aws

Guide du développeur

Amazon Data Firehose

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

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
Travailler avec AWS SDKs 4
Complétez les prérequis pour configurer Firehose 6
Inscrivez-vous pour AWS 6
(Facultatif) Téléchargez des bibliothèques et des outils 6
Tutoriel : Création d'un stream Firehose 8
Choisissez la source et la destination de votre stream Firehose 8
Configuration des paramètres de source 10
Configurer les paramètres de source pour Amazon MSK 11
Configuration des paramètres de source pour Amazon Kinesis Data Streams 12
(Facultatif) Configurer la transformation des enregistrements et la conversion des formats 14
Configuration des paramètres de destination 16
Configurer les paramètres de destination pour Amazon S3 17
Configuration des paramètres de destination pour les tables Apache Iceberg 21
Configuration des paramètres de destination pour Amazon Redshift 21
Configurer les paramètres de destination pour le OpenSearch service 28
Configuration des paramètres de destination pour OpenSearch Serverless 30
Configurer les paramètres de destination pour HTTP Endpoint 31
Configurer les paramètres de destination pour Datadog 34
Configurer les paramètres de destination pour Honeycomb 36
Configuration des paramètres de destination pour Coralogix 38
Configuration des paramètres de destination pour Dynatrace 40
Configurer les paramètres de destination pour LogicMonitor 42
Configurer les paramètres de destination pour Logz.io 44
Configurer les paramètres de destination pour MongoDB Cloud 46
Configurer les paramètres de destination pour New Relic 48
Configurer les paramètres de destination pour Snowflake 50
Configurer les paramètres de destination pour Splunk 53
Configurer les paramètres de destination pour Splunk Observability Cloud 55
Configuration des paramètres de destination pour Sumo Logic 57
Configuration des paramètres de destination pour Elastic 58

Configuration des paramètres de sauvegarde ... 60
Configurer les conseils de mise en mémoire tampon ... 62
Configurer les paramètres avancés ... 65
Testez votre stream Firehose ... 68
Prérequis ... 68
Testez avec Amazon S3 ... 68
Testez avec Amazon Redshift ... 69
Tester avec OpenSearch Service ... 69
Testez avec Splunk ... 70
Tester avec les tables Apache Iceberg ... 71
Envoyer des données vers un flux Firehose ... 72
Configurer l'agent Kinesis pour envoyer des données ... 72
Prérequis ... 73
Gérer AWS informations d'identification ... 73
Créez des fournisseurs d'informations d'identification personnalisés ... 74
Téléchargez et installez l'agent ... 75
Configuration et démarrage de l'agent ... 77
Spécifier les paramètres de configuration de l'agent ... 78
Configuration de plusieurs répertoires de fichiers et de flux ... 82
Pré-traiter les données avec des agents ... 83
Utiliser les CLI commandes courantes de l'agent ... 88
Résoudre les problèmes liés à l'envoi depuis Kinesis Agent ... 88
Envoyer des données avec AWS SDK ... 90
Opérations d'écriture uniques utilisant PutRecord ... 91
Opérations d'écriture par lots à l'aide de PutRecordBatch ... 91
Envoyer CloudWatch les logs à Firehose ... 92
Décompressez les journaux CloudWatch ... 92
Extraire le message après décompression des journaux CloudWatch ... 93
Activer la décompression sur un nouveau stream Firehose depuis la console ... 94
Activer la décompression sur un flux Firehose existant ... 95
Désactiver la décompression sur le stream Firehose ... 96
Résoudre les problèmes de décompression dans Firehose ... 96
Envoyer CloudWatch des événements à Firehose ... 97
Configuration AWS IoT pour envoyer des données à Firehose ... 98
Transformer les données sources ... 99
Comprendre le flux de transformation des données ... 99

Durée d'invocation Lambda 100
Paramètres requis pour la transformation des données 100
Blueprints Lambda pris en charge 101
Gérer les défaillances liées à la transformation des données 103
Backup source records 104
Partitionner les données de streaming 105
Activer le partitionnement dynamique 106
Comprendre les clés de partitionnement 106
Création de clés de partitionnement avec analyse syntaxique intégrée 107
Création de clés de partitionnement avec une fonction AWS Lambda 108
Utiliser le préfixe du compartiment Amazon S3 pour fournir des données 111
Ajoutez un nouveau délimiteur de ligne lors de la livraison de données vers Amazon S3 113
Appliquer le partitionnement dynamique aux données agrégées 113
Résoudre les erreurs de partitionnement dynamique 114
Données de mémoire tampon pour le partitionnement dynamique 115
Convertir le format des données d'entrée 117
Deserializer 117
Schema 119
Serializer 119
Activer la conversion du format d'enregistrement 120
Activer la conversion du format d'enregistrement depuis la console 120
Gérez la conversion des formats d'enregistrement depuis Firehose API 121
Gestion des erreurs lors de la conversion des formats de données 121
Comprendre la livraison des données 123
Comprenez la livraison entre AWS les comptes et les régions 126
Comprendre les spécifications relatives aux demandes de livraison et aux réponses des HTTP terminaux 126
Format des demandes 126
Format de la réponse 130
Exemples 133
Gérez les défaillances de livraison de données 134
Amazon S3 134
Amazon Redshift 134
Amazon OpenSearch Service et OpenSearch Serverless 135
Splunk 136
HTTPdestination du terminal 137

Snowflake 138
Configurer le format du nom d'objet Amazon S3 139
Comprendre les préfixes personnalisés pour les objets Amazon S3 148
Configurer la rotation de l'index pour le OpenSearch service 153
Suspendre et reprendre la livraison des données 154
Suspendre un stream Firehose 154
Reprendre un stream Firehose 155
Tables Apache Iceberg 156
Considérations et limites 156
Création de tables Apache Iceberg 158
Configurer le stream Firehose 159
Conditions requises pour utiliser les tables Apache Iceberg comme destination 159
Configuration de la source et de la destination 159
Configuration de la transformation des données 160
Catalogue de données Connect 160
Configurer des clés uniques 160
Spécifier la durée de la nouvelle tentative 161
Gérer les échecs de livraison ou de traitement 161
Configurer les indices de mémoire tampon 161
Configurer les paramètres avancés 161
Résoudre les erreurs 162
Surveiller les métriques 164
Formater les enregistrements d'entrée 165
Comprendre les types de données pris en charge 169
Exemples de types de données 169
Marquer un stream Firehose 175
Comprendre les principes de base des tags 175
Suivez les coûts grâce au balisage 176
Connaître les restrictions relatives aux tags 177
Sécurité 178
Protection des données 179
Chiffrement côté serveur avec Kinesis Data Streams 179
Chiffrement côté serveur avec Direct PUT ou d'autres sources de données 179
Contrôle de l'accès 181
Accordez l'accès à vos ressources Firehose 182
Accordez à Firehose l'accès à votre cluster Amazon privé MSK 182

Permettre à Firehose d'assumer un rôle IAM ... 183
Accordez à Firehose l'accès à AWS Glue pour la conversion de formats de données ... 185
Accorder à Firehose l'accès à une destination Amazon S3 ... 186
Accorder à Firehose l'accès à une destination Apache Iceberg Tables ... 189
Accorder à Firehose l'accès à une destination Amazon Redshift ... 192
Accorder à Firehose l'accès à une destination de service public OpenSearch ... 197
Accorder à Firehose l'accès à une destination OpenSearch de service dans un VPC ... 200
Accorder à Firehose l'accès à une destination publique sans serveur OpenSearch ... 202
Accordez à Firehose l'accès à une destination OpenSearch sans serveur dans un VPC ... 205
Accorder à Firehose l'accès à une destination Splunk ... 206
Accès à Splunk dans VPC ... 208
Tutoriel : Ingérer les journaux de VPC flux dans Splunk à l'aide d'Amazon Data Firehose ... 211
Accès à Snowflake ou au point de terminaison HTTP ... 211
Accordez à Firehose l'accès à une destination Snowflake ... 211
Accès à Snowflake dans VPC ... 213
Accorder à Firehose l'accès à une destination de point de terminaison HTTP ... 217
Livraison entre comptes depuis Amazon MSK ... 220
Livraison entre comptes vers une destination Amazon S3 ... 223
Livraison entre comptes vers une destination OpenSearch de service ... 224
Utilisation de balises pour contrôler l'accès ... 226
Authentifiez-vous avec AWS Secrets Manager ... 228
Comprenez les secrets ... 229
Créer un secret ... 230
Utilisez le secret ... 230
Faites pivoter le secret ... 232
Gérer les IAM rôles via la console ... 232
Choisissez un IAM rôle existant ... 233
Création d'un nouveau IAM rôle depuis la console ... 233
Modifier IAM le rôle depuis la console ... 235
Validation de conformité ... 237
Résilience ... 237
Reprise après sinistre ... 238
Comprendre la sécurité de l'infrastructure ... 238
Utiliser Firehose avec AWS PrivateLink ... 239
Mettre en œuvre les meilleures pratiques de sécurité ... 244
Implémentation d'un accès sur la base du moindre privilège ... 244

Utiliser IAM les rôles 244
Implémenter le chiffrement côté serveur dans les ressources dépendantes 245
CloudTrail À utiliser pour surveiller les API appels 245
Surveillez Amazon Data Firehose 246
Mettre en œuvre les meilleures pratiques en matière d' CloudWatch alarmes 246
Surveillance à l'aide de CloudWatch métriques 247
CloudWatch métriques pour le partitionnement dynamique 248
CloudWatch métriques pour la livraison des données 249
Métriques d'ingestion de données 261
APImétriques au niveau des -niveaux CloudWatch 268
CloudWatch Métriques de transformation des données 271
CloudWatch Métriques de décompression des journaux 271
CloudWatch Métriques de conversion de formats 272
Métriques de chiffrement côté serveur () SSE CloudWatch 273
Dimensions pour Amazon Data Firehose 273
Métriques d'utilisation d'Amazon Data Firehose 274
CloudWatch Mesures d'accès pour Amazon Data Firehose 275
Moniteur avec CloudWatch journaux 276
Erreurs de livraison de données 277
CloudWatch Journaux d'accès pour Amazon Data Firehose 314
Surveillez l'état de santé des agents 315
Moniteur avec CloudWatch 316
Enregistrez les appels Firehose API 316
Informations sur Firehose dans CloudTrail 317
Exemple : entrées du fichier journal Firehose 318
Exemples de code 324
Principes de base 324
Actions 325
Scénarios 332
Transférez des enregistrements à Firehose 332
Résoudre les erreurs 340
Problèmes courants 340
Stream Firehose non disponible 341
Aucune donnée à destination 341
Mesure de fraîcheur des données croissante ou non émise 341
La conversion du format d'enregistrement vers Apache Parquet échoue 343

Résolutions des problèmes liés à Amazon S3 344
Dépannage d'Amazon Redshift 345
Résolution des problèmes liés à Amazon OpenSearch Service 346
Dépannage de Splunk 347
Résolution des problèmes liés à Snowflake 349
La création du stream Firehose échoue 349
Résolution des problèmes d'accessibilité des terminaux Firehose 351
Résolution des problèmes liés aux HTTP terminaux 352
CloudWatch Journaux 352
Résolution des problèmes MSK en tant que source 356
Échec de la création de tuyaux 356
Tuyau suspendu 356
Tuyau contre-pressurisé 357
Actualité des données incorrecte 357
MSKproblèmes de connexion au cluster 357
Quota 361
Historique du document 365

La diffusion de flux Amazon Data Firehose vers les tables Apache Iceberg dans Amazon S3 est en version préliminaire et peut faire l'objet de modifications.

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 à des destinations telles qu'Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon Service, OpenSearch Amazon Serverless, Splunk, Apache Iceberg Tables, ainsi qu'à HTTP tout point de HTTP terminaison personnalisé appartenant à des fournisseurs de services tiers pris en charge, notamment Datadog, LogicMonitor Dynatrace, MongoDB, New c, Coralogix 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 plus d'informations, consultez Tutoriel : Création d'un stream Firehose depuis la console et Envoyer des données vers un flux Firehose.

Enregistrer

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

Producteur de données

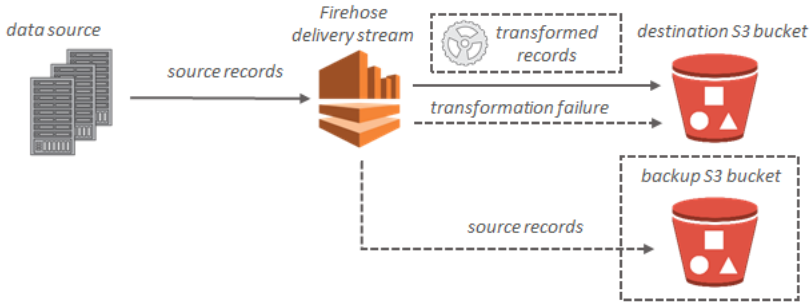
Les producteurs envoient leurs disques aux streams Firehose. Par exemple, un serveur Web qui envoie des données de journal à un flux Firehose est un producteur de données. Vous pouvez également configurer votre flux Firehose pour lire automatiquement les données d'un flux de données Kinesis existant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

Taille de la mémoire tampon et intervalle entre la mémoire tampon

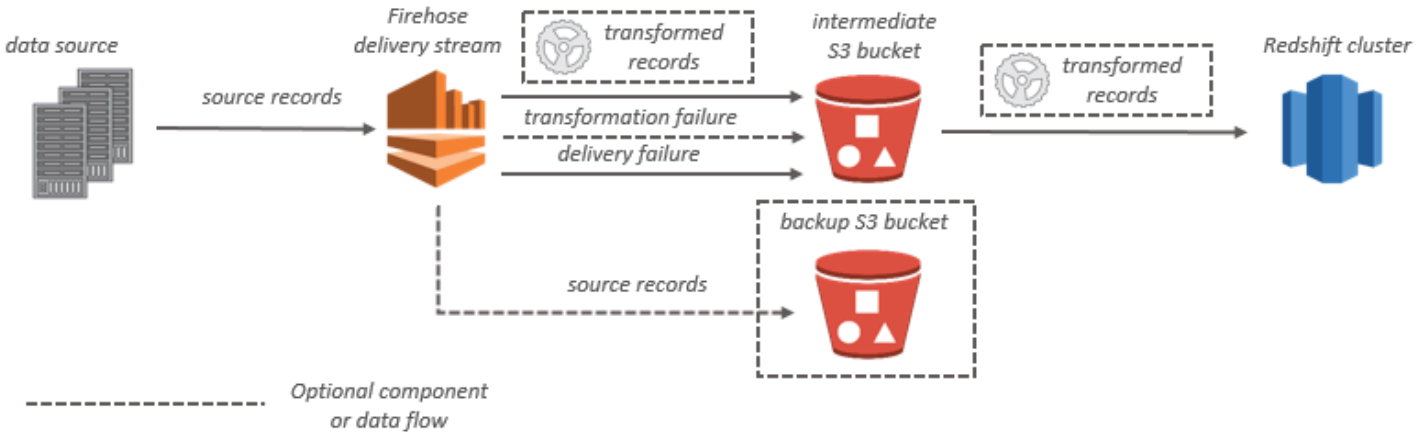
Amazon Data Firehose met en mémoire tampon les données de streaming entrantes jusqu'à une certaine taille ou pendant une certaine période avant de les diffuser vers les destinations. Buffer Sizeest en MBs et Buffer Interval est en secondes.

Comprendre le flux de données dans Amazon Data Firehose

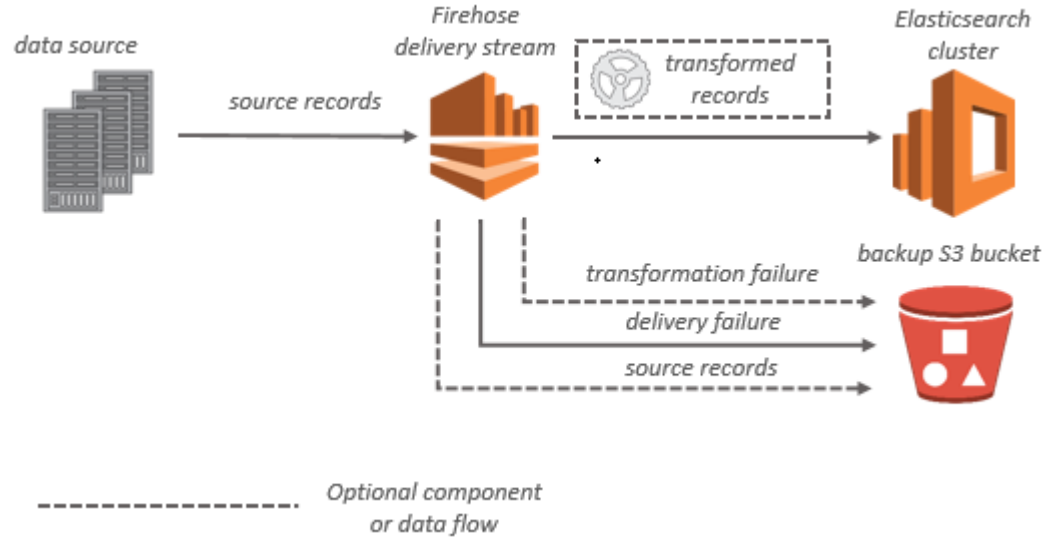
Pour les destinations Amazon S3, les données de streaming sont délivrées à votre compartiment S3. Si la transformation de données est activée, vous pouvez éventuellement sauvegarder les données source dans un autre compartiment Amazon S3.



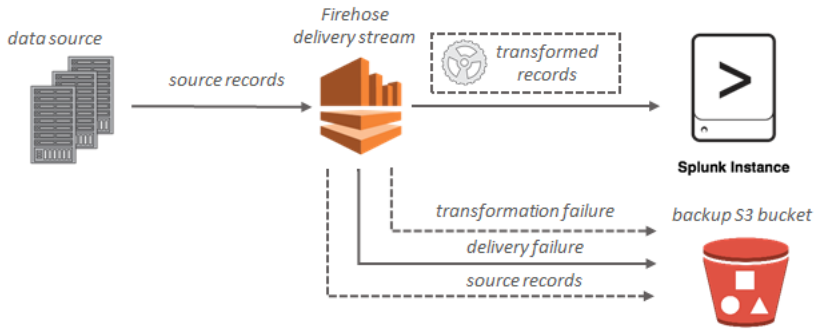
Pour les destinations Amazon Redshift, les données de streaming sont d'abord délivrées à votre compartiment S3. Amazon Data Firehose émet ensuite une commande Amazon COPY Redshift pour charger les données de votre compartiment S3 vers votre cluster Amazon Redshift. Si la transformation de données est 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.



Utiliser Firehose avec un AWS SDK

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chacun SDK fournit des exemples de code et de la documentation qui permettent aux développeurs de créer plus facilement des applications dans leur langage préféré. API

SDKdocumentation	Exemples de code
AWS SDK for C++	AWS SDK for C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK for Go	AWS SDK for Go exemples de code
AWS SDK for Java	AWS SDK for Java exemples de code
AWS SDK for JavaScript	AWS SDK for JavaScript exemples de code
Kit AWS SDK pour Kotlin	Kit AWS SDK pour Kotlin exemples de code
AWS SDK for .NET	AWS SDK for .NET exemples de code
AWS SDK for PHP	AWS SDK for PHP exemples de code
AWS Tools for PowerShell	Outils pour des exemples PowerShell de code
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemples de code

SDKdocumentation	Exemples de code
AWS SDK for Ruby	AWS SDK for Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Exemple de disponibilité

Vous n’avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Provide feedback (Fournir un commentaire) en bas de cette page.

Conditions préalables complètes pour configurer 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 le compte est automatiquement ouvert pour tous les services de AWS, y compris Amazon Data Firehose. Seuls les services que vous utilisez vous sont facturés.

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

Pour vous inscrire à un AWS compte

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
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, une Utilisateur racine d'un compte AWSest créé. L'utilisateur root a accès à tous Services AWS et les ressources du compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

(Facultatif) Téléchargez des bibliothèques et des outils

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

- Les opérations Firehose constituent l'ensemble d'APIopérations de base pris en charge par Amazon Data Firehose.
- Le AWS SDKspour Go, Java,. NET, Node.js, Python et Ruby incluent le support et des exemples d'Amazon Data Firehose.

  Si votre version du AWS SDK for Java n'inclut pas d'exemples pour Amazon Data Firehose, vous pouvez également télécharger la dernière version AWS SDKà partir de GitHub.
- L'interface AWS Command Line Interfaceprend en charge Amazon Data Firehose. Le AWS CLI vous permet de contrôler plusieurs AWS services depuis la ligne de commande et automatisez-les via des scripts.

Tutoriel : Création d'un stream Firehose depuis la console

Vous pouvez utiliser le plugin AWS Management Console ou un AWS SDKpour créer un stream 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 de plus amples informations, veuillez consulter Configurer le format du nom d'objet Amazon S3.

Suivez les étapes décrites dans les rubriques suivantes pour créer un stream Firehose.

Rubriques

- Choisissez la source et la destination de votre stream Firehose
- Configuration des paramètres de source
- (Facultatif) Configurer la transformation des enregistrements et la conversion des formats
- Configuration des paramètres de destination
- Configuration des paramètres de sauvegarde
- Configurer les paramètres avancés

Choisissez la source et la destination de votre stream Firehose

1. Ouvrez la console Firehose à l'adresse. https://console.aws.amazon.com/firehose/
2. Choisissez Create Firehose stream.
3. Sur la page Create Firehose stream, choisissez une source pour votre stream Firehose parmi les options suivantes.

    - Direct PUT — Choisissez cette option pour créer un flux Firehose dans lequel les applications du producteur peuvent écrire directement. Voici certains des AWS services, agents et services open source qui s'intègrent à Direct PUT dans Amazon Data Firehose.
        - AWS SDK

- AWS Lambda
- AWS CloudWatch Journaux
- AWS CloudWatch Évènements
- AWS Streams métriques dans le cloud
- AWS IOT
- AWS Eventbridge
- Amazon Simple Email Service
- Amazon SNS
- AWS WAFACLjournaux Web
- Amazon API Gateway - Journaux d'accès
- Amazon Pinpoint
- Journaux Amazon MSK Broker
- 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

- Amazon Kinesis Data Streams : 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 sur l'utilisation de Kinesis Data Streams comme source de données, consultez la section Envoi de données vers un flux Firehose avec 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 MSK cluster Amazon existant et les charger dans des compartiments S3

spécifiés. Pour plus d'informations, consultez Envoyer des données vers un flux Firehose avec Amazon. MSK

4. Choisissez une destination pour votre flux Firehose à partir de l'une des destinations suivantes prises en charge par Firehose.

   - Amazon OpenSearch Service
   - Amazon OpenSearch sans serveur
   - Amazon Redshift
   - Amazon S3
   - Tables Apache Iceberg
   - Coralogix
   - Datadog
   - Dynatrace
   - Elasticité
   - HTTPPoint final
   - Honeycomb
   - Logic Monitor
   - Logz.io
   - MongoDB Cloud
   - New Relic
   - Splunk
   - Splunk Observability Cloud
   - Sumo Logic
   - Snowflake

5. Pour le nom du flux Firehose, vous pouvez soit utiliser le nom que la console génère pour vous, soit ajouter le flux Firehose de votre choix.

Configuration des paramètres de source

Vous pouvez configurer les paramètres de source en fonction de la source que vous choisissez pour envoyer des informations à un flux Firehose depuis la console. Vous pouvez configurer les

paramètres de source pour Amazon MSK et Amazon Kinesis Data Streams en tant que source. Aucun paramètre de source n'est disponible pour Direct PUT en tant que source.

Configurer les paramètres de source pour Amazon MSK

Lorsque vous choisissez Amazon MSK pour envoyer des informations à un flux Firehose, vous pouvez choisir entre des clusters MSK provisionnés et MSK des clusters sans serveur. Vous pouvez ensuite utiliser Firehose pour lire facilement les données d'un MSK cluster et d'un sujet Amazon spécifiques et les charger dans la destination S3 spécifiée.

Dans la section Paramètres source de la page, indiquez des valeurs pour les champs suivants.

Connectivité au MSK cluster Amazon

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 de AWS, tandis que les courtiers privés sont destinés à être accessibles de l'intérieur AWS. Pour plus d'informations sur AmazonMSK, consultez Amazon Managed Streaming for Apache Kafka.

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

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

Pour se connecter à un MSK cluster Amazon provisionné via des courtiers Bootstrap publics, le cluster doit répondre à toutes les exigences suivantes.

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

MSKcompte de cluster

Vous pouvez choisir le compte sur lequel réside le MSK cluster Amazon. Il peut s'agir de l'une des options suivantes.

- Compte courant : vous permet d'ingérer les données d'un MSK cluster dans le courant AWS . Pour cela, vous devez spécifier le ARN MSK cluster Amazon à partir duquel votre flux Firehose lira les données.
- Compte croisé : vous permet d'ingérer des données d'un MSK cluster dans un autre AWS . Pour de plus amples informations, veuillez consulter Livraison entre comptes depuis Amazon MSK.

Rubrique

Spécifiez le sujet Apache Kafka à partir duquel vous souhaitez que votre flux Firehose ingère des données. Vous ne pouvez pas mettre à jour cette rubrique une fois la création du stream Firehose terminée.

Configuration des paramètres de source pour Amazon Kinesis Data Streams

Configurez les paramètres de source pour Amazon Kinesis Data Streams afin d'envoyer des informations à un flux Firehose comme suit.

> **Important**
>
> Si vous utilisez la bibliothèque Kinesis Producer (KPL) pour écrire des données dans un flux de données Kinesis, vous pouvez utiliser l'agrégation pour combiner les enregistrements que vous écrivez dans ce flux de données 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 à AWS Lambda. Pour plus d'informations, consultez Développement de producteurs Amazon Kinesis Data Streams à l'aide de la bibliothèque et de l'agrégation Kinesis Producer.

Dans les paramètres Source, choisissez un flux existant dans la liste des flux de données Kinesis ou entrez un flux de données ARN au format. `arn:aws:kinesis:[Region]:[AccountId]:stream/[StreamName]`

Si vous ne disposez pas d'un flux de données existant, choisissez Create pour en créer un nouveau depuis la console Amazon Kinesis. Vous aurez peut-être besoin d'un IAM rôle disposant des autorisations nécessaires sur le flux Kinesis. Pour de plus amples informations, veuillez consulter ???. Après avoir créé un nouveau flux, cliquez sur l'icône d'actualisation pour mettre à jour la liste des 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, GetShardIteratorconsultez.

Amazon Data Firehose lance l'GetRecordsopération 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.

Passez à l'étape suivante pour configurer la transformation des enregistrements et la conversion des formats.

(Facultatif) Configurer la transformation des enregistrements et 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.

Dans la section Transformer les enregistrements source avec AWS Lambda, fournissez des valeurs pour le champ suivant.

1. 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 de plus amples informations, veuillez consulter Transformez les données sources dans 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 de plus amples informations, veuillez consulter Convertir le format des données d'entrée.

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

Si vous choisissez Managed Service pour Apache, Flink ou Direct PUT comme source pour votre flux Firehose

Dans la section Paramètres de la source, fournissez les champs suivants.

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 de plus amples informations, veuillez consulter Transformez les données sources dans 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 de plus amples informations, veuillez consulter Convertir le format des données d'entrée.

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 section décrit les paramètres que vous devez configurer pour votre stream Firehose en fonction de la destination que vous sélectionnez.

Rubriques

- Configurer les paramètres de destination pour Amazon S3
- Configuration des paramètres de destination pour les tables Apache Iceberg
- 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 HTTP Endpoint
- 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 de la JSON validité ou du nouveau délimiteur de ligne spécifié.

  Si vous regroupez plusieurs événements, journaux ou enregistrements en un seul PutRecord PutRecordBatch API appel, 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. API Lorsque le flux Firehose est configuré avec Kinesis Data Stream comme source, vous pouvez également utiliser l'agrégation intégrée dans la bibliothèque Kinesis Producer (). KPL La fonctionnalité de partition des données est exécutée après la désagrégation des données. Par conséquent, chaque enregistrement de chaque API appel 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), ou désagrégation par JSON 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 entre l'une ou l'autre option JSONou 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 un AWS Fonction 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 soit l'analyse syntaxique en ligne, soit votre AWS Fonction Lambda pour créer vos clés de partitionnement. Ou vous pouvez utiliser à la fois l'analyse syntaxique en ligne et votre AWS Fonction Lambda en même temps 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 supporte 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 d'une analyse syntaxique en ligne ou selon les paramètres que vous avez spécifiés AWS Fonction Lambda. Si vous avez spécifié un AWS Fonction 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 en utilisant le 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. Lors du partitionnement de vos données à l'aide d'une analyse syntaxique intégrée ou 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 des objets Amazon S3. Par défaut, Firehose ajoute un préfixe horaire.

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 entre GZIP une compression de données Snappy, Zip ou 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 figurant dans la liste des AWS KMS des clés que vous possédez. Si vous cryptez les données avec AWS KMS touches, vous pouvez utiliser soit la valeur par défaut AWS clé gérée (aws/s3) ou clé gérée par le client. Pour plus d'informations, voir Protection des données à l'aide du chiffrement côté serveur avec AWS KMS-Clés gérées (SSE-KMS).

Configuration des paramètres de destination pour les tables Apache Iceberg

Les tables Apache Iceberg en tant que destination sont disponibles dans l'est des États-Unis (Virginie du Nord), dans l'ouest des États-Unis (Oregon), en Europe (Irlande), en Asie-Pacifique (Tokyo), au Canada (centre) et en Asie-Pacifique (Sydney) Régions AWS.

Pour plus d'informations sur les tables Apache Iceberg en tant que destination, consultezFournissez des données aux tables Apache Iceberg avec Amazon Data Firehose.

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

> **Note**
>
> Firehose ne peut pas écrire sur les clusters Amazon Redshift qui utilisent un routage amélioré. VPC

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 débloquez les adresses IP Amazon Data Firehose. Pour de plus amples informations, veuillez consulter Accorder à 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 secret auprès de 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 secret parmi AWS Secrets Manager qui contient les informations d'identification du cluster Amazon Redshift. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager pour vos informations d'identification Amazon Redshift. Pour de plus amples informations, veuillez consulter Authentifiez-vous avec 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 lGestion 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` » pour les UTC objets Amazon S3 livrés. Vous pouvez ajouter le début de ce préfixe. Pour de plus amples informations, veuillez consulter Configurer le format du nom d'objet Amazon S3.

COPYoptions

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 n'est pas dans le même AWS Région en tant que cluster Amazon Redshift. Pour plus d'informations, consultez le COPYmanuel Amazon Redshift Database Developer Guide.

COPYcommande

La commande COPY Amazon Redshift. Pour plus d'informations, consultez le COPYmanuel Amazon Redshift Database Developer Guide.

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 entre GZIP une compression de données Snappy, Zip ou 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 figurant dans la liste des AWS KMS des clés que vous possédez. Si vous cryptez les données avec AWS KMS touches, vous pouvez utiliser soit la valeur par défaut AWS clé gérée (aws/s3) ou clé gérée par le client. Pour plus d'informations, voir Protection des données à l'aide du chiffrement côté serveur avec AWS KMS-Clés gérées (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 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 secret auprès de AWS Secrets Manager pour accéder 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 secret parmi AWS Secrets Manager qui contient les informations d'identification du groupe de travail Amazon Redshift Serverless. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager pour vos informations d'identification Amazon Redshift. Pour de plus amples informations, veuillez consulter Authentifiez-vous avec 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 lGestion 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` » pour les UTC objets Amazon S3 livrés. Vous pouvez ajouter le début de ce préfixe. Pour de plus amples informations, veuillez consulter Configurer le format du nom d'objet Amazon S3.

COPYoptions

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 n'est pas dans le même AWS Région en tant que groupe de travail Amazon Redshift Serverless. Pour plus d'informations, consultez le COPYmanuel Amazon Redshift Database Developer Guide.

COPYcommande

La commande COPY Amazon Redshift. Pour plus d'informations, consultez le COPYmanuel Amazon Redshift Database Developer Guide.

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 entre GZIP une compression de données Snappy, Zip ou 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 figurant dans la liste des AWS KMS des clés que vous possédez. Si vous cryptez les données avec AWS KMS touches, vous pouvez utiliser soit la valeur par défaut AWS clé gérée (aws/s3) ou clé gérée par le client. Pour plus d'informations, voir Protection des données à l'aide du chiffrement côté serveur avec AWS KMS-Clés gérées (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 de plus amples informations, veuillez consulter 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'erreurs S3 configuré. Pour les données envoyées àDLQ, vous devez les rediriger du compartiment d'erreur S3 configuré vers la OpenSearch destination.

Si vous souhaitez empêcher Firehose Stream de fournir des données 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 soutien.

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 CPU ressources dans le domaine des OpenSearch services et améliorant ainsi les performances.

VPCConnectivité des destinations

Si votre domaine de OpenSearch service est privéVPC, utilisez cette section pour le spécifierVPC. 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 HTTPS trafic 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 HTTPS trafic provenant des groupes de sécurité que vous avez spécifiés lors de la configuration de votre stream 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 la section Règles des groupes de sécurité dans la VPC documentation Amazon.

> **Important**
>
> Lorsque vous spécifiez des sous-réseaux pour transmettre des données à la destination dans un mode privéVPC, 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 ENIs ajouter de données pour la livraison de données en mode 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

Le 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.

VPCConnectivité des destinations

Si votre collection OpenSearch Serverless est privéeVPC, utilisez cette section pour le spécifier. 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 transmettre des données à la destination dans un mode privéVPC, 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 ENIs ajouter de données pour la livraison de données en mode 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'erreurs S3 configuré. Pour les données transmises àDLQ, vous devez les rediriger depuis le compartiment d'erreur S3 configuré vers la destination OpenSearch sans serveur.

Si vous souhaitez empêcher Firehose Stream de fournir des données 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 soutien.

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 HTTP Endpoint

Cette section décrit les options d'utilisation du point de HTTPterminaison pour votre destination.

Important

Si vous choisissez un HTTP point de terminaison comme destination, consultez et suivez les instructions figurant dansComprendre les spécifications relatives aux demandes de livraison et aux réponses des HTTP terminaux.

- Fournissez des valeurs pour les champs suivants :

  HTTPnom du point de terminaison - facultatif

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

  HTTPpoint final URL

  Spécifiez URL le HTTP point de terminaison au format suivant :`https://xyz.httpendpoint.com`. Ce URL doit être un HTTPSURL.

  Authentification

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

  - (Facultatif) Clé d'accès

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

  - Secret

    Sélectionnez un secret parmi AWS Secrets Manager qui contient la clé d'accès pour le HTTP point de terminaison. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager pour la clé d'accès. Pour de plus amples informations, veuillez consulter Authentifiez-vous avec 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 GZIPou 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 sélectionné. HTTP

  Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminaison (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**
>
> En ce qui concerne les destinations des HTTP terminaux, 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 stream 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.

  HTTPpoint final URL

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

  - Logs Datadog - US1
  - Logs Datadog - US3
  - Logs Datadog - US5
  - Logs Datadog - AP1
  - Journaux Datadog – EU
  - Logs 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 - États-Unis GOV

  Authentification

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

  - APIclé

Contactez Datadog pour obtenir la API clé dont vous avez besoin pour permettre à Firehose de transmettre des données à ce point de terminaison.

- Secret

  Sélectionnez un secret parmi AWS Secrets Manager qui contient la API clé de Datadog. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultezAuthentifiez-vous avec 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 GZIPou 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 sélectionné. HTTP

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminaison (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/ /metrics//getting-data-in. aws-cloudwatch-metrics

- Fournissez des valeurs pour les champs suivants :

  Point de terminaison Kinesis Honeycomb

  Spécifiez le point URL de HTTP terminaison au format suivant : https://api.honeycom b.io/1/ kinesis_events/ {{dataset}}

  Authentification

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

  - APIclé

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

  - Secret

    Sélectionnez un secret parmi AWS Secrets Manager qui contient la API clé de Honeycomb. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets

Manager. Pour plus d'informations, consultezAuthentifiez-vous avec 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 GZIPd'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 sélectionné. HTTP

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminaison (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, voir Commencer avec Coralogix.

- Fournissez des valeurs pour les champs suivants :

  HTTPpoint final URL

  Choisissez le HTTP point URL de terminaison parmi les options suivantes dans le menu déroulant :

  - Coralogix – États-Unis
  - Coralogix - SINGAPORE
  - Coralogix - IRELAND
  - Coralogix - INDIA
  - Coralogix - STOCKHOLM

  Authentification

  Vous pouvez choisir de saisir directement la clé privée ou de récupérer le secret auprès de 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 secret parmi AWS Secrets Manager qui contient 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, consultezAuthentifiez-vous avec 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 GZIPd'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 sélectionné. HTTP

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminaison (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/ /integrations//amazon-web-services. cloudwatch-metric-streams

- 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.

  HTTPpoint final URL

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

  Authentification

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

  - APIjeton

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

  - Secret

    Sélectionnez un secret parmi AWS Secrets Manager qui contient le API jeton pour Dynatrace. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultezAuthentifiez-vous avec AWS Secrets Manager dans Amazon Data Firehose.

API URL

Fournissez le contenu API URL 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 GZIPformat.

Retry duration (Durée de la nouvelle tentative)

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

Après avoir envoyé des données, Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminal, 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 de la part 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 LogicMonitorpour votre destination. Pour plus d'informations, consultez https://www.logicmonitor.com.

- Fournissez des valeurs pour les champs suivants :

  HTTPpoint final URL

  Spécifiez URL le HTTP point de terminaison au format suivant.

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

  Authentification

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

  - APIclé

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

  - Secret

    Sélectionnez un secret parmi AWS Secrets Manager qui contient la API clé pour LogicMonitor. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultezAuthentifiez-vous avec 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 GZIPou 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 sélectionné. HTTP

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminaison (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 :

  HTTPpoint final URL

  Spécifiez URL le HTTP point de terminaison au format suivant. Ce URL doit être un HTTPSURL.

  ```
  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 secret auprès de 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 secret parmi AWS Secrets Manager qui contient 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, consultezAuthentifiez-vous avec 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 terminal. 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 HTTP terminaison (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 :

  Webhook MongoDB Realm URL

  Spécifiez URL le HTTP point de terminaison au format suivant.

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

  Ce URL doit être un HTTPSURL.

  Authentification

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

  - APIclé

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

  - Secret

    Sélectionnez un secret parmi AWS Secrets Manager qui contient la API clé 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, consultezAuthentifiez-vous avec 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 GZIPou 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 terminal. 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 HTTP terminaison (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 :

  HTTPpoint final URL

  Choisissez le HTTP point URL de terminaison parmi les options suivantes dans la liste déroulante.

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

  Authentification

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

  - APIclé

    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 API clé pour permettre la livraison de données à ce point de terminaison depuis Firehose.

  - Secret

    Sélectionnez un secret parmi AWS Secrets Manager qui contient la API clé 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, consultezAuthentifiez-vous avec 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 GZIPou 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 New Relic. HTTP

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminaison (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), l'ouest des États-Unis (Oregon), l'Europe (Irlande), l'est des États-Unis (Ohio), l'Asie-Pacifique (Tokyo), l'Europe (Francfort), l'Asie-Pacifique (Singapour), l'Asie-Pacifique (Séoul), l'Asie-Pacifique (Mumbai), l'Europe (Londres), l'Amérique du Sud (Sao Paulo), le Canada (centre), l'Europe (Paris), Asie-Pacifique (Osaka), Europe (Stockholm), Asie-Pacifique (Jakarta) Régions AWS.

Paramètres de connexion

- Fournissez des valeurs pour les champs suivants :

  Compte Snowflake URL

  Spécifiez un compte Snowflake. URL olpPar exemple : `xy12345.us-east-1.aws.snowflakecomputing.com`. Reportez-vous à la documentation de Snowflake pour savoir comment déterminer votre compte. URL 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 secret auprès de AWS Secrets Manager pour accéder à 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 PEM d'en-tête et de pied de page 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 parmi AWS Secrets Manager qui contient 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, consultezAuthentifiez-vous avec 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.

VPCEIdentifiant privé (facultatif)

L'VPCEidentifiant permettant à Firehose de se connecter en privé à Snowflake. Le format de l'identifiant est com.amazonaws.vpce. [région] .vpce-svc-*[id]*Pour de plus amples informations, veuillez consulter .AWS PrivateLink et flocon de neige.

> **Note**
>
> Assurez-vous que votre réseau Snowflake autorise l'accès à Firehose. Nous vous recommandons de désactiver la politique réseau basée sur l'IP sur votre compte Snowflake, lorsque la politique réseau `AwsVpceIds` basée sur l'adresse IP est déjà activée. Si vous activez la politique réseau basée sur l'IP dans Snowflake, cela peut bloquer le trafic de données Firehose. Si vous avez un cas particulier qui nécessite des politiques basées sur l'IP, contactez l'équipe Firehose en soumettant un ticket d'assistance. Pour obtenir la liste des VPCE IDs produits que vous pouvez utiliser, reportez-vous auAccès à Snowflake dans 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 JSON les clés comme noms de colonnes
- Utiliser des VARIANT colonnes
  - 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 si l'ouverture du canal ou la livraison à Snowflake échoue 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.

Conseils sur la 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. Pour de plus amples informations, veuillez consulter Configurer les conseils de mise en mémoire tampon.

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 auprès de 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 parmi 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, consultezAuthentifiez-vous avec AWS Secrets Manager dans Amazon Data Firehose.

HECdélai d'expiration de l'accusé de réception

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 - -api. the-splunk-observability-cloud

- Fournissez des valeurs pour les champs suivants :

  Point de terminaison Cloud Ingest URL

  Vous pouvez trouver l'ingestion de données en temps réel de votre Splunk Observability Cloud URL dans Profil > Organizations > 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 auprès de AWS Secrets Manager pour accéder à Splunk Observability Cloud.

  - Jeton d'accès

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

- Secret

  Sélectionnez un secret parmi AWS Secrets Manager qui contient 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, consultezAuthentifiez-vous avec 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 GZIPou 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 sélectionné. HTTP

Après avoir envoyé les données, Amazon Data Firehose attend d'abord un accusé de réception de la part du terminal. 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 HTTP terminaison (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 :

  HTTPpoint final URL

  Spécifiez URL le HTTP point de terminaison au format suivant :`https://deployment name.sumologic.net/receiver/v1/kinesis/dataType/access token`. Ce URL doit être un HTTPSURL.

  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 GZIPou 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 terminal. 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 HTTP terminaison (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 :

  Point de terminaison élastique URL

  Spécifiez URL le HTTP point de terminaison au format suivant :`https://<cluster-id>.es.<region>.aws.elastic-cloud.com`. Ce URL doit être un HTTPSURL.

Authentification

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

- APIclé

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

- Secret

  Sélectionnez un secret parmi AWS Secrets Manager qui contient la API clé pour Elastic. Si vous ne voyez pas votre secret dans la liste déroulante, créez-en un dans AWS Secrets Manager. Pour plus d'informations, consultezAuthentifiez-vous avec 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(c'est ce qui est 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 terminal. 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 HTTP terminaison (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 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 choisissez de spécifier un AWS Fonction Lambda pour transformer les enregistrements de données ou si vous choisissez 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 un AWS Fonction Lambda pour transformer des enregistrements de données.
- Si vous définissez l'un des services suivants comme destination pour votre flux Firehose : Amazon OpenSearch Service, Datadog, Dynatrace, HTTP Endpoint, MongoDB Cloud, New Relic, Splunk ou LogicMonitor Sumo Logic, Snowflake, Apache Iceberg Tables.

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 entre GZIP une compression de données Snappy, Zip ou 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 figurant dans la liste des AWS KMS des clés que vous possédez. Si vous cryptez les données avec AWS KMS touches, vous pouvez utiliser soit la valeur par défaut AWS clé gérée (aws/s3) ou clé gérée par le client. Pour plus d'informations, voir Protection des données à l'aide du chiffrement côté serveur avec AWS KMS-Clés gérées (SSE-KMS).

Configurer 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 secondes MBs 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 indications de mise en mémoire tampon de Firehose par le biais du AWS Management Console, AWS Command Line Interface, ou AWS SDKs. Pour les flux existants, vous pouvez reconfigurer les indications de mise en mémoire tampon avec une valeur adaptée à vos cas d'utilisation en utilisant l'option Modifier de la console ou en utilisant le. UpdateDestinationAPI Pour les nouveaux flux, vous pouvez configurer des indications de mise en mémoire tampon dans le cadre de la création de nouveaux flux à l'aide de la console ou du CreateDeliveryStreamAPI. Pour ajuster la taille de la mémoire tampon, définissez `SizeInMBs` et `IntervalInSeconds` dans le `DestinationConfiguration` paramètre spécifique à la destination du CreateDeliveryStreamou UpdateDestinationAPI.

> **Note**
>
> - Les conseils de mémoire tampon sont appliqués au niveau d'une partition ou d'une partition, tandis que les conseils de partitionnement dynamique sont appliqués au niveau du flux ou du sujet.
> - 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.
> - La fonction de mise en mémoire tampon zéro n'est pas disponible pour le partitionnement dynamique.
> - 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 PUT API coûts 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)
Amazon S3	1-128 (5)	0-900 (300)
Tables Apache Iceberg	1-128 (5)	0-900 (300)
Amazon 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)	0-900 (60)
Coralogix	1-64 (6)	0-900 (60)
Dynatrace	1-64 (5)	0-900 (60)
Elasticité	1	0-900 (60)
Honeycomb	1-64 (15)	0-900 (60)
HTTPpoint final	1-64 (5)	0-900 (60)
LogicMonitor	1-64 (5)	0-900 (60)
Logzio	1-64 (5)	0-900 (60)
MongoDB	1 à 16 ans (5)	0-900 (60)
newRelic	1-64 (5)	0-900 (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)
sumoLogic	1-64 (1)	0-900 (60)
Splunk Observability Cloud	1-64 (1)	0-900 (60)
Snowflake	1 à 128 (1)	0 - 900 (0)

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 AWS Service de gestion des 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 AWS KMS—Clés gérées (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 IAM rôles 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, y compris votre compartiment S3, AWS KMSclé (si le chiffrement des données est activé) et fonction Lambda (si la transformation des données est activée). La console peut créer un rôle avec des espaces réservés. Pour plus d'informations, voir Qu'est-ce que c'est IAM ? .

- Balises - Vous pouvez ajouter des balises pour organiser votre AWS ressources, suivi des coûts et contrôle des accès.

  Si vous spécifiez des balises dans l'`CreateDeliveryStream`action, Amazon Data Firehose octroie une autorisation supplémentaire à l'`firehose:TagDeliveryStream`action 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 IAM des balises de ressources échoueront avec un résultat `AccessDeniedException` tel que celui-ci.

```
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.

Test du 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.

Prérequis

Avant de commencer, créez un stream Firehose. Pour de plus amples informations, veuillez consulter Tutoriel : Création d'un stream Firehose depuis la console.

Testez avec Amazon S3

Utilisez la procédure suivante pour tester votre flux Firehose avec 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/
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.

Testez avec Amazon Redshift

Utilisez la procédure suivante pour tester votre flux Firehose avec 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. Connectez-vous à Amazon Redshift via une SQL interface 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/
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`.

Tester avec OpenSearch Service

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/
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.

Testez avec Splunk

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/
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 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.

Tester avec les tables Apache Iceberg

Utilisez la procédure suivante pour tester votre flux Firehose avec les tables Apache Iceberg comme destination.

Pour tester un flux Firehose à l'aide des tables Apache Iceberg

1. Ouvrez la console Firehose à l'adresse. 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 vos tables Apache Iceberg. Notez que l'apparition de nouveaux objets dans votre bucket peut prendre quelques minutes, en fonction de sa configuration de mise en mémoire tampon.
5. Si les données de test n'apparaissent pas dans vos tables Apache Iceberg, vérifiez la présence d'événements ayant échoué dans votre compartiment Amazon S3.
6. 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

Cette section décrit comment utiliser différentes sources de données pour envoyer des données à votre flux Firehose. 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.

Vous pouvez envoyer des données vers votre flux Firehose à partir des sources de données suivantes.

Rubriques

- Configurer l'agent Kinesis pour envoyer des données
- Envoyer des données avec AWS SDK
- Envoyer CloudWatch les logs à Firehose
- Envoyer CloudWatch des événements à Firehose
- Configuration AWS IoT pour envoyer des données à Firehose

Configurer l'agent Kinesis pour envoyer des données

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 Spécifier les 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.

Prérequis

Avant de commencer à utiliser Kinesis Agent, assurez-vous de remplir les conditions préalables suivantes.

- 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 JRE la version 1.8 ou ultérieure. La version 1.1.x de l'agent s'exécute à l'aide de la version JRE 1.7 ou ultérieure.
- Si vous utilisez Amazon EC2 pour exécuter votre agent, lancez votre EC2 instance.
- Le IAM rôle ou AWS les informations d'identification que vous spécifiez doivent être autorisées à effectuer l'PutRecordBatchopération Amazon Data 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' CloudWatch PutMetricDataopération est également nécessaire. Pour plus d'informations, consultez Contrôler l'accès avec Amazon Data FirehoseSurveiller l'état de santé de l'agent Kinesis, et Authentification et contrôle d'accès pour Amazon CloudWatch.

Gérer AWS informations d'identification

Gérez votre 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 “Créez des fournisseurs d'informations d'identification personnalisés”.
- Spécifiez un IAM rôle lorsque vous lancez votre EC2 instance.

- Spécifiez AWS informations d'identification lorsque vous configurez l'agent (voir les entrées pour `awsAccessKeyId` et `awsSecretAccessKey` dans le tableau de configuration ci-dessousthe section called “Spécifier les paramètres de configuration de l'agent”).
- Modifiez `/etc/sysconfig/aws-kinesis-agent` pour spécifier votre AWS Région et AWS clés d'accès.
- Si votre EC2 instance se trouve dans une autre AWS compte, créez un IAM rôle pour donner accès au service Amazon Data Firehose. Spécifiez ce rôle lorsque vous configurez l'agent (voir assumeRoleARNet assumeRoleExternalID). Utilisez l'une des méthodes précédentes pour spécifier AWS les informations d'identification d'un utilisateur de l'autre compte autorisé à assumer ce rôle.

Créez des fournisseurs d'informations d'identification personnalisés

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 “Spécifier les 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).

Téléchargez et installez l'agent

Commencez par vous connecter à votre instance. Pour plus d'informations, consultez Connect to Your Instance dans le guide de EC2 l'utilisateur Amazon. 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 EC2 l'utilisateur Amazon.

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
  ```

  L'agent v 2.0.0 ou version ultérieure est installé 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 plus d'informations concernant 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 valeur par défaut AWS La région 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 de plus amples informations, veuillez consulter Spécifier les 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`.

Spécifier les 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
`assumeRoleARN`	Le nom de ressource Amazon (ARN) du rôle à assumer par l'utilisateur. Pour plus d'informations, voir Accès délégué à travers AWS Comptes utilisant IAM des rôles dans le guide de IAM l'utilisateur
`assumeRoleExternalId`	Identifiant facultatif qui détermine qui peut assumer le rôle. Pour plus d'informations, consultez la section Comment utiliser un identifiant externe dans le guide de IAM l'utilisateur.
`awsAccessKeyId`	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.
`awsSecretAccessKey`	AWS clé secrète qui remplace les informations d'identification par défaut. Ce paramètre est prioritaire sur tous les autres fournisseurs d'informations d'identification.
`cloudwatch.emitMetrics`	Permet à l'agent d'émettre des métriques CloudWatch s'il est défini (true). Valeur par défaut : true
`cloudwatch.endpoint`	Le point de terminaison régional pour CloudWatch. Par défaut : `monitoring.us-east-1.amazonaws.com`
`firehose.endpoint`	Point de terminaison régional pour Amazon Data Firehose. Par défaut : `firehose.us-east-1.amazonaws.com`
`sts.endpoint`	Le point de terminaison régional pour AWS Service de jetons de sécurité. Par défaut : `https://sts.amazonaws.com`
`userDefinedCredentialsProvider.classname`	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 `.class` à la fin du nom de la classe.

Paramètre de configuration	Description
`userDefinedCredentialsProvider.location`	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 : `/usr/share/aws-kinesis-agent/lib/` .

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

Paramètre de configuration	Description
`aggregatedRecordSizeBytes`	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)
`dataProcessingOptions`	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 de plus amples informations, veuillez consulter Pré-traiter les données avec des agents.
`deliveryStream`	[Obligatoire] Nom du stream Firehose.
`filePattern`	[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 `aws-kinesis-agent-user` . Pour l'annuaire contenant les fichiers, accordez les autorisations de lecture et d'exécution à `aws-kinesis-agent-user` .

Paramètre de configuration	Description
	Important 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.
`initialPosition`	Position initiale à partir de laquelle le fichier a commencé à être analysé. Les valeurs valides sont `START_OF_FILE` et `END_OF_FILE` . Par défaut : `END_OF_FILE`
`maxBufferAgeMillis`	Durée maximale, en millisecondes, pendant laquelle l'agent met les données en mémoire tampon avant de les envoyer au flux Firehose. Plage de valeurs : 1 000 à 900 000 (1 seconde à 15 minutes) Par défaut : 60 000 (1 minute)
`maxBufferSizeBytes`	Taille maximale, en octets, pour laquelle l'agent met en mémoire tampon les données avant de les envoyer au flux Firehose. Plage de valeurs : 1 à 4 194 304 (4 Mo) Par défaut : 4 194 304 (4 Mo)
`maxBufferSizeRecords`	Le nombre maximum d'enregistrements pour lesquels l'agent met les données en mémoire tampon avant de les envoyer au flux Firehose. Plage de valeurs : 1 à 500 Par défaut : 500

Paramètre de configuration	Description
`minTimeBetweenFilePollsMillis`	Fréquence, en millisecondes, à laquelle l'agent interroge et analyse les fichiers surveillés pour rechercher les nouvelles données. Plage de valeurs : 1 ou plus Par défaut : 100
`multiLineStartPattern`	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.
`skipHeaderLines`	Nombre de lignes que l'agent doit ignorer lors de l'analyse au début des fichiers surveillés. Plage de valeurs : 0 ou plus Par défaut : 0 (zéro)
`truncatedRecordTerminator`	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) Par défaut : `'\n'` (nouvelle ligne)

Configuration de plusieurs répertoires de fichiers et de 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.

Pré-traiter les données avec des agents

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 d'un format séparé par un délimiteur en format. JSON

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

`customFieldNames`

[Obligatoire] Les noms de champs utilisés comme clés dans chaque JSON paire clé-valeur. 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 d'un format journal en JSON format. 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 RFC3164 Syslog. 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`

Les noms de champs personnalisés utilisés comme clés dans chaque paire JSON clé-valeur. 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.

Example : LOGTOJSON Configuration

Voici un exemple de LOGTOJSON configuration pour une entrée Apache Common Log convertie au JSON format :

```
{
    "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"}
```

Example : LOGTOJSON Configuration 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 Apache Common Log que dans l'exemple précédent est convertie au JSON format suivant :

```
{"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"}
```

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

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

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

Example : 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 aux `customFieldNames` valeurs spécifiées pour former un enregistrement d'une seule ligne au JSON format.

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

Example : LOGTOJSON Configuration avec Match Pattern

Voici un exemple de `LOGTOJSON` configuration pour une entrée Apache Common Log convertie au JSON format, le dernier champ (octets) étant omis :

```
{
    "optionName": "LOGTOJSON",
    "logFormat": "COMMONAPACHELOG",
    "matchPattern": "^([\\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"}
```

Utiliser les CLI commandes courantes de l'agent

Le tableau suivant fournit un ensemble de cas d'utilisation courants et les commandes correspondantes pour travailler avec AWS Agent Kinesis.

Cas d’utilisation	Command
Démarrage automatique de l'agent au démarrage du système	`sudo chkconfig aws-kinesis-agent on`
Vérifiez le statut de l'agent	`sudo service aws-kinesis-agent status`
Arrêtez 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`
Désinstallez l'agent	`sudo yum remove aws-kinesis-agent`

Résoudre les problèmes liés à l'envoi depuis Kinesis Agent

Ce tableau fournit des informations de dépannage et des solutions aux problèmes courants rencontrés lors de l'utilisation de l'agent Amazon Kinesis.

Problème	Solution
Pourquoi Kinesis Agent ne fonctionne-t-il pas sous 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.
Comment résoudre les `java.lang.OutOfMemoryError` exceptions ?	Cela se produit lorsque 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.
Comment résoudre les `IllegalStateException : connection pool shut down` exceptions ?	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

Problème	Solution
	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).

Envoyer des données avec AWS SDK

Vous pouvez utiliser Amazon Data Firehose API pour envoyer des données vers un flux Firehose à l'aide du AWS SDKpour 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.

Amazon Data Firehose API 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.

Opérations d'écriture uniques 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 souhaiterez 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 en savoir plus sur le contexte du 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 Operations API.

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 souhaiterez 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 en savoir plus sur le contexte du 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 Operations API.

Envoyer CloudWatch les logs à Firehose

CloudWatch Les événements des journaux peuvent être envoyés à Firehose à l'aide de filtres d' CloudWatch abonnement. 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.

Décompressez les 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) ou le partitionnement dynamique. ORC Vous devez activer la décompression pour votre flux Firehose.

Vous pouvez activer la décompression à l'aide du AWS Management Console, AWS Command Line Interface or AWS SDKs.

> **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.

Extraire le message 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"}
```

Activer la décompression sur un nouveau stream Firehose depuis la console

Pour activer la décompression sur un nouveau stream Firehose à l'aide du AWS Management Console

1. Connectez-vous au AWS Management Console et ouvrez la console Kinesis à l'adresse /kinesis. https://console.aws.amazon.com
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 peuvent écrire directement. Pour une liste des AWS services et 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.

Activer la décompression sur un flux Firehose existant

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.

Désactiver la décompression sur le stream Firehose

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

1. Connectez-vous au AWS Management Console et ouvrez la console Kinesis à l'adresse /kinesis. https://console.aws.amazon.com
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.

Résoudre les problèmes de décompression dans Firehose

Le tableau suivant montre comment Firehose gère les erreurs lors de la décompression et du traitement des données, notamment en fournissant des enregistrements à un compartiment S3 d'erreur, en enregistrant les erreurs et en émettant des métriques. Il explique également le message d'erreur renvoyé pour les opérations de saisie de données non autorisées.

Problème	Solution
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	Si Amazon Data Firehose échoue dans les étapes de traitement après la décompression, telles que le partitionnement dynamique et la conversion du format

Problème	Solution
traitement après une décompression réussie ?	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 de plus amples informations, veuillez consulter CloudWatch Métriques de décompression des journaux.

Envoyer CloudWatch des événements à Firehose

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 au AWS Management Console et ouvrez la CloudWatch console à l'adresse 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.

Configuration AWS IoT pour envoyer des données à Firehose

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

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

1. Lors de la création d'une règle dans le AWS IoT console, sur la page Créer une règle, sous Définir une ou plusieurs actions, sélectionnez Ajouter une 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. Pour IAMle nom du rôle, choisissez un IAM rôle existant ou choisissez Créer un nouveau rôle.
7. Choisissez Add action.

Pour plus d'informations sur la création AWS IoT règles, voir AWS Tutoriels sur les règles de l'IoT.

Transformez les données sources dans 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.

Comprendre le flux de transformation des 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 ProcessingConfigurationparamètre du CreateDeliveryStreamou UpdateDestinationAPIavec l'ProcessorParameterappel `BufferSizeInMBs` et`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.

Durée d'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 “Gérer les défaillances liées à la transformation des données ”

Paramètres requis pour la transformation des données

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.

For Kinesis Data Streams and Direct PUT

Les paramètres suivants sont obligatoires pour tous les enregistrements transformés à partir de Lambda.

- `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 des 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 le statut d'un enregistrement est `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`— La charge utile des données transformées, après le codage en base64.

  Voici un exemple de résultat Lambda :

```
{
   "recordId": "<recordId from the Lambda input>",
   "result": "Ok",
   "data": "<Base64 encoded Transformed data>"
```

```
}
```

For Amazon MSK

Les paramètres suivants sont obligatoires pour tous les enregistrements transformés à partir de Lambda.

- `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 des 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`— La charge utile des données transformées, après le codage en base64.

  Voici un exemple de résultat Lambda :

```
 {
    "recordId": "<recordId from the Lambda input>",
    "result": "Ok",
    "kafkaRecordValue": "<Base64 encoded Transformed data>"
}
```

Blueprints Lambda pris en charge

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/.
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 (Node.js) JSON

  Ce plan montre comment convertir les enregistrements d'entrée au format RFC3164 Syslog 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.

Gérer les défaillances liées à la 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 annuler les options de nouvelle tentative à l'aide du CreateDeliveryStreamou. `UpdateDestination` API Pour ce type d'échec, vous pouvez enregistrer les erreurs d'invocation dans Amazon CloudWatch Logs. Pour de plus amples informations, veuillez consulter Surveillez Amazon Data Firehose à l'aide des 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`

Le code HTTP d'erreur 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`

Le nom de ressource Amazon (ARN) de la fonction Lambda.

Backup source records

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.

Partitionner les données de streaming 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

- Activer le partitionnement dynamique dans Amazon Data Firehose
- Comprendre les clés de partitionnement
- Utiliser le préfixe du compartiment Amazon S3 pour fournir des données
- Appliquer le partitionnement dynamique aux données agrégées
- Résoudre les erreurs de partitionnement dynamique
- Données de mémoire tampon pour le partitionnement dynamique

Activer le partitionnement dynamique dans Amazon Data Firehose

Vous pouvez configurer le partitionnement dynamique pour vos flux Firehose via la console CLI de gestion Amazon Data Firehose, ou le. APIs

> **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.

Comprendre les 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.

Vous pouvez créer des clés de partitionnement à l'aide des méthodes 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 formatés. JSON Actuellement, nous ne prenons en charge que `jq 1.6` la version.
- AWS Fonction Lambda : cette méthode utilise une AWS Lambda fonction 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 syntaxique intégrée

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'CLIexemple.

```
{
   "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
`customer_id`	.customer_id
`device`	.type.device
`year`	.event_timestamp| strftime("%Y")
`month`	.event_timestamp| strftime("%m")
`day`	.event_timestamp| strftime("%d")
`hour`	.event_timestamp| strftime("%H")

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 avec une fonction AWS Lambda

Pour les enregistrements de données compressés ou chiffrés, ou pour les données sous un autre format de fichierJSON, vous pouvez utiliser la AWS Lambda fonction 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'),
                          "day": 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)
}
```

Utiliser le préfixe du compartiment Amazon S3 pour fournir des données

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.

Ajoutez un nouveau délimiteur de ligne lors de la livraison de données vers Amazon 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.

Appliquer le partitionnement dynamique aux données agrégées

Vous pouvez appliquer un partitionnement dynamique à des données agrégées (par exemple, plusieurs événements, journaux ou enregistrements regroupés en un seul `PutRecord` `PutRecordBatch` API appel), 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 JSON des objets 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==` La désagrégation des enregistrements par JSON ou par délimiteur est plafonnée à 500 par enregistrement.

> **Note**
>
> Lors de la désagrégation JSON des enregistrements, assurez-vous que votre entrée est toujours présentée dans le format pris en chargeJSON. JSONles objets doivent se trouver sur une seule ligne sans délimiteur ou uniquement délimités par une nouvelle ligne (). JSONL Un tableau d'JSONobjets 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 API chaque appel 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 : Déagrégation → Transformation des données via Lambda → Clés de partitionnement.

Résoudre les 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.

Données de mémoire tampon pour le 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 d'une mémoire tampon est mesurée en secondes MBs et un intervalle de mémoire tampon est mesuré en secondes.

> **Note**
>
> La fonction de mise en mémoire tampon zéro n'est pas disponible pour le partitionnement dynamique.

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 transférer ces enregistrements 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.

Convertir le format des données d'entrée dans Amazon Data Firehose

Amazon Data Firehose peut convertir le format de vos données d'entrée 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 économisent de l'espace et permettent des requêtes plus rapides par rapport aux formats orientés lignes tels que. JSON Si vous souhaitez convertir un format d'entrée autre que JSON des valeurs séparées par des virgules (CSV) ou du texte structuré, vous pouvez utiliser AWS Lambda pour le transformer en JSON premier. Pour de plus amples informations, veuillez consulter Transformer les données sources.

Vous pouvez convertir le format de vos données même si vous agrégez vos enregistrements avant de les envoyer à Amazon Data Firehose.

Amazon Data Firehose a besoin des trois éléments suivants pour convertir le format de vos données d'enregistrement :

Deserializer

Amazon Data Firehose nécessite un désérialiseur pour lire vos données d'entrée. JSON Vous pouvez choisir l'un des deux types de désérialiseur suivants.

Lorsque vous combinez plusieurs JSON documents dans le même enregistrement, assurez-vous que votre entrée est toujours présentée dans le JSON format pris en charge. Un tableau de JSON documents 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}]`

- Ruche Apache JSON SerDe
- OpenX JSON SerDe

Choisissez le JSON désérialiseur

Choisissez OpenX JSON SerDe si votre entrée JSON 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`.

L'OpenX JSON SerDe peut convertir des points (`.`) en traits de soulignement (). _ Il peut également convertir JSON les clés en minuscules avant de les désérialiser. Pour plus d'informations sur les options disponibles avec ce désérialiseur via Amazon Data Firehose, consultez O. penXJson SerDe

Si vous ne savez pas quel désérialiseur choisir, utilisez OpenX JSON SerDe, sauf si vous avez des horodatages qu'il ne supporte pas.

Si vous avez des horodatages dans des formats autres que ceux listés précédemment, utilisez Apache Hive. JSON SerDe Lorsque vous choisissez cette 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.

The Hive JSON SerDe n'autorise pas ce qui suit :

- Des points (`.`) dans les noms de colonnes.
- Les champs dont le type est `uniontype`.
- Champs dont le schéma contient des types numériques, mais qui sont des chaînes dans leJSON. Par exemple, si le schéma est (un int) et le JSON est`{"a":"123"}`, la Hive génère SerDe une erreur.

The Hive SerDe ne convertit pas les chaînes imbriquéesJSON. Par exemple, si vous avez `{"a": {"inner":1}}`, il ne traite pas `{"inner":1}` comme une chaîne.

Schema

Amazon Data Firehose a besoin d'un schéma pour déterminer comment interpréter ces données. Utiliser 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 à la fois Amazon Data Firehose et votre logiciel d'analyse. Pour plus d'informations, voir Remplissage du AWS Catalogue de données Glue dans le AWS Glue Guide du développeur.

> **Note**
>
> Le schéma créé dans AWS Glue Le catalogue de 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 NestedJSON, utilisez un STRUCT type dans le schéma qui reflète la structure de vos JSON données. Consultez cet exemple pour savoir comment gérer l'imbrication JSON avec un STRUCT type.

> **Important**
>
> Pour les types de données qui ne spécifient pas de limite de taille, il existe une limite pratique de 32 MBs pour toutes les données d'une seule ligne.
> Si vous spécifiez une longueur pour `CHAR` ou`VARCHAR`, Firehose tronque les chaînes à la longueur spécifiée lorsqu'il lit les données d'entrée. Si la chaîne de données sous-jacente est plus longue, elle reste inchangée.

Serializer

Firehose nécessite un sérialiseur pour convertir les données dans le format de stockage en colonnes cible (Parquet ouORC). Vous pouvez choisir l'un des deux types de sérialiseurs suivants.

- ORC SerDe
- Parquet SerDe

Choisissez le sérialiseur

Le sérialiseur que vous choisissez dépend de vos besoins métier. Pour en savoir plus sur les deux options du sérialiseur, voir ORC SerDeet Parquet. SerDe

Activer la conversion du format d'enregistrement

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. La section suivante explique comment activer la conversion du format d'enregistrement à partir de la console et des opérations FirehoseAPI. Pour un exemple de configuration de la conversion du format d'enregistrement avec AWS CloudFormation, voir AWS::DataFirehose::DeliveryStream.

Activer la conversion du format d'enregistrement depuis la 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 au AWS Management Console, et ouvrez la console Amazon Data Firehose 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. Tutoriel : Création d'un stream Firehose depuis la console
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 un AWS Glue table pour spécifier un schéma pour vos enregistrements source. Définissez la région, la base de données, la table et la version de la table.

Gérez la conversion des formats d'enregistrement depuis Firehose API

Si vous souhaitez qu'Amazon Data Firehose convertisse le format de vos données d'entrée au format Parquet ou ORC spécifiez l'DataFormatConversionConfigurationélément facultatif dans ExtendedS3 ou ExtendedS3 DestinationConfiguration. JSON 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 lors de la conversion des formats de données

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 enregistrement ayant échoué, Amazon Data Firehose rédige un JSON document 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
  }
}
```

Comprendre la diffusion des données dans Amazon Data Firehose

Lorsque vous envoyez des données à votre flux Firehose, elles sont automatiquement transmises à la destination que vous avez choisie. Le tableau suivant explique la livraison des données vers différentes destinations.

Destination	Détails
Amazon S3	Pour la livraison de données à Amazon S3, Firehose concatène plusieurs enregistrements entrants en fonction de la configuration de mise en mémoire tampon de votre 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 ou le paramètre de la console Firehose. API
Amazon Redshift	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 la section Paramètres du format des données de COPY commande Amazon Redshift.
OpenSearch Service et OpenSearch sans serveur	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

Destination	Détails
	demande OpenSearch groupée Service ou OpenSearch Serverles s 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 JSON objet d'une seule ligne avant de l'envoyer à Amazon Data Firehose. En outre, l'`rest.action.multi.allow_explicit_index` option 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.
Splunk	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. Pour rediriger les données qui ont été envoyées vers le compartiment d'erreur S3 (sauvegarde S3) vers Splunk, suivez les étapes mentionnées dans la documentation Splunk.
HTTPpoint final	Pour la livraison de données à un HTTP point de terminaison appartenant à un fournisseur de services tiers pris en charge, vous pouvez utiliser le service intégré Amazon Lambda pour créer une fonction permettant de transformer le ou les enregistrements entrants au format correspondant au format attendu par l'intégration du fournisseur de services. Contactez le fournisseur de services tiers dont vous avez choisi le HTTP terminal pour votre destination pour en savoir plus sur le format d'enregistrement accepté.

Destination	Détails
Snowflake	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 streaming API de Snowflake pour insérer des données dans Snowflake. Par défaut, les enregistr ements 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 JSON élément en tant que charge utile d'enregistrement et ne prend pas en charge les tableaux. JSON Assurez-vous que votre charge utile d'entrée est un JSON objet valide et qu'elle est bien formée sans guillemets, guillemets ou caractères d'échappement supplémentaires.

Chaque destination Firehose possède sa propre fréquence de livraison des données. Pour de plus amples informations, veuillez consulter Configurer les conseils de mise en mémoire tampon.

Registres en double

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.

Rubriques

- Comprenez la livraison entre AWS les comptes et les régions
- Comprendre les spécifications relatives aux demandes de livraison et aux réponses des HTTP terminaux
- Gérez les défaillances de livraison de données
- Configurer le format du nom d'objet Amazon S3
- Configurer la rotation de l'index pour le OpenSearch service
- Suspendre et reprendre la livraison des données

Comprenez la livraison entre AWS les comptes et les régions

Amazon Data Firehose prend en charge la diffusion de données vers les destinations des HTTP terminaux, quel que soit le AWS compte. Le stream Firehose et le HTTP point de terminaison que vous choisissez comme destination peuvent appartenir à des comptes différents AWS .

Amazon Data Firehose prend également en charge la diffusion de données vers des destinations de point de HTTP terminaison dans toutes les régions AWS . Vous pouvez transmettre les données d'un flux Firehose d'une AWS région à un point de HTTP terminaison d'une autre AWS région. Vous pouvez également transmettre des données d'un flux Firehose vers une destination de point de HTTP terminaison située en dehors des AWS régions, par exemple vers votre propre serveur local en configurant le point de HTTP terminaison sur la destination URL 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 ».

Comprendre les spécifications relatives aux demandes de livraison et aux réponses des HTTP terminaux

Pour qu'Amazon Data Firehose puisse transmettre des données à des points de HTTP terminaison 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 HTTP demandes que le service Amazon Data Firehose envoie aux HTTP points de terminaison personnalisés, ainsi que les spécifications de format des HTTP réponses attendues par le service Amazon Data Firehose. HTTPles points de terminaison 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.

Format des demandes

Chemin et URL paramètres

Ils sont configurés directement par vous dans le cadre d'un URL champ unique. Amazon Data Firehose les envoie tels que configurés sans modification. Seules les destinations HTTPS sont prises en charge. URLdes restrictions sont appliquées lors de la configuration du flux de diffusion.

> **Note**
>
> Actuellement, seul le port 443 est pris en charge pour la livraison des données des HTTP terminaux.

HTTPEn-têtes - 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.

HTTPEn-têtes - X-Amz-Firehose-Request-Id

La valeur de cet en-tête est une valeur opaque GUID qui peut être utilisée 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.

HTTPEn-têtes - Type de contenu

La valeur de l'en-tête Content-Type est toujours `application/json`.

HTTPEn-têtes - Encodage du contenu

Un flux Firehose peut être configuré pour être utilisé pour GZIP 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.

HTTPEn-têtes - Longueur du contenu

Ces en-têtes sont utilisés de manière standard.

HTTPEn-têtes - X-Amz-Firehose-Source-Arn :

Le ARN flux Firehose représenté sous ASCII forme de chaîne. Il ARN code la région, l'identifiant AWS du compte et le nom du flux. Par exemple, `arn:aws:firehose:us-east-1:123456789:deliverystream/testStream`.

HTTPEn-têtes - X-Amz-Firehose-Access-Key

Cet en-tête contient une API clé ou d'autres informations d'identification. Il est possible de créer ou de mettre à jour le API -key (alias 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 JWT jeton ou un ACCESS _KEY. Si un point de terminaison nécessite des informations d'identification à champs multiples (par exemple, nom d'utilisateur et 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 (ou)JSON. 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.

HTTPEn-têtes - 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 forme d'JSONobjet selon 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 JSON document avec le JSON schéma suivant (écrit enYAML) :

```
"$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 ne répond pas 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 HTTP d'état MUST doit être compris entre 2XX, 4XX ou 5XX.

Le serveur Amazon Data Firehose NOT suit les redirections (codes d'état 3XX). Seul le code de réponse 200 est considéré comme une livraison réussie des enregistrements à 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.

HTTPEn-têtes - Encodage du contenu

Le codage du contenu MUST NOT doit être utilisé. Le corps MUST ne doit pas être comprimé.

HTTPEn-têtes - Longueur du contenu

L'en-tête Content-Length MUST 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 (initial-backoff-time * (multiplicateur (2) ^ 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 = MAX (120, 2^n) * random (0,85, 1,15).

Les paramètres spécifiés dans l'équation précédente sont susceptibles d'être modifiés. Reportez-vous à la documentation 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 pour chaque nouvelle tentative, dans le meilleur des cas. Cette dernière fonctionnalité peut être utilisée à des fins de déduplication par le serveur de point de HTTP terminaison. 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 CWLog provenant d'une source.

```
{
  "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"
        }
      ]
    }
   }
  ]
}
```

Gérez les défaillances de livraison de données

Chaque destination Amazon Data Firehose dispose de sa propre gestion des défaillances de livraison de données.

Lorsque vous configurez un flux Firehose, pour de nombreuses destinations telles que OpenSearch Splunk et les HTTP points de terminaison, 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, consultez les sections de destination correspondantes sur cette page. Pour plus d'informations sur la manière 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.

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 IAM rôle 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 saisir COPY des données manuellement à l'aide de fichiers manifestes, consultez la section Utilisation d'un manifeste pour spécifier des 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, le JSON format de chaque document est le 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, le JSON format de chaque document est le 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":
 "MiAyNTE2MjAyNzIyMDkgZW5pLTA1ZjMyMmQ1IDIxOC45Mi4xODguMjE0IDE3Mi4xNi4xLjE2NyAyNTIzMyAxNDMzIDYgM
  "EventId": "49577193928114147339600778471082492393164139877200035842.0"
}
```


HTTPdestination du terminal

Lorsqu'Amazon Data Firehose envoie des données vers une destination de point de HTTP terminaison, 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 HTTP terminaison, 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 HTTP point de terminaison. 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 plus d'informations sur les autres types d'erreurs de livraison de données, consultez la section Erreurs de livraison des données des HTTP terminaux

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":"c2FtcGxlIHJhdyBkYXRh",
 "subsequenceNumber":0,
 "dataId":"49607357361271740811418664280693044274821622880012337186.0"
}
```

Snowflake

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 JSON charge utile 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 JSON charge utile 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 défaillants sont livrés dans le dossier snowflake-failed/, que vous pouvez utiliser pour le remplissage manuel.

Voici un exemple 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": "c2FtcGxlIHJhdyBkYXRh"
}
```

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 en utilisant la console Firehose ou l'UpdateDestinationAPIopération.

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 horaire est activéUTC, mais vous pouvez le modifier pour le fuseau horaire de votre choix. Par exemple, pour utiliser l'heure normale du Japon au lieu deUTC, vous pouvez configurer le fuseau horaire vers Asia/Tokyo dans le APIparamètre AWS Management Console ou dans le réglage des paramètres () CustomTimeZone. La liste suivante contient les fuseaux horaires pris en charge par Firehose pour la configuration du préfixe S3.

Fuseaux horaires pris en charge

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/Fakaofo
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 : rapide	.snappy
Compression : Hadoop-Snappy	.hsnappy

Vous pouvez également spécifier l'extension de fichier que vous préférez dans la console Firehose ou. API 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.

Comprendre les 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 FirehoseDeveloper Guide.

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 activé. UTC Mais vous pouvez spécifier le fuseau horaire que vous préférez. Par exemple, vous pouvez configurer le fuseau horaire pour Asia/Tokyo dans le AWS Management Console ou dans le API paramétrage (CustomTimeZone) si vous souhaitez utiliser

l'heure normale du Japon au lieu deUTC. 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.

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
`error-output-type`	Évalue l'une des chaînes suivantes, en fonction de la configuration de votre stream Firehose et de la raison de l'échec : {processing-failed, -failed, splunk-failed AmazonOpenSearchService,,}. format-conversion-failed http-endpoint-failed Si vous utilisez l'espace de noms plusieurs fois dans la	`myPrefix/result=!{firehose:error-output-type}/!{timestamp:yyyy/MM/dd}`	`myPrefix/result=processing-failed/2018/08/03`	La error-output-type valeur ne peut être utilisée que dans le ErrorOutputPrefix champ.

Conversion	Description	Exemple d'entrée	Exemple de sortie	Remarques
	même expression, chaque instance est évaluée sur la 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 un AWS Fonction Lambda Pour créer des clés de partitionnement pour vos données sources, vous devez spécifier une

valeur de préfixe de compartiment S3 composée d'expressions spécifiées au 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 n'`ErrorOutputPrefix`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 le plugin AWS CLI ou leAPI, vous pouvez l'utiliser `ExtendedS3DestinationConfiguration` pour spécifier une configuration de sauvegarde Amazon S3 avec son propre `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 à l'aide d'expressions, vous devez spécifier le préfixe d'erreur, y compris. `!{firehose:error-output-type}`

- Lorsque vous l'utilisez `ExtendedS3DestinationConfiguration` avec AWS CLI, leAPI, ou AWS CloudFormation, si vous spécifiez un`S3BackupConfiguration`, Firehose ne fournit pas de valeur par défaut. `ErrorOutputPrefix`
- Vous ne pouvez pas utiliser `partitionKeyFromLambda` d'`partitionKeyFromQuery`espaces de noms lorsque vous créez des ErrorOutputPrefix expressions.

Exemples de préfixe

Exemples de préfixes **`Prefix`** et **`ErrorOutputPrefix`**

Entrée	Préfixe évalué (à 10 h 30 le 27 UTC août 2018)

`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 h 30 le 27 UTC août 2018)
`ErrorOutputPrefix` : `myErrorPrefix/year=!{timestamp:yyyy}/month=!{timestamp:MM}/day=!{timestamp:dd}/hour=!{timestamp:HH}/!{firehose:error-output-type}`	`ErrorOutputPrefix` : `myErrorPrefix/year=2018/month=07/day=06/hour=23/processing-failed`

`Prefix`: `myFirehosePrefix/`
 	
`ErrorOutputPrefix` : non spécifié
	`Prefix`: `myFirehosePrefix/2018/08/27/`
	
	`ErrorOutputPrefix` : `myFirehosePrefix/processing-failed/2018/08/27/`

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'UTCarrivée 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 `OneWeek` cette option, Data Firehose crée automatiquement des index au format < YEAR >-w< WEEK NUMBER > (par exemple,`2020-w33`), où le numéro de semaine est calculé en fonction du UTC temps et conformément aux 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.

Suspendre et reprendre la livraison des données

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.

> **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 est due au fait qu'un petit nombre d'enregistrements qui n'ont pas pu être livrés à destination auparavant après plusieurs tentatives sont considérés comme ayant échoué.

Suspendre 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.

Reprendre 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.

Fournissez des données aux tables Apache Iceberg avec Amazon Data Firehose

> **Note**
>
> La diffusion du flux Firehose vers les tables Apache Iceberg dans Amazon S3 est en cours de prévisualisation. N'utilisez pas cette fonctionnalité pour les charges de travail de production.

Apache Iceberg est un format de table open source performant permettant d'effectuer des analyses de mégadonnées. Apache Iceberg apporte la fiabilité et la simplicité des SQL tables aux lacs de données Amazon S3 et permet aux moteurs d'analyse open source tels que Spark, Flink, Trino, Hive et Impala de travailler simultanément avec les mêmes données. Pour plus d'informations sur Apache Iceberg, consultezhttps://iceberg.apache.org/.

Vous pouvez utiliser Firehose pour transmettre directement des données de streaming aux tables Apache Iceberg dans Amazon S3. Grâce à cette fonctionnalité, vous pouvez acheminer les enregistrements d'un seul flux vers différentes tables Apache Iceberg et appliquer automatiquement des opérations d'insertion, de mise à jour et de suppression aux enregistrements des tables Apache Iceberg. Cette fonctionnalité nécessite l'utilisation du AWS Glue Data Catalog.

> **Note**
>
> Les tables Apache Iceberg sont disponibles dans l'est des États-Unis (Virginie du Nord), dans l'ouest des États-Unis (Oregon), en Europe (Irlande), en Asie-Pacifique (Tokyo), au Canada (centre) et en Asie-Pacifique (Sydney) Régions AWS pour un aperçu.

Considérations et limites

La prise en charge par Firehose des tables Apache Iceberg en version préliminaire comporte les considérations et limites suivantes.

- Débit — Si vous utilisez Direct PUT comme source pour fournir des données aux tables Apache Iceberg, le débit maximal par flux est de 5 Mi/seconde dans les régions USA Est (Virginie du Nord), USA Ouest (Oregon) et Europe (Irlande) et de 1 Mio par seconde dans les régions Asie-Pacifique (Tokyo), Canada (Centre) et Asie-Pacifique (Sydney). Si vous souhaitez simplement insérer des

données dans des tables Iceberg sans mises à jour ni suppressions et si vous souhaitez tester avec un débit plus élevé pour votre flux, vous pouvez utiliser le formulaire Firehose Limits pour demander une augmentation de la limite de débit.

- Colonnes — Pour les noms et les valeurs des colonnes, Firehose prend uniquement le premier niveau de nœuds d'un nœud imbriqué à plusieurs niveaux. JSON Par exemple, Firehose sélectionne les nœuds disponibles au premier niveau, y compris le champ de position. Les noms des colonnes et les types de données des données sources doivent correspondre à ceux des tables cibles pour que Firehose puisse être correctement diffusé. Dans ce cas, Firehose s'attend à ce que vous ayez une colonne de type de données de structure ou de carte dans vos tables Iceberg correspondant au champ de position. Firehose prend en charge 16 niveaux d'imbrication. Voici un exemple de modèle imbriqué. JSON

```
{
   "version":"2016-04-01",
   "deviceId":"<solution_unique_device_id>",
   "sensorId":"<device_sensor_id>",
   "timestamp":"2024-01-11T20:42:45.000Z",
   "value":"<actual_value>",
   "position":{
      "x":143.595901,
      "y":476.399628,
      "z":0.24234876
   }
}
```

  Si les noms des colonnes ou les types de données ne correspondent pas, Firehose génère une erreur et envoie les données au compartiment d'erreur S3. Si tous les noms de colonnes et types de données correspondent dans les tables Apache Iceberg, mais qu'un nouveau champ supplémentaire est présent dans l'enregistrement source, Firehose ignore le nouveau champ.
- Un JSON objet par enregistrement — Vous ne pouvez envoyer qu'un seul JSON objet dans un enregistrement Firehose. Si vous agrégez et envoyez plusieurs JSON objets dans un enregistrement, Firehose génère une erreur et envoie les données au compartiment d'erreur S3.
- Sources de streaming — Firehose ne prend actuellement pas en charge Amazon Managed Streaming pour Apache Kafka en tant que source pour les tables Apache Iceberg.
- Compaction — Chaque fois que vous écrivez avec Firehose, des fichiers de données sont générés. Le fait de disposer de milliers de petits fichiers de données augmente la surcharge de métadonnées et affecte les performances de lecture. Pour obtenir des performances de requête optimales, vous pouvez envisager une solution qui prend régulièrement de petits fichiers de

données et les réécrit dans des fichiers de données moins volumineux. Ce processus s'appelle le compactage. AWS Glue Data Catalog prend en charge le compactage automatique de vos tables Apache Iceberg. Pour plus d'informations, consultez la section Gestion du compactage dans AWS Guide de l'utilisateur de Glue. Pour plus d'informations, consultez la section Compaction automatique des tables Apache Iceberg. Vous pouvez également optimiser les tables Iceberg en réduisant la consommation de stockage grâce à une VACUUMinstruction qui effectue la maintenance des tables sur les tables Apache Iceberg.

Création de tables Apache Iceberg

Vous pouvez utiliser … AWS Glue pour créer des tables Apache Iceberg. Pour plus d'informations, consultez la section Création de tables Apache Iceberg.

> **Note**
>
> Il n'est pas nécessaire de créer des tables Iceberg avant de créer un stream Firehose. Vous pouvez également ajouter des tables et modifier le schéma des tables après avoir terminé le flux de travail de création de flux Firehose.

Si l'icône AWS Glue la console de votre région prend en charge le compactage et l'attachement d'un IAM rôle à partir de la console elle-même. Vous pouvez ensuite activer le compactage et ajouter le IAM rôle dans le cadre de AWS Glue Création de tables.

Si l'icône AWS Glue la console de votre région ne prend pas en charge le compactage et l'attachement d'un IAM rôle, puis suivez les étapes décrites dans la section Pour accorder les autorisations de Lake Formation dans la section du tableau Iceberg.

> **Note**
>
> - Version de format de tableau — Firehose ne prend en charge que le format de tableau V2. Ne créez pas de tables au format V1, sinon vous obtiendrez une erreur et les données seront envoyées au compartiment d'erreur S3 à la place.
> - Format de stockage des données : Firehose écrit les données dans les tables Apache Iceberg au format Parquet.

- Fonctionnement au niveau des lignes : Firehose prend en charge le mode Merge-on-read (MOR) pour écrire des données dans les tables Apache Iceberg.

Configurer le stream Firehose

Pour créer un flux Firehose avec les tables Apache Iceberg comme destination, vous devez configurer les éléments suivants.

Conditions requises pour utiliser les tables Apache Iceberg comme destination

Avant de commencer, remplissez les conditions préalables suivantes.

- Création d'un compartiment Amazon S3 : vous devez créer un compartiment Amazon S3 pour ajouter le chemin du fichier de métadonnées lors de la création des tables. Pour plus d'informations, consultez la section Création d'un compartiment S3.
- Créer un IAM rôle avec les autorisations requises — Firehose a besoin d'un IAM rôle avec des autorisations d'accès spécifiques AWS Glue tables et écriture de données sur Amazon S3. Le même rôle est utilisé pour octroyer AWS Glue accès aux compartiments Amazon S3. Vous avez besoin de ce IAM rôle lorsque vous créez une table Iceberg et un stream Firehose. Pour de plus amples informations, veuillez consulter the section called “Accorder à Firehose l'accès à une destination Apache Iceberg Tables ”.

Configuration de la source et de la destination

Pour fournir des données aux tables Apache Iceberg, choisissez la source de votre flux. Actuellement, nous ne prenons pas en charge Amazon MSK en tant que source de diffusion de flux vers les tables Apache Iceberg.

Si vous choisissez Amazon Kinesis Data Streams comme source, dans les paramètres Source de la console, parcourez ou créez les Kinesis Data Streams ou saisissez le ARN flux dans le format. `arn:aws:kinesis:[Region]:[AccountId]:stream/[StreamName]`

Choisissez ensuite les tables Apache Iceberg comme destination et fournissez un nom de flux Firehose.

Configuration de la transformation des données

Pour effectuer des transformations personnalisées sur vos données, telles que l'ajout ou la modification d'enregistrements dans votre flux entrant, vous pouvez ajouter une fonction Lambda à votre flux Firehose. Pour plus d'informations sur la transformation des données à l'aide de Lambda dans un flux Firehose, consultez. Transformez les données sources dans Amazon Data Firehose

Pour les tables Apache Iceberg, vous devez spécifier la manière dont vous souhaitez mapper les données de votre flux entrant à des tables et colonnes spécifiques de la destination. Vous devez également spécifier la manière dont vous souhaitez acheminer les enregistrements entrants vers les différentes tables de destination. Vous pouvez le faire de l'une des manières suivantes.

- Formatez les enregistrements dans votre producteur de données avant de les ingérer dans Firehose.
- Formatez les enregistrements à l'aide d'une fonction Lambda.

Pour de plus amples informations, veuillez consulter Formater les enregistrements d'entrée.

Catalogue de données Connect

Apache Iceberg a besoin d'un catalogue de données pour spécifier la définition des tables et des colonnes. Firehose s'intègre à AWS Glue Data Catalog pour les tables Apache Iceberg.

Si vous utilisez un métastore Hive, voir Connexion du catalogue de données à un métastore Hive externe pour savoir comment vous connecter AWS Glue Data Catalog avec votre Hive Metastore.

Vous pouvez soit utiliser AWS Glue Data Catalog dans le même compte et dans la même région que votre stream Firehose (par défaut), ou choisissez une autre région parmi Régions AWSmenu déroulant dans la console.

Configurer des clés uniques

Les clés uniques sont un ou plusieurs champs de votre enregistrement source qui identifient de manière unique une ligne dans les tables Apache Iceberg. La présence de clés uniques dans vos enregistrements source est facultative. Si vous souhaitez effectuer des opérations de mise à jour et de suppression sur certaines tables, vous devez disposer de clés uniques dans vos enregistrements source qui entrent dans ces tables et vous devez configurer des clés uniques pour les tables requises.

Vous pouvez soit configurer des clés uniques par table dans le cadre de la création du stream Firehose, soit les définir identifier-field-idsnativement dans Iceberg lors de la création ou de la modification d'une table. La configuration de clés uniques par table lors de la création du flux est facultative. Si vous ne configurez pas de clés uniques par table lors de la création du stream, Firehose vérifie `identifier-field-ids` les tables requises et les utilisera comme clés uniques. Si les deux ne sont pas configurés, la livraison des données avec les opérations de mise à jour et de suppression échoue.

Spécifier la durée de la nouvelle tentative

Vous pouvez utiliser cette configuration pour spécifier la durée en secondes pendant laquelle Firehose doit tenter de réessayer en cas d'échec lors de l'écriture dans les tables Apache Iceberg dans Amazon S3. Vous pouvez définir une valeur comprise entre 0 et 7 200 secondes pour effectuer de nouvelles tentatives. Par défaut, Firehose réessaie pendant 300 secondes.

Gérer les échecs de livraison ou de traitement

Vous devez configurer Firehose pour qu'il envoie des enregistrements à un compartiment de sauvegarde S3 au cas où il rencontrerait des échecs lors du traitement ou de la diffusion d'un flux après expiration du délai de nouvelle tentative. Pour cela, configurez le préfixe de sortie d'erreur du compartiment de sauvegarde S3 et le préfixe de sortie d'erreur du compartiment de sauvegarde S3 dans les paramètres de sauvegarde de la console.

Configurer les indices de mémoire tampon

Firehose met en mémoire tampon les données de streaming entrantes en mémoire jusqu'à une certaine taille (taille de la mémoire tampon) et pendant une certaine période (intervalle de mise en mémoire tampon) avant de les transmettre aux tables Apache Iceberg. Vous pouvez choisir une taille de mémoire tampon de 1 à 128 MiBs et un intervalle de mémoire tampon de 0 à 900 secondes. Des indices de mémoire tampon plus élevés permettent de réduire le nombre d'écritures S3, de réduire les coûts de compactage grâce à des fichiers de données plus volumineux et d'accélérer l'exécution des requêtes, mais avec une latence plus élevée. Des valeurs indicatives de mémoire tampon plus faibles fournissent les données avec une latence plus faible.

Configurer les paramètres avancés

Vous pouvez configurer le chiffrement côté serveur, la journalisation des erreurs, les autorisations et les balises pour vos tables Apache Iceberg. Pour de plus amples informations, veuillez consulter

Configurer les paramètres avancés. Dans les paramètres avancés, sous Accès au service, vous devez ajouter le IAM rôle que vous avez créé dans le cadre duthe section called “Conditions requises pour utiliser les tables Apache Iceberg comme destination”. Firehose assumera le rôle d'accès AWS Glue tables et écriture dans des compartiments Amazon S3.

La création d'un stream Firehose peut prendre plusieurs minutes. Pour toute erreur liée à la création de flux, consultezthe section called “Résoudre les erreurs ”. Après avoir créé avec succès le flux Firehose, vous pouvez commencer à y ingérer des données et les afficher dans les tables Apache Iceberg à l'aide de AWS Glue console.

Résoudre les erreurs

La section suivante décrit les différents types d'erreurs susceptibles de se produire lorsque vous utilisez les tables Apache Iceberg comme destination.

Erreurs de diffusion

Les tableaux suivants décrivent les différents types d'erreurs de diffusion.

Message d'erreur	Description
Firehose ne peut pas assumer le rôle < >roleArn. Vérifiez le rôle fourni.	Firehose ne peut pas assumer le rôle fourni.
Le rôle %s n'est pas autorisé à exécuter glue : GetTable on catalogId %s ou il n'existe pas.	Soit la table Glue n'existe pas, soit le rôle fourni n'a pas accès à la table. Pour résoudre ce problème, vérifiez vos autorisations.
Un firehose avec MSK comme source n'est pas pris en charge pour Iceberg en tant que destination.	DeliveryStreamType ne doit pas l'êtreMSKasSource.

Message d'erreur	Description
< UpdateDestination > DestinationTableConfigurationList ne peut pas être mis à jour.	DestinationTableConfigurationList ne doit pas être mis à jour pendant UpdateDestination
< UpdateDestination > Le catalogue ARN ne peut pas être mis à jour.	GlueCatalogARNne peut pas être mis à jour pendant UpdateDestination.

Erreurs de livraison d'Apache Iceberg

Firehose envoie toutes les erreurs de livraison au compartiment d'erreurs CloudWatch Logs et S3.

Code d’erreur	Message et informations sur les erreurs
Iceberg. NoSuchTable La table spécifiée n'existe pas.	Firehose essaie d'écrire dans une table qui n'existe pas.
Iceberg. InvalidTableName Nom de table non valide : null/vide ou la table Iceberg n'est pas la version v2.	Le nom de la base de données/de la table n'est pas valide conformément aux validations par défaut de la spécification Iceberg.
S3. AccessDenied Assurez-vous que la politique de confiance associée au IAM rôle fourni autorise Firehose à assumer le rôle et que la	Assurez-vous que le IAM rôle créé à l'étape des prérequis dispose des autorisations et de la politique de confiance requises.

Code d’erreur	Message et informations sur les erreurs
politique d'accès autorise l'accès au compartiment S3.	
Glue. AccessDenied Assurez-vous que la politique de confiance pour le IAM rôle fourni permet à Firehose d'assumer le rôle et que la politique d'accès autorise l'accès au compartiment S3	Assurez-vous que le IAM rôle créé à l'étape des prérequis dispose des autorisations et de la politique de confiance requises.

Surveiller les métriques

Pour la livraison de données aux tables Apache Iceberg, Firehose émet les métriques CloudWatch suivantes au niveau du flux.

Métrique	Description
`DeliveryToIceberg.Bytes`	Nombre d'octets fournis aux tables Apache Iceberg au cours de la période spécifiée. Unités : octets
`DeliveryToIceberg. IncomingRowCount`	Nombre d'enregistrements que Firehose tente de fournir aux tables Apache Iceberg. Unités : nombre
`DeliveryToIceberg. SuccessfulRowCount`	Nombre de lignes correctement envoyées aux tables Apache Iceberg. Unités : nombre

Métrique	Description
`DeliveryToIceberg.FailedRowCount`	Nombre de lignes défaillantes envoyées au compartiment de sauvegarde S3. Unités : nombre
`DeliveryToIceberg.DataFreshness`	L'âge (depuis le début de Firehose jusqu'à aujourd'hui) du plus vieux disque de Firehose. Tout enregistrement datant de plus de cet âge a été envoyé aux tables Apache Iceberg. Unités : secondes
`DeliveryToIceberg.Success`	Somme des validations réussies dans les tables Apache Iceberg.

Formater les enregistrements d'entrée

Vous pouvez ingérer des données dans Firehose avec PUT `PutRecord` Direct et/ou Amazon `PutRecordBatch` APIs Kinesis Data Streams. PutRecords API Actuellement, pour les deux sources, vous devez ajouter votre JSON objet sous forme de blob dans le champ Data, comme indiqué dans l'exemple suivant.

Exemple d'JSONobjet

```
{
    "courseid": "5",
    "coursename": "Computer Science",
    "semester": "fall",
    "grade": "A"
 }
```

JSONsous forme de blob ajouté au champ de données de l'enregistrement d'entrée

```
{
    "Data": blob // byte array of JSON string as shown in previous sample
 }
```

Lorsque vous utilisez Firehose pour envoyer des flux vers les tables Apache Iceberg, vous devez spécifier la manière dont vous souhaitez acheminer les enregistrements entrants vers les différentes tables de destination et les opérations à effectuer. Pour cela, Firehose `ADF_Metadat` a besoin de trois champs de métadonnées dans votre enregistrement de streaming, comme indiqué dans l'exemple suivant. Vous devez également ajouter votre dans JSON le cadre de`ADF_Record`, comme indiqué dans l'exemple suivant.

Exemple JSON d'objet formaté

```
{
    "ADF_Record": {
        "courseid": "5",
        "coursename": "Computer Science",
        "semester": "fall",
        "grade": "A"
    },
    "ADF_Metadata": {
        "OTF_Metadata": {
            "DestinationTableName": "iceberg_testing", //Required
            "DestinationDatabaseName": "course", // Required
            "Operation": "INSERT" // Optional (Possible values-INSERT, UPDATE, DELETE)
        }
    }
}
```

Dans l'exemple ci-dessus, `DestinationDatabaseName` il s'agit du nom de la base de données cible pour cet enregistrement et d'un champ obligatoire. `DestinationTableNameest` le nom de la table Apache Iceberg pour cet enregistrement et est un champ obligatoire. `Operationest` un champ facultatif avec des valeurs possibles telles que, *INSERT* (par défaut) UPDATE, et DELETE. Spécifiez les valeurs en fonction des opérations que vous souhaitez effectuer sur cet enregistrement. Pour effectuer UPDATE ces DELETE opérations, vous devez configurer les clés uniques pour les tables requises lorsque vous créez un flux Firehose.

> **Note**
>
> Le `DestinationTableName` et `DestinationDatabasename` doit correspondre exactement aux noms de votre base de données et de table Iceberg. S'ils ne correspondent pas, Firehose génère une erreur et transmet les données ayant échoué au compartiment de sauvegarde S3.

Après avoir formaté les données comme indiqué dans l'exemple précédent, vous devez ajouter le blob de cet JSON objet au `Data` champ comme suit.

```
{
    "Data": blob // byte array of JSON string shown in previous example
 }
```

Vous pouvez formater l'enregistrement en streaming de l'une des manières suivantes.

Formater avec Lambda

Vous pouvez formater les enregistrements de flux ingérés avec les champs obligatoires pour Firehose à l'aide d'une fonction Lambda. Pour cela, vous pouvez utiliser l'exemple de code python suivant. Cette fonction Lambda analyse les enregistrements de flux entrants et ajoute des champs obligatoires pour spécifier la manière dont les données doivent être écrites dans des tables et des colonnes spécifiques. Si vous utilisez une fonction Lambda, vous pouvez également analyser les JSON enregistrements imbriqués de votre flux de données et extraire les données que vous souhaitez écrire dans les tables et colonnes spécifiques de la table Iceberg de destination.

Par exemple, si l'enregistrement de votre flux entrant avant le formatage ressemble à cecisample JSON, après le formatage avec l'exemple de code suivant, l'enregistrement (`new_data`dans le bloc de code) ressemble à ceciformatted sample JSON.

```
import base64
import json

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 = {}
    # Create result object.
    firehose_records_output['records'] = []


    # Go through records and process them
    for firehose_record_input in firehose_records_input['records']:
        json_bytes = base64.b64decode(firehose_record_input['data']).decode('utf-8')
```

```
        original_data = json.loads(json_bytes)
        new_data = {'ADF_Record' : original_data}
        adf_metadata = json.loads('{"OTF_Metadata": {"DestinationTableName":"course",
 "DestinationDatabaseName": "iceberg_testing","Operation":"insert"}}')
        new_data["ADF_Metadata"] = adf_metadata
        firehose_record_output = {}

        # Must set proper record ID
        firehose_record_output['recordId'] = firehose_record_input['recordId']
        firehose_record_output['result'] =  'Ok'
        modified_json_bytes = json.dumps(new_data)
        firehose_record_output['data'] =
 base64.b64encode(modified_json_bytes.encode('utf-8'))
        firehose_records_output['records'].append(firehose_record_output)

    return firehose_records_output
```

Format avec producteur de données

Si vous souhaitez formater les enregistrements de flux entrants avec les champs obligatoires dans votre générateur de données avant de les intégrer dans Firehose, vous pouvez utiliser l'exemple de code python suivant.

```
import boto3
import json


delivery_stream_name = 'xxxxxx'
print("Putting data to " + delivery_stream_name)
firehose_client = boto3.client('firehose', region_name='eu-west-1')   # Replace
 'region_name' with your AWS region

def load_from_local():
    records = []
    courseids = ["5"]
    coursenames = ["Computer Science"]
    for j in range(0, 1):
        # change here to any json you need to upload to iceberg
        raw_json = '{"courseid":"%s","coursename": "%s","semester":"fall","grade":"A"}'
 % (courseids[j], coursenames[j])
        print(raw_json)

        original_data = json.loads(raw_json)
```

```
        metadata = '{"OTF_Metadata": {"DestinationTableName":"course",
 "DestinationDatabaseName": "iceberg_testing","Operation":"insert"}}'
        adf_metadata = json.loads(metadata)

        iceberg_data = {'ADF_Record' : original_data}
        iceberg_data["ADF_Metadata"] = adf_metadata

        modified_json_bytes = json.dumps(iceberg_data)
        record = {'Data': modified_json_bytes}
        records.append(record)

    response =
 firehose_client.put_record_batch(DeliveryStreamName=delivery_stream_name,
 Records=records)
    print(response)

load_from_local()
```

Par exemple, si votre enregistrement de flux entrant `original_data` dans l'exemple de code suivant ressemble à ceci sample JSON avant le formatage, puis après le formatage avec le code suivant, l'enregistrement `iceberg_data` ressemblera à ceciformatted sample JSON.

Comprendre les types de données pris en charge

Firehose prend en charge tous les types de données primitifs et complexes pris en charge par Apache Iceberg. Pour plus d'informations, consultez Schémas et types de données. Lorsque vous envoyez des données binaires sous forme de chaîne, vous devez utiliser les types de codage compatibles avec Firehose : Base64 de base, Base64, Base64, URL et MIME Hex, dont le nom de fichier est sécurisé. Pour les types de données d'horodatage, vous devez toujours envoyer en microsecondes.

Exemples de types de données

La section suivante présente des exemples de différents types de données.

MapType

```
{
    "destination_column_0":
 {"WP5o0JOkuIQcDPcsvpJJygF1xzaOSq0wUlgTwuIeCEzgVneGxA":"PO3ReF3auyDqbfonx9Cd8NTmcQnqnw7JuZOCWwI
```

```
    "destination_column_1": "{\"{\\\"destination_nested_column_0\\\": \
\\"18:56:14.974\\\", \\\"destination_nested_column_1\\\": 241.86246}\":
\"M07kAvYdHvBh61F7RzfxtEd39YQI33LnM2NbGS67DOFFsRUyUUujKT5VnK7Wtfz1mHNeIix6FAY9cYpwTdedgr9XnFwG0
\",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\
\\", \\\"destination_nested_column_1\\\": 562.56384}\":
\"9GlxhDCt95LxBo51HybBZihqOqf6EU8jrDu7NMpxtGB2dY6q6kXpvxIrFuMdqHCJKIZIcDikwggLniUm8kgE4d
\",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\
\\", \\\"destination_nested_column_1\\\": 496.03268}\":
\"keTJZYLNvLRB50DMKzEI6M0AM4mueyNnA1m2YVnYdDwyxUpPqkb72Q6LiX0B9s8gCjZ6trW6C1PFk9KNBIpxYsj5Tc5Xs
\",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\\\", \\
\"destination_nested_column_1\\\": 559.0878}\":
\"mG0ZET84BUF28E312UCIWgmypyQFSUODH9NAMAnF3LJEutbooZWcBt97PP5AHaopNvC8pQZ4mGXB9hmVmjUNmuj5Qanyx
\",\"{\\\"destination_nested_column_0\\\": \\\"18:56:14.974\
\\", \\\"destination_nested_column_1\\\": 106.845245}\":
\"aidoVYrzu8gcLRkVVUyTKCN9gqTUFYi8uJQsrXEFEYl1f9ool7JhAtg9QKG5BBu67Ngb95ENsNKQyCHNImSu5x4hMnmHU
\"}"
}
```


DecimalType


```
{
    "destination_column_0": 9455262425851.1342772,
    "destination_column_1": "9455262425851.1342772",
    "destination_column_2": 9455262425852
}
```


BinaryType (base64 par défaut, base64 mime, base64 url-safe, hexadécimal)


```
{
    "destination_column_0": "AsYhnHD\/Ra54hITl1daNV9glOjtWPEfopH
+PjgUKHYB6K7UcYi4K19b80wD4J\/93x5tyh+0y
+k5cMljVRlmfIkIuLxl9ERBiPPLhf4+yoJ2k70VavPnYWmNLs1hLDHlfeEMIfVhrqOGzJMoA
+CBAWXfIuiG420JSQP5iAx5xFG\/
mOfkM5zYothje8OGXltdthcCL6WYBiP0SlwXcE0uMeRfwclAc9fTOBz6RzdJlHhUDjoAXg
+4cvly27F82XpuGMNwpUj98AOrgbh2MoU9yvsM9ZrjD0eGVgOZP8Ky7Za4oE\/oK8j
+qABF6XV712iA6pVtTNJFvX6Ey3ssNYvno+LYF5ZsySs2rB5AbVM73RfOPqdS\/c\/
r3MEqoEqt+nPx6eGam4WSA+0swztt7aLdrlX6yK7xJeIJ0rTlIDBo0ZUaw011ykY
\/8Bvy+4byoPlmr4Z5yhN1z3ZTOkx7eDR6xMv+vDVsDBtItVazDwHgDy41r
\/hQNeNedPKrozc8TY9k7wZre\/6V2lCa3BmT8Uu9b9ydjR9z+fCSdG
+VRv35nz5kdqdKy8YIrynYs4eOcjh8jH3UwVYrYQcnWkBAfF7Xk9CoPVnL3ciHZtyiZOaTGIj9rO0xX\/
W5dGe9\/4YChs6LbD584kxLTxvHgSl4vadaTGNKci3SvNmZNsz8ducxtNXF\/Tv2DUub465hzgpaLPur3+MB
+kfdN2YXUfqb
+xJAgxThWfUe151nrH0EPow9lgSlp21rUBGznJAvPRl1ExGIAuc7JYAoUrJUkx5Hfl6PekPDhqt7+yJwCB8qxhTTryxo
```

```
+bjtai4ndRCGcuCaxT8KkOcXsS37urd3YGSDMinZdMNVc646s25415qK6nBRlqqAY8+EYmcUIVB9XcNdke4zoUfhVQoruwi
\/kFafoulo5DEoMOyaH1N2HCSxG5tZXNQocSZPaY8efZYMCpmDXsPAzkmgSkYRDSu\/r3wUqROa2tGK5\/
pQY24v+Jq0U\/jQ99GShlU283nZ85ot2ocbtMAgD\/WsrSEh6lNt9RaI3HfA7\/HcH\/
fgr9jsTtxDgZhabTBwwDwX0zjWGx1bCuTLKBN7byxg9ZvAVgqwPS4HERLer5T5UkKf74zn9Eq3HYH1Q5JpyDUx
+im7mte1sprf1+A24kksVU\/MD9aP9N8\/QDsQ13gkhOn5KwFMz3BC2Vw5gL
+gGNHFKDRL6wGIfhuYcx9LucolZ1yNy9Gbb3ioWSSufyFpyXqtndDLPI5QS1SJpJm2KDyqcH1SmRLIhd9MNRUC73EAEm
+NO5wxPzBRSjhCHZpf8SrYITWJl7K3XzGOfPFh2NgES3jMP9cvSXO6yyICcep2HBYGbFflni89+Rw==",
    "destination_column_1": "AsYhnHD\/Ra54hITl1daNV9glOjtWPEfopH
+PjgUKHYB6K7UcYi4K19b80wD4J\/93x5tyh+0y+k5c\r
\nMljVRlmfIkIuLxl9ERBiPPLhf4+yoJ2k70VavPnYWmNLs1hLDHlfeEMIfVhrqOGzJMoA+CBAWXfI\r
\nuiG420JSQP5iAx5xFG\/mOfkM5zYothje8OGXltdthcCL6WYBiP0SlwXcE0uMeRfwclAc9fTOBz6R\r
\nzdJlHhUDjoAXg+4cvly27F82XpuGMNwpUj98AOrgbh2MoU9yvsM9ZrjD0eGVgOZP8Ky7Za4oE\/oK\r
\n8j+qABF6XV712iA6pVtTNJFvX6Ey3ssNYvno+LYF5ZsySs2rB5AbVM73RfOPqdS\/c\/r3MEqoEqt+
\r\nnPx6eGam4WSA+0swztt7aLdrlX6yK7xJeIJ0rTlIDBo0ZUaw011ykY\/8Bvy+4byoPlmr4Z5yhN1z
\r\n3ZTOkx7eDR6xMv+vDVsDBtItVazDwHgDy41r\/hQNeNedPKrozc8TY9k7wZre\/6V2lCa3BmT8Uu9b
\r\n9ydjR9z+fCSdG+VRv35nz5kdqdKy8YIrynYs4eOcjh8jH3UwVYrYQcnWkBAfF7Xk9CoPVnL3ciHZ
\r\ntyiZOaTGIj9rO0xX\/W5dGe9\/4YChs6LbD584kxLTxvHgSl4vadaTGNKci3SvNmZNsz8ducxtNXF
\/\r\nTv2DUub465hzgpaLPur3+MB+kfdN2YXUfqb+xJAgxThWfUe151nrH0EPow9lgSlp21rUBGznJAvP
\r\nRl1ExGIAuc7JYAoUrJUkx5Hfl6PekPDhqt7+yJwCB8qxhTTryxo+bjtai4ndRCGcuCaxT8KkOcXs\r
\nS37urd3YGSDMinZdMNVc646s25415qK6nBRlqqAY8+EYmcUIVB9XcNdke4zoUfhVQoruwidzDU\/k\r
\nFafoulo5DEoMOyaH1N2HCSxG5tZXNQocSZPaY8efZYMCpmDXsPAzkmgSkYRDSu\/r3wUqROa2tGK5\r
\n\/pQY24v+Jq0U\/jQ99GShlU283nZ85ot2ocbtMAgD\/WsrSEh6lNt9RaI3HfA7\/HcH\/fgr9jsTtxDg
\r\nZhabTBwwDwX0zjWGx1bCuTLKBN7byxg9ZvAVgqwPS4HERLer5T5UkKf74zn9Eq3HYH1Q5JpyDUx+\r
\nim7mte1sprf1+A24kksVU\/MD9aP9N8\/QDsQ13gkhOn5KwFMz3BC2Vw5gL+gGNHFKDRL6wGIfhuYc
\r\nx9LucolZ1yNy9Gbb3ioWSSufyFpyXqtndDLPI5QS1SJpJm2KDyqcH1SmRLIhd9MNRUC73EAEm+NO\r
\n5wxPzBRSjhCHZpf8SrYITWJl7K3XzGOfPFh2NgES3jMP9cvSXO6yyICcep2HBYGbFflni89+Rw==",
    "destination_column_2": "AsYhnHD_Ra54hITl1daNV9glOjtWPEfopH-
PjgUKHYB6K7UcYi4K19b80wD4J_93x5tyh-0y-k5cMljVRlmfIkIuLxl9ERBiPPLhf4-
yoJ2k70VavPnYWmNLs1hLDHlfeEMIfVhrqOGzJMoA-
CBAWXfIuiG420JSQP5iAx5xFG_mOfkM5zYothje8OGXltdthcCL6WYBiP0SlwXcE0uMeRfwclAc9fTOBz6RzdJlHhUDjoAX
qABF6XV712iA6pVtTNJFvX6Ey3ssNYvno-LYF5ZsySs2rB5AbVM73RfOPqdS_c_r3MEqoEqt-
nPx6eGam4WSA-0swztt7aLdrlX6yK7xJeIJ0rTlIDBo0ZUaw011ykY_8Bvy-4byoPlmr4Z5yhN1z3ZTOkx7eDR6xMv-
vDVsDBtItVazDwHgDy41r_hQNeNedPKrozc8TY9k7wZre_6V2lCa3BmT8Uu9b9ydjR9z-fCSdG-
VRv35nz5kdqdKy8YIrynYs4eOcjh8jH3UwVYrYQcnWkBAfF7Xk9CoPVnL3ciHZtyiZOaTGIj9rO0xX_W5dGe9_4YChs6LbD
MB-kfdN2YXUfqb-
xJAgxThWfUe151nrH0EPow9lgSlp21rUBGznJAvPRl1ExGIAuc7JYAoUrJUkx5Hfl6PekPDhqt7-
yJwCB8qxhTTryxo-bjtai4ndRCGcuCaxT8KkOcXsS37urd3YGSDMinZdMNVc646s25415qK6nBRlqqAY8-
EYmcUIVB9XcNdke4zoUfhVQoruwidzDU_kFafoulo5DEoMOyaH1N2HCSxG5tZXNQocSZPaY8efZYMCpmDXsPAzkmgSkYRDS
Jq0U_jQ99GShlU283nZ85ot2ocbtMAgD_WsrSEh6lNt9RaI3HfA7_HcH_fgr9jsTtxDgZhabTBwwDwX0zjWGx1bCuTLKBN7
im7mte1sprf1-A24kksVU_MD9aP9N8_QDsQ13gkhOn5KwFMz3BC2Vw5gL-
gGNHFKDRL6wGIfhuYcx9LucolZ1yNy9Gbb3ioWSSufyFpyXqtndDLPI5QS1SJpJm2KDyqcH1SmRLIhd9MNRUC73EAEm-
NO5wxPzBRSjhCHZpf8SrYITWJl7K3XzGOfPFh2NgES3jMP9cvSXO6yyICcep2HBYGbFflni89-Rw==",
    "destination_column_3":
 "02c6219c70ff45ae788484e5d5d68d57d8253a3b563c47e8a47f8f8e050a1d807a2bb51c622e0ad7d6fcd300f827f
```

```
}
```

TimeType (Epoch en microsecondes, objet LocalTime Java)

```
{
    "destination_column_0": 68175096000,
    "destination_column_1": "18:56:15.096"
}
```

TimestampType. withZone (Epoch en microsecondes, objet OffsetDateTime Java, objet LocalDateTime Java)

```
{
    "destination_column_0": 1725476175099000,
    "destination_column_1": "2024-09-04T18:56:15.099Z",
    "destination_column_2": "2024-09-04T18:56:15.099"
}
```

DoubleType

```
{
    "destination_column_0": 9.18477568715142,
    "destination_column_1": "9.18477568715142"
}
```

BooleanType

```
{
    "destination_column_0": true,
    "destination_column_1": "false",
    "destination_column_2": 1,
    "destination_column_3": 0
}
```

FloatType

```
{
    "destination_column_0": 0.6242226,
    "destination_column_1": "0.6242226"
}
```

IntegerType

```
{
    "destination_column_0": 7,
    "destination_column_1": "7"
}
```

TimestampType. withoutZone (Epoch en microsecondes, objet LocalDateTime Java, objet OffsetDateTime Java, objet ZonedDateTime Java)

```
{
    "destination_column_0": 1725476175114000,
    "destination_column_1": "2024-09-04T18:56:15.114",
    "destination_column_2": "2024-09-04T18:56:15.114Z",
    "destination_column_3": "2024-09-04T18:56:15.114-07:00"
}
```

DateType

```
{
    "destination_column_0": 19970,
    "destination_column_1": "2024-09-04"
}
```

LongType

```
{
    "destination_column_0": 8,
    "destination_column_1": "8"
}
```

UUIDType(Objet UUID Java)

```
{
    "destination_column_0": "21c5521c-a6d4-48d4-b2c8-7f6d842f72c3"
}
```

ListType

```
{
```

```
    "destination_column_0":
 ["s1FSrgb0lGDxfn2iYT0EtlP47aHSjwmLZgrdr1JqRs0dmbeCcQoaLr4Xhi2KIVvmus9ppFdpWIcOHnJ0omhAPhXH0yns
    "destination_column_1": "[{\"destination_nested_column_0\":\"bb00f8e6-
db82-4241-a5c5-0d9c0d2f71a4\",\"destination_nested_column_1\":907.35345},
{\"destination_nested_column_0\":\"2c77b702-d405-4fe1-beee-fb541d7ab833\",
\"destination_nested_column_1\":544.0026},{\"destination_nested_column_0\":
\"68389200-d6b1-413d-bcd9-fdb931708395\",\"destination_nested_column_1\":153.683},
{\"destination_nested_column_0\":\"bc31cbaa-39cd-4e2f-b357-9ea9ce75532b\",
\"destination_nested_column_1\":977.5165},{\"destination_nested_column_0\":
\"b7d627f9-0d5b-41b7-903a-525488259fba\",\"destination_nested_column_1\":434.17215},
{\"destination_nested_column_0\":\"06b6ec1e-1952-4582-b285-46aaf40064b8\",
\"destination_nested_column_1\":580.33124},{\"destination_nested_column_0\":
\"f04b3bbf-61ad-4c5c-8740-6f666f57c431\",\"destination_nested_column_1\":550.75793}]"
}
```

Marquer un stream 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.

Vous pouvez spécifier des balises lorsque vous invoquez un nouveau stream Firehose CreateDeliveryStreampour 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

Comprendre les principes de base des tags

Vous pouvez utiliser les API opérations Amazon Data Firehose pour effectuer les tâches suivantes :

- Ajoutez des tags à un stream Firehose.
- Répertoriez 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 l'`CreateDeliveryStream`action, Amazon Data Firehose octroie une autorisation supplémentaire à l'`firehose:TagDeliveryStream`action 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 IAM des balises de ressources échoueront avec un résultat `AccessDeniedException` tel que celui-ci.

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

Suivez les coûts grâce au 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 .

Connaître les restrictions relatives aux tags

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 : `_ . / = + - @`.

Sécurité dans Amazon Data Firehose

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

La sécurité est une responsabilité partagée entre AWS et toi. Le modèle de responsabilité partagée décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud — AWS est chargé de protéger l'infrastructure qui fonctionne AWS services dans le AWS Nuage. 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 du AWS programmes de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Data Firehose, voir AWS Services visés par le 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 des 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 avec AWS Secrets Manager dans Amazon Data Firehose
- Gérez IAM les rôles via la console Amazon Data Firehose
- Comprendre la conformité pour Amazon Data Firehose
- Résilience dans Amazon Data Firehose
- Comprendre la sécurité de l'infrastructure dans Amazon Data Firehose
- Mettre en œuvre les meilleures 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 d'un 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 AWS Key Management Serviceet vérifie l'intégrité des données à l'aide de la vérification par somme de contrôle.

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 FIPS 140 à 2 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou unAPI, utilisez un FIPS point de terminaison. Pour plus d'informations sur les FIPS points de terminaison disponibles, voir Federal Information Processing Standard (FIPS) 140-2.

Chiffrement côté serveur avec Kinesis Data Streams

Lorsque vous envoyez des données de vos producteurs de données vers votre flux de données, Kinesis Data Streams chiffre vos données à l'aide d'un AWS Key Management Service (AWS KMS) avant de stocker les données 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 Direct PUT ou d'autres sources de données

Si vous envoyez des données à votre flux Firehose en utilisant PutRecordou 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'StartDeliveryStreamEncryptionaide de cette opération.

Pour arrêter server-side-encryption, utilisez l'StopDeliveryStreamEncryptionopération.

Vous pouvez également l'activer SSE lorsque vous créez le stream Firehose. Pour ce faire, spécifiez à DeliveryStreamEncryptionConfigurationInputquel moment vous invoquez CreateDeliveryStream.

Lorsque CMK c'est le casCUSTOMER_MANAGED_CMK, si le service Amazon Data Firehose n'est pas en mesure de déchiffrer les enregistrements à cause de aKMSNotFoundException, aKMSInvalidStateException, a KMSDisabledException ou aKMSAccessDeniedException, le service attend jusqu'à 24 heures (période de conservation) 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 la méthode symétrique. CMKs Amazon Data Firehose ne prend pas en charge l'asymétrie. CMKs Pour plus d'informations sur la symétrie et l'asymétrieCMKs, reportez-vous à la section À propos de la symétrie et de l'asymétrie dans le CMKs AWS Key Management Service guide du développeur.

> **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 une occurrence où une clé appartenant à votre AWS le compte a été utilisé, il est enregistré dans le cadre de AWS CloudTrail journaux d'événements pour votre

AWS . 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 fournir des données vers une destination appartenant à un autre AWS . La technologie permettant de gérer toutes ces formes d'accès est AWS Identity and Access Management (IAM). Pour plus d'informationsIAM, voir Qu'est-ce que c'est IAM ? .

Table des matières

- Accordez l'accès à vos ressources Firehose
- Accordez à Firehose l'accès à votre cluster Amazon privé MSK
- Permettre à Firehose d'assumer un rôle IAM
- Accordez à Firehose l'accès à AWS Glue pour la conversion de formats de données
- Accorder à Firehose l'accès à une destination Amazon S3
- Accorder à Firehose l'accès à une destination Apache Iceberg Tables
- Accorder à Firehose l'accès à une destination Amazon Redshift
- Accorder à Firehose l'accès à une destination de service public OpenSearch
- Accorder à Firehose l'accès à une destination OpenSearch de service dans un VPC
- Accorder à Firehose l'accès à une destination publique sans serveur OpenSearch
- Accordez à Firehose l'accès à une destination OpenSearch sans serveur dans un VPC
- Accorder à Firehose l'accès à une destination Splunk
- Accès à Splunk dans VPC
- Ingérez les journaux de VPC flux dans Splunk à l'aide d'Amazon Data Firehose

- Accès à Snowflake ou au point de terminaison HTTP
- Accordez à Firehose l'accès à une destination Snowflake
- Accès à Snowflake dans VPC
- Accorder à Firehose l'accès à une destination de point de terminaison HTTP
- Livraison entre comptes depuis Amazon MSK
- Livraison entre comptes vers une destination Amazon S3
- Livraison entre comptes vers une destination OpenSearch de service
- Utilisation de balises pour contrôler l'accès

Accordez l'accès à vos ressources Firehose

Pour permettre à votre application d'accéder à votre stream Firehose, utilisez une politique similaire à cet exemple. Vous pouvez ajuster les API opérations individuelles auxquelles vous accordez l'accès en modifiant la `Action` section ou en accordant 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 Firehose l'accès à votre cluster Amazon privé MSK

Si la source de votre flux Firehose est un MSK cluster Amazon 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"
    }
  ]
}
```

Vous devez ajouter une politique comme celle-ci à la politique basée sur les ressources du cluster pour accorder au principal du service Firehose l'autorisation d'invoquer l'opération Amazon. MSK `CreateVpcConnection` API

Permettre à Firehose d'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 permettant de créer un nouveau rôle, AWS attache la politique de confiance requise au rôle. Si vous souhaitez qu'Amazon Data Firehose utilise un IAM rôle 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 à remplacer *`account-id`* avec votre AWS identifiant de 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 IAM rôle 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 contextuelle de `sts:ExternalId` condition pour garantir que seule l'activité Amazon Data Firehose provient de votre AWS le compte peut assumer ce IAM rôle. Pour plus d'informations sur la prévention de l'utilisation non autorisée des IAM rôles, consultez la section Le problème de confusion des adjoints dans le guide de IAM l'utilisateur.

Si vous choisissez Amazon MSK comme source pour votre flux Firehose, vous devez spécifier un autre IAM rôle qui accorde à Amazon Data Firehose l'autorisation d'ingérer les données sources du cluster Amazon spécifié. MSK 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 MSK cluster Amazon 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"
       }]
}
```

Accordez à Firehose l'accès à AWS Glue pour la conversion de formats de données

Si votre flux Firehose effectue une conversion de format de données, Amazon Data Firehose fait référence aux définitions de tables stockées dans AWS Glue. Pour donner à Amazon Data Firehose l'accès nécessaire à AWS Glue, ajoutez la déclaration suivante à votre politique. Pour plus

d'informations sur la façon de trouver le ARN nom de la table, voir Spécification AWS Ressource Glue 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 IAMexemples de désérialiseurs dans le AWS Glue .

Accorder à 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 un 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 avoir un IAM rôle lors de la création d'un stream Firehose. Amazon Data Firehose assume ce IAM rôle 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 et AWS KMS clé. 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 ce qui suit :

```
{
   "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 pour accéder à votre AWS ressources, voir Création d'un rôle pour déléguer des autorisations à un AWS Service indiqué dans le guide de IAM l'utilisateur.

Pour savoir comment accorder à Amazon Data Firehose l'accès à une destination Amazon S3 via un autre compte, consultez. the section called “Livraison entre comptes vers une destination Amazon S3”

Accorder à Firehose l'accès à une destination Apache Iceberg Tables

Vous devez avoir un IAM rôle avant de créer un stream Firehose et des tables Apache Iceberg en utilisant AWS Glue. Suivez les étapes ci-dessous pour créer une politique et un IAM rôle. Firehose assume ce IAM rôle et exécute les actions requises.

1. Connectez-vous au AWS Management Console et ouvrez la IAM console à l'adresse https://console.aws.amazon.com/iam/.
2. Créez une politique et choisissez-la JSONdans l'éditeur de stratégie.
3. Ajoutez la politique en ligne suivante qui accorde à Amazon S3 des autorisations telles que les autorisations de lecture/écriture, les autorisations de mise à jour de la table dans le catalogue de données, etc.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetTable",
                "glue:GetDatabase",
                "glue:UpdateTable"
            ],
            "Resource": [
                "arn:aws:glue:<region>:<aws-account-id>:catalog",
                "arn:aws:glue:<region>:<aws-account-id>:database/*",
                "arn:aws:glue:<region>:<aws-account-id>:table/*/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:GetShardIterator",
                "kinesis:GetRecords",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:<region>:<aws-account-id>:stream/<stream-
name>"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:<region>:<aws-account-id>:key/<key-id>"
            ],
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3.region.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-
bucket/prefix*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:<region>:<aws-account-id>:log-group:<log-group-
name>:log-stream:<log-stream-name>"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": [
                "arn:aws:lambda:<region>:<aws-account-id>:function:<function-
name>:<function-version>"
            ]
        }
    ]
}
```

Cette politique contient une déclaration qui autorise l'accès à Amazon Kinesis Data Streams, l'appel des fonctions Lambda et l'accès aux clés. KMS Si vous n'utilisez aucune de ces ressources, vous pouvez supprimer les instructions correspondantes.

Si la journalisation des erreurs est activée, Firehose envoie également des erreurs de livraison de données à votre groupe de CloudWatch journaux et à vos flux. Pour cela, vous devez configurer les noms des groupes de journaux et des flux de journaux. Pour les noms des groupes de journaux et des flux de journaux, consultezSurveillez Amazon Data Firehose à l'aide des journaux CloudWatch .

4. Dans les politiques intégrées, remplacez *DOC-EXAMPLE-BUCKET* avec le nom de votre compartiment Amazon S3 aws-account-id et votre région avec un Compte AWS numéro et région des ressources.

> **Note**
>
> Ce rôle autorise toutes les bases de données et tables de votre catalogue de données. Si vous le souhaitez, vous ne pouvez accorder des autorisations qu'à des tables et à des bases de données spécifiques.

5. Après avoir créé la politique, ouvrez la IAMconsole et créez un IAM rôle avec Service AWSen tant que type d'entité de confiance.
6. Pour Service ou cas d'utilisation, choisissez Kinesis. Dans le cas d'utilisation, choisissez Kinesis Firehose.
7. Sur la page suivante, choisissez la politique créée à l'étape précédente à associer à ce rôle. Sur la page de révision, vous trouverez une politique de confiance déjà attachée à ce rôle qui autorise le service Firehose à assumer ce rôle. Lorsque vous créez le rôle, Amazon Data Firehose peut assumer qu'il effectue les opérations requises sur AWS Glue et des seaux S3.

Accorder à 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

- IAMrôle et politique d'accès

- VPCaccès à un cluster provisionné par Amazon Redshift ou à un groupe de travail Amazon Redshift Serverless

IAMrôle et politique 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 un 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 IAM un rôle pour accéder au bucket, à la clé, au groupe de journaux et aux flux spécifiés. CloudWatch Vous devez avoir un IAM rôle 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 et AWS KMS clé. 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 pour accéder à votre AWS ressources, voir Création d'un rôle pour déléguer des autorisations à un AWS Service indiqué dans le guide de IAM l'utilisateur.

VPCaccès à un cluster provisionné par Amazon Redshift ou à un groupe de travail Amazon Redshift Serverless

Si votre cluster provisionné Amazon Redshift ou votre groupe de travail Amazon Redshift Serverless se trouve dans un cloud privé virtuel (VPC), il doit être accessible au public 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 CIDR bloc pour chaque région disponible.

Région	CIDRblocs
USA Est (Ohio)	`13.58.135.96/27`
USA Est (Virginie du Nord)	`52.70.63.192/27`
USA Ouest (Californie du Nord)	`13.57.135.192/27`
US West (Oregon)	`52.89.255.224/27`
AWS GovCloud (USA Est)	`18.253.138.96/27`

Région	CIDRblocs
AWS GovCloud (US-Ouest)	52.61.204.160/27
Canada (Centre)	35.183.92.128/27
Canada Ouest (Calgary)	40.176.98.192/27
Asie-Pacifique (Hong Kong)	18.162.221.32/27
Asie-Pacifique (Mumbai)	13.232.67.32/27
Asie-Pacifique (Hyderabad)	18.60.192.128/27
Asie-Pacifique (Séoul)	13.209.1.64/27
Asie-Pacifique (Singapour)	13.228.64.192/27
Asie-Pacifique (Sydney)	13.210.67.224/27
Asie-Pacifique (Jakarta)	108.136.221.64/27
Asie-Pacifique (Tokyo)	13.113.196.224/27
Asie-Pacifique (Osaka)	13.208.177.192/27
Chine (Beijing)	52.81.151.32/27
Chine (Ningxia)	161.189.23.64/27
Europe (Zurich)	16.62.183.32/27
Europe (Francfort)	35.158.127.160/27
Europe (Irlande)	52.19.239.192/27
Europe (Londres)	18.130.1.96/27

Région	CIDRblocs
Europe (Paris)	35.180.1.96/27
Europe (Stockholm)	13.53.63.224/27
Moyen-Orient (Bahreïn)	15.185.91.0/27
Amérique du Sud (Sao Paulo)	18.228.1.128/27
Europe (Milan)	15.161.135.128/27
Afrique (Le Cap)	13.244.121.224/27
Moyen-Orient (UAE)	3.28.159.32/27
Israël (Tel Aviv)	51.16.102.0/27
Asie-Pacifique (Melbourne)	16.50.161.128/27
Asie-Pacifique (Malaisie)	43.216.58.0/27

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

Accorder à 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 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 IAM rôle pour accéder au domaine de OpenSearch service spécifié, le compartiment S3, AWS KMS clé, groupe de CloudWatch journaux et flux. Vous devez avoir un IAM rôle 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 de service et AWS KMS 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 pour accéder à votre AWS ressources, voir Création d'un rôle pour déléguer des autorisations à un AWS Service indiqué dans le guide de IAM l'utilisateur.

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

Si votre domaine de OpenSearch service se trouve dans unVPC, 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 à vos domaines de OpenSearch service. VPC

- `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 de l'interroger ou de le mettre à jour. ENIs

> **Important**
>
> Lorsque vous spécifiez des sous-réseaux pour transmettre des données à la destination dans un environnement privéVPC, 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 ENIs ajouter de données pour la livraison de données en mode 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 HTTPS trafic 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 HTTPS trafic 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 la section Règles des groupes de sécurité dans la VPC documentation Amazon.

Accorder à 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 IAM rôle pour accéder à la collection OpenSearch Serverless spécifiée, le compartiment S3, AWS KMS clé, groupe de CloudWatch journaux et flux. Vous devez avoir un IAM rôle 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 AWS KMS 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": [
                "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 pour accéder à votre AWS ressources, voir Création d'un rôle pour déléguer des autorisations à un AWS Service indiqué dans le guide de IAM l'utilisateur.

Accordez à Firehose l'accès à une destination OpenSearch sans serveur dans un VPC

Si votre collection OpenSearch Serverless se trouve dans unVPC, 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 à votre collection OpenSearch Serverless. VPC

- `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 de l'interroger ou de le mettre à jour. ENIs

> **Important**
>
> Lorsque vous spécifiez des sous-réseaux pour transmettre des données à la destination dans un environnement privéVPC, 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 ENIs ajouter de données pour la livraison de données en mode 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 HTTPS trafic 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 HTTPS trafic 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 stream Firehose et pour la collection OpenSearch Serverless, 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 la section Règles des groupes de sécurité dans la VPC documentation Amazon.

Accorder à Firehose l'accès à une destination Splunk

Lorsque vous utilisez une destination Splunk, Amazon Data Firehose fournit des données à votre point de terminaison HTTP Splunk Event Collector HEC (). Il sauvegarde également ces données dans le compartiment Amazon S3 que vous spécifiez, et vous pouvez éventuellement utiliser un 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 utiliser AWS Lambda pour la transformation des données.

Si vous utilisez un AWS load balancer, 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 IAM rôle lorsque vous créez un stream Firehose. Firehose assume ce IAM rôle 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 AWS Lambda pour la transformation 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 ne permet pas d'accéder IAM à Splunk. Pour accéder à Splunk, il utilise votre HEC jeton.

```
{
    "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 pour accéder à votre AWS ressources, voir Création d'un rôle pour déléguer des autorisations à un AWS Service indiqué dans le guide de IAM l'utilisateur.

Accès à Splunk dans VPC

Si votre plateforme Splunk se trouve dans unVPC, elle doit être accessible au public 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 suivantsCIDR.

Région	CIDRblocs
USA Est (Ohio)	`18.216.68.160/27, 18.216.170.64/27, 18.216.170.96/27 \`
USA Est (Virginie du Nord)	`34.238.188.128/26, 34.238.188.192/26, 34.238.195.0/26`
USA Ouest (Californie du Nord)	`13.57.180.0/26`
US West (Oregon)	`34.216.24.32/27, 34.216.24.192/27, 34.216.24.224/27`
AWS GovCloud (USA Est)	`18.253.138.192/26`
AWS GovCloud (US-Ouest)	`52.61.204.192/26`
Asie-Pacifique (Hong Kong)	`18.162.221.64/26`
Asie-Pacifique (Mumbai)	`13.232.67.64/26`
Asie-Pacifique (Séoul)	`13.209.71.0/26`
Asie-Pacifique (Singapour)	`13.229.187.128/26`
Asie-Pacifique (Sydney)	`13.211.12.0/26`
Asie-Pacifique (Tokyo)	`13.230.21.0/27, 13.230.21.32/27`
Canada (Centre)	`35.183.92.64/26`
Canada Ouest (Calgary)	`40.176.98.128/26`
Europe (Francfort)	`18.194.95.192/27, 18.194.95.224/27, 18.195.48.0/27`

Région	CIDRblocs
Europe (Irlande)	34.241.197.32/27, 34.241.197.64/27, 34.241.197.96/27
Europe (Londres)	18.130.91.0/26
Europe (Paris)	35.180.112.0/26
Europe (Stockholm)	13.53.191.0/26
Moyen-Orient (Bahreïn)	15.185.91.64/26
Amérique du Sud (Sao Paulo)	18.228.1.192/26
Europe (Milan)	15.161.135.192/26
Afrique (Le Cap)	13.244.165.128/26
Asie-Pacifique (Osaka)	13.208.217.0/26
Chine (Beijing)	52.81.151.64/26
Chine (Ningxia)	161.189.23.128/26
Asie-Pacifique (Jakarta)	108.136.221.128/26
Moyen-Orient (UAE)	3.28.159.64/26
Israël (Tel Aviv)	51.16.102.64/26
Europe (Zurich)	16.62.183.64/26
Asie-Pacifique (Hyderabad)	18.60.192.192/26
Asie-Pacifique (Melbourne)	16.50.161.192/26
Asie-Pacifique (Malaisie)	43.216.44.192/26

Ingérez les journaux de VPC flux dans Splunk à l'aide d'Amazon Data Firehose

Pour en savoir plus sur la façon de créer un abonnement aux journaux de VPC flux, de publier sur Firehose et d'envoyer les journaux de VPC flux vers une destination prise en charge, consultez Ingérer les journaux de VPC flux dans Splunk à l'aide d'Amazon Data Firehose.

Accès à Snowflake ou au point de terminaison HTTP

Il n'existe aucun sous-ensemble de AWS Plages d'adresses IP spécifiques à Amazon Data Firehose lorsque la destination est un point de HTTP terminaison ou des clusters publics Snowflake.

Pour ajouter Firehose à une liste d'autorisation pour les clusters Snowflake publics, à votre public HTTP ou à vos HTTPS points de terminaison, ajoutez tous les AWS Les plages d'adresses IP correspondent à vos règles d'entrée.

> **Note**
>
> Les notifications ne proviennent pas toujours des mêmes adresses IP AWS Région comme sujet associé. Vous devez inclure le AWS Plage d'adresses IP pour toutes les régions.

Accordez à Firehose l'accès à une destination Snowflake

Lorsque vous utilisez Snowflake comme destination, Firehose fournit des données à un compte Snowflake à l'aide de votre compte Snowflake. URL Il sauvegarde également les données d'erreur dans le bucket Amazon Simple Storage Service que vous spécifiez, et vous pouvez éventuellement utiliser un 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 IAM rôle avant de créer un stream Firehose. Firehose assume ce IAM rôle et accède au bucket, à la clé, au groupe de CloudWatch logs 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:PutObjectAcl` 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 ne permet pas d'accéder IAM 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 pour accéder à votre AWS ressources, voir Création d'un rôle pour déléguer des autorisations à un AWS Service indiqué dans le guide de IAM l'utilisateur.

Accès à Snowflake dans VPC

Si votre cluster Snowflake est activé par les liens privés, Firehose utilise des VPC points de terminaison 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 pour `AwsVpceIds` Région AWS votre cluster est intégré. Pour plus d'informations, consultez la section Création d'une règle réseau dans le guide de l'utilisateur de Snowflake.

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

Région AWS	**VPCE IDs**
USA Est (Ohio)	vpce-0d96cafcd96a50aeb vpce-0cec34343d48f537b

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

Région AWS	**VPCE IDs**
	vpce-00961862a21b033da vpce-01620b9ae33273587 vpce-078cf4ec226880ac9 vpce-0d711bf076ce56381 vpce-066b7e13cbfca6f6e vpce-0674541252d9ccc26 vpce-03540b88dedb4b000 vpce-0b1828e79ad394b95 vpce-0dc0e6f001fb1a60d vpce-0d8f82e71a244098a vpce-00e374d9e3f1af5ce vpce-0c1e3d6631ddb442f
USA Ouest (Oregon)	vpce-0f60f72da4cd1e4e7 vpce-0c60d21eb8b1669fd vpce-01c4e3e29afdafbef vpce-0cc6bf2a88da139de vpce-0797e08e169e50662 vpce-033cbe480381b5c0e vpce-00debbdd8f9eb10a5 vpce-08ec2f386c809e889 vpce-0856d14310857b545

Région AWS	**VPCE IDs**
Europe (Francfort)	vpce-068dbb7d71c9460fb vpce-0a7a7f095942d4ec9
Europe (Irlande)	vpce-06857e59c005a6276 vpce-04390f4f8778b75f2 vpce-011fd2b1f0aa172fd
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
Asie-Pacifique (Mumbai)	vpce-0d68cb822f6f0db68 vpce-0517d32692ffcbde2
Europe (Londres)	vpce-0fd1874a0ba3b9374 vpce-08091b1a85e206029
Amérique du Sud (Sao Paulo)	vpce-065169b8144e4f12e vpce-0493699f0e5762d63

Région AWS	**VPCE IDs**
Canada (Centre)	vpce-07e6ed81689d5271f vpce-0f53239730541394c
Europe (Paris)	vpce-09419680077e6488a vpce-0ea81ba2c08140c14
Asie-Pacifique (Osaka)	vpce-0a9f003e6a7e38c05 vpce-02886510b897b1c5a
Europe (Stockholm)	vpce-0d96410833219025a vpce-060a32f9a75ba969f
Asie-Pacifique (Jakarta)	vpce-00add4b9a25e5c649 vpce-004ae2de34338a856

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

Vous pouvez utiliser Amazon Data Firehose pour fournir des données à n'importe quelle destination de point de HTTP terminaison. Amazon Data Firehose sauvegarde également ces données dans le compartiment Amazon S3 que vous spécifiez, et vous pouvez éventuellement utiliser un 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 utiliser AWS Lambda pour la transformation des données.

Vous devez avoir un IAM rôle lors de la création d'un stream Firehose. Amazon Data Firehose assume ce IAM rôle 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 AWS Lambda pour la transformation 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 ne permet pas d'accéder aux destinations des HTTP terminaux appartenant IAM à des fournisseurs de services tiers pris en charge, notamment Datadog, Dynatrace, MongoDB, New Relic LogicMonitor, Splunk ou Sumo Logic. Pour accéder à une destination de point de HTTP terminaison spécifiée appartenant à un fournisseur de services tiers pris en charge, contactez ce fournisseur de services pour obtenir la API clé 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 pour accéder à votre AWS ressources, voir Création d'un rôle pour déléguer des autorisations à un AWS Service indiqué dans le guide de IAM l'utilisateur.

> **Important**
>
> Actuellement, Amazon Data Firehose prend en NOT charge la livraison de données aux HTTP points de terminaison 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 dans un autre MSK AWS compte (compte A), vous devez disposer des configurations suivantes.

Compte A :

1. Dans la MSK console Amazon, choisissez le cluster provisionné, puis choisissez Properties.
2. Sous Paramètres réseau, choisissez Modifier et activez la VPCmulticonnectivité.
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. Le 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/DO-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-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/DO-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-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/DO-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*" //topic of
 the cluster
    },
 }
```

4. Sous AWS principal, entrez l'identifiant 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. Dans Paramètres source, pour le cluster Amazon Managed Streaming for Apache Kafka, saisissez le nom ARN du MSK cluster Amazon 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 stream Firehose, vous devez avoir un IAM rôle (créé par défaut lors de l'utilisation du AWS Management Console) qui accorde au stream Firehose un accès en « lecture » au cluster MSK Amazon multi-comptes 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/DO-NOT-TOUCH-mskaas-provisioned-
privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-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/DO-NOT-TOUCH-mskaas-provisioned-
privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/mskaas_test_topic" //topic of the
 cluster
    },
    {
    "Sid": "",
    "Effect": "Allow",
    "Action": [
```

```
        "kafka-cluster:DescribeGroup"
    ],
    "Resource": "arn:aws:kafka:us-east-1:arn:group/DO-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 de plus amples informations, veuillez consulter (Facultatif) Configurer la transformation des enregistrements et la conversion des formats.

Livraison entre comptes vers une destination Amazon S3

Vous pouvez utiliser le plugin AWS CLI ou l'Amazon Data Firehose APIs pour créer un flux Firehose en un AWS compte avec une destination Amazon S3 dans 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 IAM rôle 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 depuis le IAM rôle créé précédemment, créez une politique de compartiment S3 sous 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 IAM rôle que vous avez créé à l'étape 1.

Livraison entre comptes vers une destination OpenSearch de service

Vous pouvez utiliser le plugin AWS CLI ou l'Amazon Data Firehose APIs pour créer un flux Firehose en un AWS compte avec une destination OpenSearch de service dans 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 IAM rôle sous le compte A en suivant les étapes décrites dansthe section called “Accorder à Firehose l'accès à une destination de service public OpenSearch ”.
2. Pour autoriser l'accès depuis le IAM rôle que vous avez créé à l'étape précédente, créez une politique de OpenSearch service sous le compte B. JSON Voici 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 IAM rôle que vous avez créé à l'étape 1. Lorsque vous créez le stream Firehose, utilisez AWS CLI ou Amazon Data Firehose APIs et spécifiez le `ClusterEndpoint` champ plutôt que `DomainARN` pour OpenSearch Service.

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

Utilisation de balises pour contrôler l'accès

Vous pouvez utiliser l'`Condition`élément facultatif (ou le `Condition` bloc) d'une IAM politique pour affiner l'accès aux opérations d'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'`Condition`élément et les opérateurs que vous pouvez utiliser au sein de celui-ci, voir Éléments de IAM JSON politique : 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 Comprendre les principes de base des tags.

```
{
    "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

Pour toutes les opérations Firehose autres que`CreateDeliveryStream`,, et `TagDeliveryStream` `UntagDeliveryStreamListDeliveryStreams`, 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 à AWS ressources utilisant des balises dans le guide de IAM l'utilisateur.

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

Authentifiez-vous avec AWS Secrets Manager dans Amazon Data Firehose

Amazon Data Firehose s'intègre à 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, non plus dans AWS Management Console ou API paramètres. 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 de plus amples informations, veuillez consulter le .AWS Secrets Manager Guide de l'utilisateur

Rubriques

- Comprenez les secrets
- Créer un secret
- Utilisez le secret
- Faites pivoter le secret

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 OAuth jeton ou toute autre information secrète que vous stockez sous forme cryptée dans Secrets Manager.

Pour chaque destination, vous devez spécifier la paire clé-valeur secrète JSON au format correct, comme indiqué dans la section suivante. Amazon Data Firehose ne parviendra pas à se connecter à votre destination si le JSON format 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 HTTP Endpoint, 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 AWS Secrets Manager Guide de l'utilisateur

Utilisez le secret

Nous vous recommandons d'utiliser AWS Secrets Manager pour stocker vos informations d'identification ou vos clés afin de vous connecter à des destinations de streaming telles qu'Amazon Redshift, HTTP Endpoint, 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 le AWS Console de gestion au moment de la création du stream Firehose. Pour de plus amples informations, veuillez consulter Configuration des paramètres de destination. Vous pouvez également utiliser les UpdateDestinationAPIopérations CreateDeliveryStreamet 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 API la clé 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 .AWS Secrets Managerpage de tarification.

Accordez l'accès à Firehose pour récupérer le secret

Pour que Firehose puisse récupérer un secret auprès de AWS Secrets Manager, vous devez fournir à Firehose les autorisations requises pour accéder au secret et à la clé qui le chiffre.

Lors de l'utilisation AWS Secrets Manager 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 comme votre IAM rôle et il est crypté avec la valeur par défaut AWS managed key (`aws/secretsmanager`), le IAM rôle assumé par Firehose nécessite uniquement 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 IAM les politiques, 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 CMK politique doit également permettre au IAM rôle 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 autre AWS compte plutôt que votre rôle, et il est crypté avec la valeur par défaut AWS clé gérée, cette configuration n'est pas possible car Secrets Manager n'autorise pas l'accès entre comptes lorsque le secret est chiffré avec AWS clé gérée.
- Si le secret est stocké dans un autre compte et crypté avec unCMK, le IAM rôle a besoin d'une `secretsmanager:GetSecretValue` autorisation sur le secret et d'une `kms:Decrypt` autorisation sur leCMK. La politique de ressources du secret et CMK celle de l'autre compte doivent également accorder au IAM rôle 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 automatiquement pivoter le secret pour vous 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 secrets dans le AWS Secrets Manager Guide de l'utilisateur

Gérez IAM les rôles 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 IAM rôles pour accorder des autorisations à Firehose lorsque vous créez ou modifiez un stream Firehose. Firehose utilise ce IAM rôle 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 AWS Lambda fonctionnalité activée. Dans ce cas, vous devez fournir des IAM rôles 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 IAM rôle existant
- Création d'un nouveau IAM rôle depuis la console

Choisissez un IAM rôle existant

Vous pouvez choisir parmi un IAM rôle existant. Avec cette option, assurez-vous que le IAM rôle que vous choisissez dispose d'une politique de confiance appropriée et que les autorisations requises pour votre source et votre destination sont requises. Pour de plus amples informations, veuillez consulter Contrôler l'accès avec Amazon Data Firehose.

Création d'un nouveau IAM rôle depuis la console

Vous pouvez également utiliser la console Firehose pour créer un nouveau rôle en votre nom.

Lorsque Firehose crée un IAM rôle 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 stream Firehose.

Par exemple, si vous n'avez pas activé Transformer les enregistrements source avec AWS Lambdafonctionnalité puis 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**
>
> Vous pouvez ignorer les déclarations de politique qui s'y trouvent`%FIREHOSE_POLICY_TEMPLATE_PLACEHOLDER%`, 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 IAM rôle. La politique de confiance permet à Firehose d'assumer le IAM rôle. 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 IAM rôle géré par la console pour plusieurs streams Firehose. Dans le cas contraire, le IAM rôle pourrait devenir trop permissif ou entraîner des erreurs.

- Pour utiliser différentes déclarations de politique dans une politique d'autorisation à partir d'un IAM rôle géré par la console, vous pouvez créer votre propre IAM rôle et copier les déclarations de politique dans une politique d'autorisation attachée au nouveau rôle. Pour associer le rôle au stream Firehose, sélectionnez l'option Choisir un IAM rôle existant dans l'accès au service.
- La console gère tout IAM rôle contenant la chaîne service-role dans son. ARN Lorsque vous choisissez l'option de IAM rôle existant, assurez-vous de sélectionner un IAM rôle sans la chaîne service-role ARN afin que la console n'y apporte aucune modification.

Étapes pour créer un IAM rôle depuis la console

1. Ouvrez la console Firehose à l'adresse. https://console.aws.amazon.com/firehose/
2. Choisissez Create Firehose stream.
3. Choisissez une source et une destination. Pour de plus amples informations, veuillez consulter Tutoriel : Création d'un stream Firehose depuis la console.
4. Choisissez les paramètres de destination. Pour de plus amples informations, veuillez consulter Configuration des paramètres de destination.
5. Sous Paramètres avancés, pour Accès au service, choisissez Créer ou mettre à jour IAM un rôle.

   > **Note**
   >
   > Il s'agit d'une option par défaut. Pour utiliser un rôle existant, sélectionnez l'option Choisir IAM un rôle existant. La console Firehose n'apportera aucune modification à votre propre rôle.

6. Choisissez Create Firehose stream.

Modifier IAM le rôle 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 Transform source records avec AWS Lambdafonctionnalité utilisant la dernière version de la fonction Lambda

carexampleLambdaFunction, 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 IAM rôle 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.

Étapes pour modifier IAM le rôle depuis la console

1. Ouvrez la console Firehose à l'adresse. 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 IAM rôle.

   Note

   Par défaut, la console met toujours à jour un IAM rôle en y ajoutant le modèle service-role. ARN Lorsque vous choisissez l'option de IAM rôle existant, assurez-vous de sélectionner un IAM rôle sans la chaîne service-role ARN afin que la console n'y apporte aucune modification.

5. Sélectionnez Enregistrer les modifications.

Comprendre la conformité pour Amazon Data Firehose

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon Data Firehose dans le cadre de plusieurs AWS programmes de conformité. Il s'agit SOC PCI notamment de RAMP la Fed HIPAA et d'autres.

Pour une liste des AWS services dans le cadre de programmes de conformité spécifiques, voir AWS Services visés par le programme de conformité. Pour des informations générales, voir AWS Programmes de conformité.

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 queHIPAA, ou PCI Fed, RAMP AWS fournit des ressources pour vous aider à :

- 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 au HIPAA service de la sécurité et de la conformité — Ce livre blanc décrit comment les entreprises peuvent utiliser AWS pour créer des applications HIPAA conformes.
- 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— Ceci AWS le 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— Ceci AWS le service fournit une vue complète de votre état de sécurité au sein de AWS qui vous aide à vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience dans Amazon Data Firehose

Le AWS l'infrastructure mondiale est construite autour de AWS Régions et zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées,

reliées par un réseau à latence faible, 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 Régions et zones de disponibilité, voir AWS Infrastructure mondiale.

En plus du AWS infrastructure mondiale, Data Firehose propose plusieurs fonctionnalités pour vous aider à 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 de la zone 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.

Comprendre la sécurité de l'infrastructure dans Amazon Data Firehose

En tant que service géré, Amazon Data Firehose est protégé par AWS sécurité du réseau mondial. Pour plus d'informations sur AWS services de sécurité et comment AWS protège l'infrastructure, voir AWS Sécurité dans le cloud. Pour concevoir votre AWS environnement utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, voir Protection de l'infrastructure dans le pilier de sécurité AWS Framework bien conçu.

Vous utilisez AWS APIappels publiés pour accéder à Firehose via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Sécurité de la couche de transport (TLS). Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Des suites de chiffrement parfaitement confidentielles (PFS) telles que (Ephemeral Diffie-Hellman) ou DHE 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 identifiant de clé d'accès et d'une clé d'accès secrète associés à un IAM principal. Ou vous pouvez utiliser AWS Security Token Service (AWS STS) pour générer des informations de sécurité temporaires pour signer les demandes.

> **Note**
>
> Pour les HTTPS demandes sortantes, Amazon Data Firehose utilise une HTTP bibliothèque qui sélectionne automatiquement la version de TLS protocole la plus élevée prise en charge côté destination.

Utilisation d'Amazon Data Firehose avec AWS PrivateLink

Vous pouvez utiliser un point de VPC terminaison d'interface (AWS PrivateLink) pour empêcher le trafic entre votre Amazon VPC et Amazon Data Firehose de quitter le réseau Amazon. VPCLes points de terminaison de l'interface ne nécessitent pas de passerelle Internet, NAT d'appareil, de VPN connexion ou AWS Direct Connect connexion. Les VPC points de terminaison de l'interface sont alimentés par AWS PrivateLink, une AWS technologie qui permet une communication privée entre AWS services utilisant une interface elastic network avec private IPs sur votre AmazonVPC. Pour en savoir plus, consultez Amazon Virtual Private Cloud.

Utilisation des VPC points de terminaison de l'interface (AWS PrivateLink) pour Firehose

Pour commencer, créez un point de VPC terminaison d'interface afin que le trafic Amazon Data Firehose provenant de vos VPC ressources Amazon commence à passer par le point de terminaison d'interfaceVPC. 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 VPC point de terminaison à Amazon Data Firehose, consultez la section Contrôle de l'accès aux services avec VPC des points de terminaison.

L'exemple suivant montre comment configurer un AWS Lambda fonctionne dans un VPC et créez un VPC point de terminaison 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 VPC

1. Connectez-vous au AWS Management Console et ouvrez la VPC console Amazon à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le VPC tableau de bord, choisissez Endpoints.
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 IAM rôle à utiliser avec la fonction Lambda

1. Ouvrez la IAM console 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, laissez la sélection par défaut AWS service.
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 LambdaVPCAccessExecutionRole` 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.

Créez une fonction Lambda dans VPC

1. Ouvrez le fichier AWS Lambda console à https://console.aws.amazon.com/lambda/.
2. Sélectionnez Create function (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. Sélectionnez Create function (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 Réseau, choisissez l'VPCendroit où 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 lors de sa création.
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 Details (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.
```

Pris en charge Régions AWS

Les VPC points de terminaison 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 (UAE)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Osaka)
- Israël (Tel Aviv)
- Asie-Pacifique (Malaisie)

Mettre en œuvre les meilleures 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.

Utiliser IAM les rôles

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 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 IAM rôle 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 du guide de l'IAMutilisateur :

- IAMRôles
- Scénarios courants pour les rôles : utilisateurs, applications et services.

Implémenter le chiffrement côté serveur dans les 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 Data Firehose.

CloudTrail À utiliser pour surveiller les API appels

Amazon Data Firehose est intégré à AWS CloudTrail, un service qui fournit un enregistrement des actions entreprises 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 de plus amples informations, veuillez consulter the section called “Enregistrez les appels Firehose API”.

Surveillez Amazon Data Firehose

Vous pouvez surveiller Amazon Data Firehose à l'aide des fonctionnalités suivantes :

Rubriques

- Mettre en œuvre les meilleures pratiques en matière d' CloudWatch alarmes
- Surveillez Amazon Data Firehose à l'aide de métriques CloudWatch
- CloudWatch Mesures d'accès pour Amazon Data Firehose
- Surveillez Amazon Data Firehose à l'aide des journaux CloudWatch
- CloudWatch Journaux d'accès pour Amazon Data Firehose
- Surveiller l'état de santé de l'agent Kinesis
- Enregistrez les appels Amazon Data Firehose API avec AWS CloudTrail

Mettre en œuvre les meilleures pratiques en matière d' CloudWatch alarmes

Ajoutez des CloudWatch alarmes lorsque les mesures suivantes dépassent la limite de mise en mémoire tampon (15 minutes maximum).

- `DeliveryToS3.DataFreshness`
- `DeliveryToIceberg.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 Résoudre les erreurs.

Surveillez 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 Surveiller l'état de santé de l'agent Kinesis.

Rubriques

- CloudWatch métriques pour le partitionnement dynamique
- CloudWatch métriques pour la livraison des données
- Métriques d'ingestion de données
- APImétriques au niveau des -niveaux 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
- Métriques d'utilisation d'Amazon Data Firehose

CloudWatch métriques pour le partitionnement dynamique

Si le partitionnement dynamique est activé, l'espace de noms AWS/Firehose inclut les métriques suivantes.

Métrique	Description
`ActivePartitionsLimit`	Le nombre maximum de partitions actives qu'un stream Firehose traite avant d'envoyer des données au compartiment d'erreur. Unités : nombre
`PartitionCount`	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). Unités : nombre
`PartitionCountExceeded`	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.
`JQProcessing.Duration`	Renvoie le temps nécessaire à l'exécution de l'expression JQ dans la fonction Lambda JQ. Unités : millisecondes
`PerPartitionThroughput`	Indique le débit traité par partition. Cette métrique vous permet de surveiller le débit par partition.

Métrique	Description
	Unités : StandardUnit. BytesSecond
`DeliveryToS3.ObjectCount`	Indique le nombre d'objets qui sont livrés à votre compartiment S3. Unités : nombre

CloudWatch métriques pour la livraison des 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

- De la livraison au OpenSearch service
- Livraison en mode OpenSearch Serverless
- Diffusion dans Amazon Redshift
- Diffusion dans Amazon S3
- Livraison à Snowflake
- Remise à Splunk
- Livraison aux points de HTTP terminaison

De la livraison au OpenSearch service

Métrique	Description
`DeliveryToAmazonOpenSearchService.Bytes`	Le nombre d'octets indexés sur le OpenSearch service au cours de la période spécifiée.

Métrique	Description
	Unités : octets
DeliveryToAmazonOpenSearchService.DataFreshness	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. Unités : secondes
DeliveryToAmazonOpenSearchService.Records	Le nombre d'enregistrements indexés sur OpenSearch Service au cours de la période spécifiée. Unités : nombre
DeliveryToAmazonOpenSearchService.Success	Somme des enregistrements correctement indexés.
DeliveryToS3.Bytes	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. Unités : nombre
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. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents. Unités : secondes

Métrique	Description
DeliveryToS3.Records	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. Unités : nombre
DeliveryToS3.Success	La somme des commandes put d'Amazon S3 réussies. Amazon Data Firehose émet toujours cette métrique, que la sauvegarde soit activée uniquement pour les documents défaillants ou pour tous les documents.
DeliveryToAmazonOpenSearchService.AuthFailure	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.
DeliveryToAmazonOpenSearchService.DeliveryRejected	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
DeliveryToAmazonOpenSearchServerless.Bytes	Nombre d'octets indexés sur OpenSearch Serverless au cours de la période spécifiée. Unités : octets
DeliveryToAmazonOpenSearchServerless.DataFreshness	L'âge (depuis l'entrée dans Amazon Data Firehose jusqu'à aujourd'hui) du plus ancien enregistrement dans

Métrique	Description
	Amazon Data Firehose. Tout enregistrement antérieur à cet âge a été livré à OpenSearch Serverless. Unités : secondes
`DeliveryToAmazonOpenSearchServerless.Records`	Le nombre d'enregistrements indexés sur OpenSearch Serverless au cours de la période spécifiée. Unités : nombre
`DeliveryToAmazonOpenSearchServerless.Success`	Somme des enregistrements correctement indexés.
`DeliveryToS3.Bytes`	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. Unités : nombre
`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. Amazon Data Firehose émet cette métrique uniquement lorsque vous activez la sauvegarde de tous les documents. Unités : secondes
`DeliveryToS3.Records`	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. Unités : nombre

Métrique	Description
DeliveryToS3.Success	La somme des commandes put d'Amazon S3 réussies. Amazon Data Firehose émet toujours cette métrique, que la sauvegarde soit activée uniquement pour les documents défaillants ou pour tous les documents.
DeliveryToAmazonOpenSearchServerless.AuthFailure	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 qu'il y a un échec d'authentification.
DeliveryToAmazonOpenSearchServerless.DeliveryRejected	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.

Diffusion dans Amazon Redshift

Métrique	Description
DeliveryToRedshift.Bytes	Le nombre d'octets copiés vers Amazon Redshift au cours de la période spécifiée. Unités : nombre
DeliveryToRedshift.Records	Le nombre d'enregistrements copiés vers Amazon Redshift au cours de la période spécifiée. Unités : nombre
DeliveryToRedshift.Success	La somme des commandes Amazon Redshift COPY réussies.
DeliveryToS3.Bytes	Le nombre d'octets fournis à Amazon S3 au cours de la période spécifiée.

Métrique	Description
	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 d'Amazon S3 réussies.
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 sur Amazon S3 est activée. Unités : nombre
BackupToS3.DataFreshness	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. Unités : secondes
BackupToS3.Records	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. Unités : nombre

Métrique	Description
BackupToS3.Success	Somme des commandes de sauvegarde réussies d'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 d'Amazon S3 réussies.
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 est activée (ce qui n'est possible que lorsque la transformation des données est également activée).

Métrique	Description
	Unités : nombre
`BackupToS3.DataFreshness`	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). Unités : secondes
`BackupToS3.Records`	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 transform ation des données est également activée). Unités : nombre
`BackupToS3.Success`	Somme des commandes de sauvegarde réussies d'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).

Livraison à Snowflake

Métrique	Description
`DeliveryToSnowflake.Bytes`	Le nombre d'octets fournis à Snowflake au cours de la période spécifiée. Unités : octets

Métrique	Description
`DeliveryToSnowflake.DataFreshness`	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 `DeliveryToSnowflake.DataCommitLatency` métrique. Unités : secondes
`DeliveryToSnowflake.DataCommitLatency`	Le temps nécessaire pour que les données soient validées dans Snowflake une fois que Firehose a correctement inséré des enregistrements. Unités : secondes
`DeliveryToSnowflake.Records`	Le nombre d'enregistrements livrés à Snowflake au cours de la période spécifiée. Unités : nombre
`DeliveryToSnowflake.Success`	La somme des appels d'insertion réussis adressés à Snowflake.
`DeliveryToS3.Bytes`	Le nombre d'octets 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 défaillantes sur S3. Unités : octets

Métrique	Description
DeliveryToS3.Records	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 défaillantes sur S3. Unités : nombre
DeliveryToS3.Success	La somme des commandes put d'Amazon S3 réussies. Cette métrique n'est disponible que lorsque la livraison à Snowflake échoue et que Firehose tente de sauvegarder les données défaillantes sur S3.
BackupToS3.DataFreshness	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. Unités : secondes
BackupToS3.Records	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. Unités : nombre
BackupToS3.Bytes	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. Unités : nombre

Métrique	Description
BackupToS3.Success	La somme des commandes de sauvegarde réussies d'Amazon S3. Firehose émet cette métrique lorsque le flux Firehose est configuré pour sauvegarder toutes les données.

Remise à Splunk

Métrique	Description
DeliveryToSplunk.Bytes	Nombre d'octets fournis à Splunk au cours de la période spécifiée. Unités : octets
DeliveryToSplunk.DataAckLatency	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
DeliveryToSplunk.DataFreshness	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. Unités : secondes
DeliveryToSplunk.Records	Nombre d'enregistrements fournis à Splunk au cours de la période spécifiée. Unités : nombre

Métrique	Description
DeliveryToSplunk.Success	Somme des enregistrements correctement indexés.
DeliveryToS3.Success	La somme des commandes put d'Amazon S3 réussies. Cette métrique est émise si la sauvegarde vers Amazon S3 est activée.
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 le flux Firehose est configuré pour sauvegarder tous les documents. Unités : nombre
BackupToS3.DataFreshness	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
BackupToS3.Records	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. Unités : nombre
BackupToS3.Success	Somme des commandes de sauvegarde réussies d'Amazon S3. Amazon Data Firehose émet cette métrique lorsque le flux Firehose est configuré pour sauvegarder tous les documents.

Livraison aux points de HTTP terminaison

Métrique	Description
`DeliveryToHttpEndpoint.Bytes`	Le nombre d'octets transmis avec succès au HTTP point de terminaison. Unités : octets
`DeliveryToHttpEndpoint.Records`	Le nombre d'enregistrements transmis avec succès au HTTP point de terminaison. Unités : nombres
`DeliveryToHttpEndpoint.DataFreshness`	Âge du plus ancien enregistrement dans Amazon Data Firehose. Unités : secondes
`DeliveryToHttpEndpoint.Success`	Somme de toutes les demandes de livraison de données réussies au HTTP point de terminaison. Unités : nombre
`DeliveryToHttpEndpoint.ProcessedBytes`	Le nombre de tentatives de traitement d'octets, y compris les nouvelles tentatives.
`DeliveryToHttpEndpoint.ProcessedRecords`	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
`DataReadFromKinesisStream.Bytes`	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. Unités : octets
`DataReadFromKinesisStream.Records`	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. Unités : nombre
`ThrottledDescribeStream`	Nombre total de fois où l'opération `DescribeStream` est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
`ThrottledGetRecords`	Nombre total de fois où l'opération `GetRecords` est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
`ThrottledGetShardIterator`	Nombre total de fois où l'opération `GetShardIterator` est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
`KinesisMillisBehindLatest`	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.

Métrique	Description
	Unités : millisecondes

Ingestion de données via Direct PUT

Métrique	Description
`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 transform ation des données est activée pour les destinations Amazon S3 ou Amazon Redshift. Unités : octets
`BackupToS3.DataFreshness`	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 transformation des données est activée pour les destinations Amazon S3 ou Amazon Redshift. Unités : secondes
`BackupToS3.Records`	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 transform ation des données est activée pour les destinations Amazon S3 ou Amazon Redshift. Unités : nombre
`BackupToS3.Success`	Somme des commandes de sauvegarde réussies d'Amazon S3. Amazon Data Firehose émet cette métrique lorsque la transformation des données est

Métrique	Description
	activée pour les destinations Amazon S3 ou Amazon Redshift.
BytesPerSecondLimit	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.
DeliveryToAmazonOpenSearchService.Bytes	Le nombre d'octets indexés sur le OpenSearch service au cours de la période spécifiée. Unités : octets
DeliveryToAmazonOpenSearchService.DataFreshness	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. Unités : secondes
DeliveryToAmazonOpenSearchService.Records	Le nombre d'enregistrements indexés sur OpenSearch Service au cours de la période spécifiée. Unités : nombre
DeliveryToAmazonOpenSearchService.Success	Somme des enregistrements correctement indexés.
DeliveryToRedshift.Bytes	Le nombre d'octets copiés vers Amazon Redshift au cours de la période spécifiée. Unités : octets
DeliveryToRedshift.Records	Le nombre d'enregistrements copiés vers Amazon Redshift au cours de la période spécifiée. Unités : nombre

Métrique	Description
DeliveryToRedshift.Success	La somme des commandes Amazon Redshift COPY réussies.
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 d'Amazon S3 réussies.
DeliveryToSplunk.Bytes	Nombre d'octets fournis à Splunk au cours de la période spécifiée. Unités : octets
DeliveryToSplunk.DataAckLatency	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
DeliveryToSplunk.DataFreshness	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. Unités : secondes
DeliveryToSplunk.Records	Nombre d'enregistrements fournis à Splunk au cours de la période spécifiée. Unités : nombre
DeliveryToSplunk.Success	Somme des enregistrements correctement indexés.
IncomingBytes	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 IncomingBytes . Unités : octets
IncomingPutRequests	Le nombre de PutRecordBatch demandes réussies PutRecord et le nombre de demandes sur une période donnée. Unités : nombre
IncomingRecords	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 IncomingRecords . Unités : nombre

Métrique	Description
`RecordsPerSecondLimit`	Le nombre maximum actuel d'enregistrements par seconde qu'un stream Firehose peut ingérer avant d'être limité. Unités : nombre
`ThrottledRecords`	Le nombre d'enregistrements qui ont été limités parce que l'ingestion de données a dépassé l'une des limites du flux Firehose. Unités : nombre

Ingestion de données depuis MSK

Métrique	Description
`DataReadFromSource.Records`	Le nombre d'enregistrements lus à partir de la source Rubrique Kafka. Unités : nombre
`DataReadFromSource.Bytes`	Le nombre d'octets lus à partir de la source Rubrique Kafka. Unités : octets
`SourceThrottled.Delay`	La durée pendant laquelle le cluster Kafka source est retardé pour renvoyer les enregistrements depuis la source Rubrique Kafka. Unités : millisecondes
`BytesPerSecondLimit`	Limite actuelle du débit auquel Firehose va lire chaque partition de la source Rubrique Kafka. Unités : octets/seconde

Métrique	Description
KafkaOffsetLag	Différence entre le plus grand décalage de l'enregis trement 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

APImétriques au niveau des -niveaux CloudWatch

L'AWS/Firehoseespace de noms inclut les métriques de API niveau suivantes.

Métrique	Description
DescribeDeliverySt ream.Latency	Délai nécessaire par opération DescribeDeliverySt ream , mesurée pendant la période spécifiée. Unités : millisecondes
DescribeDeliverySt ream.Requests	Nombre total de demandes DescribeDeliverySt ream .

Métrique	Description
	Unités : nombre
`ListDeliveryStreams.Latency`	Délai nécessaire par opération `ListDeliveryStream` , 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
`PutRecordBatch.Latency`	Délai nécessaire par opération `PutRecordBatch` , mesurée pendant la période spécifiée. Unités : millisecondes

Métrique	Description
PutRecordBatch.Records	Nombre total d'enregistrements issus des opérations PutRecordBatch . Unités : nombre
PutRecordBatch.Requests	Nombre total de demandes PutRecordBatch . Unités : nombre
PutRequestsPerSecondLimit	Le nombre maximum de requêtes put par seconde qu'un stream Firehose peut traiter avant le throttling. Ce nombre inclut les PutRecordBatch demandes PutRecord et les demandes. Unités : nombre
ThrottledDescribeStream	Nombre total de fois où l'opération DescribeStream est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
ThrottledGetRecords	Nombre total de fois où l'opération GetRecords est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
ThrottledGetShardIterator	Nombre total de fois où l'opération GetShardIterator est limitée lorsque la source de données est un flux de données Kinesis. Unités : nombre
UpdateDeliveryStream.Latency	Délai nécessaire par opération UpdateDeliveryStream , mesurée pendant la période spécifiée. Unités : millisecondes

Métrique	Description
`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
`ExecuteProcessing.Duration`	Le temps nécessaire pour chaque appel de fonction Lambda effectué par Firehose. Unités : millisecondes
`ExecuteProcessing.Success`	La somme des invocations de fonction Lambda réussies sur la somme du total des invocations de fonction Lambda.
`SucceedProcessing.Records`	Le nombre d'enregistrements traités avec succès au cours de la période spécifiée. Unités : nombre
`SucceedProcessing.Bytes`	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 `AWS/Firehose` noms 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
`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.

Métrique	Description
	Unités : octets

Métriques de chiffrement côté serveur () SSE CloudWatch

L'`AWS/Firehose`espace de noms inclut les métriques suivantes associées àSSE.

Métrique	Description
`KMSKeyAccessDenied`	Le nombre de fois que le service rencontre un `KMSAccessDeniedException` pour le stream Firehose. Unités : nombre
`KMSKeyDisabled`	Le nombre de fois que le service rencontre un `KMSDisabledException` pour le stream Firehose. Unités : nombre
`KMSKeyInvalidState`	Le nombre de fois que le service rencontre un `KMSInvalidStateException` pour le stream Firehose. Unités : nombre
`KMSKeyNotFound`	Le nombre de fois que le service rencontre un `KMSNotFoundException` pour le stream Firehose. Unités : nombre

Dimensions pour Amazon Data Firehose

Pour filtrer les métriques par flux Firehose, utilisez la `DeliveryStreamName` dimension.

Métriques 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
`ResourceCount`	Nombre des ressources spécifiées exécutées dans votre compte. Les ressources sont définies par les dimensions associées à la métrique. La statistique la plus utile pour cette métrique est`MAXIMUM`, qui représente le nombre maximum de ressources utilisées pendant une période d'une minute.

Les dimensions suivantes sont utilisées pour affiner les statistiques d'utilisation publiées par Amazon Data Firehose.

Dimension	Description
`Service`	Nom du AWS service contenant la ressource. Pour les métriques d'utilisation d'Amazon Data Firehose, la valeur de cette dimension est. `Firehose`
`Class`	Classe de ressource suivie. Les statistiques API d'utilisation d'Amazon Data Firehose utilisent cette dimension avec une valeur de. `None`

Dimension	Description
`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` .

CloudWatch Mesures d'accès 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 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/.
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-statisticsles 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
```

Surveillez Amazon Data Firehose à l'aide des 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 et deux flux de journaux `aws/kinesisfirehose/MyStream` nommés et. `DestinationDelivery` 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 par le biais de AWS CLI API, ou AWS CloudFormation en utilisant 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 IAM politique associée est `"logs:putLogEvents"` autorisée. Pour de plus amples informations, veuillez consulter 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 livraison de données

Erreurs de livraison 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 livraison de données Amazon S3
- Erreurs de livraison de données dans les tables Apache Iceberg
- Erreurs de livraison de données Amazon Redshift
- Erreurs de livraison des données Snowflake
- Erreurs de livraison de données Splunk
- ElasticSearch Erreurs de livraison de données
- HTTPSErreurs de livraison des données du terminal
- 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 livraison 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
`S3.KMS.NotFoundException`	« 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. »
`S3.KMS.RequestLimitExceeded`	« La limite de KMS demandes par seconde 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 .
`S3.AccessDenied`	« Accès refusé. Assurez-vous que la politique de confiance associée au IAM rôle fourni autorise Amazon Data Firehose à assumer le rôle et que la politique d'accès autorise l'accès au compartiment S3. »
`S3.AccountProblem`	« Un problème lié à votre AWS compte empêche le bon déroulement de l'opération. Contacter AWS Support. »
`S3.AllAccessDisabled`	« L'accès au compte fourni a été désactivé. Contactez AWS le Support. »
`S3.InvalidPayer`	« L'accès au compte fourni a été désactivé. Contactez AWS le Support. »
`S3.NotSignedUp`	« Le compte n'est pas inscrit à Amazon S3. S'inscrire au compte ou utiliser un autre compte. »
`S3.NoSuchBucket`	« Le compartiment spécifié n'existe pas. Créez le compartiment ou utilisez un autre compartiment existant. »
`S3.MethodNotAllowed`	« La méthode spécifiée n'est pas autorisée sur cette ressource. Modifiez la politique du compartiment pour autoriser les opérations Amazon S3 correctes. »
`InternalError`	« 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 KMS clé fournie est désactivée. Activez la clé ou utilisez-en une autre. »
`S3.KMS.InvalidStateException`	« La KMS clé fournie n'est pas valide. Veuillez utiliser une autre clé. »
`KMS.InvalidStateException`	« La KMS clé fournie n'est pas valide. Veuillez utiliser une autre clé. »
`KMS.DisabledException`	« La KMS clé 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 IAM rôle et la KMS clé (si fournis) transmis correspondent 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`	« KMS touche non configurée. Configurez KMSMasterKeyID ou désactivez le chiffrement de votre compartiment S3. »
`S3.KMS.AsymmetricCMKNotSupported`	« Amazon S3 ne prend en charge que la symétrieCMKs. Vous ne pouvez pas utiliser un système asymétrique CMK pour chiffrer vos données dans Amazon S3. Pour obtenir le type de votreCMK, utilisez l'KMS DescribeKey opération. »

Code d'erreur	Message et informations sur les erreurs
`S3.IllegalLocationConstraintException`	« 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. »
`S3.InvalidPrefixConfigurationException`	« 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. »
`DataFormatConversion.MalformedData`	« Un caractère illégal a été trouvé entre les jetons. »

Erreurs de livraison de données dans les tables Apache Iceberg

Pour les erreurs de livraison des données des tables Apache Iceberg, consultezTables Apache Iceberg.

Erreurs de livraison 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
`Redshift.TableNotFound`	« La table dans laquelle charger les données est introuvable. Vérifiez que la table spécifiée existe. » 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.

Code d'erreur	Message et informations sur les erreurs
`Redshift.SyntaxError`	« La COPY commande contient une erreur de syntaxe. Relancez la commande. »
`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 relative au IAM rôle fourni autorise Amazon Data Firehose à assumer ce rôle. »
`Redshift.S3BucketAccessDenied`	« La COPY commande n'a pas pu accéder au compartiment S3. Assurez-vous que la politique d'accès pour le IAM rôle fourni autorise l'accès au compartiment S3. »
`Redshift.DataLoadFailed`	« Échec de chargement des données dans la table. Consultez la table ERRORS système STL LOAD _ _ pour plus de détails. »
`Redshift.ColumnNotFound`	« Aucune colonne de la COPY commande n'existe dans le tableau. Spécifiez un nom de colonne valide. »
`Redshift.DatabaseNotFound`	« La base de données spécifiée dans la configuration de destination d'Amazon Redshift JDBC URL est introuvable. Spécifiez un nom de base de données valide. »
`Redshift.IncorrectCopyOptions`	« Des COPY options contradictoires ou redondantes ont été proposées. Certaines options ne sont pas compatibles dans certaines combinaisons. Consultez la référence de COPY commande pour plus d'informations. » Pour plus d'informations, consultez la COPYcommande Amazon Redshift dans le manuel Amazon Redshift Database Developer Guide.
`Redshift.MissingColumn`	« Une colonne est définie dans le schéma de la table comme étant NOT NULL sans DEFAULT valeur et non incluse dans la liste des 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 JDBC URL de destination Amazon Redshift est correct et que le cluster est disponible. »
`Redshift.ColumnMismatch`	« Le nombre de chemins JSON dans la COPY commande 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 COPY commande ou assurez-vous que le compartiment S3 se trouve dans la même région que la base de données Amazon Redshift. »
`Redshift.IncorrectJsonPathsFile`	« Le format du fichier jsonpaths fourni n'est pas pris en chargeJSON. 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 de l'utilisateur Amazon Redshift pour obtenir ce INSERT privilège. »
`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
`InternalError`	« 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. »
`Redshift.ArgumentNotSupported`	« La COPY commande contient des options non prises en charge. »
`Redshift.AnalyzeTableAccessDenied`	« 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. »
`Redshift.SchemaNotFound`	« Le schéma spécifié dans la configuration DataTableName de destination d'Amazon Redshift est introuvable. Spécifiez un nom de schéma valide. »
`Redshift.ColumnSpecifiedMoreThanOnce`	« Une colonne est spécifiée plusieurs fois dans la liste des colonnes. Assurez-vous que les colonnes dupliquées sont supprimées. »
`Redshift.ColumnNotNullWithoutDefault`	« Il existe une colonne non nulle sans DEFAULT qu'elle ne soit incluse dans la liste des colonnes. Assurez-vous que ces colonnes sont incluses dans la liste des colonnes. »
`Redshift.IncorrectBucketRegion`	« 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. »
`Redshift.S3SlowDown`	« 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 un chemin automatique ou un chemin S3 valide pour jsoncopyOption. »
`Redshift.InvalidCopyOptionJSONPathFormat`	« COPY échec avec erreur \" JSONPath Format non valide. L'index du tableau est hors de portée \». Merci de rectifier l'JSONPathexpression. »
`Redshift.InvalidCopyOptionRBACAclNotAllowed`	« COPY échec avec erreur \" Impossible d'utiliser le framework RBAC ACL lorsque 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 le prend pas en charge. SSL Vérifiez les paramètres de votre cluster. »
`Redshift.HoseNotFound`	« Le tuyau a été supprimé. Veuillez vérifier le statut de votre tuyau. »
`Redshift.Delimiter`	« Le copyOptions séparateur indiqué dans le n' copyCommand 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'COPYopération. »
`Redshift.CompressionMismatch`	« Le tuyau est configuré avecUNCOMPRESSED, mais copyOption inclut un format de compression. »
`Redshift.EncryptionCredentials`	« L'ENCRYPTEDoption nécessite des informations d'identification au format : 'aws_iam_role=... ; master_symmetric_key=... 'ou 'aws_access_key_id=... ; aws_secret_access_key=... [; jeton =...] ; master_symmetric_key=... '»
`Redshift.InvalidCopyOptions`	« Options COPY de configuration 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.CannotForkProcess`	« 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 SSL connexion s'est interrompue pendant la poignée de main. »
`Redshift.Resize`	« Le cluster Redshift se redimensionne. Firehose ne sera pas en mesure de fournir des données pendant le redimensionnement du cluster. »
`Redshift.ImproperQualifiedName`	« Le nom qualifié est incorrect (trop de noms séparés par des points). »
`Redshift.InvalidJsonPathFormat`	« JSONPath Format non valide. »
`Redshift.TooManyConnectionsException`	« Trop de connexions à Redshift. »
`Redshift.PSQLException`	« PSQlException observé depuis Redshift. »
`Redshift.DuplicateSecondsSpecification`	« Spécification des secondes dupliquées au format date/heure. »
`Redshift.RelationCouldNotBeOpened`	« 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 des 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 PEM RSA privée 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
`Snowflake.InvalidInput`	« La livraison a échoué en raison d'un format de saisie non valide. Assurez-vous que la charge utile d'entrée fournie est dans le JSON format acceptable. »
`Snowflake.IncorrectValue`	« La livraison a échoué en raison d'un type de données incorrect dans la charge utile d'entrée. Assurez-vous que les JSON valeurs spécifiées dans la charge utile d'entrée correspondent au 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
`Splunk.ProxyWithoutStickySessions`	« Si vous avez un proxy (ELBou autre) entre Amazon Data Firehose et le HEC nœud, vous devez activer les sessions persistantes pour le support HECACKs. »
`Splunk.DisabledToken`	« Le HEC jeton est désactivé. Activez le jeton pour autoriser la livraison de données à Splunk. »
`Splunk.InvalidToken`	« Le HEC jeton n'est pas valide. Mettez à jour Amazon Data Firehose avec un HEC jeton valide. »
`Splunk.InvalidDataFormat`	« Les données ne sont pas formatées correctement. Pour savoir comment formater correctement les données pour les HEC points de terminaison bruts ou événementiels, consultez Splunk Event Data. »
`Splunk.InvalidIndex`	« Le HEC jeton ou l'entrée est configuré avec un index non valide. Vérifiez la configuration de l'index et réessayez. »
`Splunk.ServerError`	« La livraison des données à Splunk a échoué en raison d'une erreur du serveur provenant du HEC nœud. Amazon Data Firehose réessaier a d'envoyer les données si la durée de la nouvelle tentative dans votre

Code d’erreur	Message et informations sur les erreurs
	Amazon Data Firehose est supérieure à 0. Si toutes les tentatives échouent, Amazon Data Firehose sauvegarde les données sur Amazon S3. »
`Splunk.DisabledAck`	« L'accusé de réception de l'indexeur est désactivé pour le HEC jeton. Activez l'accusé de réception indexeur et réessayez. Pour plus d'informations, consultez la section Activer l'accusé de réception indexeur. »
`Splunk.AckTimeout`	« Je n'ai pas reçu d'accusé de réception HEC avant l'expiration du délai HEC imparti. 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. »
`Splunk.MaxRetriesFailed`	« Échec de livraison des données à Splunk ou absence d'accusé de réception. Vérifiez votre HEC état de santé et réessayez. »
`Splunk.ConnectionTimeout`	« 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. »
`Splunk.InvalidEndpoint`	« Impossible de se connecter au HEC terminal. Assurez-vous que le HEC point de terminaison URL est valide et accessible depuis Amazon Data Firehose. »
`Splunk.ConnectionClosed`	« 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. »
`Splunk.SSLUnverified`	« Impossible de se connecter au HEC terminal. L'hôte ne correspond pas au certificat fourni par l'homologue. Vérifiez que le certificat et l'hôte sont valides . »
`Splunk.SSLHandshake`	« Impossible de se connecter au HEC terminal. Vérifiez que le certificat et l'hôte sont valides . »

Code d’erreur	Message et informations sur les erreurs
`Splunk.URLNotFound`	« La demande n'URLa pas été trouvée sur le serveur Splunk. Vérifiez le cluster Splunk et assurez-vous qu'il est correctement configuré. »
`Splunk.ServerError.ContentTooLarge`	« La livraison des données à Splunk a échoué en raison d'une erreur du serveur avec le message suivant statusCode : 413, la demande envoyée par votre client était trop importante. Consultez la documentation de Splunk pour configurer max_content_length. »
`Splunk.IndexerBusy`	« La livraison des données à Splunk a échoué en raison d'une erreur du serveur provenant du HEC nœud. Assurez-vous que le point de HEC terminaison ou l'Elastic Load Balancer est accessible et qu'il fonctionne correctement. »
`Splunk.ConnectionRecycled`	« La connexion entre Firehose et Splunk a été recyclée. La diffusion sera tentée à nouveau. »
`Splunk.AcknowledgementsDisabled`	« Je n'ai pas pu obtenir de remerciements. POST Assurez-vous que les accusés de réception sont activés sur le HEC terminal. »
`Splunk.InvalidHecResponseCharacter`	« Caractères non valides trouvés en HEC réponse, assurez-vous de vérifier le service et la HEC configuration. »

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 IAM rôle fourni associé à Firehose n'est pas supprimé. »

Code d’erreur	Message et informations sur les erreurs
`ES.ResourceNotFound`	« Le domaine AWS Elasticsearch spécifié n'existe pas. »

HTTPSErreurs de livraison des données du terminal

Amazon Data Firehose peut envoyer les erreurs suivantes HTTP liées au point de terminaison 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. »
`HttpEndpoint.ConnectionFailed`	« Impossible de se connecter au point de terminaison de destination. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »

Code d’erreur	Message et informations sur les erreurs
`HttpEndpoint.ConnectionReset`	« Impossible de maintenir la connexion avec le point de terminaison. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
`HttpEndpoint.ConnectionReset`	« Difficulté à maintenir la connexion avec le point de terminaison. Veuillez contacter le propriétaire du point de terminaison. »
`HttpEndpoint.ResponseReasonPhraseExceededLimit`	« L'expression de raison de la réponse reçue du point de terminaison dépasse la limite configurée de 64 caractères. »
`HttpEndpoint.InvalidResponseFromDestination`	« La réponse reçue du point de terminaison n'est pas valide. Consultez la section Résolution des problèmes liés HTTP aux terminaux dans la documentation de Firehose pour plus d'informations. Raison : «
`HttpEndpoint.DestinationException`	« La livraison au point de terminaison a échoué. Consultez la section Résolution des problèmes liés HTTP aux terminaux dans la documentation de Firehose pour plus d'informations. Réponse reçue avec le code d'état »
`HttpEndpoint.InvalidStatusCode`	« Code d'état de la réponse reçue non valide. »
`HttpEndpoint.SSLHandshakeFailure`	« Impossible de terminer une SSL poignée de main avec le terminal. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
`HttpEndpoint.SSLHandshakeFailure`	« Impossible de terminer une SSL poignée de main avec le terminal. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »

Code d’erreur	Message et informations sur les erreurs
`HttpEndpoint.SSLFailure`	« Impossible de terminer la poignée de TLS main avec le terminal. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
`HttpEndpoint.SSLHandshakeCertificatePathFailure`	« Impossible de terminer une SSL poignée de main avec le terminal en raison d'un chemin de certification non valide. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
`HttpEndpoint.SSLHandshakeCertificatePathValidationFailure`	« Impossible de terminer une SSL poignée de main avec le terminal en raison d'un échec de validation du chemin de certification. Contactez le propriétaire du point de terminaison pour résoudre ce problème. »
`HttpEndpoint.MakeRequestFailure.IllegalUriException`	« HttpEndpoint la demande a échoué en raison d'une saisie non valideURI. Assurez-vous que tous les caractères saisis URI sont valides. »
`HttpEndpoint.MakeRequestFailure.IllegalCharacterInHeaderValue`	« 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
`HttpEndpoint.IllegalResponseFailure`	« la HttpEndpoint demande a échoué en raison d'une erreur de réponse illégale. HTTPle message ne doit pas contenir plus d'un en-tête Content-Type. »
`HttpEndpoint.IllegalMessageStart`	« la HttpEndpoint demande a échoué en raison d'une erreur de réponse illégale. Début de HTTP message illégal. Pour plus d'informations, consultez la section Dépannage des HTTP endpoints dans la documentation de Firehose. »

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
`OS.AccessDenied`	« Accès refusé. Assurez-vous que la politique de confiance associée au IAM rôle fourni autorise Firehose à assumer le rôle et que la politique d'accès autorise l'accès à Amazon OpenSearch ServiceAPI. »

Code d’erreur	Message et informations sur les erreurs
`OS.AccessDenied`	« Accès refusé. Assurez-vous que la politique de confiance associée au IAM rôle fourni autorise Firehose à assumer le rôle et que la politique d'accès autorise l'accès à Amazon OpenSearch ServiceAPI. »
`OS.AccessDenied`	« Accès refusé. Assurez-vous que le IAM rôle fourni associé à Firehose n'est pas supprimé. »
`OS.AccessDenied`	« Accès refusé. Assurez-vous que le IAM rôle 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 IAM rôle fourni autorise Firehose à assumer le rôle et que la politique d'accès autorise l'accès à Amazon OpenSearch ServiceAPI. »
`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 IAM rôle configuré. »
`OS.ValidationException`	« Le OpenSearch cluster a renvoyé unESServiceException. 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-z][a-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
`OS.ExplicitIndexInBulkNotAllowed`	« Assurez-vous que rest.action.multi.allow_explicit_index est défini sur true sur le cluster Amazon Service. » OpenSearch
`OS.ClusterError`	« Le cluster Amazon OpenSearch Service ou la collection OpenSearch Serverless ont renvoyé une erreur non spécifiée. »
`OS.ClusterBlockException`	« Le cluster a renvoyé un ClusterBlockException. Il est peut-être surchargé. »
`OS.InvalidARN`	« Le OpenSearch service Amazon ARN fourni n'est pas valide. Vérifiez votre DeliveryStream configuration. »
`OS.MalformedData`	« Un ou plusieurs enregistrements sont mal formés. Assurez-vous que chaque enregistrement est un seul JSON objet valide et qu'il ne contient pas de nouvelles lignes. »
`OS.InternalError`	« 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. »
`OS.AliasWithMultipleIndicesNotAllowed`	« Plusieurs index sont associés à l'alias. Assurez-vous qu'un seul index est associé à l'alias. »
`OS.UnsupportedVersion`	« Amazon OpenSearch Service 6.0 n'est actuellement pas pris en charge par Amazon Data Firehose. Contactez AWS le Support pour plus d'informations. »
`OS.CharConversionException`	« Un ou plusieurs enregistrements contenaient un caractère non valide. »

Code d’erreur	Message et informations sur les erreurs
`OS.InvalidDomainNameLength`	« La longueur du nom de domaine n'est pas dans les limites valides du système d'exploitation. »
`OS.VPCDomainNotSupported`	« Les domaines Amazon OpenSearch Service inclus ne VPCs sont actuellement pas pris en charge. »
`OS.ConnectionError`	« Le serveur HTTP a fermé la connexion de façon inattendue. Vérifiez l'état du cluster Amazon OpenSearch Service ou de la collection OpenSearch Serverless. »
`OS.LargeFieldData`	« Le cluster Amazon OpenSearch Service a abandonné la demande car elle contenait des données de champ plus grandes que celles autorisées. »
`OS.BadGateway`	« Le cluster Amazon OpenSearch Service ou la collection OpenSearch Serverless a abandonné la demande avec la réponse 502 Bad Gateway. »
`OS.ServiceException`	« Erreur reçue du cluster Amazon OpenSearch Service ou de la collection OpenSearch Serverless. Si le cluster ou la collection se trouve derrière unVPC, assurez-vous que la configuration réseau permet la connectivité. »
`OS.GatewayTimeout`	« Firehose a rencontré des erreurs de temporisation lors de la connexion au cluster Amazon OpenSearch Service ou à la collection OpenSearch Serverless. »
`OS.MalformedData`	« Amazon Data Firehose ne prend pas en charge les API commandes Amazon OpenSearch Service Bulk dans l'enregistrement Firehose. »
`OS.ResponseEntryCountMismatch`	« La réponse du Bulk API contenait plus d'entrées que le nombre d'enregistrements envoyés. Assurez-vous que chaque enregistrement ne contient qu'un seul JSON objet et qu'il n'y a pas de nouvelles lignes. »

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
`Lambda.AssumeRoleAccessDenied`	« Accès refusé. Assurez-vous que la politique de confiance relative au IAM rôle fourni autorise Amazon Data Firehose à assumer ce rôle. »
`Lambda.InvokeAccessDenied`	« Accès refusé. Vérifiez que la stratégie d'accès permet d'accéder à la fonction Lambda. »
`Lambda.JsonProcessingException`	« 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. » Pour de plus amples informations, veuillez consulter Paramètres requis pour la transformation des données.
`Lambda.InvokeLimitExceeded`	« La limite appliquée aux exécutions simultanées Lambda est dépassée. Augmentez-la. » Pour plus d'informations, consultez la section Limites AWS Lambda dans le Guide du développeur AWS Lambda .
`Lambda.DuplicatedRecordId`	« Plusieurs enregistrements ont été retournés avec le même ID d'enregistrement. Assurez-vous que la fonction Lambda renvoie un enregistrement unique IDs pour chaque enregistrement. » Pour de plus amples informations, veuillez consulter Paramètres requis pour la transformation des données.
`Lambda.MissingRecordId`	« Un ou plusieurs enregistrements n'IDsont pas été renvoyés. Assurez-vous que la fonction Lambda renvoie tous les enregistrements IDs reçus. »

Code d'erreur	Message et informations sur les erreurs
	Pour de plus amples informations, veuillez consulter Paramètres requis pour la transformation des données.
`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 de la VPC 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 VPC configuration 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'VPCaccès 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 plus d'informations, consultez Amazon VPC Limits - VPC and Subnets dans le guide de l'VPCutilisateur Amazon.
`Lambda.ENILimitReachedException`	« n'AWS Lambda a pas pu créer d'interface réseau élastique (ENI) dans leVPC, spécifié 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 plus d'informations, consultez Amazon VPC Limits - Network Interfaces dans le guide de VPC l'utilisateur Amazon.
`Lambda.FunctionTimedOut`	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
`Lambda.FunctionError`	Cela peut être dû à l'une des erreurs suivantes : • 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 `Dropped`, `Ok`, ou `ProcessingFailed` . • 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'KMSaccès lui a été refusé. Vérifiez les KMS principaux paramètres de la fonction ainsi que la politique des touches. Pour de plus amples informations, consultez Troubleshooting Key Access.
`Lambda.FunctionRequestTimedOut`	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.
`Lambda.TargetServerFailedToRespond`	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.
`Lambda.InvalidZipFileException`	Amazon Data Firehose s'est produit InvalidZipFileException lors de l'appel de la fonction Lambda. Vérifiez les paramètres de configuration de votre fonction Lambda et le fichier zip du code Lambda.

Code d'erreur	Message et informations sur les erreurs
`Lambda.InternalServerError`	« Amazon Data Firehose s'est rencontré InternalServerError 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 annuler les options de nouvelle tentative à l'aide de la `CreateDeliveryStream` touche ou. `UpdateDestination` APIs Si l'erreur persiste, contactez l'équipe d'assistance AWS Lambda.
`Lambda.ServiceUnavailable`	Amazon Data Firehose s'est produit ServiceUnavailableException lors de l'appel du service Lambda AWS . Amazon Data Firehose réessaier a d'envoyer des données un nombre fixe de fois. Vous pouvez spécifier ou annuler les options de nouvelle tentative à l'aide de la `CreateDeliveryStream` touche ou. `UpdateDestination` APIs Si l'erreur persiste, contactez le support AWS Lambda.
`Lambda.InvalidSecurityToken`	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.
`Lambda.InvocationFailure`	Cela peut être dû à l'une des erreurs suivantes : • 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 message de KMSInvalidStateException Lambda. Lambda n'a pas pu déchiffrer les variables d'environnement car la KMS clé utilisée n'est pas valide pour Decrypt. Vérifiez la KMS clé de la fonction Lambda. • Amazon Data Firehose a rencontré un homme provenant de AWS LambdaException 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.JsonMappingException`	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.AccessDenied`	« L'accès a été refusé lors de l'appel de Kinesis. Assurez-vous que la politique d'accès relative au IAM rôle utilisé autorise l'accès au Kinesis APIs approprié. »
`Kinesis.ResourceNotFound`	« 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.SubscriptionRequired`	« L'accès a été refusé lors de l'appel de Kinesis. Assurez-vous que le IAM rôle transmis pour l'accès au flux Kinesis est associé à un abonnement AWS Kinesis. »
`Kinesis.Throttling`	« Une erreur de limitation s'est produite lors de l'appel de Kinesis. Cela peut être dû au fait que d'autres applications appellent le même flux APIs que le flux Firehose, ou parce que vous avez créé trop de flux Firehose avec le même flux Kinesis comme source. »
`Kinesis.Throttling`	« Une erreur de limitation s'est produite lors de l'appel de Kinesis. Cela peut être dû au fait que d'autres applications appellent le même flux APIs que le flux Firehose, ou parce que vous avez créé trop de flux Firehose avec le même flux Kinesis comme source. »

Code d’erreur	Message et informations sur les erreurs
`Kinesis.AccessDenied`	« L'accès a été refusé lors de l'appel de Kinesis. Assurez-vous que la politique d'accès relative au IAM rôle utilisé autorise l'accès au Kinesis APIs approprié. »
`Kinesis.AccessDenied`	« L'accès a été refusé alors que j'essayais d'appeler API des opérations sur le Kinesis Stream sous-jacent. Assurez-vous que le IAM rôle est propagé et valide. »
`Kinesis.KMS.AccessDeniedException`	« Firehose n'a pas accès à la KMS clé utilisée pour crypter/déchiffrer le Kinesis Stream. Veuillez accorder au rôle de diffusion Firehose l'accès à la clé. »
`Kinesis.KMS.KeyDisabled`	« Firehose ne parvient pas à lire le Kinesis Stream source car la KMS clé utilisée pour le chiffrer/déchiffrer est désactivée. Activez la touche pour que les lectures puissent continuer. »
`Kinesis.KMS.InvalidStateException`	« Firehose ne parvient pas à lire depuis la source Kinesis Stream car la KMS clé utilisée pour le chiffrer n'est pas valide. »
`Kinesis.KMS.NotFoundException`	« Firehose ne parvient pas à lire depuis la source Kinesis Stream car la KMS clé utilisée pour le chiffrer est introuvable. »

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
`Firehose.KMS.Acces`	« Firehose n'a pas accès à la KMS clé. Veuillez vérifier la stratégie de clé. »

Code d’erreur	Message et informations sur les erreurs
`sDeniedEx ception`	
`Firehose. KMS.Inval idStateEx ception`	« Firehose n'est pas en mesure de déchiffrer les données car la KMS clé utilisée pour les chiffrer n'est pas valide. »
`Firehose. KMS.NotFo undException`	« Firehose ne parvient pas à déchiffrer les données car la KMS clé utilisée pour les chiffrer est introuvable. »
`Firehose. KMS.KeyDi sabled`	« Firehose ne parvient pas à déchiffrer les données car la KMS clé utilisée pour chiffrer les données est 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
`DataForma tConversi on.Invali dSchema`	« Le schéma n'est pas valide. »
`DataForma tConversi on.Entity NotFound`	« 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. »
`DataForma tConversi`	« 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
on.Invali dInput	
DataForma tConversi on.Invali dInput	« Impossible de trouver un schéma correspondant dans Glue. Assurez-v ous que le document transmis ARN est au bon format. »
DataForma tConversi on.Invali dInput	« Impossible de trouver un schéma correspondant dans Glue. Veuillez vous assurer que les catalogId informations fournies sont valides. »
DataForma tConversi on.Invali dVersionId	« Impossible de trouver un schéma correspondant dans Glue. Assurez-v ous que la version spécifiée de la table existe. »
DataForma tConversi on.NonExi stentColumns	« Impossible de trouver un schéma correspondant dans Glue. Assurez-v ous que la table est configurée avec un descripteur de stockage non nul contenant les colonnes cibles. »
DataForma tConversi on.Access Denied	« 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. »
DataForma tConversi on.Thrott ledByGlue	« 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 IAM rôle 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
`DataFirehose.InternalError`	« 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. »
`DataFormatConversion.GlueEncryptionException`	« 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
`DataFormatConversion.InvalidSchema`	« Le schéma n'est pas valide. »
`DataFormatConversion.ValidationException`	« Les noms et types de colonnes doivent être des chaînes non vides. »
`DataFormatConversion.ParseError`	« J'ai rencontré une malformationJSON. »
`DataFormatConversi`	« Les données ne correspondent pas au schéma. »

Code d’erreur	Message et informations sur les erreurs
on.Malfor medData	
DataForma tConversi on.Malfor medData	« La longueur de la clé JSON ne doit pas être supérieure à 262 144 »
DataForma tConversi on.Malfor medData	« Les données ne peuvent pas être décodées sous la forme UTF -8. »
DataForma tConversi on.Malfor medData	« Un caractère illégal a été trouvé entre les jetons. »
DataForma tConversi on.Invali dTypeFormat	« Le format de type n'est pas valide. Vérifiez la syntaxe du type. »
DataForma tConversi on.Invali dSchema	« Schéma non valide. Assurez-vous qu'il n'y a pas de caractères spéciaux ou d'espaces blancs dans les noms de colonnes. »
DataForma tConversi on.Invali dRecord	« 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
`DataFormatConversion.MalformedData`	« L'entrée JSON contenait une primitive au niveau supérieur. Le niveau supérieur doit être un objet ou un tableau. »
`DataFormatConversion.MalformedData`	« L'entrée JSON contenait une primitive au niveau supérieur. Le niveau supérieur doit être un objet ou un tableau. »
`DataFormatConversion.MalformedData`	« L'enregistrement était vide ou ne contenait que des espaces. »
`DataFormatConversion.MalformedData`	« Caractères non valides rencontrés. »
`DataFormatConversion.MalformedData`	« 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. »
`DataFormatConversion.MalformedData`	« Un type scalaire a été trouvé dans les données, mais un type complexe a été spécifié dans le schéma. »
`DataFormatConversion.MalformedData`	« Les données ne correspondent pas au schéma. »

Code d'erreur	Message et informations sur les erreurs
`DataFormatConversion.MalformedData`	« Un type scalaire a été trouvé dans les données, mais un type complexe a été spécifié dans le schéma. »
`DataFormatConversion.ConversionFailureException`	"ConversionFailureException"
`DataFormatConversion.DataFormatConversionCustomerErrorException`	"DataFormatConversionCustomerErrorException"
`DataFormatConversion.DataFormatConversionCustomerErrorException`	"DataFormatConversionCustomerErrorException"
`DataFormatConversion.MalformedData`	« Les données ne correspondent pas au schéma. »

Code d'erreur	Message et informations sur les erreurs
`DataFormatConversion.InvalidSchema`	« Le schéma n'est pas valide. »
`DataFormatConversion.MalformedData`	« Les données ne correspondent pas au schéma. Format non valide pour une ou plusieurs dates. »
`DataFormatConversion.MalformedData`	« Les données contiennent une JSON structure hautement imbriquée qui n'est pas prise en charge. »
`DataFormatConversion.EntityNotFound`	« 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. »
`DataFormatConversion.InvalidInput`	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la base de données spécifiée avec l'ID de catalogue fourni existe. »
`DataFormatConversion.InvalidInput`	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que le document transmis ARN est au bon format. »
`DataFormatConversion.InvalidInput`	« Impossible de trouver un schéma correspondant dans Glue. Veuillez vous assurer que les catalogId informations fournies sont valides. »

Code d’erreur	Message et informations sur les erreurs
`DataFormatConversion.InvalidVersionId`	« Impossible de trouver un schéma correspondant dans Glue. Assurez-vous que la version spécifiée de la table existe. »
`DataFormatConversion.NonExistentColumns`	« 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. »
`DataFormatConversion.AccessDenied`	« 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. »
`DataFormatConversion.ThrottledByGlue`	« 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. »
`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.GlueNotAvailableInRegion`	« 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
`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. »
`DataFirehose.InternalError`	« 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. »
`DataFormatConversion.MalformedData`	« Le format d'un ou de plusieurs champs est incorrect. »

CloudWatch Journaux d'accès 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 / firehose https://console.aws.amazon.com
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/.
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.

Surveiller l'état de santé de l'agent Kinesis

Kinesis Agent publie des CloudWatch métriques personnalisées avec un espace de noms de. AWS KinesisAgent Cela permet de déterminer si l'agent est en bonne santé, s'il soumet les données à Amazon Data Firehose conformément aux spécifications et si le producteur de données consomme la quantité CPU et les ressources de mémoire appropriées.

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. Des indicateurs tels que la consommation sur l'hôte CPU et la consommation de mémoire et les compteurs d'erreurs des agents indiquent l'utilisation des ressources par le producteur de données et fournissent des informations sur les erreurs de configuration ou d'hôte potentielles. 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 de plus amples informations, veuillez consulter Spécifier les 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.

Moniteur 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 nouvelle 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

Enregistrez les appels Amazon Data Firehose API 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 API appels à 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 Amazon Data Firehose. API Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements 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 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 SNS notifications Amazon 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'CloudTrail userIdentity élément.

Exemple : entrées du fichier journal 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 comprend 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 API appels 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-amzn-s3-demo-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-amzn-s3-demo-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-amzn-s3-demo-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"
    }
  ]
}
```

Exemples de code pour Firehose à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Firehose avec un AWS kit de développement logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Les actions vous indiquent comment appeler des fonctions de service individuelles, mais vous pouvez les visualiser dans leur contexte dans les scénarios correspondants.

Les scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un service ou en les combinant avec d'autres Services AWS.

Pour une liste complète des AWS SDKguides du développeur et exemples de code, voirUtiliser Firehose avec un AWS SDK. Cette rubrique inclut également des informations sur la mise en route et des détails sur SDK les versions précédentes.

Exemples de code

- Exemples de base pour l'utilisation de Firehose AWS SDKs
  - Actions pour Firehose utilisant AWS SDKs
    - À utiliser PutRecord avec un AWS SDKou CLI
    - À utiliser PutRecordBatch avec un AWS SDKou CLI
- Scénarios d'utilisation de Firehose AWS SDKs
  - Utilisez Amazon Data Firehose pour traiter les enregistrements individuels et par lots

Exemples de base pour l'utilisation de Firehose AWS SDKs

Les exemples de code suivants montrent comment utiliser les bases d'Amazon Data Firehose avec AWS SDKs.

Exemples

- Actions pour Firehose utilisant AWS SDKs
  - À utiliser PutRecord avec un AWS SDKou CLI
  - À utiliser PutRecordBatch avec un AWS SDKou CLI

Actions pour Firehose utilisant AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions Firehose individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Ces extraits, appelés API Firehose, sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Vous pouvez voir les actions dans leur contexte dansScénarios d'utilisation de Firehose AWS SDKs .

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour une liste complète, consultez le Amazon Data Firehose API Reference.

Exemples

- À utiliser PutRecord avec un AWS SDKou CLI
- À utiliser PutRecordBatch avec un AWS SDKou CLI

À utiliser **`PutRecord`** avec un AWS SDKou CLI

Les exemples de code suivants montrent comment utiliser`PutRecord`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Transférez des enregistrements à Firehose

CLI

AWS CLI

Pour écrire un enregistrement dans un flux

L'`put-record`exemple suivant écrit des données dans un flux. Les données sont codées au format Base64.

```
aws firehose put-record \
    --delivery-stream-name my-stream \
    --record '{"Data":"SGVsbG8gd29ybGQ="}'
```

Sortie :

```
{
    "RecordId": "RjB5K/nnoGFHqwTsZlNd/
TTqvjE8V5dsyXZTQn2JXrdpMTOwssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/
j1Xujv31iDhnPdaLw4BOyM9Amv7PcCuB2079RuM0NhoakbyUymlwY8yt20G8X2420wu1jlFafhci4erAt7QhDEvpw
    "Encrypted": false
}
```

Pour plus d'informations, consultez la section Envoi de données vers un flux de diffusion Amazon Kinesis Data Firehose dans le manuel du développeur Amazon Kinesis Data Firehose.

- Pour API plus de détails, voir PutRecordAWS CLI Référence de commande.

Python

SDKpour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Consultez l'exemple complet et apprenez à configurer et à exécuter dans AWS Référentiel d'exemples de code.

```
class FirehoseClient:
    """
    AWS Firehose client to send records and monitor metrics.

    Attributes:
        config (object): Configuration object with delivery stream name and
 region.
        delivery_stream_name (str): Name of the Firehose delivery stream.
        region (str): AWS region for Firehose and CloudWatch clients.
        firehose (boto3.client): Boto3 Firehose client.
        cloudwatch (boto3.client): Boto3 CloudWatch client.
    """

    def __init__(self, config):
        """
        Initialize the FirehoseClient.
```

```
        Args:
            config (object): Configuration object with delivery stream name and
region.
        """
        self.config = config
        self.delivery_stream_name = config.delivery_stream_name
        self.region = config.region
        self.firehose = boto3.client("firehose", region_name=self.region)
        self.cloudwatch = boto3.client("cloudwatch", region_name=self.region)


    @backoff.on_exception(
        backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
    )
    def put_record(self, record: dict):
        """
        Put individual records to Firehose with backoff and retry.

        Args:
            record (dict): The data record to be sent to Firehose.

        This method attempts to send an individual record to the Firehose
delivery stream.
        It retries with exponential backoff in case of exceptions.
        """
        try:
            entry = self._create_record_entry(record)
            response = self.firehose.put_record(
                DeliveryStreamName=self.delivery_stream_name, Record=entry
            )
            self._log_response(response, entry)
        except Exception:
            logger.info(f"Fail record: {record}.")
            raise
```

- Pour API plus de détails, voir PutRecordAWS SDKpour Python (Boto3) Reference. API

Pour une liste complète des AWS SDKguides du développeur et exemples de code, voirUtiliser Firehose avec un AWS SDK. Cette rubrique inclut également des informations sur la mise en route et des détails sur SDK les versions précédentes.

À utiliser **PutRecordBatch** avec un AWS SDKou CLI

Les exemples de code suivants montrent comment utiliser`PutRecordBatch`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Transférez des enregistrements à Firehose

CLI

AWS CLI

Pour écrire plusieurs enregistrements dans un flux

L'`put-record-batch`exemple suivant écrit trois enregistrements dans un flux. Les données sont codées au format Base64.

```
aws firehose put-record-batch \
    --delivery-stream-name my-stream \
    --records file://records.json
```

Contenu de `myfile.json` :

```
[
    {"Data": "Rmlyc3QgdGhpbmc="},
    {"Data": "U2Vjb25kIHRoaW5n"},
    {"Data": "VGhpcmQgdGhpbmc="}
]
```

Sortie :

```
{
    "FailedPutCount": 0,
    "Encrypted": false,
```

```
    "RequestResponses": [
        {
            "RecordId": "9D2OJ6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhYOaWKPSmZy/
CGlRVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn
+3boDnURDEaD42V7GiixpOyLJkYZcae1i7HzlCEoy9LJhMr8EjDSi4Om/9Vc2uhwwuAtGE0XKpxJ2WD7ZRWtAnYlK
        },
        {
            "RecordId": "jFirejqxCLlK5xjH/UNmlMVcjktEN76I7916X9PaZ
+PVaOSXDfU1WGOqEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/
GsuF37Uhg67GKmR5z90l6XKJ+/
+pDloFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCUkBphR2QVzhP9OiHLbzGwy8/
DfH8sqWEUYASNJKS8GXP5s"
        },
        {
            "RecordId":
 "oy0amQ40o5Y2YV4vxzufdcMOOw6n3EPr3tpPJGoYVNKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3
DTBt3qBlmTj7Xq8SKVbO1S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/sOkQLe+qh0sYKXWl"
        }
    ]
}
```

Pour plus d'informations, consultez la section Envoi de données vers un flux de diffusion Amazon Kinesis Data Firehose dans le manuel du développeur Amazon Kinesis Data Firehose.

- Pour API plus de détails, voir PutRecordBatchAWS CLI Référence de commande.

Python

SDKpour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Consultez l'exemple complet et apprenez à configurer et à exécuter dans AWS Référentiel d'exemples de code.

```
class FirehoseClient:
    """
    AWS Firehose client to send records and monitor metrics.

    Attributes:
```

```
        config (object): Configuration object with delivery stream name and
region.
        delivery_stream_name (str): Name of the Firehose delivery stream.
        region (str): AWS region for Firehose and CloudWatch clients.
        firehose (boto3.client): Boto3 Firehose client.
        cloudwatch (boto3.client): Boto3 CloudWatch client.
    """

    def __init__(self, config):
        """
        Initialize the FirehoseClient.

        Args:
            config (object): Configuration object with delivery stream name and
region.
        """
        self.config = config
        self.delivery_stream_name = config.delivery_stream_name
        self.region = config.region
        self.firehose = boto3.client("firehose", region_name=self.region)
        self.cloudwatch = boto3.client("cloudwatch", region_name=self.region)


    @backoff.on_exception(
        backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
    )
    def put_record_batch(self, data: list, batch_size: int = 500):
        """
        Put records in batches to Firehose with backoff and retry.

        Args:
            data (list): List of data records to be sent to Firehose.
            batch_size (int): Number of records to send in each batch. Default is
500.

        This method attempts to send records in batches to the Firehose delivery
stream.
        It retries with exponential backoff in case of exceptions.
        """
        for i in range(0, len(data), batch_size):
            batch = data[i : i + batch_size]
            record_dicts = [{"Data": json.dumps(record)} for record in batch]
            try:
                response = self.firehose.put_record_batch(
```

```
                    DeliveryStreamName=self.delivery_stream_name,
 Records=record_dicts
                )
                self._log_batch_response(response, len(batch))
            except Exception as e:
                logger.info(f"Failed to send batch of {len(batch)} records.
 Error: {e}")
```

- Pour API plus de détails, voir PutRecordBatchAWS SDKpour Python (Boto3) Reference. API

Rust

SDKpour Rust

> **Note**
>
> Il y en a plus à ce sujet GitHub. Consultez l'exemple complet et apprenez à configurer et à exécuter dans AWS Référentiel d'exemples de code.

```
async fn put_record_batch(
    client: &Client,
    stream: &str,
    data: Vec<Record>,
) -> Result<PutRecordBatchOutput, SdkError<PutRecordBatchError>> {
    client
        .put_record_batch()
        .delivery_stream_name(stream)
        .set_records(Some(data))
        .send()
        .await
}
```

- Pour API plus de détails, voir PutRecordBatchAWS SDKpour API référence à Rust.

Pour une liste complète des AWS SDKguides du développeur et exemples de code, voirUtiliser Firehose avec un AWS SDK. Cette rubrique inclut également des informations sur la mise en route et des détails sur SDK les versions précédentes.

Scénarios d'utilisation de Firehose AWS SDKs

Les exemples de code suivants vous montrent comment implémenter des scénarios courants dans Firehose avec AWS SDKs. Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions dans Firehose ou en les combinant avec d'autres Services AWS. Chaque scénario inclut un lien vers le code source complet, où vous trouverez des instructions sur la façon de configurer et d'exécuter le code.

Les scénarios ciblent un niveau d'expérience intermédiaire pour vous aider à comprendre les actions de service dans leur contexte.

Exemples

- Utilisez Amazon Data Firehose pour traiter les enregistrements individuels et par lots

Utilisez Amazon Data Firehose pour traiter les enregistrements individuels et par lots

L'exemple de code suivant montre comment utiliser Firehose pour traiter des enregistrements individuels et par lots.

Python

SDKpour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Consultez l'exemple complet et apprenez à configurer et à exécuter dans AWS Référentiel d'exemples de code.

Ce script place des enregistrements individuels et par lots dans Firehose.

```
import json
import logging
import random
```

```
from datetime import datetime, timedelta

import backoff
import boto3

from config import get_config


def load_sample_data(path: str) -> dict:
    """
    Load sample data from a JSON file.

    Args:
        path (str): The file path to the JSON file containing sample data.

    Returns:
        dict: The loaded sample data as a dictionary.
    """
    with open(path, "r") as f:
        return json.load(f)


# Configure logging
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)


class FirehoseClient:
    """
    AWS Firehose client to send records and monitor metrics.

    Attributes:
        config (object): Configuration object with delivery stream name and
 region.
        delivery_stream_name (str): Name of the Firehose delivery stream.
        region (str): AWS region for Firehose and CloudWatch clients.
        firehose (boto3.client): Boto3 Firehose client.
        cloudwatch (boto3.client): Boto3 CloudWatch client.
    """

    def __init__(self, config):
        """
        Initialize the FirehoseClient.
```

```
        Args:
            config (object): Configuration object with delivery stream name and
region.
        """
        self.config = config
        self.delivery_stream_name = config.delivery_stream_name
        self.region = config.region
        self.firehose = boto3.client("firehose", region_name=self.region)
        self.cloudwatch = boto3.client("cloudwatch", region_name=self.region)


    @backoff.on_exception(
        backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
    )
    def put_record(self, record: dict):
        """
        Put individual records to Firehose with backoff and retry.

        Args:
            record (dict): The data record to be sent to Firehose.

        This method attempts to send an individual record to the Firehose
delivery stream.
        It retries with exponential backoff in case of exceptions.
        """
        try:
            entry = self._create_record_entry(record)
            response = self.firehose.put_record(
                DeliveryStreamName=self.delivery_stream_name, Record=entry
            )
            self._log_response(response, entry)
        except Exception:
            logger.info(f"Fail record: {record}.")
            raise


    @backoff.on_exception(
        backoff.expo, Exception, max_tries=5, jitter=backoff.full_jitter
    )
    def put_record_batch(self, data: list, batch_size: int = 500):
        """
        Put records in batches to Firehose with backoff and retry.

        Args:
```

```
            data (list): List of data records to be sent to Firehose.
            batch_size (int): Number of records to send in each batch. Default is
500.

        This method attempts to send records in batches to the Firehose delivery
stream.
        It retries with exponential backoff in case of exceptions.
        """
        for i in range(0, len(data), batch_size):
            batch = data[i : i + batch_size]
            record_dicts = [{"Data": json.dumps(record)} for record in batch]
            try:
                response = self.firehose.put_record_batch(
                    DeliveryStreamName=self.delivery_stream_name,
Records=record_dicts
                )
                self._log_batch_response(response, len(batch))
            except Exception as e:
                logger.info(f"Failed to send batch of {len(batch)} records.
Error: {e}")


    def get_metric_statistics(
        self,
        metric_name: str,
        start_time: datetime,
        end_time: datetime,
        period: int,
        statistics: list = ["Sum"],
    ) -> list:
        """
        Retrieve metric statistics from CloudWatch.

        Args:
            metric_name (str): The name of the metric.
            start_time (datetime): The start time for the metric statistics.
            end_time (datetime): The end time for the metric statistics.
            period (int): The granularity, in seconds, of the returned data
points.
            statistics (list): A list of statistics to retrieve. Default is
['Sum'].

        Returns:
            list: List of datapoints containing the metric statistics.
```

```
        """
        response = self.cloudwatch.get_metric_statistics(
            Namespace="AWS/Firehose",
            MetricName=metric_name,
            Dimensions=[
                {"Name": "DeliveryStreamName", "Value":
self.delivery_stream_name},
            ],
            StartTime=start_time,
            EndTime=end_time,
            Period=period,
            Statistics=statistics,
        )
        return response["Datapoints"]

    def monitor_metrics(self):
        """
        Monitor Firehose metrics for the last 5 minutes.

        This method retrieves and logs the 'IncomingBytes', 'IncomingRecords',
and 'FailedPutCount' metrics
        from CloudWatch for the last 5 minutes.
        """
        end_time = datetime.utcnow()
        start_time = end_time - timedelta(minutes=10)
        period = int((end_time - start_time).total_seconds())

        metrics = {
            "IncomingBytes": self.get_metric_statistics(
                "IncomingBytes", start_time, end_time, period
            ),
            "IncomingRecords": self.get_metric_statistics(
                "IncomingRecords", start_time, end_time, period
            ),
            "FailedPutCount": self.get_metric_statistics(
                "FailedPutCount", start_time, end_time, period
            ),
        }

        for metric, datapoints in metrics.items():
            if datapoints:
                total_sum = sum(datapoint["Sum"] for datapoint in datapoints)
                if metric == "IncomingBytes":
                    logger.info(
```

```
                        f"{metric}: {round(total_sum)} ({total_sum / (1024 *
1024):.2f} MB)"
                    )
                else:
                    logger.info(f"{metric}: {round(total_sum)}")
            else:
                logger.info(f"No data found for {metric} over the last 5
minutes")


    def _create_record_entry(self, record: dict) -> dict:
        """
        Create a record entry for Firehose.

        Args:
            record (dict): The data record to be sent.

        Returns:
            dict: The record entry formatted for Firehose.

        Raises:
            Exception: If a simulated network error occurs.
        """
        if random.random() < 0.2:
            raise Exception("Simulated network error")
        elif random.random() < 0.1:
            return {"Data": '{"malformed": "data"'}
        else:
            return {"Data": json.dumps(record)}

    def _log_response(self, response: dict, entry: dict):
        """
        Log the response from Firehose.

        Args:
            response (dict): The response from the Firehose put_record API call.
            entry (dict): The record entry that was sent.
        """
        if response["ResponseMetadata"]["HTTPStatusCode"] == 200:
            logger.info(f"Sent record: {entry}")
        else:
            logger.info(f"Fail record: {entry}")

    def _log_batch_response(self, response: dict, batch_size: int):
```

```
        """
        Log the batch response from Firehose.

        Args:
            response (dict): The response from the Firehose put_record_batch API
 call.
            batch_size (int): The number of records in the batch.
        """
        if response.get("FailedPutCount", 0) > 0:
            logger.info(
                f'Failed to send {response["FailedPutCount"]} records in batch of
 {batch_size}'
            )
        else:
            logger.info(f"Successfully sent batch of {batch_size} records")


if __name__ == "__main__":
    config = get_config()
    data = load_sample_data(config.sample_data_file)
    client = FirehoseClient(config)

    # Process the first 100 sample network records
    for record in data[:100]:
        try:
            client.put_record(record)
        except Exception as e:
            logger.info(f"Put record failed after retries and backoff: {e}")
    client.monitor_metrics()

    # Process remaining records using the batch method
    try:
        client.put_record_batch(data[100:])
    except Exception as e:
        logger.info(f"Put record batch failed after retries and backoff: {e}")
    client.monitor_metrics()
```

Ce fichier contient la configuration du script ci-dessus.

```
class Config:
    def __init__(self):
        self.delivery_stream_name = "ENTER YOUR DELIVERY STREAM NAME HERE"
```

```
        self.region = "us-east-1"
        self.sample_data_file = (
            "../../../../../workflows/firehose/resources/sample_records.json"
        )


def get_config():
    return Config()
```

- Pour API plus de détails, consultez les rubriques suivantes dans AWS SDKpour Python (Boto3) Reference. API
  - PutRecord
  - PutRecordBatch

Pour une liste complète des AWS SDKguides du développeur et exemples de code, voirUtiliser Firehose avec un AWS SDK. Cette rubrique inclut également des informations sur la mise en route et des détails sur SDK les versions précédentes.

Résoudre les erreurs dans 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 plus d'informations sur le suivi des erreurs de livraison à CloudWatch l'aide dethe section called “Moniteur avec CloudWatch journaux”.

Direct PUT

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. Dans ce cas, Firehose réessaie indéfiniment les opérations suivantes :`DescribeStream`, et`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 le `PutRecordBatch`API, assurez-vous que votre code vérifie la valeur de FailedPutCountdans la réponse, même en cas de réussite de l'APIappel.

Kinesis Data Stream

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.

Problèmes courants

Vous trouverez ci-dessous des conseils de dépannage qui vous aideront à résoudre les problèmes courants lorsque vous travaillez avec un stream Firehose.

Stream Firehose non disponible

Le flux Firehose n'est pas disponible en tant que cible pour les CloudWatch journaux, les CloudWatch événements ou AWS Les actions liées à l'IoT en guise d'exemple AWS les services ne peuvent envoyer des messages et des événements qu'à un flux Firehose qui se trouve dans le même Région AWS. Vérifiez que votre stream Firehose se trouve dans la même région que vos autres services.

Aucune donnée à destination

S'il n'y a aucun problème d'ingestion de données et que les métriques émises pour le flux Firehose semblent correctes, 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.

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 de prendre du retard. 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. Pensez également à augmenter la taille de la mémoire de la fonction Lambda afin que l'CPUallocation puisse augmenter en conséquence. Ces modifications peuvent entraîner des invocations Lambda plus rapides. Pour plus d'informations sur la configuration des fonctions Lambda, voir Configuration AWS Fonctions Lambda.
- 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, consultezthe section called “Moniteur avec 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 l'`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, consultezthe section called “Moniteur avec 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 AWS Support et demande 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 un AWS Glue Data Catalog pour convertir le format d'enregistrement en Apache Parquet.

Lorsque le AWS Glue le robot d'exploration indexe les types de données de l'ensemble DynamoDB (`StringSet`,, et`BinarySet`)`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 AWS Glue catalogue de données

1. Connectez-vous au AWS Management Console et ouvrez le AWS Glue console à 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 Save (Enregistrer).

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 de plus amples informations, veuillez consulter Surveillez 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 de plus amples informations, veuillez consulter Surveillez 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 de plus amples informations, veuillez consulter Surveillez 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 de plus amples informations, veuillez consulter Surveillez Amazon Data Firehose à l'aide des 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 AWS Support pour 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 IAM rôle 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 IAM rôle a accès au groupe de CloudWatch journaux et aux flux de journaux pour vérifier les journaux d'erreurs. Pour de plus amples informations, veuillez consulter Accorder à 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 FirehoseData 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 de plus amples informations, veuillez consulter Surveillez 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 de plus amples informations, veuillez consulter Surveillez Amazon Data Firehose à l'aide des 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 la vérification précédente indique que des connexions sont en cours d'établissement, consultez le `STL_LOAD_ERRORS` tableau Amazon Redshift pour vérifier la raison de l'échec. 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 IAM rôle 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 IAM rôle a accès au groupe de CloudWatch journaux et aux flux de journaux pour vérifier les journaux d'erreurs. Pour de plus amples informations, veuillez consulter Accorder à 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 de plus amples informations, veuillez consulter Accorder à 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 FirehoseData 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 de plus amples informations, veuillez consulter Surveillez 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 de plus amples informations, veuillez consulter Surveillez 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 de plus amples informations, veuillez consulter Surveillez 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 de plus amples informations, veuillez consulter Surveillez Amazon Data Firehose à l'aide des 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 IAM rôle 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 IAM rôle spécifié dans votre flux Firehose peut accéder à votre cluster de OpenSearch services, à votre compartiment de sauvegarde S3 et à votre fonction Lambda (si la transformation des données est activée). Assurez-vous également que le IAM rôle 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 unVPC, assurez-vous que Firehose peut y accéder. Pour plus d'informations, consultez Accès à Splunk dans VPC.
- Si vous utilisez un AWS load balancer, 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 HEC réception (). ACKs
- Assurez-vous que vous utilisez un HEC jeton valide.
- Assurez-vous que le HEC jeton est activé.
- Vérifiez si les données que vous envoyez à Splunk sont correctement formatées. Pour plus d'informations, voir Formater les événements pour HTTP Event Collector.
- Assurez-vous que le HEC jeton et l'événement d'entrée sont configurés avec un index valide.
- Lorsqu'un téléchargement vers Splunk échoue en raison d'une erreur du serveur provenant du HEC nœud, la demande est automatiquement réessayée. 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.
- Assurez-vous d'avoir activé l'accusé de réception de l'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 votre HEC état de santé.
- 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 à l'assistance de Splunk d'activer le collecteur d'HTTPévénements, de le configurer sur `ackIdleCleanup` `true` in`inputs.conf`, et de créer ou de modifier 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 IAM rôle 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 IAM rôle a accès au groupe CloudWatch Logs et aux flux de journaux pour consulter les journaux d'erreurs. Pour plus d'informations, voir Subvention FirehoseAccess à une destination Splunk.
- Pour rediriger les données qui ont été envoyées vers le compartiment d'erreur S3 (sauvegarde S3) vers Splunk, suivez les étapes mentionnées dans la documentation 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 VPCE n'a pas accès au -ID. Cela peut être dû à l'une des raisons suivantes :

- VPCE-ID incorrect. Vérifiez qu'il n'y a aucune erreur typographique.
- Firehose ne prend pas en charge Snowflake sans région en version préliminaire. URLs Fournissez le localisateur de compte Snowflake à l'URLaide. Consultez la documentation de Snowflake pour plus de détails.
- Vérifiez que le stream Firehose est créé dans le même AWS Région en tant que région du flocon de neige.
- Si le problème persiste, contactez AWS soutien.

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, consultezErreurs de livraison des 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 valide. JSON Assurez-vous que la charge utile JSON est bien formée, qu'elle ne contient pas de guillemets, de guillemets, de caractères d'échappement, etc. Firehose ne prend actuellement en charge qu'un seul JSON élément en tant que charge utile record, les JSON tableaux 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 JSON valeurs spécifiées dans la charge utile d'entrée correspondent au 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 le Firehose atteint un délai API d'expiration, effectuez les étapes suivantes pour tester l'accessibilité des terminaux :

- Vérifiez si les API demandes proviennent d'un hôte dans unVPC. Tout le trafic provenant d'un VPC nécessite la configuration d'un point de terminaison FirehoseVPC. Pour plus d'informations, voir Utilisation de Firehose avec AWS PrivateLink.
- Si le trafic provient d'un réseau public ou VPC si le point de VPC terminaison Firehose est 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, API réessayez pour voir si une réponse est renvoyée par leAPI.

Résolution des problèmes liés aux HTTP terminaux

Cette section décrit les étapes de résolution des problèmes courants lors de la transmission de données par Amazon Data Firehose vers des destinations de HTTP points de terminaison génériques et vers des destinations partenaires, notamment Datadog, Dynatrace, LogicMonitor MongoDB, New Relic, Splunk ou Sumo Logic. Aux fins de cette section, toutes les destinations applicables sont appelées HTTP points de terminaison. Assurez-vous que le IAM rôle 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 IAM rôle 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 Firehose l'accès à une destination de point de HTTP terminaison.

> **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 terminal et de faire une HTTP demande, 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 d'Amazon Data Firehose. HTTP 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 :

- Champs non valides JSON ou inattendus : indique que la réponse ne peut pas être correctement désérialisée sous forme de champs inattendus JSON ou qu'elle contient des champs inattendus. Assurez-vous que le contenu de la réponse n'est pas codé.
- RequestIdManquant : indique que la réponse ne contient pas derequestId.
- RequestId ne correspond pas : indique que le contenu requestId de la réponse ne correspond pas au contenu sortantrequestId.
- 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 se trouve `https://aws-kinesis-http-intake.logs.datadoghq.com/` alors que son point de terminaison public l'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'une SSL poignée de main n'a pas pu être correctement effectuée avec le point de terminaison configuré.

Résolution des problèmes MSK en tant que source

Cette section décrit les étapes de dépannage courantes lors de l'utilisation de MSK As 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 As Source échoue

- Vérifiez que le MSK cluster 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. Voir également : MSKautorisations entre comptes
- 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 VPC de sécurité autorisent le trafic entrant sur les ports utilisés par les serveurs bootstrap du cluster

Tuyau suspendu

Vérifiez les points suivants si votre tuyau est en bon SUSPENDED état

- Vérifiez que le MSK cluster 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`

MSKproblèmes de connexion au cluster

La procédure suivante explique comment valider la connectivité aux MSK clusters. Pour en savoir plus sur la configuration MSK du client Amazon, consultez Getting started using Amazon MSK dans le manuel Amazon Managed Streaming for Apache Kafka Developer Guide.

Pour valider la connectivité aux MSK clusters

1. Créez une instance AL2 Amazon EC2 basée sur Unix (de préférence). Si seule la VPC connectivité est activée sur votre cluster, assurez-vous que votre EC2 instance s'exécute dans le même clusterVPC. SSHdans l'instance une fois qu'elle est disponible. Pour plus d'informations, consultez ce didacticiel dans le guide de EC2 l'utilisateur Amazon.

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. Installer 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 `kafka_2.12-2.6.2/libs` répertoire, puis exécutez la commande suivante pour télécharger le MSK IAM JAR fichier Amazon.

```
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 AWS utilisateur client devant 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 MSK cluster au sein de votre environnement local VPC et que les serveurs bootstrap fonctionnent correctement sur le port spécifié.

```
Connected to ..
```

10. Si la demande échoue, vérifiez les règles de trafic entrant de votre groupe VPC de sécurité. 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 soutien.

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 stream Firehose, la taille d'enregistrement maximale est de 6 Mo si AWS Lambda est activé, et la taille d'enregistrement maximale est 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 un enregistrement d'Amazon MSK d'une taille 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 AWS Limites Lambda, voir : 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), AWS GovCloud (USA Ouest), Canada (Ouest), Canada (Centre) : 2 000 streams 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 (USA Est), Afrique (Le Cap) : 500 streams de Firehose
- Europe (Zurich), Europe (Espagne), Asie-Pacifique (Hyderabad), Asie-Pacifique (Jakarta), Asie-Pacifique (Melbourne), Moyen-Orient (), Israël (Tel AvivUAE), Canada Ouest (Calgary), Canada (Centre), Asie-Pacifique (Malaisie) : 100 ruisseaux Firehose
- Si vous dépassez ce nombre, un appel à CreateDeliveryStreamentraîne 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 PutRecordet PutRecordBatchles 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 (USA Est), AWS GovCloud (US-Ouest), Asie-Pacifique (Hong Kong), Asie-Pacifique (Mumbai), Asie-Pacifique (Séoul), Asie-Pacifique (Singapour), Chine (Pékin), Chine (Ningxia), Asie-Pacifique (Sydney), Asie-Pacifique (Tokyo), Canada (Centre), Canada Ouest (Calgary), Europe (Francfort), Europe (Londres), Europe (Paris), Europe (Stockholm), Moyen-Orient (Bahreïn), Amérique du Sud (São Paulo)), Afrique (Le Cap), Asie-Pacifique (Malaisie) 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 comme 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 KDS votre configuration.
- La taille maximale d'un enregistrement envoyé à Amazon Data Firehose, avant le codage base64, est de 1 000 KiB.
- L'PutRecordBatchopération 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.
- Chacune des opérations suivantes peut fournir jusqu'à cinq appels par seconde, ce qui constitue une 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 un fichier CMK de type CUSTOMER _ MANAGED _ CMK pour chiffrer jusqu'à 500 streams Firehose.

Historique du document

Le tableau suivant décrit les modifications importantes apportées à la documentation Amazon Data Firehose.

Modification	Description	Date de modification
Exemples de types de données ajoutés	Ajout d'exemples de types de données pris en charge pour les tables Apache Iceberg. Consultez Comprendre les types de données pris en charge.	22 août 2024
Lancement d'une nouvelle région	Amazon Data Firehose est désormais disponible en Asie-Pacifique (Malaisie). Consultez Quota Amazon Data Firehose.	22 août 2024
Ajout des tables Apache Iceberg en tant que destination (version préliminaire publique)	Vous pouvez créer un flux Firehose avec les tables Apache Iceberg comme destination. Consultez Tables Apache Iceberg.	25 juillet 2024
Conseils de mise en mémoire tampon pour Snowflake	Snowflake prend désormais en charge la mise en mémoire tampon des indices. Consultez the section called “Configurer les paramètres de destination pour Snowflake”.	25 juillet 2024
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”.	25 juillet 2024
Sections du guide de l'utilisateur restructurées	Navigation simplifiée pour les sections du guide de l'utilisateur. Tu vois, Envoyer des données vers un flux Firehose etRésoudre les erreurs.	5 juillet 2024

Modification	Description	Date de modification
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. Consultez 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
Ajout de Snowflake comme destination (aperçu public)	Vous pouvez créer un stream Firehose avec Snowflake comme destination. Consultez 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. Consultez the section called “Envoyer CloudWatch les logs à Firehose”.	15 décembre 2023

Modification	Description	Date de modification
Ajout de Splunk Observability Cloud comme destination	Vous pouvez créer un stream Firehose avec Splunk Observability Cloud comme destination. Consultez 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. Consultez the section called “Configurer les paramètres de source pour 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. Consultez 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. Consultez Partitionner les données de streaming.	31 août 2021
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. Consultez the section called “Comprendre les préfixes personnalisés pour les objets Amazon S3”.	20 décembre 2018

Modification	Description	Date de modification
Ajout d'un nouveau didacticiel Amazon Data Firehose	Ajout d'un didacticiel expliquant comment envoyer des journaux de VPC flux Amazon à Splunk via Amazon Data Firehose. Consultez Ingérez les journaux de VPC flux 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 de plus amples informations, veuillez consulter Quota Amazon Data Firehose.	27 juin 2018
Ajout de deux nouvelles régions Amazon Data Firehose	Ajout de Séoul et Montréal. Pour de plus amples informations, veuillez consulter 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 de plus amples informations, veuillez consulter Choisissez la source et la destination de votre stream Firehose.	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 de plus amples informations, veuillez consulter Tutoriel : Création d'un stream Firehose depuis la console.	19 juillet 2017
Transformation de nouvelles données	Vous pouvez configurer Amazon Data Firehose pour transformer vos données avant leur livraison. Pour de plus amples informations, veuillez consulter Transformez les données sources dans Amazon Data Firehose.	19 décembre 2016

Modification	Description	Date de modification
Nouvelle tentative d'Amazon Redshift COPY	Vous pouvez configurer Amazon Data Firehose pour réessayer une COPY commande sur votre cluster Amazon Redshift en cas d'échec. Pour plus d'informations, consultez Tutoriel : Création d'un stream Firehose depuis la console, Comprendre la diffusion des données dans 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 Tutoriel : Création d'un stream Firehose depuis la console, Comprendre la diffusion des données dans Amazon Data Firehose et Accorder à Firehose l'accès à une destination de service public OpenSearch .	19 avril 2016
Nouvelles fonctionnalités de CloudWatch mesure et de résolution des problèmes améliorées	Surveillez Amazon Data Firehose mis à jour et Résoudre les erreurs dans Amazon Data Firehose.	19 avril 2016
Nouvel agent Kinesis amélioré	Mis à jour Configurer l'agent Kinesis pour envoyer des données.	11 avril 2016
Nouveaux agents Kinesis	Ajouté Configurer l'agent Kinesis pour envoyer des données.	2 octobre 2015
Première version	Première publication du guide du développeur Amazon Data Firehose.	4 octobre 2015