Assurez-vous que le rôle IAM fourni associé à Firehose n'est pas supprimé. »
OS.ResourceNotFound	« Le domaine Amazon OpenSearch Service spécifié n'existe pas. »
OS.ResourceNotFound	« Le domaine Amazon OpenSearch Service spécifié n'existe pas. »
OS.AccessDenied	« Accès refusé. Assurez-vous que la politique de confiance associée au rôle IAM fourni autorise Firehose à assumer ce rôle et que la politique d'accès autorise l'accès à l'API OpenSearch Amazon Service. »
OS.RequestTimeout	« La demande adressée au cluster Amazon OpenSearch Service ou à la collecte OpenSearch sans serveur a expiré. Assurez-vous que le cluster ou la collection dispose d'une capacité suffisante pour la charge de travail actuelle. »
OS.ClusterError	« Le cluster Amazon OpenSearch Service a renvoyé une erreur non spécifiée. »
OS.RequestTimeout	« La demande adressée au cluster Amazon OpenSearch Service a expiré. Assurez-vous que le cluster dispose d'une capacité suffisante pour la charge de travail actuelle. »
OS.ConnectionFailed	« Problème de connexion au cluster Amazon OpenSearch Service ou à la collection OpenSearch Serverless. Assurez-vous que le cluster ou la collection est sain et accessible. »

Code d'erreur	Message et informations sur les erreurs
OS.ConnectionReset	« Impossible de maintenir la connexion avec le cluster Amazon OpenSearch Service ou la collection OpenSearch Serverless. Contactez le propriétaire du cluster ou de la collection pour résoudre ce problème. »
OS.ConnectionReset	« Problème lors du maintien de la connexion avec le cluster Amazon OpenSearch Service ou la collection OpenSearch Serverless. Assurez-vous que le cluster ou la collection est sain et dispose d'une capacité suffisante pour la charge de travail actuelle. »
OS.ConnectionReset	« Problème lors du maintien de la connexion avec le cluster Amazon OpenSearch Service ou la collection OpenSearch Serverless. Assurez-vous que le cluster ou la collection est sain et dispose d'une capacité suffisante pour la charge de travail actuelle. »
OS.AccessDenied	« Accès refusé. Assurez-vous que la politique d'accès du cluster Amazon OpenSearch Service autorise l'accès au rôle IAM configuré. »
OS.ValidationException	« Le OpenSearch cluster a renvoyé un ESServiceException. L'une des raisons est que le cluster a été mis à niveau vers OS 2.x ou supérieur, mais que le TypeName paramètre du tuyau est toujours configuré. Mettez à jour la configuration du TypeName tuyau en définissant une chaîne vide, ou remplacez le point de terminaison par le cluster, qui prend en charge le paramètre Type. »
OS.ValidationException	« Le membre doit satisfaire le modèle d'expression régulière suivant : [a-zA-Z0-9\-_]+
OS.JsonParseException	« Le cluster Amazon OpenSearch Service a renvoyé un JsonParse Exception. Assurez-vous que les données saisies sont valides. »
OS.AmazonOpenSearchServiceParseException	« Le cluster Amazon OpenSearch Service a renvoyé un AmazonOpenSearchServiceParseException. Assurez-vous que les données saisies sont valides. »

Code d'erreur	Message et informations sur les erreurs
<code>OS.ExplicitIndexInBulkNotAllowed</code>	« Assurez-vous que <code>rest.action.multi.allow_explicit_index</code> est défini sur <code>true</code> sur le cluster Amazon Service. » OpenSearch
<code>OS.ClusterError</code>	« Le cluster Amazon OpenSearch Service ou la collection OpenSearch Serverless ont renvoyé une erreur non spécifiée. »
<code>OS.ClusterBlockException</code>	« Le cluster a renvoyé un <code>ClusterBlockException</code> . Il est peut-être surchargé. »
<code>OS.InvalidARN</code>	« L'ARN du OpenSearch service Amazon fourni n'est pas valide. Vérifiez votre <code>DeliveryStream</code> configuration. »
<code>OS.MalformedData</code>	« Un ou plusieurs enregistrements sont mal formés. Assurez-vous que chaque enregistrement est un seul objet JSON valide et qu'il ne contient pas de nouvelles lignes. »
<code>OS.InternalError</code>	« Une erreur interne s'est produite lors de la tentative de diffusion des données. La livraison sera réessayée ; si l'erreur persiste, elle sera signalée AWS pour résolution. »
<code>OS.AliasWithMultipleIndicesNotAllowed</code>	« Plusieurs index sont associés à l'alias. Assurez-vous qu'un seul index est associé à l'alias. »
<code>OS.UnsupportedVersion</code>	« Amazon OpenSearch Service 6.0 n'est actuellement pas pris en charge par Amazon Data Firehose. Contactez AWS le Support pour plus d'informations. »
<code>OS.CharacterConversionException</code>	« Un ou plusieurs enregistrements contenaient un caractère non valide. »

Code d'erreur	Message et informations sur les erreurs
<code>OS.InvalidDomainNameLength</code>	« La longueur du nom de domaine n'est pas dans les limites valides du système d'exploitation. »
<code>OS.VPCDomainNotSupported</code>	« Les domaines Amazon OpenSearch Service au sein de VPC ne sont actuellement pas pris en charge. »
<code>OS.ConnectionError</code>	« Le serveur HTTP a fermé la connexion de façon inattendue. Vérifiez l'état du cluster Amazon OpenSearch Service ou de la collection OpenSearch Serverless. »
<code>OS.LargeFieldData</code>	« Le cluster Amazon OpenSearch Service a abandonné la demande car elle contenait des données de champ plus grandes que celles autorisées. »
<code>OS.BadGateway</code>	« Le cluster Amazon OpenSearch Service ou la collection OpenSearch Serverless a abandonné la demande avec la réponse 502 Bad Gateway. »
<code>OS.ServiceException</code>	« Erreur reçue du cluster Amazon OpenSearch Service ou de la collection OpenSearch Serverless. Si le cluster ou la collection se trouve derrière un VPC, assurez-vous que la configuration réseau autorise la connectivité. »
<code>OS.GatewayTimeout</code>	« Firehose a rencontré des erreurs de temporisation lors de la connexion au cluster Amazon OpenSearch Service ou à la collection OpenSearch Serverless. »
<code>OS.MalformedData</code>	« Amazon Data Firehose ne prend pas en charge les commandes de l'API Amazon OpenSearch Service Bulk dans l'enregistrement Firehose. »
<code>OS.ResponseEntryCountMismatch</code>	« La réponse de l'API Bulk contenait plus d'entrées que le nombre d'enregistrements envoyés. Assurez-vous que chaque enregistrement ne contient qu'un seul objet JSON et qu'il n'y a aucune nouvelle ligne. »

Erreurs d'invocation Lambda

Amazon Data Firehose peut envoyer les erreurs d'appel Lambda suivantes à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
<code>Lambda.AssumeRoleAccessDenied</code>	« Accès refusé. Assurez-vous que la politique de confiance associée au rôle IAM fourni autorise Amazon Data Firehose à assumer ce rôle. »
<code>Lambda.InvokeAccessDenied</code>	« Accès refusé. Vérifiez que la stratégie d'accès permet d'accéder à la fonction Lambda. »
<code>Lambda.JsonProcessingException</code>	<p>« Une erreur s'est produite lors de l'analyse des enregistrements retournés par la fonction Lambda. Assurez-vous que les enregistrements renvoyés suivent le modèle de statut requis par Amazon Data Firehose. »</p> <p>Pour plus d'informations, consultez Transformation de données et modèle d'état.</p>
<code>Lambda.InvokeLimitExceeded</code>	<p>« La limite appliquée aux exécutions simultanées Lambda est dépassée. Augmentez-la. »</p> <p>Pour plus d'informations, consultez la section Limites AWS Lambda dans le Guide du développeur AWS Lambda .</p>
<code>Lambda.DuplicatedRecordId</code>	<p>« Plusieurs enregistrements ont été retournés avec le même ID d'enregistrement. Vérifiez que la fonction Lambda retourne des ID d'enregistrements uniques pour chaque enregistrement. »</p> <p>Pour plus d'informations, consultez Transformation de données et modèle d'état.</p>
<code>Lambda.MissingRecordId</code>	« Un ou plusieurs ID d'enregistrements n'ont pas été retournés. Vérifiez que la fonction Lambda retourne tous les ID d'enregistrements reçus. »

Code d'erreur	Message et informations sur les erreurs
	Pour plus d'informations, consultez Transformation de données et modèle d'état .
Lambda.ResourceNotFound	« La fonction Lambda spécifiée n'existe pas. Utilisez une autre fonction existante. »
Lambda.InvalidSubnetIDException	« L'ID de sous-réseau spécifié dans la configuration VPC de la fonction Lambda n'est pas valide. Vérifiez que l'ID de sous-réseau est valide. »
Lambda.InvalidSecurityGroupIDException	« L'ID de groupe de sécurité spécifié dans la configuration VPC de la fonction Lambda n'est pas valide. Vérifiez que l'ID de groupe de sécurité est valide. »
Lambda.SubnetIPAddressLimitReachedException	« n'AWS Lambda a pas pu configurer l'accès VPC pour la fonction Lambda car un ou plusieurs sous-réseaux configurés n'ont aucune adresse IP disponible. Augmentez la limite appliquée aux adresses IP. » Pour de plus amples informations, consultez Limites Amazon VPC – VPC et sous-réseaux dans le Guide de l'utilisateur Amazon VPC.
Lambda.ENILimitReachedException	« n'AWS Lambda a pas pu créer d'interface réseau élastique (ENI) dans le VPC, spécifiée dans le cadre de la configuration de la fonction Lambda, car la limite d'interfaces réseau a été atteinte. Augmentez la limite appliquée à l'interface réseau. » Pour de plus amples informations, consultez Limites Amazon VPC – Interfaces réseau dans le Guide de l'utilisateur Amazon VPC.
Lambda.FunctionTimeout	L'invocation de la fonction Lambda a expiré. Augmentez le paramètre du délai d'expiration dans la fonction Lambda. Pour plus d'informations, consultez Configuration du délai d'expiration d'une fonction .

Code d'erreur	Message et informations sur les erreurs
<code>Lambda.FunctionError</code>	<p>Cela peut être dû à l'une des erreurs suivantes :</p> <ul style="list-style-type: none">• Structure de sortie non valide. Vérifiez votre fonction et assurez-vous que la sortie est au format requis. Assurez-vous également que les enregistrements traités contiennent un statut de résultat valide, à savoir <code>Dropped</code>, <code>Ok</code>, ou <code>ProcessingFailed</code> .• La fonction Lambda a été invoquée avec succès, mais elle a renvoyé un résultat d'erreur.• Lambda n'a pas pu déchiffrer les variables d'environnement, car l'accès KMS a été refusé. Vérifiez les paramètres de clé KMS de la fonction ainsi que la stratégie de clé. Pour de plus amples informations, consultez Troubleshooting Key Access.
<code>Lambda.FunctionRequestTimeout</code>	<p>Amazon Data Firehose a rencontré une erreur de configuration de la requête « Request did not complete before the request timeout » lors de l'appel de Lambda. Revoyez le code Lambda pour vérifier s'il est censé s'exécuter au-delà du délai d'expiration configuré. Si tel est le cas, envisagez d'ajuster les paramètres de configuration de Lambda, y compris la mémoire, le délai d'expiration. Pour plus d'informations, consultez Configuration des options de fonction Lambda.</p>
<code>Lambda.TargetServerFailedToRespond</code>	<p>Amazon Data Firehose a rencontré une erreur. Le serveur cible n'a pas répondu à l'erreur lors de l'appel du AWS service Lambda.</p>
<code>Lambda.InvalidZipFileException</code>	<p>Amazon Data Firehose s'est produit <code>InvalidZipFileException</code> lors de l'appel de la fonction Lambda. Vérifiez les paramètres de configuration de votre fonction Lambda et le fichier zip du code Lambda.</p>

Code d'erreur	Message et informations sur les erreurs
<code>Lambda.InternalServerError</code>	« Amazon Data Firehose s'est rencontré <code>InternalServerError</code> lors de l'appel du service Lambda AWS . Amazon Data Firehose réessaiera d'envoyer des données un nombre fixe de fois. Vous pouvez spécifier ou remplacer les options de tentative d'appel à l'aide des API <code>CreateDeliveryStream</code> ou <code>UpdateDestination</code> . Si l'erreur persiste, contactez l'équipe d'assistance AWS Lambda.
<code>Lambda.ServiceUnavailable</code>	Amazon Data Firehose s'est produit <code>ServiceUnavailableException</code> lors de l'appel du service Lambda AWS . Amazon Data Firehose réessaiera d'envoyer des données un nombre fixe de fois. Vous pouvez spécifier ou remplacer les options de tentative d'appel à l'aide des API <code>CreateDeliveryStream</code> ou <code>UpdateDestination</code> . Si l'erreur persiste, contactez le support AWS Lambda.
<code>Lambda.InvalidSecurityToken</code>	Impossible d'invoquer la fonction Lambda en raison d'un jeton de sécurité non valide. L'invocation Lambda entre partitions n'est pas pris en charge.
<code>Lambda.InvocationFailure</code>	Cela peut être dû à l'une des erreurs suivantes : <ul style="list-style-type: none">• Amazon Data Firehose a rencontré des erreurs lors de l'appel de Lambda AWS . L'opération sera tentée à nouveau ; si l'erreur persiste, elle sera reportée à AWS pour être résolue. »• Amazon Data Firehose a rencontré un KMS provenant de <code>InvalidStateException</code> Lambda. Lambda n'a pas pu déchiffrer les variables d'environnement parce que la clé KMS utilisée est dans un état non valide pour le déchiffrement. Vérifiez la clé KMS de la fonction Lambda.• Amazon Data Firehose a rencontré un homme provenant de <code>AWS LambdaException</code> Lambda. Lambda n'a pas pu initialiser l'image du conteneur fournie. Vérifiez l'image.• Amazon Data Firehose a rencontré des erreurs de temporisation lors de l'appel de Lambda. AWS Le délai maximum de fonctionnement pris en charge est de cinq minutes. Pour plus d'informations, consultez Data Transformation Execution Duration.

Code d'erreur	Message et informations sur les erreurs
Lambda . Js onMapping Exception	Une erreur s'est produite lors de l'analyse des enregistrements retournés par la fonction Lambda. Assurez-vous que le champ de données est codé en base 64.

Erreurs d'invocation Kinesis

Amazon Data Firehose peut envoyer les erreurs d'invocation Kinesis suivantes à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
Kinesis . A ccessDenied	« L'accès a été refusé lors de l'appel de Kinesis. Assurez-vous que la stratégie d'accès du rôle IAM utilisé autorise l'accès aux API Kinesis appropriées. »
Kinesis . R esourceNo tFound	« Firehose n'a pas réussi à lire le flux. Si Firehose est connecté à Kinesis Stream, le flux n'existe peut-être pas, ou la partitionnée a peut-être été fusionnée ou divisée. Si le Firehose est de DirectPut type Firehose, il se peut qu'il n'existe plus. »
Kinesis . S ubscripti onRequired	« L'accès a été refusé lors de l'appel de Kinesis. Assurez-vous que le rôle IAM transmis pour l'accès au flux Kinesis est associé à un abonnement AWS Kinesis. »
Kinesis . T hrottling	« Une erreur de limitation s'est produite lors de l'appel de Kinesis. Cela peut être dû au fait que d'autres applications appellent les mêmes API que le flux Firehose, ou au fait que vous avez créé trop de flux Firehose avec le même flux Kinesis comme source. »
Kinesis . T hrottling	« Une erreur de limitation s'est produite lors de l'appel de Kinesis. Cela peut être dû au fait que d'autres applications appellent les mêmes API que le flux Firehose, ou au fait que vous avez créé trop de flux Firehose avec le même flux Kinesis comme source. »

Code d'erreur	Message et informations sur les erreurs
<code>Kinesis.AccessDenied</code>	« L'accès a été refusé lors de l'appel de Kinesis. Assurez-vous que la stratégie d'accès du rôle IAM utilisé autorise l'accès aux API Kinesis appropriées. »
<code>Kinesis.AccessDenied</code>	« L'accès a été refusé alors que j'essayais d'appeler des opérations d'API sur le Kinesis Stream sous-jacent. Assurez-vous que le rôle IAM est propagé et valide. »
<code>Kinesis.KMS.AccessDeniedException</code>	« Firehose n'a pas accès à la clé KMS utilisée pour crypter/déchiffrer le Kinesis Stream. Veuillez accorder au rôle de diffusion Firehose l'accès à la clé. »
<code>Kinesis.KMS.KeyDisabled</code>	« Firehose ne parvient pas à lire le Kinesis Stream source, car la clé KMS utilisée pour le chiffrer/déchiffrer est désactivée. Activez la touche pour que les lectures puissent continuer. »
<code>Kinesis.KMS.InvalidStateException</code>	« Firehose ne parvient pas à lire le Kinesis Stream source, car la clé KMS utilisée pour le chiffrer est dans un état non valide. »
<code>Kinesis.KMS.NotFoundException</code>	« Firehose ne parvient pas à lire le Kinesis Stream source, car la clé KMS utilisée pour le chiffrer n'a pas été trouvée. »

Erreurs d'invocation Kinesis DirectPut

Amazon Data Firehose peut envoyer les erreurs d'invocation DirectPut Kinesis suivantes à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
<code>Firehose.KMS.Access</code>	« Firehose n'a pas accès à la clé KMS. Veuillez vérifier la stratégie de clé. »

Code d'erreur	Message et informations sur les erreurs
<code>sDeniedException</code>	
<code>Firehose.KMS.InvalidStateException</code>	« Firehose ne parvient pas à déchiffrer les données, car la clé KMS utilisée pour le chiffrer est dans un état non valide. »
<code>Firehose.KMS.NotFoundException</code>	« Firehose ne parvient pas à déchiffrer les données, car la clé KMS utilisée pour le chiffrer n'a pas été trouvée. »
<code>Firehose.KMS.KeyDisabled</code>	« Firehose ne parvient pas à déchiffrer les données, car la clé KMS utilisée pour chiffrer les données a été désactivée. » Activez la clé pour que la diffusion des données puisse se poursuivre. »

AWS Glue Erreurs d'invocation

Amazon Data Firehose peut envoyer les erreurs d' AWS Glue invocation suivantes à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
<code>DataFormatConversion.InvalidSchema</code>	« Le schéma n'est pas valide. »
<code>DataFormatConversion.EntityNotFound</code>	« La table/base de données spécifiée est introuvable. Assurez-vous que la table/base de données existe et que les valeurs fournies dans la configuration du schéma sont correctes, en particulier en ce qui concerne la casse. »
<code>DataFormatConversion</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la base de données spécifiée avec l'ID de catalogue fourni existe. »

Code d'erreur	Message et informations sur les erreurs
<code>on.InvalidInput</code>	
<code>DataFormatConversion.InvalidInput</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que le format de l'ARN transmis est correct. »
<code>DataFormatConversion.InvalidInput</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que le catalogId fourni est valide. »
<code>DataFormatConversion.InvalidVersionId</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la version spécifiée de la table existe. »
<code>DataFormatConversion.NonExistentColumns</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la table est configurée avec un descripteur de stockage non nul contenant les colonnes cibles. »
<code>DataFormatConversion.AccessDenied</code>	« Accès refusé en assumant le rôle. Assurez-vous que le rôle spécifié dans la configuration de conversion du format des données a accordé au service Firehose l'autorisation de l'assumer. »
<code>DataFormatConversion.ThrottledByGlue</code>	« Une erreur de limitation s'est produite lors de l'appel de Glue. Augmentez la limite du taux de demandes ou réduisez le taux actuel d'appels via d'autres applications. »

Code d'erreur	Message et informations sur les erreurs
DataFormatConversion.AccessDenied	« L'accès a été refusé lors de l'appel de Glue. Assurez-vous que le rôle spécifié dans la configuration de conversion du format des données dispose des autorisations nécessaires. »
DataFormatConversion.InvalidGlueRole	« Rôle non valide. Assurez-vous que le rôle spécifié dans la configuration de conversion du format des données existe. »
DataFormatConversion.InvalidGlueRole	« Le jeton de sécurité inclus dans la demande n'est pas valide. Assurez-vous que le rôle IAM fourni associé à Firehose n'est pas supprimé. »
DataFormatConversion.GlueNotAvailableInRegion	« AWS Glue n'est pas encore disponible dans la région que vous avez spécifiée ; veuillez en spécifier une autre. »
DataFormatConversion.GlueEncryptionException	« Une erreur s'est produite lors de la récupération de la clé principale. Assurez-vous que la clé existe et qu'elle dispose des autorisations d'accès appropriées. »
DataFormatConversion.SchemaValidationTimeout	« Le délai imparti a expiré lors de la récupération de la table sur Glue. Si vous possédez un grand nombre de versions de table Glue, veuillez ajouter l'autorisation « glue : GetTableVersion » (recommandée) ou supprimer les versions de table non utilisées. Si vous n'avez pas un grand nombre de tables dans Glue, veuillez contacter le AWS Support. »

Code d'erreur	Message et informations sur les erreurs
<code>DataFirehose.InternalError</code>	« Le délai imparti a expiré lors de la récupération de la table sur Glue. Si vous possédez un grand nombre de versions de table Glue, veuillez ajouter l'autorisation « glue : GetTableVersion » (recommandée) ou supprimer les versions de table non utilisées. Si vous n'avez pas un grand nombre de tables dans Glue, veuillez contacter le AWS Support. »
<code>DataFormatConversion.GlueEncryptionException</code>	« Une erreur s'est produite lors de la récupération de la clé principale. Assurez-vous que la clé existe et que l'état est correct. »

DataFormatConversion Erreurs d'invocation

Amazon Data Firehose peut envoyer les erreurs d' `DataFormatConversion` invocation suivantes à Logs. CloudWatch

Code d'erreur	Message et informations sur les erreurs
<code>DataFormatConversion.InvalidSchema</code>	« Le schéma n'est pas valide. »
<code>DataFormatConversion.ValidationException</code>	« Les noms et types de colonnes doivent être des chaînes non vides. »
<code>DataFormatConversion.ParseError</code>	« Un JSON malformé a été rencontré. »
<code>DataFormatConversion</code>	« Les données ne correspondent pas au schéma. »

Code d'erreur	Message et informations sur les erreurs
<code>on.MalformedData</code>	
<code>DataFormatConversion.MalformedData</code>	« La longueur de la clé JSON ne doit pas être supérieure à 262 144 »
<code>DataFormatConversion.MalformedData</code>	« Les données ne peuvent pas être décodées en UTF-8. »
<code>DataFormatConversion.MalformedData</code>	« Un caractère illégal a été trouvé entre les jetons. »
<code>DataFormatConversion.InvalidTypeFormat</code>	« Le format de type n'est pas valide. Vérifiez la syntaxe du type. »
<code>DataFormatConversion.InvalidSchema</code>	« Schéma non valide. Assurez-vous qu'il n'y a pas de caractères spéciaux ou d'espaces blancs dans les noms de colonnes. »
<code>DataFormatConversion.InvalidRecord</code>	« L'enregistrement n'est pas conforme au schéma. Une ou plusieurs clés de mappage n'étaient pas valides pour le mappage <string,string>. »

Code d'erreur	Message et informations sur les erreurs
<code>DataFormatConversion.MalformedData</code>	« Le JSON d'entrée contenait un objet ou un tableau primitif au niveau supérieur. Le niveau supérieur doit être un objet ou un tableau. »
<code>DataFormatConversion.MalformedData</code>	« Le JSON d'entrée contenait un objet ou un tableau primitif au niveau supérieur. Le niveau supérieur doit être un objet ou un tableau. »
<code>DataFormatConversion.MalformedData</code>	« L'enregistrement était vide ou ne contenait que des espaces. »
<code>DataFormatConversion.MalformedData</code>	« Caractères non valides rencontrés. »
<code>DataFormatConversion.MalformedData</code>	« Le format d'horodatage rencontré n'est pas valide ou n'est pas pris en charge. Consultez le guide du développeur Firehose pour connaître les formats d'horodatage pris en charge. »
<code>DataFormatConversion.MalformedData</code>	« Un type scalaire a été trouvé dans les données, mais un type complexe a été spécifié dans le schéma. »
<code>DataFormatConversion.MalformedData</code>	« Les données ne correspondent pas au schéma. »

Code d'erreur	Message et informations sur les erreurs
<code>DataFormatConversion.MalformedData</code>	« Un type scalaire a été trouvé dans les données, mais un type complexe a été spécifié dans le schéma. »
<code>DataFormatConversion.ConversionFailureException</code>	"ConversionFailureException"
<code>DataFormatConversion.DataFormatConversionCustomerErrorException</code>	"DataFormatConversionCustomerErrorException"
<code>DataFormatConversion.DataFormatConversionCustomerErrorException</code>	"DataFormatConversionCustomerErrorException"
<code>DataFormatConversion.MalformedData</code>	« Les données ne correspondent pas au schéma. »

Code d'erreur	Message et informations sur les erreurs
<code>DataFormatConversion.InvalidSchema</code>	« Le schéma n'est pas valide. »
<code>DataFormatConversion.MalformedData</code>	« Les données ne correspondent pas au schéma. Format non valide pour une ou plusieurs dates. »
<code>DataFormatConversion.MalformedData</code>	« Les données contiennent une structure JSON hautement imbriquée qui n'est pas prise en charge. »
<code>DataFormatConversion.EntityNotFound</code>	« La table/base de données spécifiée est introuvable. Assurez-vous que la table/base de données existe et que les valeurs fournies dans la configuration du schéma sont correctes, en particulier en ce qui concerne la casse. »
<code>DataFormatConversion.InvalidInput</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la base de données spécifiée avec l'ID de catalogue fourni existe. »
<code>DataFormatConversion.InvalidInput</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que le format de l'ARN transmis est correct. »
<code>DataFormatConversion.InvalidInput</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que le catalogId fourni est valide. »

Code d'erreur	Message et informations sur les erreurs
<code>DataFormatConversion.InvalidVersionId</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la version spécifiée de la table existe. »
<code>DataFormatConversion.NonExistentColumns</code>	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la table est configurée avec un descripteur de stockage non nul contenant les colonnes cibles. »
<code>DataFormatConversion.AccessDenied</code>	« Accès refusé en assumant le rôle. Assurez-vous que le rôle spécifié dans la configuration de conversion du format des données a accordé au service Firehose l'autorisation de l'assumer. »
<code>DataFormatConversion.ThrottledByGlue</code>	« Une erreur de limitation s'est produite lors de l'appel de Glue. Augmentez la limite du taux de demandes ou réduisez le taux actuel d'appels via d'autres applications. »
<code>DataFormatConversion.AccessDenied</code>	« L'accès a été refusé lors de l'appel de Glue. Assurez-vous que le rôle spécifié dans la configuration de conversion du format des données dispose des autorisations nécessaires. »
<code>DataFormatConversion.InvalidGlueRole</code>	« Rôle non valide. Assurez-vous que le rôle spécifié dans la configuration de conversion du format des données existe. »
<code>DataFormatConversion.GlueNotAvailableInRegion</code>	« AWS Glue n'est pas encore disponible dans la région que vous avez spécifiée ; veuillez en spécifier une autre. »

Code d'erreur	Message et informations sur les erreurs
<code>DataFormatConversion.GlueEncryptionException</code>	« Une erreur s'est produite lors de la récupération de la clé principale. Assurez-vous que la clé existe et qu'elle dispose des autorisations d'accès appropriées. »
<code>DataFormatConversion.SchemaValidationTimeout</code>	« Le délai imparti a expiré lors de la récupération de la table sur Glue. Si vous possédez un grand nombre de versions de table Glue, veuillez ajouter l'autorisation « glue : GetTableVersion » (recommandée) ou supprimer les versions de table non utilisées. Si vous n'avez pas un grand nombre de tables dans Glue, veuillez contacter le AWS Support. »
<code>DataFirehose.InternalError</code>	« Le délai imparti a expiré lors de la récupération de la table sur Glue. Si vous possédez un grand nombre de versions de table Glue, veuillez ajouter l'autorisation « glue : GetTableVersion » (recommandée) ou supprimer les versions de table non utilisées. Si vous n'avez pas un grand nombre de tables dans Glue, veuillez contacter le AWS Support. »
<code>DataFormatConversion.MalformedData</code>	« Le format d'un ou de plusieurs champs est incorrect. »

Accès aux CloudWatch journaux pour Amazon Data Firehose

Vous pouvez consulter les journaux d'erreurs liés à l'échec de livraison des données Amazon Data Firehose à l'aide de la console Amazon Data Firehose ou de la console CloudWatch. Les procédures suivantes vous montrent comment accéder aux journaux d'erreurs à l'aide de ces deux méthodes.

Pour accéder aux journaux d'erreurs à l'aide de la console Amazon Data Firehose

1. Connectez-vous à la console Firehose AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/firehose>
2. Dans la barre de navigation, choisissez une AWS région.
3. Choisissez un nom de stream Firehose pour accéder à la page de détails du stream Firehose.

4. Choisissez Error Log pour afficher la liste des journaux d'erreurs liés aux défaillances de diffusion de données.

Pour accéder aux journaux d'erreurs à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).
2. Sélectionnez une région dans la barre de navigation.
3. Dans le panneau de navigation, sélectionnez Logs (Journaux).
4. Choisissez un groupe et un flux de journaux pour afficher la liste des journaux d'erreurs liés aux défaillances de diffusion de données.

Surveillance de l'état de Kinesis Agent

Kinesis Agent publie des CloudWatch métriques personnalisées avec un espace de noms de. AWS KinesisAgent Cela permet de déterminer si l'agent est sain, s'il soumet les données à Amazon Data Firehose conformément aux spécifications et s'il consomme la quantité appropriée de ressources CPU et mémoire sur le producteur de données.

Les métriques telles que le nombre d'enregistrements et d'octets envoyés sont utiles pour comprendre le rythme auquel l'agent soumet des données au flux Firehose. Lorsque ces métriques deviennent inférieurs d'un certain pourcentage aux seuils attendus ou nulles, elles peuvent traduire des problèmes de configuration, des erreurs réseau ou des problèmes d'état de l'agent. Les métriques comme l'utilisation de l'UC ou de la mémoire sur l'hôte et les compteurs d'erreurs de l'agent indiquent l'utilisation des ressources de l'application productrice de données et donnent des informations sur la configuration potentielle ou les erreurs de l'hôte. Enfin, l'agent consigne également les exceptions au niveau du service pour mieux examiner les problèmes de l'agent.

Ces métriques d'agent sont présentées dans la région spécifiée dans le paramètre de configuration `cloudwatch.endpoint`. Pour plus d'informations, consultez [Paramètres de configuration de l'agent](#).

Les métriques Cloudwatch qui sont publiées par plusieurs Kinesis Agent sont agrégées ou combinées.

Il y a une charge nominale pour les métriques émises par Kinesis Agent qui sont activées par défaut. Pour plus d'informations, consultez [Amazon CloudWatch Pricing](#).

Surveillance avec CloudWatch

Kinesis Agent envoie les métriques suivantes à CloudWatch

Métrique	Description
BytesSent	Le nombre d'octets envoyés au flux Firehose au cours de la période spécifiée. Unités : octets
RecordSendAttempts	Nombre d'enregistrements tentés (une première fois ou lors d'une nouvelle tentative) dans un appel de PutRecordBatch au cours de la période spécifiée. Unités : nombre
RecordSendErrors	Nombre d'enregistrements qui ont renvoyé un état d'échec dans un appel de PutRecordBatch, y compris les nouvelles tentatives, au cours de la période spécifiée. Unités : nombre
ServiceErrors	Nombre d'appels de PutRecordBatch qui ont provoqué une erreur de service (autre qu'une erreur de limitation) au cours de la période spécifiée. Unités : nombre

Journalisation des appels d'API Amazon Data Firehose avec AWS CloudTrail

Amazon Data Firehose est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un AWS service dans Amazon Data Firehose. CloudTrail capture tous les appels d'API pour Amazon Data Firehose sous forme d'événements. Les appels capturés incluent des appels provenant de la console Amazon Data Firehose et des appels de code vers les opérations de l'API Amazon Data Firehose. Si vous créez un suivi, vous pouvez activer la diffusion continue d'événements CloudTrail vers un compartiment Amazon S3, y compris

des événements pour Amazon Data Firehose. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Data Firehose, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le [guide de AWS CloudTrail l'utilisateur](#).

Informations sur Amazon Data Firehose dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité événementielle prise en charge se produit dans Amazon Data Firehose, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS . Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#).

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements liés à Amazon Data Firehose, créez un historique. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- [Vue d'ensemble de la création d'un journal d'activité](#)
- [CloudTrail Services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

Amazon Data Firehose prend en charge l'enregistrement des actions suivantes sous forme d'événements dans des fichiers CloudTrail journaux :

- [CreateDeliveryStream](#)
- [DeleteDeliveryStream](#)

- [DescribeDeliveryStream](#)
- [ListDeliveryStreams](#)
- [ListTagsForDeliveryStream](#)
- [TagDeliveryStream](#)
- [StartDeliveryStreamEncryption](#)
- [StopDeliveryStreamEncryption](#)
- [UntagDeliveryStream](#)
- [UpdateDestination](#)

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été faite avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez l'élément [CloudTrailUserIdentity](#).

Exemple : entrées dans le fichier journal Amazon Data Firehose

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre les `DeleteDeliveryStream` actions `CreateDeliveryStream` `DescribeDeliveryStream` `ListDeliveryStreams`, `UpdateDestination`, et.

```
{  
  "Records": [  

```

```

{
  "eventVersion":"1.02",
  "userIdentity":{
    "type":"IAMUser",
    "principalId":"AKIAIOSFODNN7EXAMPLE",
    "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
    "accountId":"111122223333",
    "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
    "userName":"CloudTrail_Test_User"
  },
  "eventTime":"2016-02-24T18:08:22Z",
  "eventSource":"firehose.amazonaws.com",
  "eventName":"CreateDeliveryStream",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"127.0.0.1",
  "userAgent":"aws-internal/3",
  "requestParameters":{
    "deliveryStreamName":"TestRedshiftStream",
    "redshiftDestinationConfiguration":{
      "s3Configuration":{
        "compressionFormat":"GZIP",
        "prefix":"prefix",
        "bucketARN":"arn:aws:s3:::firehose-cloudtrail-test-bucket",
        "roleARN":"arn:aws:iam::111122223333:role/Firehose",
        "bufferingHints":{
          "sizeInMBs":3,
          "intervalInSeconds":900
        },
        "encryptionConfiguration":{
          "kMSEncryptionConfig":{
            "aWSKMSKeyARN":"arn:aws:kms:us-east-1:key"
          }
        }
      }
    },
    "clusterJDBCURL":"jdbc:redshift://example.abc123.us-
west-2.redshift.amazonaws.com:5439/dev",
    "copyCommand":{
      "copyOptions":"copyOptions",
      "dataTableName":"dataTable"
    },
    "password":"","
    "username":"","
    "roleARN":"arn:aws:iam::111122223333:role/Firehose"
  }
}

```

```

    },
    "responseElements":{
      "deliveryStreamARN":"arn:aws:firehose:us-
east-1:111122223333:deliverystream/TestRedshiftStream"
    },
    "requestID":"958abf6a-db21-11e5-bb88-91ae9617edf5",
    "eventID":"875d2d68-476c-4ad5-bbc6-d02872cfc884",
    "eventType":"AwsApiCall",
    "recipientAccountId":"111122223333"
  },
  {
    "eventVersion":"1.02",
    "userIdentity":{
      "type":"IAMUser",
      "principalId":"AKIAIOSFODNN7EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
      "accountId":"111122223333",
      "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
      "userName":"CloudTrail_Test_User"
    },
    "eventTime":"2016-02-24T18:08:54Z",
    "eventSource":"firehose.amazonaws.com",
    "eventName":"DescribeDeliveryStream",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"aws-internal/3",
    "requestParameters":{
      "deliveryStreamName":"TestRedshiftStream"
    },
    "responseElements":null,
    "requestID":"aa6ea5ed-db21-11e5-bb88-91ae9617edf5",
    "eventID":"d9b285d8-d690-4d5c-b9fe-d1ad5ab03f14",
    "eventType":"AwsApiCall",
    "recipientAccountId":"111122223333"
  },
  {
    "eventVersion":"1.02",
    "userIdentity":{
      "type":"IAMUser",
      "principalId":"AKIAIOSFODNN7EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
      "accountId":"111122223333",
      "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
      "userName":"CloudTrail_Test_User"
    }
  }
}

```

```

    },
    "eventTime": "2016-02-24T18:10:00Z",
    "eventSource": "firehose.amazonaws.com",
    "eventName": "ListDeliveryStreams",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3",
    "requestParameters": {
      "limit": 10
    }
  },
  "responseElements": null,
  "requestID": "d1bf7f86-db21-11e5-bb88-91ae9617edf5",
  "eventID": "67f63c74-4335-48c0-9004-4ba35ce00128",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
},
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/CloudTrail_Test_User",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "CloudTrail_Test_User"
  },
  "eventTime": "2016-02-24T18:10:09Z",
  "eventSource": "firehose.amazonaws.com",
  "eventName": "UpdateDestination",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-internal/3",
  "requestParameters": {
    "destinationId": "destinationId-000000000001",
    "deliveryStreamName": "TestRedshiftStream",
    "currentDeliveryStreamVersionId": "1",
    "redshiftDestinationUpdate": {
      "roleARN": "arn:aws:iam::111122223333:role/Firehose",
      "clusterJDBCURL": "jdbc:redshift://example.abc123.us-west-2.redshift.amazonaws.com:5439/dev",
      "password": "",
      "username": "",
      "copyCommand": {
        "copyOptions": "copyOptions",

```

```

        "dataTableName":"dataTable"
    },
    "s3Update":{
        "bucketARN":"arn:aws:s3:::firehose-cloudtrail-test-bucket-update",
        "roleARN":"arn:aws:iam::111122223333:role/Firehose",
        "compressionFormat":"GZIP",
        "bufferingHints":{
            "sizeInMBs":3,
            "intervalInSeconds":900
        },
        "encryptionConfiguration":{
            "kMSEncryptionConfig":{
                "aWSKMSKeyARN":"arn:aws:kms:us-east-1:key"
            }
        },
        "prefix":"arn:aws:s3:::firehose-cloudtrail-test-bucket"
    }
}
},
"responseElements":null,
"requestID":"d549428d-db21-11e5-bb88-91ae9617edf5",
"eventID":"1cb21e0b-416a-415d-bbf9-769b152a6585",
"eventType":"AwsApiCall",
"recipientAccountId":"111122223333"
},
{
    "eventVersion":"1.02",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"AKIAIOSFODNN7EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
        "accountId":"111122223333",
        "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
        "userName":"CloudTrail_Test_User"
    },
    "eventTime":"2016-02-24T18:10:12Z",
    "eventSource":"firehose.amazonaws.com",
    "eventName":"DeleteDeliveryStream",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"aws-internal/3",
    "requestParameters":{
        "deliveryStreamName":"TestRedshiftStream"
    }
},

```

```
    "responseElements":null,  
    "requestID":"d85968c1-db21-11e5-bb88-91ae9617edf5",  
    "eventID":"dd46bb98-b4e9-42ff-a6af-32d57e636ad1",  
    "eventType":"AwsApiCall",  
    "recipientAccountId":"111122223333"  
  }  
]  
}
```

Préfixes personnalisés pour les objets Amazon S3

Les objets livrés à Amazon S3 suivent le [format de nom](#) de <evaluated prefix><suffix>. Vous pouvez spécifier votre préfixe personnalisé qui inclut les expressions évaluées lors de l'exécution. Le préfixe personnalisé que vous spécifiez remplacera le préfixe par défaut de. YYYY/MM/dd/HH

Vous pouvez utiliser les expressions ayant le format suivant dans votre préfixe personnalisé : ! {namespace: *value*}, où namespace peut être l'un des éléments suivants, comme expliqué dans les sections suivantes.

- firehose
- timestamp
- partitionKeyFromQuery
- partitionKeyFromLambda

Si un préfixe se termine par une barre oblique, il apparaît comme dossier dans le compartiment Amazon S3. Pour plus d'informations, consultez le [format du nom d'objet Amazon S3](#) dans le Amazon Data Firehose Developer Guide.

L'espace de noms **timestamp**

Les valeurs valides pour cet espace de noms sont des DateTimeFormatter chaînes [Java](#) valides. Par exemple, au cours de l'année 2018, l'expression ! {timestamp: yyyy} est évaluée sur 2018.

Lors de l'évaluation des horodatages, Firehose utilise l'horodatage d'arrivée approximatif du plus ancien enregistrement contenu dans l'objet Amazon S3 en cours d'écriture.

Par défaut, l'horodatage est exprimé en UTC. Mais vous pouvez spécifier le fuseau horaire que vous préférez. Par exemple, vous pouvez configurer le fuseau horaire vers Asie/Tokyo dans le réglage des paramètres AWS Management Console ou dans l'API ([CustomTimeZone](#)) si vous souhaitez utiliser l'heure normale du Japon au lieu de l'heure UTC. Pour consulter la liste des fuseaux horaires pris en charge, consultez [Amazon S3 Object Name Format](#).

Si vous utilisez l'espace de noms `timestamp` plusieurs fois dans la même expression de préfixe, chaque instance est évaluée sur le même instantané dans le temps.

L'espace de noms **firehose**

Il existe deux valeurs que vous pouvez utiliser avec cet espace de noms : `error-output-type` et `random-string`. Le tableau suivant explique comment les utiliser.

Les valeurs de l'espace de noms **firehose**

Conversion	Description	Exemple d'entrée	Exemple de sortie	Remarques
<code>error-output-type</code>	Évalue l'une des chaînes suivantes, en fonction de la configuration de votre stream Firehose et de la raison de l'échec : <code>{processing-failed, -failed, splunk-failed AmazonOpenSearchService,,}.format-conversion-failed http-endpoint-failed</code> Si vous utilisez l'espace de noms plusieurs fois dans la même expression, chaque instance est évaluée sur la	<code>myPrefix/result=!{firehose:error-output-type}/!{timestamp:yyyy/MM/dd}</code>	<code>myPrefix/result=processing-failed/2018/08/03</code>	La <code>error-output-type</code> valeur ne peut être utilisée que dans le <code>ErrorOutputPrefix</code> champ.

Conversion	Description	Exemple d'entrée	Exemple de sortie	Remarques
	même chaîne d'erreur.			
random-string	Évaluation sur une chaîne aléatoire de 11 caractères. Si vous utilisez l'espace de noms plusieurs fois dans la même expression, chaque instance est évaluée sur une nouvelle chaîne aléatoire.	myPrefix/!{firehose:random-string}/	myPrefix/046b6c7f-0b/	Vous pouvez l'utiliser avec les deux types de préfixe. Vous pouvez le placer au début de la chaîne de format pour obtenir un préfixe aléatoire, ce qui est parfois nécessaire pour atteindre un débit extrêmement élevé avec Amazon S3.

Espaces de noms `partitionKeyFromLambda` et `partitionKeyFromQuery`

Pour le [partitionnement dynamique](#), vous devez utiliser le format d'expression suivant dans le préfixe de votre compartiment S3 : `!{namespace:value}`, où l'espace de noms peut être `partitionKeyFromQuery` ou `partitionKeyFromLambda`, ou les deux. Si vous utilisez l'analyse en ligne pour créer les clés de partitionnement de vos données sources, vous devez spécifier une valeur de préfixe de compartiment S3 qui consiste en des expressions spécifiées dans le format suivant : `"partitionKeyFromQuery:keyID"`. Si vous utilisez une fonction AWS Lambda pour créer les clés de partitionnement de vos données sources, vous devez spécifier une valeur de préfixe de compartiment S3 qui consiste en des expressions spécifiées dans le format suivant :

"partitionKeyFromLambda:keyID". Pour plus d'informations, consultez la section « Choisissez Amazon S3 pour votre destination » dans [Création d'un flux Amazon Firehose](#).

Règles sémantiques

Les règles suivantes s'appliquent aux expressions `Prefix` et `ErrorOutputPrefix`.

- Pour l'espace de noms `timestamp`, n'importe quel caractère autre que des guillemets simples est évalué. En d'autres termes, n'importe quelle chaîne dans une séquence d'échappement avec des guillemets simples dans le champ `value` est prise littéralement.
- Si vous spécifiez un préfixe qui ne contient pas d'expression d'espace de noms d'horodatage, Firehose ajoute l'expression `!{timestamp:yyyy/MM/dd/HH/}` à la valeur du champ `Prefix`.
- La séquence `!{` peut uniquement apparaître dans les expressions `!{namespace: value}`.
- `ErrorOutputPrefix` peut être null uniquement si `Prefix` ne contient pas d'expressions. Dans ce cas, `Prefix` correspond à `<specified-prefix>yyyy/MM/DDD/HH/` et `ErrorOutputPrefix` correspond à `<specified-prefix><error-output-type>YYYY/MM/DDD/HH/`. DDD représente le jour de l'année.
- Si vous spécifiez une expression pour `ErrorOutputPrefix`, vous devez inclure au moins une instance de `!{firehose:error-output-type}`.
- `Prefix` ne peut pas contenir `!{firehose:error-output-type}`.
- Ni `Prefix` ni `ErrorOutputPrefix` ne peuvent être supérieurs à 512 caractères après leur évaluation.
- Si la destination est Amazon Redshift, `Prefix` ne doit pas contenir d'expressions et `ErrorOutputPrefix` doit être null.
- Lorsque la destination est Amazon OpenSearch Service ou Splunk, et qu'aucune `ErrorOutputPrefix` n'est spécifiée, Firehose utilise `Prefix` le champ pour les enregistrements ayant échoué.
- Lorsque la destination est Amazon S3, les préfixes `Prefix` et `ErrorOutputPrefix` dans la configuration de destination Amazon S3 sont utilisés pour les enregistrements ayant réussi et les enregistrements ayant échoué, respectivement. Si vous utilisez AWS CLI ou l'API, vous pouvez utiliser `ExtendedS3DestinationConfiguration` pour spécifier une configuration de sauvegarde Amazon S3 avec ses propres préfixes `Prefix` et `ErrorOutputPrefix`.
- Lorsque vous utilisez le AWS Management Console et définissez la destination sur Amazon S3, Firehose utilise le `Prefix` et `ErrorOutputPrefix` dans la configuration de destination pour les enregistrements réussis et les enregistrements échoués, respectivement. Si vous spécifiez

un préfixe mais aucun préfixe d'erreur, Firehose définit automatiquement le préfixe d'erreur sur. !
`{firehose:error-output-type}/`

- Lorsque vous utilisez `ExtendedS3DestinationConfiguration` l'API ou AWS CLI, si vous spécifiez une `AWS CloudFormationS3BackupConfiguration`, Firehose ne fournit pas de valeur par défaut. `ErrorOutputPrefix`
- Vous ne pouvez pas utiliser `partitionKeyFromLambda` d'`partitionKeyFromQueryespaces` de noms lorsque vous créez des `ErrorOutputPrefix` expressions.

Exemples de préfixe

Exemples de préfixes **Prefix** et **ErrorOutputPrefix**

Entrée	Préfixe évalué (à 10:30 AM UTC le 27août2018)
Prefix : non spécifié ErrorOutputPrefix : myFirehoseFailures/!{firehose:error-output-type}/	Prefix: 2018/08/27/10 ErrorOutputPrefix : myFirehoseFailures/processing-failed/
Prefix: !{timestamp:yyyy/MM/dd} ErrorOutputPrefix : non spécifié	Saisie non valide : ErrorOutputPrefix ne peut pas être vide si le préfixe contient des expressions
Prefix: myFirehose/DeliveredYear=!{timestamp:yyyy}/anyMonth/rand=!{firehose:random-string} ErrorOutputPrefix : myFirehoseFailures/!{firehose:error-output-type}/!{timestamp:yyyy}/anyMonth/!{timestamp:dd}	Prefix: myFirehose/DeliveredYear=2018/anyMonth/rand=5abf82daaa5 ErrorOutputPrefix : myFirehoseFailures/processing-failed/2018/anyMonth/10
Prefix: myPrefix/year=!{timestamp:yyyy}/month=!{timestamp:MM}/day=!{timestamp:dd}/hour=!{timestamp:HH}/	Prefix: myPrefix/year=2018/month=07/day=06/hour=23/

Entrée	Préfixe évalué (à 10:30 AM UTC le 27août2018)
<pre>ErrorOutputPrefix : myErrorPrefix/ year={!{timestamp:yyyy}/month=! {timestamp:MM}/day={!{timesta mp:dd}/hour={!{timestamp:HH}/! {firehose:error-output-type}</pre>	<pre>ErrorOutputPrefix : myErrorPrefix/ year=2018/month=07/day=06/hour= 23/processing-failed</pre>
<pre>Prefix: myFirehosePrefix/ ErrorOutputPrefix : non spécifié</pre>	<pre>Prefix: myFirehosePrefix/2 018/08/27/ ErrorOutputPrefix : myFirehos ePrefix/processing-failed/2 018/08/27/</pre>

Utilisation d'Amazon Data Firehose avec AWS PrivateLink

Points de terminaison VPC d'interface ()AWS PrivateLink pour Amazon Data Firehose

Vous pouvez utiliser un point de terminaison VPC d'interface pour empêcher le trafic entre votre Amazon VPC et Amazon Data Firehose de quitter le réseau Amazon. Les points de terminaison VPC d'interface ne nécessitent pas de passerelle Internet, de périphérique NAT, de connexion VPN ou de connexion. AWS Direct Connect Les points de terminaison VPC d'interface sont alimentés par AWS PrivateLink une AWS technologie qui permet une communication privée entre les AWS services à l'aide d'une interface réseau élastique avec des adresses IP privées dans votre Amazon VPC. Pour en savoir plus, consultez [Amazon Virtual Private Cloud](#).

Utilisation des points de terminaison VPC d'interface ()AWS PrivateLink pour Amazon Data Firehose

Pour commencer, créez un point de terminaison VPC d'interface afin que le trafic Amazon Data Firehose provenant de vos ressources Amazon VPC commence à circuler via le point de terminaison VPC d'interface. Lorsque vous créez un point de terminaison, vous pouvez y associer une politique de point de terminaison qui contrôle l'accès à Amazon Data Firehose. Pour en savoir plus sur l'utilisation de politiques pour contrôler l'accès d'un point de terminaison VPC à Amazon Data Firehose, consultez la section [Contrôle de l'accès aux services avec](#) des points de terminaison VPC.

L'exemple suivant montre comment configurer une AWS Lambda fonction dans un VPC et créer un point de terminaison VPC pour permettre à la fonction de communiquer en toute sécurité avec le service Amazon Data Firehose. Dans cet exemple, vous utilisez une politique qui permet à la fonction Lambda de répertorier les flux Firehose dans la région actuelle, mais pas de décrire un flux Firehose.

Création d'un point de terminaison de VPC

1. [Connectez-vous à la console Amazon VPC AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/vpc/](https://console.aws.amazon.com/vpc/).
2. Dans le tableau de bord VPC, choisissez Endpoints (Points de terminaison).
3. Choisissez Créer un point de terminaison.

4. Dans la liste des noms de service, choisissez `com.amazonaws.your_region.kinesis-firehose`.
5. Choisissez le VPC et un ou plusieurs sous-réseaux dans lesquels créer le point de terminaison.
6. Choisissez un ou plusieurs groupes de sécurité à associer au point de terminaison.
7. Pour Policy (Stratégie), choisissez Custom (Personnalisée) et collez la stratégie suivante :

```
{
  "Statement": [
    {
      "Sid": "Allow-only-specific-PrivateAPIs",
      "Principal": "*",
      "Action": [
        "firehose:ListDeliveryStreams"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Allow-only-specific-PrivateAPIs",
      "Principal": "*",
      "Action": [
        "firehose:DescribeDeliveryStream"
      ],
      "Effect": "Deny",
      "Resource": [
        "*"
      ]
    }
  ]
}
```

8. Choisissez Créer un point de terminaison.

Création d'un rôle IAM à utiliser avec la fonction Lambda

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, choisissez Rôles, puis Créer un rôle.
3. Sous Sélectionner le type d'entité de confiance, conservez la sélection par défaut Service AWS .

4. Sous Choose the service that will use this role (Choisir le service qui utilisera ce rôle), choisissez Lambda.
5. Sélectionnez Next: Permissions (Suivant : autorisations).
6. Dans la liste des stratégies, recherchez et ajoutez les deux stratégies nommées AWS LambdaVPCLambdaAccessExecutionRole et AmazonDataFirehoseReadOnlyAccess.

 Important

Il s'agit d'un exemple. Il se peut que vous ayez besoin de stratégies plus strictes pour votre environnement de production.

7. Choisissez Suivant : Balises. Vous n'avez pas besoin d'ajouter des balises dans le cadre de cet exercice. Choisissez Suivant : vérification.
8. Entrez un nom pour le rôle, puis choisissez Créer un rôle.

Création d'une fonction Lambda au sein du VPC

1. Ouvrez la AWS Lambda console à l'[adresse https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/).
2. Choisissez Créer une fonction.
3. Choisissez Créer à partir de zéro.
4. Entrez le nom de la fonction, puis définissez Runtime sur Python 3.9 ou une valeur supérieure.
5. Sous Permissions (Autorisations), développez Choose or create an execution role (Choisir ou créer un rôle d'exécution).
6. Dans la liste Execution role (Rôle d'exécution), choisissez Use an existing role (Utiliser un rôle existant).
7. Dans la liste Existing role (Rôle existant), choisissez le rôle que vous avez créé ci-dessus.
8. Choisissez Créer une fonction.
9. Sous Function code (Code de fonction), Collez le code suivant.

```
import json
import boto3
import os
from botocore.exceptions import ClientError
```

```
def lambda_handler(event, context):
    REGION = os.environ['AWS_REGION']
    client = boto3.client(
        'firehose',
        REGION
    )
    print("Calling list_delivery_streams with ListDeliveryStreams allowed
policy.")
    delivery_stream_request = client.list_delivery_streams()
    print("Successfully returned list_delivery_streams request %s." % (
        delivery_stream_request
    ))
    describe_access_denied = False
    try:
        print("Calling describe_delivery_stream with DescribeDeliveryStream
denied policy.")
        delivery_stream_info =
client.describe_delivery_stream(DeliveryStreamName='test-describe-denied')
    except ClientError as e:
        error_code = e.response['Error']['Code']
        print ("Caught %s." % (error_code))
        if error_code == 'AccessDeniedException':
            describe_access_denied = True

    if not describe_access_denied:
        raise
    else:
        print("Access denied test succeeded.")
```

10. Sous Basic settings (Paramètres de base), définissez le délai d'expiration sur 1 minute.
11. Sous Network (Réseau), sélectionnez le VPC dans lequel vous avez créé le point de terminaison ci-dessus, puis choisissez les sous-réseaux et le groupe de sécurité que vous avez associés au point de terminaison lorsque vous l'avez créé.
12. Dans le haut de la page, choisissez Enregistrer.
13. Sélectionnez Tester).
14. Entrez un nom d'événement, puis choisissez Créer.
15. Choisissez Test à nouveau. La fonction est alors exécutée. Une fois que le résultat d'exécution s'affiche, développez Détails (Détails) et comparez la sortie de journal au code de la fonction. Les résultats positifs indiquent une liste des streams Firehose de la région, ainsi que le résultat suivant :

Calling `describe_delivery_stream`.

`AccessDeniedException`

Access denied test succeeded.

Disponibilité

Les points de terminaison de VPC d'interface sont actuellement pris en charge dans les régions suivantes :

- USA Est (Ohio)
- USA Est (Virginie du Nord)
- USA Ouest (Californie du Nord)
- USA Ouest (Oregon)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asia Pacific (Tokyo)
- Asie-Pacifique (Hong Kong)
- Canada (Centre)
- Canada Ouest (Calgary)
- Chine (Beijing)
- China (Ningxia)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Paris)
- Amérique du Sud (São Paulo)
- AWS GovCloud (USA Est)
- AWS GovCloud (US-Ouest)

- Europe (Espagne)
- Moyen-Orient (EAU)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Osaka)
- Israël (Tel Aviv)

Marquer vos streams Firehose dans Amazon Data Firehose

Vous pouvez attribuer vos propres métadonnées aux flux Firehose que vous créez dans Amazon Data Firehose sous forme de balises. Une balise est une paire clé-valeur que vous définissez pour un flux. L'utilisation de balises est un moyen simple mais puissant de gérer les AWS ressources et d'organiser les données, y compris les données de facturation.

Rubriques

- [Principes de base des identifications](#)
- [Suivi des coûts à l'aide du balisage](#)
- [Restrictions liées aux balises](#)
- [Marquage de flux Firehose à l'aide de l'API Amazon Data Firehose](#)

Principes de base des identifications

Vous pouvez utiliser l'API Amazon Data Firehose pour effectuer les tâches suivantes :

- Ajoutez des tags à un stream Firehose.
- Répertoirez les tags de vos streams Firehose.
- Supprimez les tags d'un stream Firehose.

Vous pouvez utiliser des tags pour classer vos streams Firehose. Par exemple, vous pouvez classer les streams Firehose par objectif, propriétaire ou environnement. Dans la mesure où vous avez défini la clé et la valeur de chaque balise, vous pouvez créer un ensemble personnalisé de catégories répondant à vos besoins spécifiques. Par exemple, vous pouvez définir un ensemble de balises qui vous aident à suivre les flux Firehose par propriétaire et par application associée.

Voici plusieurs exemples de balises :

- Project: *Project name*
- Owner: *Name*
- Purpose: Load testing
- Application: *Application name*
- Environment: Production

Si vous spécifiez des balises dans `CreateDeliveryStreamAction`, Amazon Data Firehose octroie une autorisation supplémentaire à `firehose:TagDeliveryStreamAction` afin de vérifier si les utilisateurs sont autorisés à créer des balises. Si vous ne fournissez pas cette autorisation, les demandes de création de nouveaux flux Firehose avec des balises de ressources IAM échoueront avec l'erreur suivante `AccessDeniedException`.

AccessDeniedException

```
User: arn:aws:sts::x:assumed-role/x/x is not authorized to perform:
  firehose:TagDeliveryStream on resource: arn:aws:firehose:us-east-1:x:deliverystream/x
  with an explicit deny in an identity-based policy.
```

L'exemple suivant illustre une politique qui permet aux utilisateurs de créer un flux Firehose et d'appliquer des balises.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "firehose:CreateDeliveryStream",
      "Resource": "*",
    },
    {
      "Effect": "Allow",
      "Action": "firehose:TagDeliveryStream",
      "Resource": "*",
    }
  ]
}
```

Suivi des coûts à l'aide du balisage

Vous pouvez utiliser des balises pour classer et suivre vos AWS coûts. Lorsque vous appliquez des balises à vos AWS ressources, notamment aux streams Firehose, votre rapport de répartition des AWS coûts inclut l'utilisation et les coûts agrégés par balises. Vous pouvez organiser vos coûts entre plusieurs services en appliquant des balises qui représentent des catégories métier (telles que les centres de coûts, les noms d'applications ou les propriétaires). Pour plus d'informations, consultez

[Utilisation des identifications de répartition des coûts pour les rapports de facturation personnalisés](#) dans le Guide de l'utilisateur AWS Billing .

Restrictions liées aux balises

Les restrictions suivantes s'appliquent aux balises dans Amazon Data Firehose.

Restrictions de base

- Le nombre maximum de balises par ressource (flux) est de 50.
- Les clés et valeurs d'étiquette sont sensibles à la casse.
- Vous ne pouvez pas changer ou modifier les balises d'un flux supprimé.

Restrictions relatives aux clés de balise

- Chaque clé de balise doit être unique. Si vous ajoutez une balise avec une clé qui est déjà en cours d'utilisation, la nouvelle balise remplacera la paire clé-valeur existante.
- Vous ne pouvez pas démarrer une clé de balise avec, `aws :` car ce préfixe est réservé à une utilisation par AWS. AWS crée des balises qui commencent par ce préfixe en votre nom, mais vous ne pouvez pas les modifier ou les supprimer.
- Les clés de balise doivent comporter entre 1 et 128 caractères Unicode.
- Les clés de balise doivent comporter les caractères suivants : lettres Unicode, chiffres, espaces et les caractères spéciaux suivants : `_ . / = + - @`.

Restrictions relatives à la valeur de balise

- Les valeurs de balise doivent comporter entre 0 et 255 caractères Unicode.
- Les valeurs de balise peuvent être vides. Si tel n'est pas le cas, elles doivent être composées des caractères suivants : lettres Unicode, chiffres, espaces et les caractères spéciaux suivants : `_ . / = + - @`.

Marquage de flux Firehose à l'aide de l'API Amazon Data Firehose

Vous pouvez spécifier des balises lorsque vous invoquez [CreateDeliveryStream](#) pour créer un nouveau stream Firehose. Pour les streams Firehose existants, vous pouvez ajouter, répertorier et supprimer des tags en effectuant les trois opérations suivantes :

- [TagDeliveryStream](#)
- [ListTagsForDeliveryStream](#)
- [UntagDeliveryStream](#)

Tutoriel : Ingérer les journaux de flux VPC dans Splunk à l'aide d'Amazon Data Firehose

Pour un didacticiel, consultez [Ingérer les journaux de flux VPC dans Splunk à l'aide d'Amazon Data Firehose](#).

Résolution des problèmes liés à Amazon Data Firehose

Si Firehose rencontre des erreurs lors de la livraison ou du traitement des données, il réessaie jusqu'à ce que la durée de tentative configurée expire. Si la durée de la nouvelle tentative prend fin avant que les données ne soient correctement livrées, Firehose sauvegarde les données dans le compartiment de sauvegarde S3 configuré. Si la destination est Amazon S3 et que la livraison échoue ou si la livraison vers le compartiment S3 de sauvegarde échoue, Firehose continue de réessayer jusqu'à la fin de la période de rétention. Pour les streams `DirectPut` Firehose, Firehose conserve les enregistrements pendant 24 heures. Pour un flux Firehose dont la source de données est un flux de données Kinesis, vous pouvez modifier la période de conservation comme décrit dans [Modification de la période de conservation des données](#).

Si la source de données est un flux de données Kinesis, Firehose réessaie indéfiniment les opérations suivantes : `DescribeStream`, `GetRecords`, `GetShardIterator`

Si le stream Firehose l'utilise `DirectPut`, vérifiez les `IncomingRecords` métriques `IncomingBytes` et pour voir s'il y a du trafic entrant. Si vous utilisez `PutRecord` ou `PutRecordBatch`, assurez-vous de tenir compte des exceptions et de réessayer. Nous recommandons une stratégie incluant une interruption exponentielle avec instabilité et plusieurs tentatives. De plus, si vous utilisez l'`PutRecordBatchAPI`, assurez-vous que votre code vérifie la valeur de `FailedPutCount` dans la réponse, même lorsque l'appel d'API aboutit.

Si le flux Firehose utilise un flux de données Kinesis comme source, vérifiez les `IncomingRecords` métriques `IncomingBytes` et pour le flux de données source. De plus, assurez-vous que les `DataReadFromKinesisStream.Records` métriques `DataReadFromKinesisStream.Bytes` et sont émises pour le flux Firehose.

Pour plus d'informations sur le suivi des erreurs de livraison à CloudWatch l'aide de [the section called "Surveillance à l'aide de CloudWatch journaux"](#).

Problèmes courants

Voici quelques problèmes courants et comment les résoudre.

- Le flux Firehose n'est pas disponible en tant que cible pour les CloudWatch journaux, les CloudWatch événements ou les actions AWS IoT. Certains AWS services peuvent uniquement envoyer des messages et des événements à un flux Firehose qui se trouve dans le même flux.

Région AWS Vérifiez que votre stream Firehose se trouve dans la même région que vos autres services.

- Aucune donnée à destination malgré de bons indicateurs — S'il n'y a aucun problème d'ingestion de données et que les indicateurs émis pour le flux Firehose semblent bons, mais que vous ne voyez pas les données à destination, vérifiez la logique du lecteur. Assurez-vous que le lecteur analyse correctement toutes les données.

Résolutions des problèmes liés à Amazon S3

Vérifiez ce qui suit si les données ne sont pas diffusées à votre compartiment Amazon Simple Storage Service (Amazon S3).

- Vérifiez le `Firehose IncomingBytes` et `IncomingRecords` les métriques pour vous assurer que les données sont correctement envoyées à votre flux Firehose. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#).
- Si la transformation des données avec Lambda est activée, vérifiez la `ExecuteProcessingSuccess` métrique Firehose pour vous assurer que Firehose a essayé d'appeler votre fonction Lambda. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#).
- Vérifiez la `DeliveryToS3.Success` métrique Firehose pour vous assurer que Firehose a essayé d'insérer des données dans votre compartiment Amazon S3. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#).
- Activez la journalisation des erreurs si elle n'est pas déjà activée et vérifiez les échecs de diffusion dans les journaux d'erreur. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de journaux CloudWatch](#).
- Si vous voyez un message d'erreur dans le journal indiquant « Firehose a été détecté `InternalServerError` lors de l'appel du service Amazon S3 ». L'opération sera réessayée ; si l'erreur persiste, contactez S3 pour obtenir une résolution. », cela pourrait être dû à l'augmentation significative des taux de requêtes sur une seule partition dans S3. Vous pouvez optimiser les modèles de conception des préfixes S3 pour atténuer le problème. Pour plus d'informations, consultez la section [Modèles de conception des meilleures pratiques : optimisation des performances d'Amazon S3](#). Si cela ne résout pas le problème, contactez le AWS Support pour obtenir une assistance supplémentaire.
- Assurez-vous que le compartiment Amazon S3 spécifié dans votre flux Firehose existe toujours.

- Si la transformation des données avec Lambda est activée, assurez-vous que la fonction Lambda spécifiée dans votre flux Firehose existe toujours.
- Assurez-vous que le rôle IAM spécifié dans votre flux Firehose a accès à votre compartiment S3 et à votre fonction Lambda (si la transformation des données est activée). Assurez-vous également que le rôle IAM a accès au groupe de CloudWatch journaux et aux flux de journaux pour vérifier les journaux d'erreurs. Pour plus d'informations, consultez [Accorder à Amazon Data Firehose l'accès à une destination Amazon S3](#).
- Si vous utilisez la transformation de données, assurez-vous que votre fonction Lambda ne renvoie jamais les réponses dont la taille de charge utile dépasse 6 Mo. Pour plus d'informations, consultez [Amazon Data Firehose Data Transformation](#).

Dépannage d'Amazon Redshift

Vérifiez les points suivants si les données ne sont pas diffusées dans votre cluster provisionné Amazon Redshift ou votre groupe de travail Amazon Redshift sans serveur.

Les données sont diffusées dans votre compartiment S3 avant leur chargement dans Amazon Redshift. Si les données n'ont pas été diffusées à votre compartiment S3, consultez la rubrique [Résolutions des problèmes liés à Amazon S3](#).

- Vérifiez la `DeliveryToRedshift.Success` métrique Firehose pour vous assurer que Firehose a essayé de copier les données de votre compartiment S3 vers le cluster provisionné Amazon Redshift ou le groupe de travail Amazon Redshift Serverless. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#).
- Activez la journalisation des erreurs si elle n'est pas déjà activée et vérifiez les échecs de diffusion dans les journaux d'erreur. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de journaux CloudWatch](#).
- Consultez le `STL_CONNECTION_LOG` tableau Amazon Redshift pour voir si Firehose peut établir des connexions réussies. Dans cette table, vous devriez pouvoir afficher les connexions et leur statut en fonction d'un nom d'utilisateur. Pour en savoir plus, consultez [STL_CONNECTION_LOG](#) dans le Guide du développeur de base de données Amazon Redshift.
- Si le contrôle précédent révèle que les connexions sont établies, vérifiez la table `STL_LOAD_ERRORS` d'Amazon Redshift pour connaître la raison de l'échec de l'opération COPY. Pour en savoir plus, consultez [STL_LOAD_ERRORS](#) dans le Guide du développeur de base de données Amazon Redshift.
- Assurez-vous que la configuration Amazon Redshift de votre stream Firehose est précise et valide.

- Assurez-vous que le rôle IAM spécifié dans votre flux Firehose peut accéder au compartiment S3 à partir duquel Amazon Redshift copie les données, ainsi qu'à la fonction Lambda pour la transformation des données (si la transformation des données est activée). Assurez-vous également que le rôle IAM a accès au groupe de CloudWatch journaux et aux flux de journaux pour vérifier les journaux d'erreurs. Pour plus d'informations, consultez [Accorder à Amazon Data Firehose l'accès à une destination Amazon Redshift](#) .
- Si votre cluster provisionné Amazon Redshift ou votre groupe de travail Amazon Redshift Serverless se trouve dans un cloud privé virtuel (VPC), assurez-vous que le cluster autorise l'accès depuis les adresses IP Firehose. Pour plus d'informations, consultez [Accorder à Amazon Data Firehose l'accès à une destination Amazon Redshift](#) .
- Assurez-vous que le cluster Amazon Redshift provisionné ou le groupe de travail Amazon Redshift sans serveur est accessible au public.
- Si vous utilisez la transformation de données, assurez-vous que votre fonction Lambda ne renvoie jamais les réponses dont la taille de charge utile dépasse 6 Mo. Pour plus d'informations, consultez [Amazon Data Firehose Data Transformation](#).

Résolution des problèmes liés à Amazon OpenSearch Service

Vérifiez les points suivants si les données ne sont pas livrées à votre domaine OpenSearch de service.

Les données peuvent être sauvegardées dans votre compartiment Amazon S3 simultanément. Si les données n'ont pas été diffusées à votre compartiment S3, consultez la rubrique [Résolutions des problèmes liés à Amazon S3](#).

- Vérifiez le `Firehose IncomingBytes` et `IncomingRecords` les métriques pour vous assurer que les données sont correctement envoyées à votre flux Firehose. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#) .
- Si la transformation des données avec Lambda est activée, vérifiez la `ExecuteProcessingSuccess` métrique Firehose pour vous assurer que Firehose a essayé d'appeler votre fonction Lambda. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#) .
- Vérifiez la `DeliveryToAmazonOpenSearchService` . `Success` métrique Firehose pour vous assurer que Firehose a essayé d'indexer les données sur le cluster de services. OpenSearch Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de métriques CloudWatch](#) .

- Activez la journalisation des erreurs si elle n'est pas déjà activée et vérifiez les échecs de diffusion dans les journaux d'erreur. Pour plus d'informations, consultez [Surveillance d'Amazon Data Firehose à l'aide de journaux CloudWatch](#) .
- Assurez-vous que la configuration du OpenSearch service dans votre stream Firehose est précise et valide.
- Si la transformation des données avec Lambda est activée, assurez-vous que la fonction Lambda spécifiée dans votre flux Firehose existe toujours. Assurez-vous également que le rôle IAM a accès au groupe de CloudWatch journaux et aux flux de journaux pour vérifier les journaux d'erreurs. Pour plus d'informations, voir [Subvention FirehoseAccess à une destination OpenSearch de service public](#).
- Assurez-vous que le rôle IAM spécifié dans votre flux Firehose peut accéder à votre cluster de services, à OpenSearch votre bucket de sauvegarde S3 et à votre fonction Lambda (si la transformation des données est activée). Assurez-vous également que le rôle IAM a accès au groupe de CloudWatch journaux et aux flux de journaux pour vérifier les journaux d'erreurs. Pour plus d'informations, voir [Subvention FirehoseAccess à une destination OpenSearch de service public](#).
- Si vous utilisez la transformation de données, assurez-vous que votre fonction Lambda ne renvoie jamais les réponses dont la taille de charge utile dépasse 6 Mo. Pour plus d'informations, consultez [Amazon Data FirehoseData Transformation](#).
- Amazon Data Firehose ne prend actuellement pas en charge la livraison de journaux CloudWatch vers la OpenSearch destination Amazon Service, car Amazon CloudWatch combine plusieurs événements de journal dans un seul enregistrement Firehose et Amazon OpenSearch Service ne peut pas accepter plusieurs événements de journal dans un seul enregistrement. Vous pouvez également envisager d'[utiliser un filtre d'abonnement pour Amazon OpenSearch Service in CloudWatch Logs](#).

Dépannage de Splunk

Lisez les explications suivantes si les données ne sont pas transmises à votre point de terminaison Splunk.

- Si votre plateforme Splunk se trouve dans un VPC, assurez-vous que Firehose peut y accéder. Pour plus d'informations, consultez [Accès à Splunk dans un VPC](#).
- Si vous utilisez un équilibreur de AWS charge, assurez-vous qu'il s'agit d'un Classic Load Balancer ou d'un Application Load Balancer. Activez également les sessions persistantes basées sur la

durée avec l'expiration des cookies désactivée pour Classic Load Balancer et l'expiration fixée au maximum (7 jours) pour Application Load Balancer. [Pour plus d'informations sur la procédure à suivre, consultez la section Stickiness de session basée sur la durée pour un Classic Load Balancer ou un Application Load Balancer.](#)

- Passez en revue les exigences de la plate-forme Splunk. Le module complémentaire Splunk pour Firehose nécessite la version 6.6.X ou ultérieure de la plateforme Splunk. Pour plus d'informations, consultez [Module complémentaire Splunk pour Amazon Kinesis Firehose](#).
- Si vous disposez d'un proxy (Elastic Load Balancing ou autre) entre Firehose et le nœud HTTP Event Collector (HEC), activez les sessions persistantes pour prendre en charge les accusés de réception HEC (ACK).
- Vérifiez que vous utilisez bien un jeton HEC valide.
- Assurez-vous que le jeton HEC est activé. Voir [Activer et désactiver les jetons Event Collector](#).
- Vérifiez si les données que vous envoyez à Splunk sont correctement formatées. Pour plus d'informations, consultez [Format des événements pour HTTP Event Collector](#).
- Assurez-vous que le jeton HEC et l'événement d'entrée sont configurés avec un index valide.
- Si un chargement sur Splunk échoue en raison d'une erreur de serveur sur le nœud HEC, la demande fait automatiquement l'objet d'une nouvelle tentative. Si toutes les tentatives échouent, les données sont sauvegardées sur Amazon S3. Vérifiez si vos données figurent dans Amazon S3, ce qui indiquerait un échec de ce type.
- Vérifiez que vous avez bien activé l'accusé de réception indexeur sur votre jeton HEC. Pour plus d'informations, consultez [Activer l'accusé de réception indexeur](#).
- Augmentez la valeur de `HECAcknowledgmentTimeoutInSeconds` dans la configuration de destination Splunk de votre stream Firehose.
- Augmentez la valeur de `DurationInSeconds` under `RetryOptions` dans la configuration de destination Splunk de votre stream Firehose.
- Vérifiez l'état de votre HEC.
- Si vous utilisez la transformation de données, assurez-vous que votre fonction Lambda ne renvoie jamais les réponses dont la taille de charge utile dépasse 6 Mo. Pour plus d'informations, consultez [Amazon Data FirehoseData Transformation](#).
- Assurez-vous que le paramètre Splunk nommé `ackIdleCleanup` est défini sur `true`. Il est défini sur `false` par défaut. Pour définir ce paramètre sur `true`, procédez comme suit :
 - Pour un déploiement Splunk Cloud [géré](#), soumettez une demande à l'aide du portail de support Splunk. Dans ce cas, demandez au support de Splunk d'activer le collecteur d'événements

HTTP, définissez `ackIdleCleanup` sur `true` dans `inputs.conf`, et créez ou modifiez un équilibreur de charge à utiliser avec ce module complémentaire.

- Pour un [déploiement Splunk Enterprise distribué](#), définissez le paramètre `ackIdleCleanup` sur `true` dans le fichier `inputs.conf`. Pour les utilisateurs *nix, ce fichier se trouve sous `$SPLUNK_HOME/etc/apps/splunk_httpinput/local/`. Pour les utilisateurs Windows, il est sous `%SPLUNK_HOME%\etc\apps\splunk_httpinput\local\`.
- Pour un [déploiement Splunk Enterprise à instance unique](#), définissez le paramètre `ackIdleCleanup` sur `true` dans le fichier `inputs.conf`. Pour les utilisateurs *nix, ce fichier se trouve sous `$SPLUNK_HOME/etc/apps/splunk_httpinput/local/`. Pour les utilisateurs Windows, il est sous `%SPLUNK_HOME%\etc\apps\splunk_httpinput\local\`.
- Assurez-vous que le rôle IAM spécifié dans votre flux Firehose peut accéder au compartiment de sauvegarde S3 et à la fonction Lambda pour la transformation des données (si la transformation des données est activée). Assurez-vous également que le rôle IAM a accès au groupe CloudWatch Logs et aux flux de journaux pour vérifier les journaux d'erreurs. Pour plus d'informations, consultez la section [Subvention FirehoseAccess à une destination Splunk](#).
- Consultez [Troubleshoot the Splunk Add-on for Amazon Kinesis Firehose](#).

Résolution des problèmes liés à Snowflake

Cette section décrit les étapes de dépannage courantes lors de l'utilisation de Snowflake comme destination.

La création du stream Firehose échoue

Si la création d'un flux Firehose échoue pour un flux fournissant des données à un cluster Snowflake PrivateLink activé, cela indique que Firehose n'a pas accès au VPCE-ID. Cela peut être dû à l'une des raisons suivantes :

- Le VPCE-ID est incorrect. Vérifiez qu'il n'y a aucune erreur typographique.
- Firehose ne prend pas en charge les URL Snowflake sans région dans la version préliminaire. Fournissez l'URL à l'aide du localisateur de compte Snowflake. Consultez la [documentation de Snowflake](#) pour plus de détails.
- Vérifiez que le stream Firehose est créé dans la même AWS région que la région Snowflake.
- Si le problème persiste, contactez l' AWS assistance.

Défaillances de livraison

Vérifiez les points suivants si les données ne sont pas transmises à votre table Snowflake. Les données ayant échoué à la livraison de Snowflake seront envoyées au compartiment d'erreur S3 avec un code d'erreur et un message d'erreur correspondant à la charge utile. Voici quelques scénarios d'erreur courants. Pour obtenir la liste complète des codes d'erreur, consultez [Erreurs de livraison de données Snowflake](#).

- Code d'erreur : Snowflake. DefaultRoleMissing: indique que le rôle Snowflake n'est pas configuré lors de la création du stream Firehose. Si le rôle Snowflake n'est pas configuré, assurez-vous de définir un rôle par défaut pour l'utilisateur Snowflake spécifié.
- Code d'erreur : Snowflake. ExtraColumns: indique que l'insertion dans Snowflake est rejetée en raison de colonnes supplémentaires dans la charge utile d'entrée. Les colonnes absentes du tableau ne doivent pas être spécifiées. Notez que les noms des colonnes Snowflake distinguent les majuscules et minuscules. Si la livraison échoue avec cette erreur alors que la colonne est présente dans la table, assurez-vous que le cas du nom de colonne dans la charge utile d'entrée correspond au nom de colonne déclaré dans la définition de la table.
- Code d'erreur : Snowflake. MissingColumns: indique que l'insertion dans Snowflake est rejetée en raison de colonnes manquantes dans la charge utile d'entrée. Assurez-vous que des valeurs sont spécifiées pour toutes les colonnes non nullables.
- Code d'erreur : Snowflake. InvalidInput: Cela peut se produire lorsque Firehose ne parvient pas à analyser la charge utile d'entrée fournie dans un format JSON valide. Assurez-vous que la charge utile JSON est bien formée, qu'elle ne contient pas de guillemets doubles, de guillemets, de caractères d'échappement, etc. Actuellement, Firehose ne prend en charge qu'un seul élément JSON comme charge utile d'enregistrement, les tableaux JSON ne sont pas pris en charge.
- Code d'erreur : Snowflake. InvalidValue: indique que la livraison a échoué en raison d'un type de données incorrect dans la charge utile d'entrée. Assurez-vous que les valeurs JSON spécifiées dans la charge utile d'entrée respectent le type de données déclaré dans la définition de la table Snowflake.
- Code d'erreur : Snowflake. InvalidTableType: indique que le type de table configuré dans le flux Firehose n'est pas pris en charge. Reportez-vous aux limitations (voir [Limitations](#)) du streaming Snowpipe pour connaître les tables, les colonnes et les types de données pris en charge.

Note

Quelle que soit la raison, si la définition de la table ou les autorisations de rôle sont modifiées sur votre destination Snowflake après la création du stream Firehose, Firehose peut mettre plusieurs minutes à détecter ces modifications. Si vous constatez des erreurs de livraison à cause de cela, essayez de supprimer et de recréer le stream Firehose.

Résolution des problèmes d'accessibilité des terminaux Firehose

Si l'API Firehose atteint un délai d'expiration, effectuez les étapes suivantes pour tester l'accessibilité des terminaux :

- Vérifiez si les demandes d'API proviennent d'un hôte dans un VPC. Tout le trafic provenant d'un VPC nécessite la configuration d'un point de terminaison Firehose VPC. Pour plus d'informations, consultez la section [Utilisation de Firehose](#) avec AWS PrivateLink
- Si le trafic provient d'un réseau public ou d'un VPC avec le point de terminaison Firehose VPC configuré dans un sous-réseau particulier, exécutez les commandes suivantes depuis l'hôte pour vérifier la connectivité réseau. Le point de terminaison Firehose se trouve dans la section Points de terminaison et quotas [Firehose](#).
- Utilisez des outils tels que traceroute ou tcping pour vérifier si la configuration du réseau est correcte. En cas d'échec, vérifiez les paramètres de votre réseau :

Par exemple :

```
traceroute firehose.us-east-2.amazonaws.com
```

or

```
tcping firehose.us-east-2.amazonaws.com 443
```

- S'il apparaît que les paramètres réseau sont corrects et que la commande suivante échoue, vérifiez si l'[autorité de certification Amazon fait partie](#) de la chaîne de confiance.

Par exemple :

```
curl firehose.us-east-2.amazonaws.com
```

Si les commandes ci-dessus aboutissent, réessayez l'API pour voir si l'API renvoie une réponse.

Dépannage des points de terminaison HTTP

Cette section décrit les étapes de résolution des problèmes courants lors de la transmission de données par Amazon Data Firehose vers des destinations de points de terminaison HTTP génériques et vers des destinations partenaires, notamment Datadog, Dynatrace, LogicMonitor MongoDB, New Relic, Splunk ou Sumo Logic. Dans la présente section, toutes les destinations applicables sont appelées points de terminaison HTTP. Assurez-vous que le rôle IAM spécifié dans votre flux Firehose peut accéder au compartiment de sauvegarde S3 et à la fonction Lambda pour la transformation des données (si la transformation des données est activée). Assurez-vous également que le rôle IAM a accès au groupe de CloudWatch journaux et aux flux de journaux pour vérifier les journaux d'erreurs. Pour plus d'informations, voir [Accorder à Firehose l'accès à une destination de point de terminaison HTTP](#).

Note

Les informations contenues dans cette section ne s'appliquent pas aux destinations suivantes : Splunk, OpenSearch Service, S3 et Redshift.

CloudWatch Journaux

Il est vivement recommandé d'activer la [CloudWatch journalisation pour Firehose](#). Les journaux ne sont publiés que lorsqu'il y a des erreurs de diffusion vers votre destination.

Exceptions de destination

ErrorCode: HttpEndpoint.DestinationException

```
{
  "deliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/ronald-test",
  "destination": "custom.firehose.endpoint.com...",
  "deliveryStreamVersionId": 1,
  "message": "The following response was received from the endpoint destination.
413: {\"requestId\": \"43b8e724-dbac-4510-adb7-ef211c6044b9\", \"timestamp\":
1598556019164, \"errorMessage\": \"Payload too large\"}",
```

```
"errorCode": "HttpEndpoint.DestinationException",
"processor": "arn:aws:lambda:us-east-1:379522611494:function:httpLambdaProcessing"
}
```

Les exceptions de destination indiquent que Firehose est capable d'établir une connexion avec votre point de terminaison et d'effectuer une requête HTTP, mais n'a pas reçu de code de réponse 200. Les réponses 2xx qui ne sont pas des 200 entraîneront également une exception de destination. Amazon Data Firehose enregistre le code de réponse et une charge utile de réponse tronquée reçus du point de terminaison configuré dans Logs. CloudWatch Dans la mesure où Amazon Data Firehose enregistre le code de réponse et la charge utile sans modification ni interprétation, il appartient au terminal de fournir la raison exacte pour laquelle il a rejeté la demande de livraison HTTP d'Amazon Data Firehose. Voici les recommandations de dépannage les plus courantes pour ces exceptions :

- 400 : indique que vous envoyez une mauvaise demande en raison d'une mauvaise configuration de votre Amazon Data Firehose. Assurez-vous que vous disposez de l'[URL](#), des [attributs communs](#), du [codage de contenu](#), de la [clé d'accès](#) et des [indices de mise en mémoire tampon](#) corrects pour votre destination. Consultez la documentation spécifique à la destination sur la configuration requise.
- 401 : Indique que la clé d'accès que vous avez configurée pour votre stream Firehose est incorrecte ou manquante.
- 403 : indique que la clé d'accès que vous avez configurée pour votre stream Firehose n'est pas autorisée à fournir des données au point de terminaison configuré.
- 413 : indique que la charge utile des demandes qu'Amazon Data Firehose envoie au point de terminaison est trop importante pour que celui-ci puisse la gérer. Essayez de [réduire l'indice de mise en mémoire tampon](#) à la taille recommandée pour votre destination.
- 429 : indique qu'Amazon Data Firehose envoie des demandes à une vitesse supérieure à celle que la destination peut gérer. Ajustez votre indice de mise en mémoire tampon en augmentant le temps de mise en mémoire tampon ou en augmentant la taille de la mémoire tampon (tout en respectant les limites de votre destination).
- 5xx : indique qu'il y a un problème avec la destination. Le service Amazon Data Firehose fonctionne toujours correctement.

⚠ Important

Important : bien qu'il s'agisse des recommandations de dépannage les plus courantes, les raisons pour lesquelles des points de terminaison spécifiques fournissent des codes de réponse peuvent être différentes et les recommandations spécifiques aux points de terminaison doivent être suivies en premier lieu.

Réponse non valide

ErrorCode: HttpEndpoint.InvalidResponseFromDestination

```
{
  "deliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/ronald-test",
  "destination": "custom.firehose.endpoint.com...",
  "deliveryStreamVersionId": 1,
  "message": "The response received from the specified endpoint is invalid. Contact the owner of the endpoint to resolve the issue. Response for request 2de9e8e9-7296-47b0-bea6-9f17b133d847 is not recognized as valid JSON or has unexpected fields. Raw response received: 200 {\"requestId\": null}\",
  "errorCode": "HttpEndpoint.InvalidResponseFromDestination",
  "processor": "arn:aws:lambda:us-east-1:379522611494:function:httpLambdaProcessing"
}
```

Les exceptions de réponse non valides indiquent qu'Amazon Data Firehose a reçu une réponse non valide de la part de la destination du point de terminaison. La réponse doit être conforme aux [spécifications de la réponse](#), sinon Amazon Data Firehose considérera la tentative de livraison comme un échec et redistribuera les mêmes données jusqu'à ce que la durée de nouvelle tentative configurée soit dépassée. Amazon Data Firehose considère les réponses qui ne respectent pas les spécifications de réponse comme des échecs, même si le statut de la réponse est de 200. Si vous développez un point de terminaison compatible avec Amazon Data Firehose, suivez les spécifications de réponse pour vous assurer que les données sont correctement livrées.

Vous trouverez ci-dessous quelques-uns des types de réponses non valides les plus courants et la manière d'y remédier :

- JSON non valide ou champs inattendus : indique que la réponse ne peut pas être correctement désérialisée en JSON ou comporte des champs inattendus. Assurez-vous que le contenu de la réponse n'est pas codé.
- RequestIdManquant : indique que la réponse ne contient pas de RequestID.
- RequestId ne correspond pas : indique que le RequestID dans la réponse ne correspond pas au RequestID sortant.
- Horodatage manquant : indique que la réponse ne contient aucun champ d'horodatage. Le champ d'horodatage doit être un nombre et non une chaîne.
- En-tête Content-Type manquant : indique que la réponse ne contient pas d'en-tête « content-type: application/json ». Aucun autre content-type n'est accepté.

Important

Important : Amazon Data Firehose ne peut fournir des données qu'aux points de terminaison conformes aux spécifications de demande et de réponse Firehose. Si vous configurez votre destination pour un service tiers, assurez-vous d'utiliser le bon point de terminaison compatible avec Amazon Data Firehose, qui sera probablement différent du point de terminaison d'ingestion public. Par exemple, le point de terminaison Amazon Data Firehose de Datadog est <https://aws-kinesis-http-intake.logs.datadoghq.com/> tandis que son point de terminaison public est <https://api.datadoghq.com/>.

Autres erreurs courantes

D'autres codes d'erreur et définitions sont répertoriés ci-dessous.

- Code d'erreur : HttpEndpoint. RequestTimeout- Indique que le terminal a mis plus de 3 minutes à répondre. Si vous êtes le propriétaire de la destination, réduisez le temps de réponse du point de terminaison de destination. Si vous n'êtes pas le propriétaire de la destination, contactez le propriétaire et demandez-lui s'il est possible de réduire le temps de réponse (par exemple, en diminuant l'indice de mise en mémoire tampon afin que moins de données soient traitées par demande).
- Code d'erreur : HttpEndpoint. ResponseTooLarge- Indique que la réponse est trop importante. La réponse doit être inférieure à 1 Mio, en-têtes compris.
- Code d'erreur : HttpEndpoint. ConnectionFailed- Indique qu'une connexion n'a pas pu être établie avec le point de terminaison configuré. Cela peut être dû à une faute de frappe dans l'URL.

configurée, au fait que le point de terminaison n'est pas accessible à Amazon Data Firehose ou au fait que le point de terminaison met trop de temps à répondre à la demande de connexion.

- Code d'erreur : `HttpEndpoint.ConnectionReset` - Indique qu'une connexion a été établie mais réinitialisée ou fermée prématurément par le terminal.
- Code d'erreur : `HttpEndpoint.SSLHandshakeFailure` - Indique qu'un handshake SSL n'a pas pu être correctement effectué avec le point de terminaison configuré.

Dépannage de MSK comme source

Cette section décrit les étapes de dépannage courantes lors de l'utilisation de MSK comme source

Note

Pour résoudre les problèmes de traitement, de transformation ou de diffusion du S3, reportez-vous aux sections précédentes.

Échec de la création de tuyaux

Vérifiez les points suivants si la création de votre tuyau avec MSK comme source échoue

- Vérifiez que le cluster MSK source est à l'état actif.
- Si vous utilisez une connectivité privée, assurez-vous que [Private Link est activé sur le cluster](#)
- Si vous utilisez une connectivité publique, assurez-vous que [l'accès public sur le cluster est activé](#)
- Si vous utilisez une connectivité privée, assurez-vous d'ajouter une [politique basée sur les ressources qui autorise Firehose à créer un Private Link](#). Reportez-vous également à : [MSK cross account permissions](#)
- Assurez-vous que le rôle dans la configuration de la source est [autorisé à ingérer les données de la rubrique du cluster](#)
- Assurez-vous que vos groupes de sécurité VPC autorisent le trafic entrant sur les [ports utilisés par les serveurs d'amorçage du cluster](#)

Tuyau suspendu

Si votre tuyau est en état de `SUSPENDED`, vérifiez les points suivants

- Vérifiez que le cluster MSK source est à l'état actif.
- Vérifiez que la rubrique source existe. Si le sujet a été supprimé et recréé, vous devrez également supprimer et recréer le stream Firehose.

Tuyau contre-pressurisé

La valeur de `DataReadFromSource .Backpressured` sera de 1 en cas de dépassement `BytesPerSecondLimit` par partition ou lorsque le flux normal de livraison est lent ou arrêté.

- Si vous touchez, `BytesPerSecondLimit` veuillez vérifier la métrique `DataReadFromSource .Bytes` et demander une augmentation de la limite.
- Consultez les CloudWatch journaux, les métriques de destination, les métriques de transformation des données et les métriques de conversion de format pour identifier les goulots d'étranglement.

Actualité des données incorrecte

L'actualité des données semble incorrecte

- Firehose calcule l'actualité des données en fonction de l'horodatage de l'enregistrement consommé. Pour vous assurer que cet horodatage est correctement enregistré lorsque l'enregistrement du producteur est conservé dans les journaux de l'agent Kafka, définissez la configuration du type d'horodatage de la rubrique Kafka sur `message.timestamp.type=LogAppendTime`

Problèmes de connexion au cluster MSK

La procédure suivante explique comment valider la connectivité aux clusters MSK. Pour en savoir plus sur la configuration du client Amazon MSK, consultez [Getting started using Amazon MSK](#) dans le guide du développeur Amazon Managed Streaming for Apache Kafka.

Pour valider la connectivité aux clusters MSK

1. Créez une instance Amazon EC2 basée sur Unix (de préférence AL2). Si seule la connectivité VPC est activée sur votre cluster, assurez-vous que votre instance EC2 s'exécute dans le même VPC. Connectez-vous à l'instance en SSH une fois qu'elle est disponible. Pour plus d'informations, consultez [ce didacticiel](#) dans le guide de l'utilisateur Amazon EC2.

2. Installez Java à l'aide du gestionnaire de packages Yum en exécutant la commande suivante. Pour plus d'informations, consultez les [instructions d'installation](#) du guide de l'utilisateur d'Amazon Corretto 8.

```
sudo yum install java-1.8.0
```

3. Installez le [AWS client](#) en exécutant la commande suivante.

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install
```

4. Téléchargez la version 2.6* du client Apache Kafka en exécutant la commande suivante.

```
wget https://archive.apache.org/dist/kafka/2.6.2/kafka_2.12-2.6.2.tgz  
tar -xzf kafka_2.12-2.6.2.tgz
```

5. Accédez au répertoire `kafka_2.12-2.6.2/libs`, puis exécutez la commande suivante pour télécharger le fichier Amazon MSK IAM JAR.

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v1.1.3/aws-msk-iam-auth-1.1.3-all.jar
```

6. Créez un `client.properties` fichier dans le dossier Kafka bin.
7. `awsRoleArn` Remplacez-le par le rôle ARN que vous avez utilisé dans votre Firehose SourceConfiguration et vérifiez l'emplacement du certificat. Autorisez votre utilisateur AWS client à assumer le rôle `awsRoleArn`. AWS L'utilisateur client tentera d'assumer le rôle que vous avez spécifié ici.

```
[ec2-user@ip-xx-xx-xx-xx bin]$ cat client.properties  
security.protocol=SASL_SSL  
sasl.mechanism=AWS_MSK_IAM  
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required  
  awsRoleArn="<role arn>" awsStsRegion="<region name>";  
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler  
awsDebugCreds=true  
ssl.truststore.location=/usr/lib/jvm/java-1.8.0-  
openjdk-1.8.0.342.b07-1.amzn2.0.1.x86_64/jre/lib/security/cacerts  
ssl.truststore.password=changeit
```

8. Exécutez la commande Kafka suivante pour répertorier les sujets. Si votre connexion est publique, utilisez les serveurs Bootstrap du point de terminaison public. Si votre connexion est privée, utilisez les serveurs Bootstrap du point de terminaison privé.

```
bin/kafka-topics.sh --list --bootstrap-server <bootstrap servers> --command-config bin/client.properties
```

Si la demande aboutit, vous devriez voir un résultat similaire à l'exemple suivant.

```
[ec2-user@ip-xx-xx-xx-xx kafka_2.12-2.6.2]$ bin/kafka-topics.sh --list --bootstrap-server <bootstrap servers> --command-config bin/client.properties

[xxxx-xx-xx 05:49:50,877] WARN The configuration 'awsDebugCreds' was supplied but isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'ssl.truststore.location' was supplied but isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'sasl.jaas.config' was supplied but isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'sasl.client.callback.handler.class' was supplied but isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'ssl.truststore.password' was supplied but isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:50:21,629] WARN [AdminClient clientId=adminclient-1] Connection to node...
__amazon_msk_canary
__consumer_offsets
```

9. Si vous rencontrez des problèmes lors de l'exécution du script précédent, vérifiez que les serveurs bootstrap que vous avez fournis sont accessibles sur le port spécifié. Pour ce faire, vous pouvez télécharger et utiliser telnet ou un utilitaire similaire, comme indiqué dans la commande suivante.

```
sudo yum install telnet
telnet <bootstrap servers><port>
```

Si la demande aboutit, vous obtiendrez le résultat suivant. Cela signifie que vous pouvez vous connecter à votre cluster MSK au sein de votre VPC local et que les serveurs bootstrap sont sains sur le port spécifié.

```
Connected to ..
```

10. Si la demande échoue, vérifiez les règles de trafic entrant sur votre groupe de sécurité [VPC](#). Par exemple, vous pouvez utiliser les propriétés suivantes sur la règle entrante.

```
Type: All traffic
Port: Port used by the bootstrap server (e.g. 14001)
Source: 0.0.0.0/0
```

Réessayez la connexion Telnet comme indiqué à l'étape précédente. [Si vous ne parvenez toujours pas à vous connecter ou si votre connexion Firehose échoue toujours, contactez le AWS support.](#)

Mesure de fraîcheur des données croissante ou non émise

La fraîcheur des données est une mesure de l'actualité de vos données dans votre flux Firehose. Il s'agit de l'âge du plus ancien enregistrement de données du flux Firehose, mesuré depuis le moment où Firehose a ingéré les données jusqu'à aujourd'hui. Firehose fournit des métriques que vous pouvez utiliser pour surveiller l'actualité des données. Pour identifier la métrique d'actualité des données pour une destination spécifique, consultez [the section called "Surveillance à l'aide de CloudWatch métriques"](#).

Si vous activez la sauvegarde pour tous les événements ou tous les documents, surveillez deux métriques d'actualité des données distinctes : l'une pour la destination principale et l'autre pour la sauvegarde.

Si la métrique de fraîcheur des données n'est pas émise, cela signifie qu'il n'y a pas de diffusion active pour le flux Firehose. Ce cas de figure se présente lorsque la livraison des données est complètement bloquée ou lorsqu'il n'y a pas de données entrantes.

Si la métrique d'actualité des données augmente constamment, cela signifie que la livraison des données prend du retard. Plusieurs raisons sont possibles.

- La destination ne parvient pas à gérer le taux de livraison. Si Firehose rencontre des erreurs transitoires en raison d'un trafic élevé, la livraison risque d'être retardée. Cela peut se produire pour des destinations autres qu'Amazon S3 (cela peut se produire pour OpenSearch Service, Amazon Redshift ou Splunk). Assurez-vous que la destination dispose de suffisamment de capacité pour gérer le trafic entrant.

- La destination est lente. La livraison des données peut prendre du retard si Firehose rencontre une latence élevée. Surveillez la métrique de latence de la destination.
- La fonction Lambda est lente. Cela peut entraîner un taux de livraison de données inférieur au taux d'ingestion de données pour le flux Firehose. Si possible, améliorez l'efficacité de la fonction Lambda. Par exemple, si la fonction effectue des E/S réseau, utilisez plusieurs threads ou des E/S asynchrones pour accroître le parallélisme. Envisagez également d'augmenter la taille de la mémoire de la fonction Lambda afin que l'allocation du CPU puisse croître en conséquence. Ces modifications peuvent entraîner des invocations Lambda plus rapides. Pour plus d'informations sur la configuration des fonctions Lambda, voir Configuration des fonctions [Lambda AWS](#).
- Des échecs affectent la livraison des données. Pour plus d'informations sur la façon de surveiller les erreurs à l'aide d'Amazon CloudWatch Logs, consultez [the section called “Surveillance à l'aide de CloudWatch journaux”](#).
- Si la source de données du flux Firehose est un flux de données Kinesis, une régulation est peut-être en cours. Vérifiez les métriques `ThrottledGetRecords`, `ThrottledGetShardIterator` et `ThrottledDescribeStream`. Si plusieurs consommateurs sont associés au flux de données Kinesis, tenez compte des points suivants :
 - Si les métriques `ThrottledGetRecords` et `ThrottledGetShardIterator` sont élevées, nous vous recommandons d'augmenter le nombre de partitions provisionnées pour le flux de données.
 - Si la `ThrottledDescribeStream` valeur est élevée, nous vous recommandons d'ajouter `kinesis:listshards` autorisation au rôle configuré dans [KinesisStreamSourceConfiguration](#).
- La mise en mémoire tampon est insuffisante pour la destination. Cela peut augmenter le nombre d'allers-retours que Firehose doit effectuer pour se rendre à destination, ce qui peut entraîner un retard de livraison. Pensez à augmenter la valeur des indicateurs de mise en mémoire tampon. Pour plus d'informations, consultez [BufferingHints](#).
- Une durée élevée configurée pour les nouvelles tentatives peut entraîner un retard de livraison lorsque les erreurs sont fréquentes. Envisagez de réduire cette durée. En outre, surveillez les erreurs et essayez de les corriger. Pour plus d'informations sur la façon de surveiller les erreurs à l'aide d'Amazon CloudWatch Logs, consultez [the section called “Surveillance à l'aide de CloudWatch journaux”](#).
- Si la destination est Splunk et que la métrique `DeliveryToSplunk.DataFreshness` est élevée, mais que `DeliveryToSplunk.Success` semble correct, le cluster Splunk est peut-être occupé. Libérez le cluster Splunk si possible. Vous pouvez également contacter le AWS Support et demander une augmentation du nombre de canaux utilisés par Firehose pour communiquer avec le cluster Splunk.

La conversion du format d'enregistrement vers Apache Parquet échoue

Cela se produit si vous prenez des données DynamoDB qui incluent Set le type, que vous les diffusez via Lambda vers un flux Firehose et que vous utilisez AWS Glue Data Catalog an pour convertir le format d'enregistrement en Apache Parquet.

Lorsque le AWS Glue robot d'exploration indexe les types de données de l'ensemble DynamoDB (`StringSet`, `etBinarySet`)`NumberSet`, il les stocke dans le catalogue de données sous `SET<STRING>` les formes, et, respectivement. `SET<BIGINT>` `SET<BINARY>` Cependant, pour que Firehose puisse convertir les enregistrements de données au format Apache Parquet, les types de données Apache Hive sont nécessaires. Les types définis n'étant pas des types de données Apache Hive valides, la conversion échoue. Pour que la conversion fonctionne, mettez à jour le catalogue de données avec les types de données Apache Hive. Pour cela, remplacez `set` par `array` dans le catalogue de données.

Pour modifier un ou plusieurs types de données de **set** à **array** dans un catalogue AWS Glue de données

1. Connectez-vous à la AWS Glue console AWS Management Console et ouvrez-la à l'[adresse](https://console.aws.amazon.com/glue/) `https://console.aws.amazon.com/glue/`.
2. Dans le volet de gauche, sous l'en-tête Data catalog (Catalogue de données) choisissez Tables.
3. Dans la liste des tables, choisissez le nom de la table dans laquelle vous devez modifier un ou plusieurs types de données. Vous accédez alors à la page des détails de la table.
4. Choisissez le bouton Modifier le schéma dans le coin supérieur droit de la page des détails.
5. Dans la colonne Data type (Type de données), choisissez le premier type de données set.
6. Dans la liste déroulante Column type (Type de colonne) remplacez le type set par array.
7. Dans le ArraySchemachamp, entrez `array<string>``array<int>`, ou `array<binary>`, selon le type de données approprié à votre scénario.
8. Choisissez Mettre à jour.
9. Répétez les étapes précédentes pour convertir d'autres types set en types array.
10. Choisissez Enregistrer.

Quota Amazon Data Firehose

Le quota d'Amazon Data Firehose est le suivant.

- Amazon MSK étant la source du flux Firehose, chaque flux Firehose possède un quota par défaut de 10 Mo/sec de débit de lecture par partition et une taille d'enregistrement maximale de 10 Mo. Vous pouvez utiliser l'[augmentation du quota de service](#) pour demander une augmentation du quota par défaut de 10 Mo/sec de débit de lecture par partition.
- Amazon MSK étant la source du flux Firehose, la taille d'enregistrement maximale est de 6 Mo si AWS Lambda est activée, et de 10 Mo si Lambda est désactivé. AWS Lambda limite son enregistrement entrant à 6 Mo, et Amazon Data Firehose transmet les enregistrements supérieurs à 6 Mo vers un compartiment S3 d'erreur. Si Lambda est désactivé, Firehose limite son enregistrement entrant à 10 Mo. Si Amazon Data Firehose reçoit une taille d'enregistrement d'Amazon MSK supérieure à 10 Mo, Amazon Data Firehose envoie cet enregistrement au compartiment d'erreur S3 et envoie les métriques Cloudwatch à votre compte. [Pour plus d'informations sur les limites AWS Lambda, consultez : https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html](#).
- Lorsque le [partitionnement dynamique](#) d'un flux Firehose est activé, un quota par défaut de 500 partitions actives peut être créé pour ce flux Firehose. Le nombre de partitions actives est le nombre total de partitions actives dans le tampon de diffusion. Par exemple, si la requête de partitionnement dynamique crée trois partitions par seconde et que vous avez une configuration d'indice de mémoire tampon qui déclenche la livraison toutes les 60 secondes, vous aurez en moyenne 180 partitions actives. Une fois que les données sont livrées dans une partition, celle-ci n'est plus active. Vous pouvez utiliser le [formulaire Amazon Data Firehose Limits](#) pour demander une augmentation de ce quota jusqu'à 5 000 partitions actives par flux Firehose donné. Si vous avez besoin de plus de partitions, vous pouvez créer davantage de flux Firehose et répartir les partitions actives entre eux.
- Lorsque le [partitionnement dynamique](#) sur un flux Firehose est activé, un débit maximum de 1 Go par seconde est pris en charge pour chaque partition active.
- Chaque compte aura le quota suivant pour le nombre de streams Firehose par région :
 - USA Est (Virginie du Nord), USA Est (Ohio), USA Ouest (Oregon), Europe (Irlande), Asie-Pacifique (Tokyo) : 5 000 streams Firehose
 - Europe (Francfort), Europe (Londres), Asie-Pacifique (Singapour), Asie-Pacifique (Sydney), Asie-Pacifique (Séoul), Asie-Pacifique (Mumbai), (États-Unis ouest), Canada AWS GovCloud (ouest), Canada (ouest), Canada (centre) : 2 000 flux Firehose

- Europe (Paris), Europe (Milan), Europe (Stockholm), Asie-Pacifique (Hong Kong), Asie-Pacifique (Osaka), Amérique du Sud (Sao Paulo), Chine (Ningxia), Chine (Pékin), Moyen-Orient (Bahreïn), AWS GovCloud (États-Unis-Est), Afrique (Le Cap) : 500 Firehose streams
- Europe (Zurich), Europe (Espagne), Asie-Pacifique (Hyderabad), Asie-Pacifique (Jakarta), Asie-Pacifique (Melbourne), Moyen-Orient (Émirats arabes unis), Israël (Tel Aviv), Canada Ouest (Calgary), Canada (Centre) : 100 flux Firehose
- Si vous dépassez ce nombre, un appel à [CreateDeliveryStream](#) entraîne une `LimitExceededException` exception. Pour augmenter ce quota, utilisez les [Service Quotas](#), s'ils sont disponibles dans votre région. Pour plus d'informations sur Service Quotas, consultez [Requesting a Quota Increase](#). Si les quotas de Service ne sont pas disponibles dans votre région, vous pouvez utiliser le [formulaire Amazon Data Firehose Limits](#) pour demander une augmentation.
- Lorsque Direct PUT est configuré comme source de données, chaque flux Firehose fournit le quota [PutRecord](#) et [PutRecordBatch](#) les demandes combinés suivants :
 - Pour USA Est (Virginie du Nord), USA Ouest (Oregon) et Europe (Irlande) : 500 000 enregistrements/seconde, 2 000 demandes/seconde et 5 Mio/seconde.
 - Pour l'est des États-Unis (Ohio), l'ouest des États-Unis (Californie du Nord), AWS GovCloud AWS GovCloud (États-Unis est), l'Asie-Pacifique (Hong Kong), l'Asie-Pacifique (Mumbai), l'Asie-Pacifique (Séoul), l'Asie-Pacifique (Singapour), la Chine (Pékin), la Chine (Ningxia), l'Asie-Pacifique (Sydney), l'Asie-Pacifique (Tokyo), le Canada (centre), le Canada ouest (Calgary), l'Europe (Francfort), l'Europe (Londres), l'Europe (Paris), Europe (Stockholm), Moyen-Orient (Bahreïn), Amérique du Sud (São Paulo), Afrique (Le Cap) et Europe (Milan) : 100 000 enregistrements/seconde, 1 000 demandes/seconde et 1 Mib/seconde.

Pour demander une augmentation du quota, utilisez le formulaire [Amazon Data Firehose Limits](#). Les trois quotas évoluent proportionnellement. Par exemple, si vous augmentez le quota de débit de USA Est (Virginie du Nord), USA Ouest (Oregon) ou Europe (Irlande) de 10 Mio/seconde, les deux autres quotas vont augmenter de 4 000 transactions/seconde et de 1 000 000 enregistrements/seconde.

Important

Si le quota augmenté est bien supérieur au trafic en cours d'exécution, cela entraîne la diffusion de petits lots aux destinations. Ce processus est inefficace et peut s'avérer coûteux pour les services de destination. Assurez-vous d'augmenter le quota appliqué de

manière à le faire correspondre au trafic en cours d'exécution et de l'augmenter en fonction du trafic.

Important

Notez que des enregistrements de données plus petits peuvent entraîner des coûts plus élevés. Le [prix d'ingestion de Firehose](#) est basé sur le nombre d'enregistrements de données que vous envoyez au service, multiplié par la taille de chaque enregistrement arrondie aux 5 Ko (5 120 octets) les plus proches. Ainsi, pour un même volume de données entrantes (octets), s'il y a un plus grand nombre d'enregistrements entrants, le coût engendré sera plus élevé. Par exemple, si le volume total de données entrantes est de 5 Mio, l'envoi de 5 Mio de données sur 5 000 enregistrements coûte plus cher que l'envoi de la même quantité de données en utilisant 1 000 enregistrements. [Pour plus d'informations, consultez Amazon Data Firehose dans le AWS calculateur.](#)

Note

Lorsque Kinesis Data Streams est configuré en tant que source de données, ce quota ne s'applique pas et Amazon Data Firehose augmente ou diminue sans limite.

- Chaque stream Firehose stocke des enregistrements de données pendant 24 heures au maximum au cas où la destination de livraison ne serait pas disponible et si la source l'était. DirectPut Si la source est Kinesis Data Streams (KDS) et que la destination n'est pas disponible, les données seront conservées en fonction de votre configuration KDS.
- La taille maximale d'un enregistrement envoyé à Amazon Data Firehose, avant le codage en base64, est de 1 000 KiB.
- L'[PutRecordBatch](#) opération peut prendre jusqu'à 500 enregistrements par appel ou 4 MiB par appel, la valeur la plus petite étant retenue. Ce quota ne peut pas être modifié.
- Les opérations suivantes peuvent fournir jusqu'à cinq invocations par seconde (limite stricte) : [CreateDeliveryStream](#), [DeleteDeliveryStream](#), [DescribeDeliveryStream](#), [ListDeliveryStreams](#), [UpdateDestination](#), [TagDeliveryStream](#), [UntagDeliveryStream](#), [ListTagsForDeliveryStream](#), [StartDeliveryStreamEncryption](#), [StopDeliveryStreamEncryption](#).
- Les intervalles de temps tampon vont de 60 à 900 secondes.

- Pour la livraison depuis Amazon Data Firehose vers Amazon Redshift, seuls les clusters Amazon Redshift accessibles au public sont pris en charge.
- La durée des nouvelles tentatives est comprise entre 0 seconde et 7 200 secondes pour Amazon Redshift OpenSearch et Service Delivery.
- Firehose prend en charge les versions 1.5, 2.3, 5.1, 5.3, 5.5, 5.6 d'Elasticsearch, ainsi que toutes les versions 6.* et 7.* et Amazon OpenSearch Service 2.x à 2.11.
- Lorsque la destination est Amazon S3, Amazon Redshift ou OpenSearch Service, Amazon Data Firehose autorise jusqu'à 5 appels Lambda en attente par partition. Pour Splunk, le quota est de 10 invocations Lambda exceptionnelles par partition.
- Vous pouvez utiliser une clé CMK de type CUSTOMER_MANAGED_CMK pour chiffrer jusqu'à 500 flux Firehose.

Annexe : spécifications des demandes et réponses de livraison de points de terminaison HTTP

Pour qu'Amazon Data Firehose puisse transmettre des données à des points de terminaison HTTP personnalisés, ces points de terminaison doivent accepter les demandes et envoyer des réponses en utilisant certains formats de demande et de réponse Amazon Data Firehose. Cette section décrit les spécifications de format des requêtes HTTP que le service Amazon Data Firehose envoie aux points de terminaison HTTP personnalisés, ainsi que les spécifications de format des réponses HTTP attendues par le service Amazon Data Firehose. Les points de terminaison HTTP disposent de 3 minutes pour répondre à une demande avant qu'Amazon Data Firehose n'expire cette demande. Amazon Data Firehose considère les réponses qui ne respectent pas le format approprié comme des échecs de livraison.

Rubriques

- [Format des demandes](#)
- [Format de la réponse](#)
- [Exemples](#)

Format des demandes

Paramètres de chemin d'accès et d'URL

Ils sont configurés directement par vous dans le cadre d'un champ d'URL unique. Amazon Data Firehose les envoie tels que configurés, sans modification. Seules les destinations HTTPS sont prises en charge. Les restrictions d'URL sont appliquées lors de la configuration du flux de diffusion.

Note

Actuellement, seul le port 443 est pris en charge pour la diffusion des données des points de terminaison HTTP.

En-têtes HTTP : X-Amz-Firehose-Protocol-Version

Cet en-tête est utilisé pour indiquer la version des formats de demande et de réponse. Actuellement, la seule version est la version 1.0.

En-têtes HTTP : X-Amz-Firehose-Request-Id

La valeur de cet en-tête est un GUID opaque qui peut être utilisé à des fins de débogage et de déduplication. Les implémentations des points de terminaison doivent enregistrer la valeur de cet en-tête si possible, à la fois pour les requêtes réussies et pour celles qui ont échoué. L'ID de la demande reste le même entre plusieurs tentatives de la même demande.

En-têtes HTTP : Content-Type

La valeur de l'en-tête Content-Type est toujours `application/json`.

En-têtes HTTP : Content-Encoding

Un flux Firehose peut être configuré pour utiliser GZIP afin de compresser le corps lors de l'envoi de requêtes. Lorsque cette compression est activée, la valeur de l'en-tête Content-Encoding est définie sur `gzip`, conformément à la pratique standard. Si la compression n'est pas activée, l'en-tête Content-Encoding est totalement absent.

En-têtes HTTP : Content-Length

Ces en-têtes sont utilisés de manière standard.

En-têtes HTTP : X-Amz-Firehose-Source-Arn :

L'ARN du flux Firehose représenté au format de chaîne ASCII. L'ARN code la région, l'ID AWS du compte et le nom du flux. Par exemple, `arn:aws:firehose:us-east-1:123456789:deliverystream/testStream`.

En-têtes HTTP : X-Amz-Firehose-Access-Key

Cet en-tête contient une clé d'API ou d'autres informations d'identification. Vous avez la possibilité de créer ou de mettre à jour la clé API (ou jeton d'autorisation) lors de la création ou de la mise à jour de votre flux de diffusion. Amazon Data Firehose limite la taille de la clé d'accès à 4 096 octets. Amazon Data Firehose n'essaie en aucun cas d'interpréter cette clé. La clé configurée est copiée exactement dans la valeur de cet en-tête.

Le contenu peut être arbitraire et peut potentiellement représenter un jeton JWT ou un `ACCESS_KEY`. Si un point de terminaison nécessite des informations d'identification à champs multiples (par exemple, le nom d'utilisateur et le mot de passe), les valeurs de tous les champs

doivent être stockées ensemble dans une seule clé d'accès dans un format que le point de terminaison comprend (JSON ou CSV). Ce champ peut être codé en base 64 si le contenu d'origine est binaire. Amazon Data Firehose ne modifie et/ou ne code pas la valeur configurée et utilise le contenu tel quel.

En-têtes HTTP : X-Amz-Firehose-Common-Attributes

Cet en-tête contient les attributs communs (métadonnées) relatifs à l'ensemble de la demande ou à tous les enregistrements de la demande. Vous les configurez directement lors de la création d'un stream Firehose. La valeur de cet attribut est codée sous la forme d'un objet JSON avec le schéma suivant :

```
"$schema": http://json-schema.org/draft-07/schema#

properties:
  commonAttributes:
    type: object
    minProperties: 0
    maxProperties: 50
    patternProperties:
      "^.{1,256}$":
        type: string
        minLength: 0
        maxLength: 1024
```

Voici un exemple :

```
"commonAttributes": {
  "deployment -context": "pre-prod-gamma",
  "device-types": ""
}
```

Corps : taille maximale

Vous définissez vous-même la taille maximale du corps, qui peut aller jusqu'à 64 Mio, avant compression.

Corps : schéma

Le corps contient un seul document JSON avec le schéma JSON suivant (écrit en YAML) :

```
"$schema": http://json-schema.org/draft-07/schema#

title: FirehoseCustomHttpsEndpointRequest
description: >
  The request body that the Firehose service sends to
  custom HTTPS endpoints.
type: object
properties:
  requestId:
    description: >
      Same as the value in the X-Amz-Firehose-Request-Id header,
      duplicated here for convenience.
    type: string
  timestamp:
    description: >
      The timestamp (milliseconds since epoch) at which the Firehose
      server generated this request.
    type: integer
  records:
    description: >
      The actual records of the Firehose stream, carrying
      the customer data.
    type: array
    minItems: 1
    maxItems: 10000
    items:
      type: object
      properties:
        data:
          description: >
            The data of this record, in Base64. Note that empty
            records are permitted in Firehose. The maximum allowed
            size of the data, before Base64 encoding, is 1024000
            bytes; the maximum length of this field is therefore
            1365336 chars.
          type: string
          minLength: 0
          maxLength: 1365336

required:
  - requestId
  - records
```

Voici un exemple :

```
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090901599
  "records": [
    {
      "data": "aGVsbG8="
    },
    {
      "data": "aGVsbG8gd29ybGQ="
    }
  ]
}
```

Format de la réponse

Comportement par défaut en cas d'erreur

Si une réponse n'est pas conforme aux exigences ci-dessous, le serveur Firehose la traite comme si elle avait un code d'état 500 sans corps.

Code d'état

Le code d'état HTTP DOIT être dans la plage 2XX, 4XX ou 5XX.

Le serveur Amazon Data Firehose ne suit PAS les redirections (codes d'état 3XX). Seul le code de réponse 200 est considéré comme une diffusion réussie des enregistrements vers HTTP/EP. Le code de réponse 413 (taille dépassée) est considéré comme une défaillance permanente et le lot d'enregistrements n'est pas envoyé au compartiment d'erreurs s'il est configuré. Tous les autres codes de réponse sont considérés comme des erreurs réitérables et sont soumis à l'algorithme de tentative de retardement expliqué plus loin.

En-têtes HTTP : type de contenu

Le seul type de contenu acceptable est application/json.

En-têtes HTTP : Content-Encoding

Content-Encoding NE DOIT PAS être utilisé. Le corps DOIT être décompressé.

En-têtes HTTP : Content-Length

L'en-tête Content-Length DOIT être présent si la réponse a un corps.

Corps : taille maximale

La taille du corps de la réponse doit être inférieure ou égale à 1 Mio.

```
"$schema": http://json-schema.org/draft-07/schema#

title: FirehoseCustomHttpsEndpointResponse

description: >
  The response body that the Firehose service sends to
  custom HTTPS endpoints.
type: object
properties:
  requestId:
    description: >
      Must match the requestId in the request.
    type: string

  timestamp:
    description: >
      The timestamp (milliseconds since epoch) at which the
      server processed this request.
    type: integer

  errorMessage:
    description: >
      For failed requests, a message explaining the failure.
      If a request fails after exhausting all retries, the last
      Instance of the error message is copied to error output
      S3 bucket if configured.
    type: string
    minLength: 0
    maxLength: 8192
required:
  - requestId
  - timestamp
```

Voici un exemple :

```
Failure Case (HTTP Response Code 4xx or 5xx)
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": "1578090903599",
  "errorMessage": "Unable to deliver records due to unknown error."
}
Success case (HTTP Response Code 200)
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090903599
}
```

Gestion des réponses d'erreur

Dans tous les cas d'erreur, le serveur Amazon Data Firehose tente à nouveau de livrer le même lot d'enregistrements à l'aide d'un algorithme de réduction exponentielle. Les nouvelles tentatives sont annulées en utilisant un temps d'attente initial (1 seconde) avec un facteur de gigue de (15 %) et chaque nouvelle tentative suivante est annulée en utilisant la formule ($\text{initial-backoff-time} * (\text{multiplicateur} (2) ^ \text{retry_count})$) avec une instabilité accrue. Le temps de retard est limité à un intervalle maximal de deux minutes. Par exemple, lors de la neuvième tentative, le temps de retour est = $\text{MAX}(120, 2^n) * \text{random}(0,85, 1,15)$.

Les paramètres spécifiés dans l'équation précédente sont susceptibles d'être modifiés. Reportez-vous à la documentation de AWS Firehose pour connaître le temps d'arrêt initial exact, le temps d'arrêt maximal, le multiplicateur et les pourcentages de gigue utilisés dans l'algorithme de ralentissement exponentiel.

À chaque nouvelle tentative suivante, la clé d'accès et/ou la destination vers laquelle les enregistrements sont livrés peuvent changer en fonction de la configuration mise à jour du flux Firehose. Le service Amazon Data Firehose utilise le même identifiant de demande lors de chaque nouvelle tentative, de la manière la plus efficace possible. Cette dernière fonctionnalité peut être utilisée à des fins de déduplication par le serveur de point de terminaison HTTP. Si la demande n'est toujours pas livrée après le délai maximum autorisé (selon la configuration du flux

Firehose), le lot d'enregistrements peut éventuellement être envoyé vers un compartiment d'erreur basé sur la configuration du flux.

Exemples

Exemple de demande sourcée de CWLog :

```
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090901599,
  "records": [
    {
      "data": {
        "messageType": "DATA_MESSAGE",
        "owner": "123456789012",
        "logGroup": "log_group_name",
        "logStream": "log_stream_name",
        "subscriptionFilters": [
          "subscription_filter_name"
        ],
        "logEvents": [
          {
            "id": "01234567890123456789012345678901234567890123456789012345",
            "timestamp": 1510109208016,
            "message": "log message 1"
          },
          {
            "id": "01234567890123456789012345678901234567890123456789012345",
            "timestamp": 1510109208017,
            "message": "log message 2"
          }
        ]
      }
    }
  ]
}
```

Historique du document

Le tableau suivant décrit les modifications importantes apportées à la documentation Amazon Data Firehose.

Modification	Description	Date de modification
Snowflake comme destination dans de nouvelles régions	Snowflake est désormais disponible en tant que destination en Asie-Pacifique (Singapour), en Asie-Pacifique (Séoul) et en Asie-Pacifique (Sydney). Consultez the section called “Configurer les paramètres de destination pour Snowflake” .	19 juin 2024
Amazon Data Firehose s'intègre à AWS Secrets Manager	Vous pouvez désormais accéder à vos secrets et automatiser la rotation des identifiants en toute sécurité avec Secrets Manager. Consultez the section called “Authentifiez-vous avec AWS Secrets Manager” .	6 juin 2024
Ajout de la prise en charge de l'ingestion des journaux pour Dynatrace	Vous pouvez désormais envoyer des journaux et des événements à Dynatrace pour une analyse plus approfondie. Consultez the section called “Configuration des paramètres de destination pour Dynatrace” .	18 avril 2024
Version de disponibilité générale (GA) pour Snowflake en tant que destination	Snowflake est désormais généralement disponible en tant que destination. veuillez consulter the section called “Configurer les paramètres de destination pour Snowflake” .	17 avril 2024
Amazon Kinesis Data Firehose est désormais connu sous le nom d'Amazon Data Firehose	Amazon Kinesis Data Firehose a été renommé Amazon Data Firehose. Consultez Qu'est-ce qu'Amazon Data Firehose ? .	9 février 2024

Modification	Description	Date de modification
Ajout de Snowflake comme destination (aperçu public)	Vous pouvez créer un stream Firehose avec Snowflake comme destination. veuillez consulter the section called “Configurer les paramètres de destination pour Snowflake” .	19 janvier 2024
Ajout de la décompression automatique des journaux CloudWatch	Vous pouvez activer la décompression sur des flux nouveaux ou existants pour envoyer les données des CloudWatch journaux décompressés aux destinations Firehose. veuillez consulter the section called “Écrire à l'aide de CloudWatch journaux” .	15 décembre 2023
Ajout de Splunk Observability Cloud comme destination	Vous pouvez créer un stream Firehose avec Splunk Observability Cloud comme destination. veuillez consulter the section called “Configurer les paramètres de destination pour Splunk Observability Cloud” .	3 octobre 2023
Ajout d'Amazon Managed Streaming pour Apache Kafka en tant que source de données	Vous pouvez désormais configurer Amazon MSK pour envoyer des informations à un flux Firehose. veuillez consulter the section called “Écrire à l'aide d'Amazon MSK” .	26 septembre 2023
Ajout de la prise en charge du type DocumentID pour la destination du service OpenSearch	Si OpenSearch Service est la destination de votre stream Firehose, le type DocumentID indique la méthode de configuration de l'ID du document. Les méthodes prises en charge sont l'ID de document généré par Firehose et l'ID de document généré par le OpenSearch Service. veuillez consulter the section called “Configuration des paramètres de destination” .	10 mai 2023
Ajout de la prise en charge du partitionnement dynamique	Ajout de la prise en charge du partitionnement dynamique continu des données de streaming dans Amazon Data Firehose. veuillez consulter Partitionnement dynamique .	31 août 2021

Modification	Description	Date de modification
Ajout d'une rubrique sur les préfixes personnalisés.	Ajout d'une rubrique sur les expressions que vous pouvez utiliser lors de la création d'un préfixe personnalisé pour les données livrées à Amazon S3. veuillez consulter Préfixes Amazon S3 personnalisés .	20 décembre 2018
Ajout d'un nouveau didacticiel Amazon Data Firehose	Ajout d'un didacticiel expliquant comment envoyer les journaux de flux Amazon VPC à Splunk via Amazon Data Firehose. veuillez consulter Tutoriel : Ingérer les journaux de flux VPC dans Splunk à l'aide d'Amazon Data Firehose .	30 octobre 2018
Ajout de quatre nouvelles régions Amazon Data Firehose	Ajout de Paris, Mumbai, Sao Paulo et Londres. Pour plus d'informations, consultez Quota Amazon Data Firehose .	27 juin 2018
Ajout de deux nouvelles régions Amazon Data Firehose	Ajout de Séoul et Montréal. Pour plus d'informations, consultez Quota Amazon Data Firehose .	13 juin 2018
Nouvelle fonctionnalité : flux Kinesis en tant que source	Kinesis Streams a été ajouté en tant que source potentielle d'enregistrements pour un stream Firehose. Pour plus d'informations, consultez Configuration de la source et de la destination .	18 août 2017
Mise à jour de la documentation de la console	L'assistant de création de streams Firehose a été mis à jour. Pour plus d'informations, consultez Création d'un stream Firehose .	19 juillet 2017
Transformation de nouvelles données	Vous pouvez configurer Amazon Data Firehose pour transformer vos données avant leur livraison. Pour plus d'informations, consultez Transformation des données avec Amazon Data Firehose .	19 décembre 2016

Modification	Description	Date de modification
Nouvelle tentative de la commande COPY Amazon Redshift	Vous pouvez configurer Amazon Data Firehose pour réessayer une commande COPY sur votre cluster Amazon Redshift en cas d'échec. Pour plus d'informations, consultez Création d'un stream Firehose , Comprendre la livraison des données Amazon Data Firehose et Quota Amazon Data Firehose .	18 mai 2016
Nouvelle destination Amazon Data Firehose, Amazon Service OpenSearch	Vous pouvez créer un flux Firehose avec Amazon OpenSearch Service comme destination. Pour plus d'informations, consultez Création d'un stream Firehose , Comprendre la livraison des données Amazon Data Firehose et Accorder à Amazon Data Firehose l'accès à une destination de service public OpenSearch .	19 avril 2016
Nouvelles fonctionnalités de CloudWatch mesure et de résolution des problèmes améliorées	Surveillance d'Amazon Data Firehose mis à jour et Résolution des problèmes liés à Amazon Data Firehose .	19 avril 2016
Nouvel agent Kinesis amélioré	Mis à jour Écrire sur Amazon Data Firehose à l'aide de Kinesis Agent .	11 avril 2016
Nouveaux agents Kinesis	Ajouté Écrire sur Amazon Data Firehose à l'aide de Kinesis Agent .	2 octobre 2015
Première version	Première publication du guide du développeur Amazon Data Firehose.	4 octobre 2015

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le [Glossaire AWS](#) dans la Référence Glossaire AWS.