Guide de l'utilisateur

Amazon CloudWatch Logs

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon CloudWatch Logs: Guide de l’utilisateur

Copyright © 2025 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

Qu'est-ce qu'Amazon CloudWatch Logs ? ... 1
Fonctionnalités ... 1
AWS Services connexes ... 3
Tarification ... 4
Concepts ... 4
Facturation et coûts ... 6
Classes de log ... 7
Fonctionnalités prises en charge ... 7
Premiers pas ... 11
Prérequis ... 11
Inscrivez-vous pour un Compte AWS ... 11
Création d'un utilisateur doté d'un accès administratif ... 12
Configuration de l'interface de ligne de commande ... 13
Utilisation de l' CloudWatch agent unifié ... 14
Utilisation de l' CloudWatch agent précédent ... 14
CloudWatch Conditions préalables requises pour l'agent de journalisation ... 15
Démarrage rapide : installation de l'agent sur une instance EC2 Linux en cours d'exécution ... 16
Démarrage rapide : installer l'agent sur une instance EC2 Linux au lancement ... 23
Démarrage rapide : utilisation CloudWatch des journaux avec les instances de Windows Server 2016 ... 27
Démarrage rapide : utilisation CloudWatch des journaux avec les instances de Windows Server 2012 et Windows Server 2008 ... 39
Démarrage rapide : installez l'agent à l'aide de OpsWorks ... 50
Signaler le statut de l'agent CloudWatch Logs ... 56
Démarrez l'agent CloudWatch Logs ... 57
Arrêter l'agent CloudWatch Logs ... 57
Démarrage rapide avec AWS CloudFormation ... 58
Travailler avec AWS SDKs ... 60
Analyse des données des CloudWatch journaux avec Logs Insights ... 62
Langages de requête pris en charge ... 66
CloudWatch Langage de requête Logs Insights (Logs Insights QL) ... 66
OpenSearch Langage de traitement canalisé (PPL) ... 128
OpenSearch Langage de requête structuré (SQL) ... 137

Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Logs Insights ... 148
Exemples de requêtes ... 149
Refus d'utiliser vos données pour améliorer le service ... 152
Journaux pris en charge et champs découverts ... 153
Champs des journaux JSON ... 155
Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation ... 157
Syntaxe et quotas d'index de champs ... 159
Création d'une politique d'indexation des champs au niveau du compte ... 163
Création d'une politique d'index de champs au niveau du groupe de journaux ... 164
Options de sélection de groupes de journaux lors de la création d'une requête ... 165
Effets de la suppression d'une politique d'index de champs ... 166
Analyse de modèles ... 167
Commencer à utiliser l'analyse de modèles ... 168
Détails sur la commande pattern ... 170
Enregistrer et réexécuter des requêtes ... 171
Ajouter une requête au tableau de bord ou exporter les résultats de la requête ... 173
Afficher les requêtes en cours d'exécution ou l'historique des requêtes ... 174
Chiffrer les résultats de requête avec AWS Key Management Service ... 175
Limites ... 175
Étape 1 : Création d'un AWS KMS key ... 176
Étape 2 : Définition des autorisations sur la clé KMS ... 176
Étape 3 : Associer une clé KMS à vos résultats de requête ... 178
Étape 4 : Dissocier une clé des résultats de requête dans le compte ... 178
Génération d'un résumé en langage naturel à partir des résultats des requêtes de CloudWatch Logs Insights ... 179
Comment ça marche ... 179
Disponibilité régionale et traitement des données ... 179
Premiers pas ... 180
Autorisations ... 180
Confidentialité des données ... 181
Détection des anomalies du journal ... 182
Gravité et priorité des anomalies et des modèles ... 183
Durée de visibilité des anomalies ... 183
Supprimer une anomalie ... 184

Questions fréquentes (FAQ) ... 184
Activer la détection des anomalies sur un groupe de journaux ... 185
Afficher les anomalies détectées ... 187
Créez des alarmes sur les détecteurs d'anomalies du journal ... 190
Métriques publiées par les détecteurs d'anomalies logarithmiques ... 193
Chiffrez un détecteur d'anomalie et ses résultats avec AWS KMS ... 193
Limites ... 193
Résoudre les problèmes avec CloudWatch Logs Live Tail ... 198
Démarrez une session Live Tail à l'aide du AWS CLI ... 198
impression uniquement ... 199
interactif ... 199
Démarrer une session Live Tail dans la console ... 200
Utilisation des groupes de journaux et des flux de journaux ... 204
Création d'un groupe de journaux ... 204
Envoi de journaux à un groupe de journaux ... 205
Affichage des données de journal ... 205
Recherche de données journal au moyen de modèles de filtres ... 206
Recherche d'entrées de journal à l'aide de la console ... 206
Recherchez les entrées du journal à l'aide du AWS CLI ... 207
Transition des métriques aux journaux ... 208
Résolution des problèmes ... 208
Modification de la conservation des données de journaux ... 209
Étiquetage des groupes de journaux ... 210
Principes de base des balises ... 211
Suivi des coûts à l'aide d'étiquettes ... 211
Restrictions liées aux étiquettes ... 212
Marquage de groupes de journaux à l'aide du AWS CLI ... 212
Marquage de groupes de journaux à l'aide de l'API CloudWatch Logs ... 213
Chiffrez les données du journal à l'aide de AWS KMS ... 213
Limites ... 215
Étape 1 : Création d'une AWS KMS clé ... 176
Étape 2 : Définition des autorisations sur la clé KMS ... 176
Étape 3 : Association d'une clé KMS à un groupe de journaux ... 197
Étape 4 : Dissociation de la clé d'un groupe de journaux ... 197
Clés KMS et contexte de chiffrement ... 221
Aider à protéger les données sensibles des journaux grâce au masquage ... 224

Comprendre les politiques de protection des données ... 228
Autorisations IAM requises pour créer ou utiliser une politique de protection des données ... 230
Création d'une politique de protection des données à l'échelle du compte ... 236
Création d'une politique de protection des données pour un seul groupe de journaux ... 239
Affichage de données non masquées ... 243
Rapports de résultats d'audit ... 243
Types de données que vous pouvez protéger ... 245
Transformez les journaux lors de l'ingestion ... 290
Création et gestion de transformateurs de log ... 291
Création d'une politique de transformation au niveau du compte ... 293
Modifier ou supprimer une politique de transformation au niveau du compte ... 294
Créez un transformateur de log-group-level bûches à partir de zéro ... 295
Créez un log-group-level transformateur en copiant un transformateur existant ... 296
Modifier un log-group-level transformateur ... 297
Supprimer un log-group-level transformateur ... 298
Processeurs que vous pouvez utiliser ... 298
Processeurs configurables de type analyseur ... 300
Processeurs intégrés pour les AWS journaux vendus ... 337
Processeurs String Mutate ... 344
Processeurs de mutation JSON ... 354
Processeurs de conversion de types de données ... 368
Métriques de transformation et erreurs ... 372
Analyser avec Amazon OpenSearch Service ... 373
Étape 1 : Création de l'intégration avec le OpenSearch service ... 375
Autorisations requises ... 376
Création de l'intégration ... 383
Étape 2 : Création de tableaux de bord de journaux vendus ... 385
Afficher, modifier ou supprimer les tableaux de bord des journaux vendus ... 386
Afficher les tableaux de bord des journaux vendus dans CloudWatch Logs or Service OpenSearch ... 386
Accorder l'accès à des rôles IAM ou à des utilisateurs IAM supplémentaires pour consulter le tableau de bord ... 386
Modifier la configuration du tableau de bord ... 387
Supprimer un tableau de bord de vended log ... 387
Supprimer toutes les intégrations du tableau de bord des journaux automatiques avec Service OpenSearch ... 388

Politiques IAM pour les utilisateurs ... 389
Autorisations nécessaires à l'intégration ... 390
Filtres de métriques ... 393
Concepts ... 394
Syntaxe des modèles de filtres pour les filtres de métriques ... 395
Configuration des valeurs de métriques pour un filtre de métriques ... 397
Publication de dimensions avec des métriques à partir des événements du journal ... 397
Utilisation de valeurs dans des événements du journal pour incrémenter la valeur d'une métrique ... 401
Création de filtres de métriques ... 402
Créer un filtre de métrique pour un groupe de journaux ... 402
Exemple : Comptage des événements du journal ... 404
Exemple : Comptage des occurrences d'un terme ... 405
Exemple : Comptage du nombre de codes HTTP 404 ... 407
Exemple : comptage de codes HTTP 4xx ... 410
Exemple : Extraction des champs d'un journal Apache et attribution de dimensions ... 411
Liste des filtres de métriques ... 413
Suppression d'un filtre de métrique ... 414
Filtres d'abonnements ... 416
Concepts ... 417
Filtres d'abonnement au niveau des groupes de journaux ... 419
Exemple 1 : filtres d'abonnement avec Kinesis Data Streams ... 419
Exemple 2 : filtres d'abonnement avec AWS Lambda ... 426
Exemple 3 : filtres d'abonnement avec Amazon Data Firehose ... 429
Exemple 4 : filtres d'abonnement avec Amazon OpenSearch Service ... 437
Filtres d'abonnement au niveau du compte ... 438
Exemple 1 : filtres d'abonnement avec Kinesis Data Streams ... 439
Exemple 2 : filtres d'abonnement avec AWS Lambda ... 446
Exemple 3 : filtres d'abonnement avec Amazon Data Firehose ... 450
Abonnements entre comptes et entre régions ... 457
Partage de données de journal entre comptes et entre régions à l'aide de Kinesis Data Streams ... 458
Partage de données de journal entre comptes et entre régions à l'aide de Firehose ... 479
Abonnements entre comptes et entre régions à l'aide de Kinesis Data Streams ... 494
Abonnements entre comptes et entre régions à l'aide de Firehose ... 513
Prévention du député confus ... 526

Prévention de la récursivité dans les journaux ... 527
Syntaxe des modèles de filtres ... 529
Expressions régulières prises en charge ... 530
Faire correspondre les termes à l'aide d'expressions régulières ... 533
Faire correspondre les termes dans les événements du journal non structurés ... 534
Faire correspondre les termes dans les événements du journal JSON ... 537
Faire correspondre les termes dans les événements du journal délimités par des espaces ... 546
Activer la journalisation à partir AWS des services ... 552
Journalisation nécessitant des autorisations supplémentaires [V1] ... 563
Logs envoyés à CloudWatch Logs ... 564
Journaux envoyés à Amazon S3 ... 566
Logs envoyés à Firehose ... 571
Journalisation nécessitant des autorisations supplémentaires [V2] ... 573
Logs envoyés à CloudWatch Logs ... 574
Journaux envoyés à Amazon S3 ... 577
Logs envoyés à Firehose ... 581
Traces envoyées à X-Ray ... 584
Autorisations spécifiques au service ... 587
Autorisations spécifiques à la console ... 588
Exemple de livraison entre comptes ... 589
Création d'une source de livraison ... 590
Configuration de la livraison vers un compartiment Amazon S3 ... 590
Configurer la diffusion vers un flux Firehose ... 594
Prévention du cas de figure de l'adjoint désorienté entre services ... 596
Exporter les données du journal vers Amazon S3 ... 598
Concepts ... 600
Exporter les données du journal vers Amazon S3 à l'aide de la console ... 600
Exportation vers le même compte ... 601
Exportation intercomptes ... 608
Exportez les données du journal vers Amazon S3 à l'aide du AWS CLI ... 618
Exportation vers le même compte ... 618
Exportation intercomptes ... 626
Décrire les tâches d'exportation ... 635
Annuler une tâche d'exportation ... 636
Transmission de données vers le OpenSearch service ... 637
Prérequis ... 637

Abonnement d'un groupe de logs au OpenSearch Service ... 638
Exemples de code ... 640
Principes de base ... 641
Actions ... 642
Scénarios ... 701
Exécuter une requête volumineuse ... 702
Utilisent des événements planifiés pour invoquer une fonction Lambda ... 717
Sécurité ... 720
Protection des données ... 721
Chiffrement au repos ... 722
Chiffrement en transit ... 722
Gestion des identités et des accès ... 722
Authentification ... 723
Contrôle d'accès ... 723
Présentation de la gestion des accès ... 724
Utilisation des politiques basées sur une identité (politiques IAM) ... 730
Validation de conformité ... 756
Résilience ... 757
Sécurité de l'infrastructure ... 757
Points de terminaison de VPC d'Interface ... 758
Disponibilité ... 759
Création d'un point de terminaison VPC pour les journaux CloudWatch ... 759
Test de la connexion entre votre VPC et Logs CloudWatch ... 759
Contrôle de l'accès à votre point de CloudWatch terminaison Logs VPC ... 760
Prise en charge des clés de contexte de VPC ... 761
Journalisation des opérations d'API et de console avec AWS CloudTrail ... 762
Informations de génération de requêtes dans CloudTrail ... 767
Présentation des entrées des fichiers journaux ... 768
Référence de l'agent ... 770
Fichier de configuration de l'agent ... 770
Utilisation de l'agent CloudWatch Logs avec des proxys HTTP ... 777
Compartimentation des fichiers de configuration de l'agent CloudWatch Logs ... 778
CloudWatch FAQ sur les agents de journalisation ... 778
Surveillance de l'utilisation à l'aide de CloudWatch métriques ... 783
CloudWatch Métriques des journaux ... 783
Dimensions pour les métriques CloudWatch Logs ... 787

Métriques et dimensions des transformateurs de log 788
CloudWatch Statistiques d'utilisation du service de journalisation 789
Quotas de service 792
Gestion des quotas de votre service CloudWatch Logs 800
Historique du document 802
AWS Glossaire 816
..... dcccxvii

Qu'est-ce qu'Amazon CloudWatch Logs ?

Vous pouvez utiliser Amazon CloudWatch Logs pour surveiller, stocker et accéder à vos fichiers journaux à partir d'instances Amazon Elastic Compute Cloud (Amazon EC2) AWS CloudTrail, de Route 53 et d'autres sources.

CloudWatch Les journaux vous permettent de centraliser les journaux de tous les systèmes, applications et AWS services que vous utilisez, au sein d'un seul service hautement évolutif. Vous pouvez ensuite facilement les consulter, y rechercher des codes ou modèles d'erreur spécifiques, les filtrer en fonction de champs spécifiques ou les archiver en toute sécurité pour une analyse future. CloudWatch Les journaux vous permettent de voir tous vos journaux, quelle que soit leur source, sous la forme d'un flux unique et cohérent d'événements classés par ordre chronologique.

CloudWatch Logs permet également d'interroger vos journaux à l'aide d'un langage de requête puissant, d'auditer et de masquer les données sensibles dans les journaux, et de générer des métriques à partir des journaux à l'aide de filtres ou d'un format de journal intégré.

CloudWatch Logs prend en charge deux classes de journaux. Les groupes de journaux de la classe de CloudWatch journaux Logs Standard prennent en charge toutes les fonctionnalités CloudWatch des journaux. Les groupes de CloudWatch journaux de la classe Logs Infrequent Access entraînent des frais d'ingestion moins élevés et prennent en charge un sous-ensemble des fonctionnalités de la classe Standard. Pour de plus amples informations, veuillez consulter Classes de log.

Fonctionnalités

- Deux classes de CloudWatch journaux pour plus de flexibilité — Logs propose deux classes de journaux, ce qui vous permet de disposer d'une option rentable pour les journaux auxquels vous accédez rarement. Vous disposez également d'une option complète pour les journaux qui nécessitent une surveillance en temps réel ou d'autres fonctionnalités. Pour de plus amples informations, veuillez consulter Classes de log.
- Interrogez les données de vos journaux : vous pouvez utiliser CloudWatch Logs Insights pour rechercher et analyser de manière interactive les données de vos journaux. Vous pouvez effectuer des requêtes pour vous aider à répondre de manière plus efficace aux problèmes opérationnels. CloudWatch Logs Insights inclut un langage de requête spécialement conçu avec quelques commandes simples mais puissantes. Nous fournissons des exemples de requête, des descriptions de commande, la saisie automatique de requête et la découverte des champs de

journal pour vous aider à démarrer. Des exemples de requêtes sont inclus pour plusieurs types de journaux de AWS service. Consultez Analyse des données des CloudWatch journaux avec Logs Insights pour démarrer.

- Créez des index de champs pour rendre les requêtes plus efficaces — Vous pouvez créer des index de champs dans votre journal des événements. Lorsque vous utilisez ensuite un index de champ dans une requête CloudWatch Logs Insights, la requête tente d'ignorer le traitement des événements du journal dont on sait qu'ils n'incluent pas le champ indexé. Cette requête réduit le volume d'analyse de vos requêtes, ce qui permet de renvoyer des résultats plus rapidement. Consultez Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation pour démarrer.
- Détecter et déboguer à l'aide de Live Tail : vous pouvez utiliser Live Tail pour résoudre rapidement les incidents en consultant une liste des nouveaux événements du journal au fur et à mesure de leur ingestion. Vous pouvez afficher, filtrer et mettre en évidence les journaux ingérés en temps quasi réel, ce qui vous permet de détecter et de résoudre rapidement les problèmes. Vous pouvez filtrer les journaux en fonction des termes que vous spécifiez et mettre en évidence les journaux qui contiennent les termes spécifiés pour vous aider à trouver rapidement ce que vous cherchez. Pour de plus amples informations, veuillez consulter Résoudre les problèmes avec CloudWatch Logs Live Tail.
- Surveillez les journaux des EC2 instances Amazon : vous pouvez utiliser CloudWatch les journaux pour surveiller les applications et les systèmes à l'aide des données des journaux. Par exemple, CloudWatch Logs peut suivre le nombre d'erreurs qui se produisent dans les journaux de vos applications et vous envoyer une notification chaque fois que le taux d'erreurs dépasse un seuil que vous spécifiez. CloudWatch Logs utilise les données de vos journaux à des fins de surveillance ; aucune modification de code n'est donc requise. Par exemple, vous pouvez surveiller les journaux des applications pour détecter des termes littéraux spécifiques (tels que NullReferenceException « ») ou compter le nombre d'occurrences d'un terme littéral à une position donnée dans les données des journaux (tels que les codes d'état « 404 » dans un journal d'accès Apache). Lorsque le terme que vous recherchez est trouvé, CloudWatch Logs rapporte les données selon une CloudWatch métrique que vous spécifiez. Les données des journaux sont chiffrées, pendant le transit et pendant le repos. Consultez Commencer à utiliser CloudWatch Logs pour démarrer.
- Surveiller les événements AWS CloudTrail enregistrés : vous pouvez créer des alarmes CloudWatch et recevoir des notifications concernant une activité d'API particulière telle qu'elle est capturée, CloudTrail et utiliser la notification pour résoudre les problèmes. Pour commencer, consultez la section Envoyer CloudTrail des événements aux CloudWatch journaux dans le guide de AWS CloudTrail l'utilisateur.

- Auditez et masquez les données sensibles : si vos journaux contiennent des données sensibles, vous pouvez les protéger grâce à des politiques de protection des données. Ces politiques vous permettent d'auditer et de masquer les données sensibles. Si vous activez la protection des données, les données sensibles correspondant aux identifiants de données que vous sélectionnez sont masquées par défaut. Pour de plus amples informations, veuillez consulter Aider à protéger les données sensibles des journaux grâce au masquage.
- Conservation des journaux : par défaut, les journaux sont conservés indéfiniment et n'expirent jamais. Vous pouvez ajuster la stratégie de conservation pour chaque groupe de journaux. Elle peut être indéfinie ou comprise entre 10 ans et un jour.
- Archiver les données du journal : vous pouvez utiliser CloudWatch les journaux pour stocker les données de vos journaux dans un espace de stockage hautement durable. L'agent CloudWatch Logs permet d'envoyer rapidement des données de journal avec ou sans rotation depuis un hôte vers le service de journalisation. Vous pouvez ensuite accéder aux données brutes des journaux lorsque vous en avez besoin.
- Consigner les requêtes DNS de Route 53 : vous pouvez utiliser CloudWatch les journaux pour enregistrer les informations relatives aux requêtes DNS reçues par Route 53. Pour plus d'informations, consultez Consignation des requêtes DNS dans le Guide du développeur Amazon Route 53.

AWS Services connexes

Les services suivants sont utilisés conjointement avec CloudWatch Logs :

- AWS CloudTrailest un service Web qui vous permet de surveiller les appels passés à l'API CloudWatch Logs pour votre compte, y compris les appels passés par le AWS Management Console, AWS Command Line Interface (AWS CLI) et d'autres services. Lorsque la CloudTrail journalisation est activée, elle CloudTrail capture les appels d'API dans votre compte et envoie les fichiers journaux au compartiment Amazon S3 que vous spécifiez. Chaque fichier journal peut contenir un ou plusieurs enregistrements, selon le nombre d'actions à effectuer afin de satisfaire une demande. Pour plus d'informations AWS CloudTrail, voir Qu'est-ce que c'est AWS CloudTrail ? dans le guide de AWS CloudTrail l'utilisateur. Pour obtenir un exemple du type de données enregistrées CloudWatch dans les fichiers CloudTrail journaux, consultezLogging CloudWatch Logs, API et opérations de console dans AWS CloudTrail.
- AWS Identity and Access Management (IAM) est un service Web qui vous permet de contrôler en toute sécurité l'accès aux AWS ressources pour vos utilisateurs. Utilisez IAM pour contrôler qui peut utiliser vos ressources AWS (authentification) et quelles ressources pourront être utilisées de

quelle manière (autorisation). Pour plus d'informations, consultez Qu'est-ce qu'IAM ? dans le Guide de l'utilisateur IAM.

- Amazon Kinesis Data Streams est un service Web que vous pouvez utiliser pour une extraction et un regroupement de données rapides et en continu. Le type de données utilisées inclut des données de journaux d'infrastructure informatique, des journaux d'applications, des réseaux sociaux, des flux de données du marché et des données de flux de clics Web. Comme le temps de réponse pour la récupération et le traitement des données est en temps réel, le traitement est généralement léger. Pour plus d'informations, consultez Présentation d'Amazon Kinesis Data Streams dans le Guide du développeur Amazon Kinesis Data Streams.
- AWS Lambda est un service Web que vous pouvez utiliser pour créer des applications qui apporteront une réponse rapide à de nouvelles informations. Téléchargez votre code d'application sous la forme de fonctions Lambda pour que Lambda l'exécute sur une infrastructure de calcul à haute disponibilité et effectue toute l'administration des ressources de calcul, y compris la maintenance du serveur et du système d'exploitation, l'approvisionnement des capacités et la scalabilité automatique, le déploiement du code et des correctifs de sécurité, ainsi que la surveillance et la journalisation du code. Il vous suffit de fournir votre code dans l'un des langages pris en charge par Lambda. Pour plus d'informations, voir Qu'est-ce que c'est AWS Lambda ? dans le Guide AWS Lambda du développeur.

Tarification

Lorsque vous vous inscrivez AWS, vous pouvez commencer à utiliser CloudWatch Logs gratuitement en utilisant le niveau AWS gratuit.

Les tarifs standard s'appliquent aux journaux stockés par d'autres services à l'aide de CloudWatch journaux (par exemple, les journaux de flux Amazon VPC et les journaux Lambda).

Pour plus d'informations sur les tarifs, consultez Amazon CloudWatch Pricing.

Pour plus d'informations sur la manière d'analyser vos coûts et votre utilisation CloudWatch des journaux et CloudWatch pour connaître les meilleures pratiques en matière de réduction des coûts, consultez la section CloudWatch facturation et coûts.

Concepts CloudWatch d'Amazon Logs

La terminologie et les concepts essentiels à votre compréhension et à votre utilisation des CloudWatch journaux sont décrits ci-dessous.

Classe de log

CloudWatch Logs propose deux classes de groupes de journaux. La classe de journal standard est une option complète pour les journaux qui nécessitent une surveillance en temps réel ou pour les journaux auxquels vous accédez fréquemment. La classe de journaux Infrequent Access est une option moins coûteuse pour les journaux auxquels vous accédez moins fréquemment. Il prend en charge un sous-ensemble des fonctionnalités de la classe log standard.

Événements de journaux

Un événement de journal est l'enregistrement d'une activité effectué par l'application ou la ressource sous surveillance. L'enregistrement d'événements du journal compris par CloudWatch Logs contient deux propriétés : l'horodatage de l'événement et le message d'événement brut. Les messages d'événements doivent être codé en UTF-8.

Flux de journaux

Un flux de journal est une séquence d'événements du journaux qui partagent la même source. Plus précisément, un flux de journal est généralement destiné à représenter la séquence des événements provenant de l'instance d'application ou de la ressource sous surveillance. Par exemple, un flux de journal peut être associé à un journal des accès Apache sur un hôte spécifique. Lorsque vous n'avez plus besoin d'un flux de journal, vous pouvez le supprimer à l'aide de la delete-log-stream commande aws logs.

Groupes de journaux

Les groupes de journaux définissent des groupes de flux de journaux qui partagent les mêmes paramètres de conservation, de surveillance et de contrôle d'accès. Chaque flux de journal doit appartenir à un groupe de journaux. Par exemple, si vous disposez d'un flux de journal distinct pour les journaux d'accès Apache de chaque hôte, vous pouvez regrouper ces flux de journaux dans un seul groupe de journaux appelé `MyWebsite.com/Apache/access_log`.

Le nombre de flux de journaux pouvant appartenir à un groupe de journaux est illimité.

Filtres de métriques

Vous pouvez utiliser les filtres de métriques pour extraire les observations de métriques à partir des événements obtenus et les transformer en points de données d'une métrique CloudWatch . Les filtres de métriques sont associés aux groupes de journaux, et tous les filtres affectés à un groupe sont appliqués à ses flux de journaux.

Paramètres de conservation

Les paramètres de conservation peuvent être utilisés pour spécifier la durée pendant laquelle les événements du journal sont conservés dans CloudWatch les journaux. Les événements de journaux arrivés à expiration sont supprimés automatiquement. Comme les filtres de métriques, les paramètres de conservation sont également affectés aux groupes de journaux et la conservation associée à un groupe est appliquée à ses flux de journaux.

Facturation et coûts d'Amazon CloudWatch Logs

Pour obtenir des informations détaillées sur la manière d'analyser vos coûts et votre utilisation CloudWatch des journaux et CloudWatch pour connaître les meilleures pratiques en matière de réduction des coûts, consultez la section CloudWatch facturation et coûts.

Pour plus d'informations sur les tarifs, consultez Amazon CloudWatch Pricing.

Lorsque vous vous inscrivez AWS, vous pouvez commencer à utiliser CloudWatch Logs gratuitement en utilisant le niveau AWS gratuit.

Les tarifs standard s'appliquent aux journaux stockés par d'autres services à l'aide de CloudWatch journaux (par exemple, les journaux de flux Amazon VPC et les journaux Lambda).

Classes de log

CloudWatch Logs propose deux catégories de groupes de journaux :

- La classe de CloudWatch journaux Logs Standard est une option complète pour les journaux qui nécessitent une surveillance en temps réel ou pour les journaux auxquels vous accédez fréquemment.
- La classe de CloudWatch journaux Logs Infrequent Access est une nouvelle classe de journaux que vous pouvez utiliser pour consolider vos journaux de manière rentable. Cette classe de CloudWatch journaux propose un sous-ensemble de fonctionnalités de journalisation, notamment l'ingestion gérée, le stockage, l'analyse des journaux entre comptes et le chiffrement, avec un prix d'ingestion inférieur par Go. La classe de journaux Infrequent Access est idéale pour les requêtes ad hoc et les analyses after-the-fact médico-légales sur les journaux rarement consultés.

> **Note**
>
> En ce qui concerne les frais, les classes de log d'accès standard et peu fréquent diffèrent uniquement en termes de coûts d'ingestion. Les frais de stockage et CloudWatch les frais de Logs Insights sont les mêmes dans chaque classe de journaux.

Pour plus d'informations sur la tarification des CloudWatch journaux, consultez Amazon CloudWatch Pricing.

> **Important**
>
> Une fois qu'un groupe de journaux est créé, sa classe de journal ne peut pas être modifiée.

Fonctionnalités prises en charge

Le tableau suivant répertorie les fonctionnalités de chaque classe de log.

Fonctionnalité	Standard	Accès peu fréquent
Gestion complète de l'ingestion et du stockage des journaux	Oui ✓	Oui ✓
Fonctionnalités multi-comptes	Oui ✓	Oui ✓
Chiffrement avec AWS KMS	Oui ✓	Oui ✓
CloudWatch Commandes de requête Logs Insights	Oui ✓	Oui ✓ (La plupart des commandes, voirCommandes Logs Insights QL prises en charge dans les classes de journaux.)
CloudWatch Champs découverts par Logs Insights	Oui ✓	Oui ✓
Utiliser OpenSearch PPL ou OpenSearch SQL pour effectuer des requêtes dans CloudWatch Logs Insights ;	Oui ✓	Non
Assistance aux requêtes en langage naturel	Oui ✓	Non
CloudWatch Détection des anomalies dans les journaux	Oui ✓	Non
Tail en direct	Oui ✓	Non
Indexation des champs	Oui ✓	Non
Comparer avec la plage horaire précédente	Oui ✓	Non
Filtres d'abonnement	Oui ✓	Non

Fonctionnalité	Standard	Accès peu fréquent
Exporter vers Amazon S3	Oui ✓	Non
GetLogEventset opérations FilterLog Eventsd'API	Oui ✓	Non pris en charge. Utilisez CloudWatch Logs Insights pour afficher les événements de journal stockés dans des groupes de journaux de la classe de journaux Infrequent Access.
Filtres métriques	Oui ✓	Non
Container Insights enregistre l'ingestion	Oui ✓	Non
Ingestion du journal Lambda Insights	Oui ✓	Non
Protection des données sensibles grâce au masquage	Oui ✓	Non
Format de métriques intégré	Oui ✓	Non

> **Note**
>
> Outre ces deux classes de log, il existe une classe de `Delivery` log. Utilisez la classe de `Delivery` log uniquement pour fournir AWS Lambda des journaux à stocker dans Amazon S3 ou Amazon Data Firehose. Les événements de journal des groupes de journaux de

la classe Delivery ne sont conservés que CloudWatch Logs pendant une journée. Cette classe de log n'offre pas de CloudWatch Logs fonctionnalités riches telles que les requêtes CloudWatch Logs Insights.

Commencer à utiliser CloudWatch Logs

Pour collecter les journaux de vos EC2 instances Amazon et de vos serveurs sur site dans CloudWatch Logs, utilisez l' CloudWatch agent unifié. Il vous permet de collecter à la fois les journaux et les métriques avancées avec un agent. Il propose une assistance sur les systèmes d'exploitation, y compris les serveurs exécutant Windows Server. Cet agent propose également de meilleures performances.

Si vous utilisez l' CloudWatch agent unifié pour collecter des CloudWatch métriques, il permet de collecter des métriques système supplémentaires, pour une visibilité auprès des clients. Il prend également en charge la collecte des métriques personnalisées à l'aide de `StatsD` ou `collectd`.

Pour plus d'informations, consultez la section Installation de l' CloudWatch agent dans le guide de CloudWatch l'utilisateur Amazon.

L'ancien agent CloudWatch Logs, qui prend uniquement en charge la collecte de journaux à partir de serveurs exécutant Linux, est obsolète et n'est plus pris en charge. Pour plus d'informations sur la migration de l'ancien agent CloudWatch Logs vers l'agent unifié, voir Création du fichier de configuration de l' CloudWatch agent avec l'assistant.

Table des matières

- Prérequis
- Utilisez l' CloudWatch agent unifié pour démarrer avec CloudWatch Logs
- Utiliser l' CloudWatch agent précédent pour commencer à utiliser CloudWatch Logs
- Démarrage rapide : AWS CloudFormation à utiliser pour commencer à utiliser les CloudWatch journaux

Prérequis

Pour utiliser Amazon CloudWatch Logs, vous avez besoin d'un AWS compte. Votre AWS compte vous permet d'utiliser des services (par exemple, Amazon EC2) pour générer des journaux que vous pouvez consulter dans la CloudWatch console, une interface Web. En outre, vous pouvez installer et configurer le AWS Command Line Interface (AWS CLI).

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

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 ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce 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.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu'utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d'un ensemble d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l'utilisateur AWS IAM Identity Center .

Configuration de l'interface de ligne de commande

Vous pouvez utiliser le AWS CLI pour effectuer des opérations de CloudWatch journalisation.

Pour plus d'informations sur l'installation et la configuration du AWS CLI, consultez la section Mise en place de l'interface de ligne de AWS commande dans le guide de AWS Command Line Interface l'utilisateur.

Utilisez l' CloudWatch agent unifié pour démarrer avec CloudWatch Logs

Pour plus d'informations sur l'utilisation de l' CloudWatch agent unifié pour démarrer avec les CloudWatch journaux, consultez la section Collecter des métriques et des journaux à partir d' EC2 instances Amazon et de serveurs sur site avec l' CloudWatch agent dans le guide de l' CloudWatch utilisateur Amazon. Effectuez les étapes indiquées dans cette section pour installer, configurer et démarrer l'agent. Si vous n'utilisez pas l'agent pour collecter également des CloudWatch métriques, vous pouvez ignorer les sections qui font référence aux métriques.

Si vous utilisez actuellement l'ancien agent CloudWatch Logs et que vous souhaitez migrer vers le nouvel agent unifié, nous vous recommandons d'utiliser l'assistant inclus dans le nouveau package d'agent. Cet assistant peut lire le fichier de configuration actuel de votre agent CloudWatch Logs et configurer l' CloudWatchagent pour qu'il collecte les mêmes journaux. Pour plus d'informations sur l'assistant, consultez la section Créer le fichier de configuration de l' CloudWatch agent avec l'assistant dans le guide de CloudWatch l'utilisateur Amazon.

Utiliser l' CloudWatch agent précédent pour commencer à utiliser CloudWatch Logs

> **Important**
>
> CloudWatch inclut un CloudWatch agent unifié capable de collecter à la fois des journaux et des métriques à partir d' EC2 instances et de serveurs sur site. L'ancien agent de journaux uniquement est obsolète et n'est plus pris en charge.
>
> Pour plus d'informations sur la migration de l'ancien agent uniquement basé sur les journaux vers l'agent unifié, voir Création du fichier de configuration de l' CloudWatch agent avec l'assistant.
>
> Le reste de cette section explique l'utilisation de l'ancien agent CloudWatch Logs pour les clients qui l'utilisent toujours.

À l'aide de l'agent CloudWatch Logs, vous pouvez publier des données de journal à partir d' EC2 instances Amazon exécutant Linux ou Windows Server, ainsi que des événements enregistrés à partir de AWS CloudTrail. Nous vous recommandons plutôt d'utiliser l'agent CloudWatch unifié pour publier les données de vos journaux. Pour plus d'informations sur le nouvel agent, consultez la section Collecter des métriques et des journaux à partir d' EC2 instances Amazon et de serveurs sur site avec l' CloudWatch agent dans le guide de l' CloudWatch utilisateur Amazon.

Table des matières

- CloudWatch Conditions préalables requises pour l'agent de journalisation
- Démarrage rapide : installation et configuration de l'agent CloudWatch Logs sur une instance EC2 Linux en cours d'exécution
- Démarrage rapide : installer et configurer l'agent CloudWatch Logs sur une instance EC2 Linux au lancement
- Démarrage rapide : autorisez vos EC2 instances Amazon exécutant Windows Server 2016 à envoyer des journaux à Logs à CloudWatch l'aide de l'agent CloudWatch Logs
- Démarrage rapide : autorisez vos EC2 instances Amazon exécutant Windows Server 2012 et Windows Server 2008 à envoyer des CloudWatch journaux vers Logs
- Démarrage rapide : installez l'agent CloudWatch Logs à l'aide de AWS OpsWorks et Chef
- Signaler le statut de l'agent CloudWatch Logs
- Démarrez l'agent CloudWatch Logs
- Arrêter l'agent CloudWatch Logs

CloudWatch Conditions préalables requises pour l'agent de journalisation

L'agent CloudWatch Logs nécessite la version 2.7, 3.0 ou 3.3 de Python et l'une des versions suivantes de Linux :

- Amazon Linux version 2014.03.02 ou ultérieure. Amazon Linux 2 n'est pas pris en charge
- Ubuntu Server version 12.04, 14.04 ou 16.04
- CentOS 6, 6.3, 6.4, 6.5 ou 7.0
- Red Hat Enterprise Linux (RHEL) 6.5 ou 7.0
- Debian 8.0

Démarrage rapide : installation et configuration de l'agent CloudWatch Logs sur une instance EC2 Linux en cours d'exécution

> **Important**
>
> L'ancien agent de journalisation est obsolète. CloudWatch inclut un agent unifié capable de collecter à la fois des journaux et des métriques à partir d' EC2 instances et de serveurs sur site. Pour de plus amples informations, veuillez consulter Commencer à utiliser CloudWatch Logs.
>
> Pour plus d'informations sur la migration de l'ancien agent CloudWatch Logs vers l'agent unifié, voir Création du fichier de configuration de l' CloudWatch agent avec l'assistant.
>
> L'ancien agent de journalisation ne prend en charge que les versions 2.6 à 3.5 de Python. En outre, l'ancien agent CloudWatch Logs ne prend pas en charge le service de métadonnées d'instance version 2 (IMDSv2). Si votre serveur l'utilise IMDSv2, vous devez utiliser le nouvel agent unifié au lieu de l'ancien agent CloudWatch Logs.
>
> Le reste de cette section explique l'utilisation de l'ancien agent CloudWatch Logs pour les clients qui l'utilisent toujours.

> **Tip**
>
> CloudWatch inclut un nouvel agent unifié capable de collecter à la fois des journaux et des métriques à partir d' EC2 instances et de serveurs sur site. Si vous n'utilisez pas encore l'ancien agent CloudWatch Logs, nous vous recommandons d'utiliser le nouvel CloudWatch agent unifié. Pour de plus amples informations, veuillez consulter Commencer à utiliser CloudWatch Logs.
>
> En outre, l'ancien agent ne prend pas en charge le service de métadonnées d'instance version 2 (IMDSv2). Si votre serveur l'utilise IMDSv2, vous devez utiliser le nouvel agent unifié au lieu de l'ancien agent CloudWatch Logs.
>
> Le reste de cette section explique l'utilisation de l'ancien agent CloudWatch Logs.

Configurer l'ancien agent CloudWatch Logs sur une instance EC2 Linux en cours d'exécution

Vous pouvez utiliser le programme d'installation de l'agent CloudWatch Logs sur une EC2 instance existante pour installer et configurer l'agent CloudWatch Logs. Une fois l'installation terminée, les

journaux sont automatiquement transmis à partir de l'instance vers le flux de journaux que vous créez lors de l'installation de l'agent. L'agent confirme qu'il a commencé à s'exécuter et ce, jusqu'à ce que vous le désactiviez.

Outre l'agent, vous pouvez également publier les données des journaux à l'aide du AWS CLI SDK CloudWatch Logs ou de l'API CloudWatch Logs. AWS CLI Il convient parfaitement à la publication de données en ligne de commande ou via des scripts. Le SDK CloudWatch Logs convient parfaitement à la publication de données de journaux directement à partir d'applications ou à la création de votre propre application de publication de journaux.

Étape 1 : Configuration de votre rôle ou utilisateur IAM pour Logs CloudWatch

L'agent CloudWatch Logs prend en charge les rôles et les utilisateurs IAM. Si votre instance est déjà associée à un rôle IAM, assurez-vous d'inclure la politique IAM ci-dessous. Si aucun rôle IAM n'est déjà attribué à votre instance, vous pouvez utiliser vos informations d'identification IAM pour les prochaines étapes ou attribuer un rôle IAM à cette instance. Pour plus d'informations, consultez Attachement d'un rôle IAM à une instance.

Pour configurer votre rôle ou utilisateur IAM pour Logs CloudWatch

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Choisissez le rôle en sélectionnant le nom du rôle (ne cochez pas la case en regard du nom).
4. Choisissez Attach Policies, Create Policy.

   Un nouvel onglet (ou une nouvelle fenêtre) s'ouvre dans votre navigateur.
5. Choisissez l'onglet JSON et copiez le document de stratégie JSON suivant.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
```

```
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "*"
    ]
  }
 ]
}
```

6. Lorsque vous avez terminé, sélectionnez Examiner une politique. Le programme de validation de stratégie signale les éventuelles erreurs de syntaxe.
7. Dans la page Examiner une politique, saisissez un nom et une description (facultatif) pour la politique que vous êtes en train de créer. Vérifiez le récapitulatif de politique pour voir les autorisations accordées par votre politique. Sélectionnez ensuite Créer une politique pour enregistrer votre travail.
8. Fermez la fenêtre ou l'onglet du navigateur, et retournez dans la page Ajouter des autorisations pour votre rôle. Choisissez Refresh, puis choisissez la nouvelle stratégie pour l'attacher à votre rôle.
9. Choisissez Attach Policy (Attacher une politique).

Étape 2 : Installation et configuration CloudWatch des journaux sur une EC2 instance Amazon existante

Le processus d'installation de l'agent CloudWatch Logs varie selon que votre EC2 instance Amazon exécute Amazon Linux, Ubuntu, CentOS ou Red Hat. Utilisez les étapes correspondant à la version de Linux utilisée par votre instance.

Pour installer et configurer CloudWatch les journaux sur une instance Amazon Linux existante

À partir de l'AMI Amazon Linux 2014.09, l'agent CloudWatch Logs est disponible sous forme d'installation RPM avec le package awslogs. Les versions antérieures d'Amazon Linux peuvent accéder au package awslogs en mettant à jour leur instance avec la commande `sudo yum update -y`. En installant le package awslogs sous forme de RPM au lieu d'utiliser le programme d'installation de CloudWatch Logs, votre instance reçoit régulièrement des mises à jour du package et des correctifs AWS sans avoir à réinstaller manuellement l'agent CloudWatch Logs.

> **Warning**
>
> Ne mettez pas à jour l'agent CloudWatch Logs à l'aide de la méthode d'installation RPM si vous avez déjà utilisé le script Python pour installer l'agent. Cela peut entraîner des problèmes de configuration qui empêcheront l'agent CloudWatch Logs d'envoyer vos journaux à CloudWatch.

1. Connectez-vous à votre instance Amazon Linux. Pour plus d'informations, consultez Connect to Your Instance dans le guide de EC2 l'utilisateur Amazon.

   Pour plus d'informations sur les problèmes de connexion, consultez la section Résolution des problèmes de connexion à votre instance dans le guide de EC2 l'utilisateur Amazon.
2. Mettez à jour votre instance Amazon Linux pour récupérer les dernières modifications des référentiels de package.

   ```
   sudo yum update -y
   ```
3. Installez le package `awslogs`. Il est recommandé d'utiliser cette méthode pour l'installation du package awslogs sur des instances Amazon Linux.

   ```
   sudo yum install -y awslogs
   ```
4. Modifiez le fichier `/etc/awslogs/awslogs.conf` pour configurer les journaux à suivre. Pour plus d'informations sur la modification de ce fichier, consultez CloudWatch Référence de l'agent de journalisation.
5. Par défaut, `/etc/awslogs/awscli.conf` pointe vers la région us-east-1. Pour transmettre vos journaux vers une autre région, modifiez le fichier `awscli.conf` et spécifiez cette région.
6. Lancez le service `awslogs`.

   ```
   sudo service awslogs start
   ```

   Si vous utilisez Amazon Linux 2, lancez le service `awslogs` à l'aide de la commande suivante.

   ```
   sudo systemctl start awslogsd
   ```
7. (Facultatif) Vérifiez si le fichier `/var/log/awslogs.log` contient des erreurs lorsque vous lancez le service.

8. (Facultatif) Exécutez la commande suivante pour lancer le service `awslogs` à chaque démarrage du système.

```
sudo chkconfig awslogs on
```

   Si vous utilisez Amazon Linux 2, utilisez la commande suivante pour lancer le service à chaque démarrage système.

```
sudo systemctl enable awslogsd.service
```

9. Le nouveau groupe de journaux et le nouveau flux de journal doivent être visibles dans la console CloudWatch une fois que l'agent s'exécute depuis quelques instants.

   Pour de plus amples informations, veuillez consulter Afficher les données du journal envoyées à CloudWatch Logs.

Pour installer et configurer CloudWatch les journaux sur une instance Ubuntu Server, CentOS ou Red Hat existante

Si vous utilisez une AMI exécutant Ubuntu Server, CentOS ou Red Hat, suivez la procédure suivante pour installer manuellement l'agent CloudWatch Logs sur votre instance.

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

   Pour plus d'informations sur les problèmes de connexion, consultez la section Résolution des problèmes de connexion à votre instance dans le guide de EC2 l'utilisateur Amazon.

2. Exécutez le programme d'installation de l'agent CloudWatch Logs à l'aide de l'une des deux options suivantes. Vous pouvez l'exécuter directement à partir d'Internet ou télécharger les fichiers et l'exécuter de manière autonome.

   > **Note**
   >
   > Si vous utilisez CentOS 6.x, Red Hat 6.x ou Ubuntu 12.04, suivez les étapes de téléchargement et d'exécution du programme d'installation autonome. L'installation de l'agent CloudWatch Logs directement depuis Internet n'est pas prise en charge sur ces systèmes.

> **Note**
>
> Sur Ubuntu, exécutez `apt-get update` avant d'exécuter les commandes ci-dessous.

Pour l'exécuter directement à partir d'Internet, utilisez les commandes suivantes et suivez les instructions :

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py -O
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

Si la commande précédente ne fonctionne pas, essayez ce qui suit :

```
sudo python3 ./awslogs-agent-setup.py --region us-east-1
```

Pour le télécharger et l'exécuter de manière autonome, utilisez les commandes suivantes et suivez les instructions :

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py -O
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/
AgentDependencies.tar.gz -O
```

```
tar xvf AgentDependencies.tar.gz -C /tmp/
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/
AgentDependencies
```

Vous pouvez installer l'agent CloudWatch Logs en spécifiant us-east-1, us-west-1, us-west-2, ap-southeast-1, ap-southeast-2, ap-southeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-2, ap-northeast-2, ap-northeast-2 Régions ap-northeast-1, eu-central-1, eu-west-1 ou sa-east-1.

> **Note**
>
> Pour plus d'informations sur la version actuelle et l'historique de versions de `awslogs-agent-setup`, consultez CHANGELOG.txt.

Le programme d'installation de l'agent CloudWatch Logs nécessite certaines informations lors de l'installation. Avant de commencer, vous devez savoir quel fichier journal vous souhaitez surveiller et son format d'horodatage. Vous devez également préparer les informations suivantes.

Élément	Description
AWS ID de clé d'accès	Appuyez sur Entrée si vous utilisez un rôle IAM. Dans le cas contraire, entrez l'identifiant de votre clé d' AWS accès.
AWS clé d'accès secrète	Appuyez sur Entrée si vous utilisez un rôle IAM. Dans le cas contraire, entrez votre clé d'accès AWS secrète.
Nom de la région par défaut	Appuyez sur Entrée. La région par défaut est us-east-2. Vous pouvez la définir sur us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, ou sa-east-1.
Format de sortie par défaut	Laissez ce champ vide et appuyez sur Entrée.
Chemin d'accès du fichier journal à charger	Emplacement du fichier qui contient les données des journaux à envoyer. Le programme d'installation propose un chemin d'accès.
Nom du groupe de journaux de destination	Nom de votre groupe de journaux. Le programme d'installation propose un nom de groupe de journaux.
Nom du flux de journaux de destination	Par défaut, il s'agit du nom de l'hôte. Le programme d'installation propose un nom d'hôte.

Élément	Description
Format d'horodatage	Spécifiez le format de l'horodatage du fichier journal spécifié. Choisissez l'option Custom pour spécifier votre propre format.
Position initiale	Indique comment les données sont chargées. Choisissez start_of_file pour charger toutes les informations du fichier de données. Choisissez end_of_file pour charger uniquement les données nouvellement ajoutées.

Une fois ces étapes réalisées, le programme d'installation vous demande si vous souhaitez configurer un autre fichier journal. Vous pouvez exécuter le processus autant de fois que vous le souhaitez pour chaque fichier journal. Si vous n'avez plus de fichiers journaux à surveiller, choisissez N lorsque vous êtes invité à configurer un autre journal par le programme d'installation. Pour plus d'informations sur les paramètres du fichier de configuration de l'agent, consultez CloudWatch Référence de l'agent de journalisation.

> **Note**
>
> La configuration de plusieurs sources de journal pour envoyer des données dans un flux de journal unique n'est pas prise en charge.

3. Le nouveau groupe de journaux et le nouveau flux de journal doivent être visibles dans la console CloudWatch une fois que l'agent s'exécute depuis quelques instants.

   Pour de plus amples informations, veuillez consulter Afficher les données du journal envoyées à CloudWatch Logs.

Démarrage rapide : installer et configurer l'agent CloudWatch Logs sur une instance EC2 Linux au lancement

> **Tip**
>
> L'ancien agent CloudWatch Logs décrit dans cette section est sur le point de devenir obsolète. Nous vous recommandons vivement d'utiliser plutôt le nouvel CloudWatch agent unifié capable de collecter à la fois des journaux et des métriques. En outre, l'ancien agent

CloudWatch Logs nécessite Python 3.3 ou une version antérieure, et ces versions ne sont pas installées par défaut sur EC2 les nouvelles instances. Pour plus d'informations sur l' CloudWatch agent unifié, consultez la section Installation de l' CloudWatch agent.
Le reste de cette section explique l'utilisation de l'ancien agent CloudWatch Logs.

Installation de l'ancien agent CloudWatch Logs sur une instance EC2 Linux au lancement

Vous pouvez utiliser les données EC2 utilisateur d'Amazon, une fonctionnalité d'Amazon EC2 qui permet de transmettre des informations paramétriques à l'instance lors du lancement, pour installer et configurer l'agent CloudWatch Logs sur cette instance. Pour transmettre les informations d'installation et de configuration de l'agent CloudWatch Logs à Amazon EC2, vous pouvez fournir le fichier de configuration dans un emplacement réseau tel qu'un compartiment Amazon S3.

La configuration de plusieurs sources de journal pour envoyer des données dans un flux de journal unique n'est pas prise en charge.

Prérequis

Créez un fichier de configuration de l'agent qui décrit tous les groupes de journaux et les flux de journaux. Il s'agit d'un fichier de texte qui décrit les fichiers journaux à surveiller, ainsi que les groupes de journaux et les flux de journaux vers lesquels les charger. L'agent utilise ce fichier de configuration et commence à surveiller et charger tous les fichiers journaux qui y sont décrits. Pour plus d'informations sur les paramètres du fichier de configuration de l'agent, consultez CloudWatch Référence de l'agent de journalisation.

Voici un exemple de fichier de configuration d'agent pour Amazon Linux 2

```
[general]
state_file = /var/lib/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Voici un exemple de fichier de configuration d'agent pour Ubuntu

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Pour configurer votre rôle IAM

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, choisissez Policies, puis Create Policy.
3. Sur la page Créer une stratégie, dans Créer votre propre stratégie, choisissez Sélectionner. Pour plus d'informations sur la création de politiques personnalisées, consultez la section Politiques IAM pour Amazon EC2 dans le guide de l' EC2 utilisateur Amazon.
4. Sur la page Review Policy, tapez un nom pour la stratégie dans Policy Name.
5. Pour Policy Document, collez la stratégie suivante :

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

6. Choisissez Create Policy (Créer une politique).
7. Dans le volet de navigation, choisissez Roles, puis Create New Role.
8. Dans la page Set Role Name, saisissez le nom du rôle et sélectionnez Next Step.
9. Sur la page Sélectionner le type de rôle, choisissez Sélectionner à côté d'Amazon EC2.
10. Sur la page Attacher la stratégie, dans l'en-tête de la table, choisissez Type de stratégie, Géré par le client.
11. Sélectionnez la politique IAM que vous avez créée, puis choisissez Next Step (Étape suivante).
12. Choisissez Create Role (Créer un rôle).

    Pour plus d'informations sur les utilisateurs et les politiques, consultez Utilisateurs et groupes IAM et Gestion des politiques IAM dans le Guide de l'utilisateur IAM.

Pour lancer une nouvelle instance et activer CloudWatch Logs

1. Ouvrez la EC2 console Amazon à l'adresse https://console.aws.amazon.com/ec2/.
2. Choisissez Launch Instances (Lancer les instances).

   Pour plus d'informations, consultez le guide de l' EC2 utilisateur Amazon sur le lancement d'une instance.
3. Sur la page Step 1: Choose an Amazon Machine Image (AMI), sélectionnez le type d'instance Linux que vous souhaitez démarrer, puis sur la page Step 2: Choose an Instance Type, choisissez Next: Configure Instance Details.

   Vérifiez que la commande cloud-init est comprise dans Amazon Machine Image (AMI). Amazon Linux AMIs, ainsi que AMIs pour Ubuntu et RHEL, incluent déjà cloud-init, mais CentOS et d'autres AMIs sites ne le sont peut-être pas. AWS Marketplace
4. Sur la page Étape 3 : Configurer les détails de l'instance, dans IAM role (Rôle IAM), sélectionnez le rôle IAM que vous avez créé.

5. Sous Détails avancés, dans Données utilisateur, collez le script ci-dessous. Mettez ensuite à jour ce script en changeant la valeur de l'option -c par l'emplacement du fichier de configuration de votre agent :

```
#!/bin/bash
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py -O
chmod +x ./awslogs-agent-setup.py
./awslogs-agent-setup.py -n -r us-east-1 -c s3://amzn-s3-demo-bucket/my-config-file
```

6. Apportez toutes les modifications nécessaires à l'instance, vérifiez vos paramètres de lancement et choisissez Launch.
7. Vous devriez voir le groupe de journaux et le flux de journaux nouvellement créés dans la CloudWatch console après quelques instants d'exécution de l'agent.

   Pour de plus amples informations, veuillez consulter Afficher les données du journal envoyées à CloudWatch Logs.

Démarrage rapide : autorisez vos EC2 instances Amazon exécutant Windows Server 2016 à envoyer des journaux à Logs à CloudWatch l'aide de l'agent CloudWatch Logs

> **Tip**
>
> CloudWatch inclut un nouvel agent unifié capable de collecter à la fois des journaux et des métriques à partir d' EC2 instances et de serveurs sur site. Nous vous recommandons d'utiliser le nouvel agent CloudWatch unifié. Pour de plus amples informations, veuillez consulter Commencer à utiliser CloudWatch Logs.
> Le reste de cette section explique l'utilisation de l'ancien agent CloudWatch Logs.

Permettez à vos EC2 instances Amazon exécutant Windows Server 2016 d'envoyer des journaux à Logs à CloudWatch l'aide de l'ancien agent CloudWatch Logs

Vous pouvez utiliser plusieurs méthodes pour permettre aux instances exécutant Windows Server 2016 d'envoyer des CloudWatch journaux à Logs. Les étapes de cette section utilisent la fonctionnalité Run Command de Systems Manager. Pour plus d'informations sur les autres méthodes

possibles, consultez la section Envoi de journaux, d'événements et de compteurs de performance à Amazon CloudWatch.

Étapes

- Téléchargement de l'exemple de fichier de configuration
- Configurez le fichier JSON pour CloudWatch
- Création d'un rôle IAM pour Systems Manager
- Vérification des prérequis de Systems Manager
- Vérification de l'accès Internet
- Activer CloudWatch les journaux à l'aide de la commande Run de Systems Manager

Téléchargement de l'exemple de fichier de configuration

Téléchargez le fichier d'exemple suivant sur votre ordinateur : `AWS.EC2.Windows.CloudWatch.json`.

Configurez le fichier JSON pour CloudWatch

Vous déterminez les journaux à envoyer CloudWatch en spécifiant vos choix dans un fichier de configuration. Le processus de création de ce fichier et la spécification des options peuvent prendre 30 minutes ou plus. Après avoir effectué cette tâche une fois, vous pouvez réutiliser le fichier de configuration sur toutes vos instances.

Étapes

- Étape 1 : activer CloudWatch les journaux
- Étape 2 : Configuration des paramètres pour CloudWatch
- Étape 3 : Configurer les données à envoyer
- Étape 4 : Configurer le contrôle de flux
- Étape 5 : Enregistrer le contenu JSON

Étape 1 : activer CloudWatch les journaux

Dans la partie supérieure du fichier JSON, remplacez « false » par « true » pour `IsEnabled` :

```
"IsEnabled": true,
```

Étape 2 : Configuration des paramètres pour CloudWatch

Indiquez les informations d'identification, la région, le nom du groupe de journaux et l'espace de noms du flux de journaux. Cela permet à l'instance d'envoyer des données de journal à CloudWatch Logs. Pour envoyer les mêmes données de journal à différents emplacements, vous pouvez ajouter des sections supplémentaires avec un identifiant unique IDs (par exemple, CloudWatchLogs « 2 » CloudWatchLogs et 3) et une région différente pour chaque identifiant.

Pour configurer les paramètres d'envoi des données de journal à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `CloudWatchLogs`.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Laissez les champs `AccessKey` et `SecretKey` vides. Vous configurez les informations d'identification à l'aide d'un rôle IAM.
3. Pour `Region`, saisissez la région vers laquelle envoyer les données des journaux (par exemple, `us-east-2`).
4. Pour `LogGroup`, saisissez le nom de votre groupe de journaux. Ce nom s'affiche sur l'écran Groupes de journaux dans la console CloudWatch.
5. Pour `LogStream`, saisissez le flux de journal de destination. Ce nom apparaît sur l'écran Log Groups > Streams de la CloudWatch console.

   Si vous utilisez `{instance_id}`, la valeur par défaut, le nom de flux de journal est l'ID de l'instance.

   Si vous spécifiez un nom de flux de journal qui n'existe pas encore, CloudWatch Logs le crée automatiquement pour vous. Vous pouvez définir un nom de flux de journal à l'aide d'une chaîne

littérale, des variables prédéfinies `{instance_id}`, `{hostname}` et `{ip_address}`, ou d'une combinaison de celles-ci.

Étape 3 : Configurer les données à envoyer

Vous pouvez envoyer des données du journal des événements, des données de suivi des événements pour Windows (ETW) et d'autres données du journal à CloudWatch Logs.

Pour envoyer les données du journal des événements des applications Windows à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :

   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer les données du journal de sécurité à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `SecurityEventLog`.

```
{
```

```
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Pour `Levels`, saisissez **7** pour charger tous les messages.

Pour envoyer les données du journal des événements du système à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :

   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer d'autres types de données du journal des événements à CloudWatch Logs

1. Dans le fichier JSON, ajoutez une nouvelle section. Chaque section doit avoir un `Id` unique.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Pour `Id`, tapez un nom pour le journal à charger (par exemple, **`WindowsBackup`**).
3. Pour `LogName`, saisissez le nom du journal à charger. Vous pouvez trouver le nom du journal comme suit.
   a. Ouvrez l'Observateur d'événements
   b. Dans le panneau de navigation, cliquez sur Journaux d'applications et de services.
   c. Naviguez jusqu'au journal, puis choisissez Actions, Propriétés.
4. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :
   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer des données de suivi des événements pour Windows à CloudWatch Logs

ETW (Event Tracing for Windows, suivi d'événements pour Windows) fournit un mécanisme de journalisation efficace et détaillé dans lequel les applications peuvent consigner des journaux.

Chaque ETW est contrôlé par un gestionnaire de session qui peut démarrer et arrêter la session de journalisation. Chaque session a un fournisseur et un ou plusieurs utilisateurs.

1. Dans le fichier JSON, localisez la section `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Pour `LogName`, saisissez le nom du journal à charger.
3. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :

   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer des journaux personnalisés (n'importe quel fichier journal texte) à Logs CloudWatch

1. Dans le fichier JSON, localisez la section `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
```

```
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Pour `LogDirectoryPath`, saisissez le chemin d'accès où les journaux sont stockés sur votre instance.
3. Pour `TimestampFormat`, saisissez le format d'horodatage que vous voulez utiliser. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique Chaînes de format de date et d'heure personnalisées sur MSDN.

> **Important**
>
> Votre fichier journal source doit avoir l'horodatage au début de chaque ligne de journal et un espace après l'horodatage.

4. Pour `Encoding`, tapez l'encodage de fichier à utiliser (par exemple, UTF-8). Pour obtenir une liste des valeurs prises en charge, consultez la rubrique Classe d'encodage sur MSDN.

> **Note**
>
> Utilisez le nom d'encodage, pas le nom complet.

5. (Facultatif) Pour `Filter`, tapez le préfixe des noms de journaux. Laissez ce paramètre vide de façon à surveiller tous les fichiers. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique FileSystemWatcherFilter Propriétés sur MSDN.
6. (Facultatif) Pour `CultureName`, saisissez les paramètres régionaux où l'horodatage est consigné. Si `CultureName` est vide, il prend par défaut les mêmes paramètres régionaux que ceux actuellement utilisés par votre instance de Windows. Pour plus d'informations, consultez la colonne `Language tag` du tableau de la rubrique Product Behavior sur MSDN.

> **Note**
>
> Les valeurs `div`, `div-MV`, hu et `hu-HU` ne sont pas prises en charge.

7. (Facultatif) Pour `TimeZoneKind`, tapez `Local` ou `UTC`. Vous pouvez définir ce paramètre pour fournir des informations de fuseau horaire si aucune n'est comprise dans l'horodatage de vos journaux. Si ce paramètre est laissé vide et si votre horodatage n'inclut aucune information de fuseau horaire, CloudWatch Logs utilise par défaut le fuseau horaire local. Ce paramètre est ignoré si votre horodatage contient déjà des informations de fuseau horaire.
8. (Facultatif) Pour `LineCount`, saisissez le nombre de lignes dans l'en-tête pour identifier le fichier journal. Par exemple, les fichiers journaux IIS ont des en-têtes presque identiques. Vous pouvez entrer **5**, qui lit les trois premières lignes de l'en-tête du fichier journal pour l'identifier. Dans les fichiers journaux IIS, la troisième ligne contient la date et l'horodatage, mais l'horodatage n'est pas toujours différent d'un fichier journal à l'autre. Pour cette raison, nous recommandons d'inclure au moins une ligne de données de journaux réelles pour affecter une empreinte unique au fichier journal.

Pour envoyer les données du journal IIS à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Pour `LogDirectoryPath`, saisissez le dossier où les journaux IIS sont stockés pour un site individuel (par exemple, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> Seul le format de journal W3C est pris en charge. Les formats IIS, NCSA et Personnalisé ne sont pas pris en charge.

3. Pour `TimestampFormat`, saisissez le format d'horodatage que vous voulez utiliser. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique Chaînes de format de date et d'heure personnalisées sur MSDN.
4. Pour `Encoding`, tapez l'encodage de fichier à utiliser (par exemple, UTF-8). Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique Classe d'encodage sur MSDN.

> **Note**
>
> Utilisez le nom d'encodage, pas le nom complet.

5. (Facultatif) Pour `Filter`, tapez le préfixe des noms de journaux. Laissez ce paramètre vide de façon à surveiller tous les fichiers. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique FileSystemWatcherFilter Propriétés sur MSDN.
6. (Facultatif) Pour `CultureName`, saisissez les paramètres régionaux où l'horodatage est consigné. Si `CultureName` est vide, il prend par défaut les mêmes paramètres régionaux que ceux actuellement utilisés par votre instance de Windows. Pour plus d'informations sur les valeurs prises en charge, consultez la colonne `Language tag` du tableau de la rubrique Product Behavior sur MSDN.

> **Note**
>
> Les valeurs `div`, `div-MV`, hu et `hu-HU` ne sont pas prises en charge.

7. (Facultatif) Pour `TimeZoneKind`, saisissez `Local` ou `UTC`. Vous pouvez définir ce paramètre pour fournir des informations de fuseau horaire si aucune n'est comprise dans l'horodatage de vos journaux. Si ce paramètre est laissé vide et si votre horodatage n'inclut aucune information de fuseau horaire, CloudWatch Logs utilise par défaut le fuseau horaire local. Ce paramètre est ignoré si votre horodatage contient déjà des informations de fuseau horaire.
8. (Facultatif) Pour `LineCount`, saisissez le nombre de lignes dans l'en-tête pour identifier le fichier journal. Par exemple, les fichiers journaux IIS ont des en-têtes presque identiques. Vous pouvez entrer **5**, qui lirait les cinq premières lignes de l'en-tête du fichier journal pour l'identifier. Dans les fichiers journaux IIS, la troisième ligne contient la date et l'horodatage, mais l'horodatage n'est pas toujours différent d'un fichier journal à l'autre. Pour cette raison, nous recommandons d'inclure au moins une seule ligne de données de journaux réelles pour affecter une empreinte unique au fichier journal.

Étape 4 : Configurer le contrôle de flux

Chaque type de données doit avoir une destination correspondante dans la section `Flows`. Par exemple, pour envoyer le journal personnalisé, le journal ETW et le journal système à CloudWatch Logs, ajoutez-les `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` à la `Flows` section.

> **Warning**
>
> L'ajout d'une étape qui n'est pas valide bloque le flux. Par exemple, si vous ajoutez une étape de métrique de disque, mais que votre instance ne comporte pas de disque, toutes les étapes du flux sont bloquées.

Vous pouvez envoyer le même fichier journal plusieurs destinations. Par exemple, pour envoyer le journal d'application à deux destinations différentes que vous avez définies dans la section `CloudWatchLogs`, ajoutez `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` à la section `Flows`.

Pour configurer le contrôle de flux

1. Dans le fichier `AWS.EC2.Windows.CloudWatch.json`, recherchez la section `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. Pour `Flows`, ajoutez chaque type de données à charger (par exemple, `ApplicationEventLog`) et sa destination (par exemple, `CloudWatchLogs`).

Étape 5 : Enregistrer le contenu JSON

Vous avez maintenant terminé de modifier le fichier JSON. Enregistrez et collez le contenu du fichier dans un éditeur de texte dans une autre fenêtre. Vous aurez besoin du contenu du fichier lors d'une étape ultérieure de cette procédure.

Création d'un rôle IAM pour Systems Manager

Un rôle IAM pour les informations d'identification de l'instance est obligatoire lorsque vous utilisez la fonctionnalité Run Command de Systems Manager. Ce rôle permet à Systems Manager d'exécuter des actions sur l'instance. Pour plus d'informations, consultez Configuration des rôles de sécurité pour Systems Manager dans le Guide de l'utilisateur AWS Systems Manager . Pour plus d'informations sur la façon d'associer un rôle IAM à une instance existante, consultez Attacher un rôle IAM à une instance dans le guide de EC2 l'utilisateur Amazon.

Vérification des prérequis de Systems Manager

Avant d'utiliser Systems Manager Run Command pour configurer l'intégration avec CloudWatch Logs, vérifiez que vos instances répondent aux exigences minimales. Pour plus d'informations, consultez Systems Manager Prerequisites (Prérequis de Systems Manager) dans le Guide de l'utilisateur AWS Systems Manager .

Vérification de l'accès Internet

Vos instances Amazon EC2 Windows Server et vos instances gérées doivent disposer d'un accès Internet sortant pour pouvoir envoyer des données de journal et d'événement à CloudWatch. Pour plus d'informations sur la configuration de l'accès à Internet, consultez Internet Gateways (Passerelles Internet) dans le Guide de l'utilisateur Amazon VPC.

Activer CloudWatch les journaux à l'aide de la commande Run de Systems Manager

La fonctionnalité Exécuter la commande vous permet de gérer la configuration de vos instances à la demande. Spécifiez un document Systems Manager, fournissez des paramètres et exécutez la commande sur une ou plusieurs instances. Le SSM Agent sur l'instance traite la commande et configure l'instance comme indiqué.

Pour configurer l'intégration aux CloudWatch journaux à l'aide de la commande Run

1. Ouvrez la EC2 console Amazon à l'adresse https://console.aws.amazon.com/ec2/.
2. Ouvrez la console SSM à https://console.aws.amazon.com/systems-manager/l'adresse.
3. Dans le panneau de navigation, choisissez Fonctionnalité Exécuter la commande.
4. Choisissez Run a Command.
5. Pour le document de commande, choisissez AWS- ConfigureCloudWatch.
6. Pour les instances Target, choisissez les instances à intégrer à CloudWatch Logs. Si vous ne voyez pas une instance dans cette liste, c'est qu'elle n'est peut-être pas configurée pour

l'exécution de la commande. Pour plus d'informations, consultez les prérequis de Systems Manager dans le guide de l' EC2 utilisateur Amazon.

7. Pour Statut, sélectionnez Enabled.
8. Pour Propriétés, copiez et collez le contenu JSON que vous avez créé lors des tâches précédentes.
9. Complétez les champs facultatifs restants et choisissez Run.

Utilisez la procédure suivante pour afficher les résultats de l'exécution des commandes dans la EC2 console Amazon.

Pour afficher la sortie de commande dans la console

1. Sélectionnez une commande.
2. Choisissez l'onglet Output.
3. Choisissez View Output. La page de sortie de la commande illustre les résultats de l'exécution de votre commande.

Démarrage rapide : autorisez vos EC2 instances Amazon exécutant Windows Server 2012 et Windows Server 2008 à envoyer des CloudWatch journaux vers Logs

> **Tip**
>
> CloudWatch inclut un nouvel agent unifié capable de collecter à la fois des journaux et des métriques à partir d' EC2 instances et de serveurs sur site. Nous vous recommandons d'utiliser le nouvel agent CloudWatch unifié. Pour de plus amples informations, veuillez consulter Commencer à utiliser CloudWatch Logs.
> Le reste de cette section explique l'utilisation de l'ancien agent CloudWatch Logs.

Permettez à vos EC2 instances Amazon exécutant Windows Server 2012 et Windows Server 2008 d'envoyer des CloudWatch journaux vers Logs

Procédez comme suit pour permettre à vos instances exécutant Windows Server 2012 et Windows Server 2008 d'envoyer des CloudWatch journaux vers Logs.

Téléchargement de l'exemple de fichier de configuration

Téléchargez le fichier d'exemple JSON suivant sur votre ordinateur : `AWS.EC2.Windows.CloudWatch.json`. Vous y apporterez des modifications lors des étapes suivantes.

Configurez le fichier JSON pour CloudWatch

Vous déterminez les journaux à envoyer CloudWatch en spécifiant vos choix dans le fichier de configuration JSON. Le processus de création de ce fichier et la spécification des options peuvent prendre 30 minutes ou plus. Après avoir effectué cette tâche une fois, vous pouvez réutiliser le fichier de configuration sur toutes vos instances.

Étapes

- Étape 1 : activer CloudWatch les journaux
- Étape 2 : Configuration des paramètres pour CloudWatch
- Étape 3 : Configurer les données à envoyer
- Étape 4 : Configurer le contrôle de flux

Étape 1 : activer CloudWatch les journaux

Dans la partie supérieure du fichier JSON, remplacez « false » par « true » pour `IsEnabled` :

```
"IsEnabled": true,
```

Étape 2 : Configuration des paramètres pour CloudWatch

Indiquez les informations d'identification, la région, le nom du groupe de journaux et l'espace de noms du flux de journaux. Cela permet à l'instance d'envoyer des données de journal à CloudWatch Logs. Pour envoyer les mêmes données de journal à différents emplacements, vous pouvez ajouter des sections supplémentaires avec un identifiant unique IDs (par exemple, CloudWatchLogs « 2 » CloudWatchLogs et 3) et une région différente pour chaque identifiant.

Pour configurer les paramètres d'envoi des données de journal à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `CloudWatchLogs`.

   ```
   {
   ```

```
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Laissez les champs `AccessKey` et `SecretKey` vides. Vous configurez les informations d'identification à l'aide d'un rôle IAM.
3. Pour `Region`, saisissez la région vers laquelle envoyer les données des journaux (par exemple, `us-east-2`).
4. Pour `LogGroup`, saisissez le nom de votre groupe de journaux. Ce nom s'affiche sur l'écran Groupes de journaux dans la console CloudWatch.
5. Pour `LogStream`, saisissez le flux de journal de destination. Ce nom apparaît sur l'écran Log Groups > Streams de la CloudWatch console.

   Si vous utilisez `{instance_id}`, la valeur par défaut, le nom de flux de journal est l'ID de l'instance.

   Si vous spécifiez un nom de flux de journal qui n'existe pas encore, CloudWatch Logs le crée automatiquement pour vous. Vous pouvez définir un nom de flux de journal à l'aide d'une chaîne littérale, des variables prédéfinies `{instance_id}`, `{hostname}` et `{ip_address}`, ou d'une combinaison de celles-ci.

Étape 3 : Configurer les données à envoyer

Vous pouvez envoyer des données du journal des événements, des données de suivi des événements pour Windows (ETW) et d'autres données du journal à CloudWatch Logs.

Pour envoyer les données du journal des événements des applications Windows à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
```

```
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :

   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer les données du journal de sécurité à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Pour `Levels`, saisissez **7** pour charger tous les messages.

Pour envoyer les données du journal des événements du système à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :

   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer d'autres types de données du journal des événements à CloudWatch Logs

1. Dans le fichier JSON, ajoutez une nouvelle section. Chaque section doit avoir un `Id` unique.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Pour `Id`, tapez un nom pour le journal à charger (par exemple, **`WindowsBackup`**).

3. Pour `LogName`, saisissez le nom du journal à charger. Vous pouvez trouver le nom du journal comme suit.

   a. Ouvrez l'Observateur d'événements

   b. Dans le panneau de navigation, cliquez sur Journaux d'applications et de services.

   c. Naviguez jusqu'au journal, puis choisissez Actions, Propriétés.

4. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :

   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer des données de suivi des événements pour Windows à CloudWatch Logs

ETW (Event Tracing for Windows, suivi d'événements pour Windows) fournit un mécanisme de journalisation efficace et détaillé dans lequel les applications peuvent consigner des journaux. Chaque ETW est contrôlé par un gestionnaire de session qui peut démarrer et arrêter la session de journalisation. Chaque session a un fournisseur et un ou plusieurs utilisateurs.

1. Dans le fichier JSON, localisez la section `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Pour `LogName`, saisissez le nom du journal à charger.

3. Pour `Levels`, spécifiez le type de messages à charger. Vous pouvez spécifier l'une des valeurs suivantes :

   - **1** - Chargez uniquement les messages d'erreur.
   - **2** - Chargez uniquement les messages d'avertissement.
   - **4** - Chargez uniquement les messages d'information.

   Vous pouvez associer des valeurs pour inclure plusieurs types de message. Par exemple, la valeur **3** charge les messages d'erreur (**1**) et les messages d'avertissement (**2**). La valeur **7** charge les messages d'erreur (**1**), les messages d'avertissement (**2**) et les messages d'information (**4**).

Pour envoyer des journaux personnalisés (n'importe quel fichier journal texte) à Logs CloudWatch

1. Dans le fichier JSON, localisez la section `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Pour `LogDirectoryPath`, saisissez le chemin d'accès où les journaux sont stockés sur votre instance.
3. Pour `TimestampFormat`, saisissez le format d'horodatage que vous voulez utiliser. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique Chaînes de format de date et d'heure personnalisées sur MSDN.

> **Important**
>
> Votre fichier journal source doit avoir l'horodatage au début de chaque ligne de journal et un espace après l'horodatage.

4. Pour `Encoding`, tapez l'encodage de fichier à utiliser (par exemple, UTF-8). Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique Classe d'encodage sur MSDN.

> **Note**
>
> Utilisez le nom d'encodage, pas le nom complet.

5. (Facultatif) Pour `Filter`, tapez le préfixe des noms de journaux. Laissez ce paramètre vide de façon à surveiller tous les fichiers. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique FileSystemWatcherFilter Propriétés sur MSDN.
6. (Facultatif) Pour `CultureName`, saisissez les paramètres régionaux où l'horodatage est consigné. Si `CultureName` est vide, il prend par défaut les mêmes paramètres régionaux que ceux actuellement utilisés par votre instance de Windows. Pour plus d'informations sur les valeurs prises en charge, consultez la colonne `Language tag` du tableau de la rubrique Product Behavior sur MSDN.

> **Note**
>
> Les valeurs `div`, `div-MV`, hu et `hu-HU` ne sont pas prises en charge.

7. (Facultatif) Pour `TimeZoneKind`, tapez `Local` ou `UTC`. Vous pouvez définir ce paramètre pour fournir des informations de fuseau horaire si aucune n'est comprise dans l'horodatage de vos journaux. Si ce paramètre est laissé vide et si votre horodatage n'inclut aucune information de fuseau horaire, CloudWatch Logs utilise par défaut le fuseau horaire local. Ce paramètre est ignoré si votre horodatage contient déjà des informations de fuseau horaire.
8. (Facultatif) Pour `LineCount`, saisissez le nombre de lignes dans l'en-tête pour identifier le fichier journal. Par exemple, les fichiers journaux IIS ont des en-têtes presque identiques. Vous pouvez entrer **5**, qui lit les trois premières lignes de l'en-tête du fichier journal pour l'identifier. Dans les fichiers journaux IIS, la troisième ligne contient la date et l'horodatage, mais l'horodatage n'est pas toujours différent d'un fichier journal à l'autre. Pour cette raison, nous recommandons

d'inclure au moins une ligne de données de journaux réelles pour affecter une empreinte unique au fichier journal.

Pour envoyer les données du journal IIS à CloudWatch Logs

1. Dans le fichier JSON, localisez la section `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Pour `LogDirectoryPath`, saisissez le dossier où les journaux IIS sont stockés pour un site individuel (par exemple, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> Seul le format de journal W3C est pris en charge. Les formats IIS, NCSA et Personnalisé ne sont pas pris en charge.

3. Pour `TimestampFormat`, saisissez le format d'horodatage que vous voulez utiliser. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique Chaînes de format de date et d'heure personnalisées sur MSDN.
4. Pour `Encoding`, tapez l'encodage de fichier à utiliser (par exemple, UTF-8). Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique Classe d'encodage sur MSDN.

> **Note**
>
> Utilisez le nom d'encodage, pas le nom complet.

5. (Facultatif) Pour `Filter`, tapez le préfixe des noms de journaux. Laissez ce paramètre vide de façon à surveiller tous les fichiers. Pour plus d'informations sur les valeurs prises en charge, consultez la rubrique FileSystemWatcherFilter Propriétés sur MSDN.
6. (Facultatif) Pour `CultureName`, saisissez les paramètres régionaux où l'horodatage est consigné. Si `CultureName` est vide, il prend par défaut les mêmes paramètres régionaux que ceux actuellement utilisés par votre instance de Windows. Pour plus d'informations sur les valeurs prises en charge, consultez la colonne `Language tag` du tableau de la rubrique Product Behavior sur MSDN.

> **Note**
>
> Les valeurs `div`, `div-MV`, hu et `hu-HU` ne sont pas prises en charge.

7. (Facultatif) Pour `TimeZoneKind`, saisissez `Local` ou `UTC`. Vous pouvez définir ce paramètre pour fournir des informations de fuseau horaire si aucune n'est comprise dans l'horodatage de vos journaux. Si ce paramètre est laissé vide et si votre horodatage n'inclut aucune information de fuseau horaire, CloudWatch Logs utilise par défaut le fuseau horaire local. Ce paramètre est ignoré si votre horodatage contient déjà des informations de fuseau horaire.
8. (Facultatif) Pour `LineCount`, saisissez le nombre de lignes dans l'en-tête pour identifier le fichier journal. Par exemple, les fichiers journaux IIS ont des en-têtes presque identiques. Vous pouvez entrer **5**, qui lirait les cinq premières lignes de l'en-tête du fichier journal pour l'identifier. Dans les fichiers journaux IIS, la troisième ligne contient la date et l'horodatage, mais l'horodatage n'est pas toujours différent d'un fichier journal à l'autre. Pour cette raison, nous recommandons d'inclure au moins une seule ligne de données de journaux réelles pour affecter une empreinte unique au fichier journal.

Étape 4 : Configurer le contrôle de flux

Chaque type de données doit avoir une destination correspondante dans la section `Flows`. Par exemple, pour envoyer le journal personnalisé, le journal ETW et le journal système à CloudWatch Logs, ajoutez-les `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` à la `Flows` section.

> **Warning**
>
> L'ajout d'une étape qui n'est pas valide bloque le flux. Par exemple, si vous ajoutez une étape de métrique de disque, mais que votre instance ne comporte pas de disque, toutes les étapes du flux sont bloquées.

Vous pouvez envoyer le même fichier journal plusieurs destinations. Par exemple, pour envoyer le journal d'application à deux destinations différentes que vous avez définies dans la section `CloudWatchLogs`, ajoutez `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` à la section `Flows`.

Pour configurer le contrôle de flux

1. Dans le fichier `AWS.EC2.Windows.CloudWatch.json`, recherchez la section `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. Pour `Flows`, ajoutez chaque type de données à charger (par exemple, `ApplicationEventLog`) et sa destination (par exemple, `CloudWatchLogs`).

Vous avez maintenant terminé de modifier le fichier JSON. Vous l'utiliserez dans une étape ultérieure.

Démarrage de l'agent

Pour permettre à une EC2 instance Amazon exécutant Windows Server 2012 ou Windows Server 2008 d'envoyer des CloudWatch journaux à Logs, utilisez le service EC2 Config (`EC2Config.exe`). Votre instance doit disposer de EC2 Config 4.0 ou d'une version ultérieure, et vous pouvez utiliser cette procédure.

Pour configurer à CloudWatch l'aide de EC2 Config 4.x

1. Vérifiez l'encodage du fichier `AWS.EC2.Windows.CloudWatch.json` que vous avez déjà modifié dans cette procédure. Seul l'encodage UTF-8 sans BOM est pris en charge. Enregistrez ensuite le fichier dans le dossier suivant de votre instance Windows Server 2008 - 2012 R2 : `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.
2. Démarrez ou redémarrez l'agent SSM (`AmazonSSMAgent.exe`) à l'aide du panneau de configuration Windows Services ou à l'aide de la PowerShell commande suivante :

```
PS C:\> Restart-Service AmazonSSMAgent
```

Après le redémarrage de l'agent SSM, il détecte le fichier de configuration et configure l'instance pour l'intégration. CloudWatch Si vous modifiez les paramètres dans le fichier de configuration local, vous devez redémarrer le SSM Agent pour appliquer les modifications. Pour désactiver CloudWatch l'intégration sur l'instance, modifiez `IsEnabled false` et enregistrez vos modifications dans le fichier de configuration.

Démarrage rapide : installez l'agent CloudWatch Logs à l'aide de AWS OpsWorks et Chef

Vous pouvez installer l'agent CloudWatch Logs et créer des flux de journaux à l'aide de AWS OpsWorks and Chef, un outil tiers d'automatisation des systèmes et de l'infrastructure cloud. Chef utilise des « recettes », que vous écrivez pour installer et configurer des logiciels sur votre ordinateur, et des « livres de recettes » pour exécuter les tâches de configuration et de distribution de stratégie. Pour plus d'informations, consultez Chef.

Les exemples de recettes Chef ci-dessous montrent comment surveiller un fichier journal sur chaque EC2 instance. Les recettes utilisent le nom de la pile en tant que groupe de journaux et le nom d'hôte de l'instance en tant que nom de flux de journaux. Pour surveiller plusieurs fichiers journaux, vous devez étendre les recettes afin de créer plusieurs groupes de journaux et flux de journaux.

Étape 1 : Créer des recettes personnalisées

Créez un référentiel pour stocker vos recettes. AWS OpsWorks supporte Git et Subversion, ou vous pouvez stocker une archive dans Amazon S3. La structure du référentiel de votre livre de recettes est décrite dans Référentiels de livre de recette dans le Guide de l'utilisateur AWS OpsWorks .
Les exemples ci-dessous supposent que le nom du livre de recettes est `logs`. La recette install.rb

installe l'agent Logs. CloudWatch Vous pouvez également télécharger l'exemple du livre de recettes (CloudWatchLogs-Cookbooks.zip).

Créez un fichier nommé metadata.rb contenant le code suivant :

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

Créez le fichier de configuration CloudWatch des journaux :

```
#config.rb

template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Téléchargez et installez l'agent CloudWatch Logs :

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end

remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py"
  mode "0755"
end

  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> Dans l'exemple ci-dessus, remplacez *region* par l'un des suivants : us-east-1, us-west-1, us-west-2, ap-southeast-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1 ou sa-east-1.

Si l'installation de l'agent échoue, assurez-vous que le package `python-dev` est installé. Si ce n'est pas le cas, utilisez la commande suivante, puis réessayez d'installer l'agent :

```
sudo apt-get -y install python-dev
```

Cette recette utilise un fichier de modèle cwlogs.cfg.erb que vous pouvez modifier pour spécifier plusieurs attributs, par exemple les fichiers à conserver. Pour obtenir plus d'informations sur ces attributs, consultez CloudWatch Référence de l'agent de journalisation.

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state


## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
#
#log_stream_name = {instance_id}
#
## Format specifier for timestamp parsing.
#
```

```
#datetime_format = %b %d %H:%M:%S
#
#

[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

Le modèle obtient le nom de la pile et le nom de l'hôte en référençant les attributs correspondants dans la configuration de la pile et dans le déploiement JSON. L'attribut qui spécifie le fichier à enregistrer est défini dans le fichier d'attributs .rb par défaut du livre de recettes cwlogs (.rb). logs/attributes/default

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

Étape 2 : créer une OpsWorks pile

1. Ouvrez la OpsWorks console à l'adresse https://console.aws.amazon.com/opsworks/.
2. Sur le OpsWorks tableau de bord, choisissez Ajouter une pile pour créer une OpsWorks pile.
3. Sur l'écran Add stack, choisissez Chef 11 stack.
4. Pour Stack name, saisissez un nom de pile.
5. Pour Use custom Chef Cookbooks, choisissez Yes.
6. Pour Repository type, sélectionnez le type de référentiel que vous utilisez. Si vous utilisez l'exemple ci-dessus, choisissez Http Archive.
7. Pour Repository URL, saisissez le référentiel dans lequel vous avez enregistré le livre de recettes créé lors de l'étape précédente. Si vous utilisez l'exemple ci-dessus, saisissez **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`**.
8. Choisissez Add Stack pour créer la pile.

Étape 3 : Étendre votre rôle IAM

Pour utiliser CloudWatch Logs avec vos OpsWorks instances, vous devez étendre le rôle IAM utilisé par vos instances.

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, choisissez Policies, puis Create Policy.
3. Sur la page Create Policy (Créer une politique), sous Create Your Own Policy (Créer votre propre politique), choisissez Select (Sélectionner). Pour plus d'informations sur la création de politiques personnalisées, consultez la section Politiques IAM pour Amazon EC2 dans le guide de l' EC2 utilisateur Amazon.
4. Sur la page Review Policy, tapez un nom pour la stratégie dans Policy Name.
5. Pour Policy Document, collez la stratégie suivante :

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

6. Choisissez Create Policy (Créer une politique).
7. Dans le volet de navigation, choisissez Rôles, puis dans le volet de contenu, pour Nom du rôle, sélectionnez le nom du rôle d'instance utilisé par votre OpsWorks pile. Vous pouvez identifier le rôle utilisé par votre pile dans les paramètres de la pile (la valeur par défaut est `aws-opsworks-ec2-role`).

   > **Note**
   >
   > Choisissez le nom du rôle, et non pas la case à cocher.

8. Dans l'onglet Permissions, sous Managed Policies, choisissez Attach Policy.
9. Sur la page Attach Policy, dans l'en-tête de la table (à côté de Filter et Search), choisissez Policy Type, Customer Managed Policies.
10. Pour Customer Managed Policies (Politiques gérées par le client), sélectionnez la politique IAM que vous avez créée ci-dessus et cliquez sur Attach Policy (Attacher une politique).

    Pour plus d'informations sur les utilisateurs et les politiques, consultez Utilisateurs et groupes IAM et Gestion des politiques IAM dans le Guide de l'utilisateur IAM.

Étape 4 : Ajouter une couche

1. Ouvrez la OpsWorks console à l'adresse https://console.aws.amazon.com/opsworks/.
2. Choisissez Layers dans le volet de navigation.
3. Dans le volet de contenu, sélectionnez une couche, puis choisissez Add layer.
4. OpsWorksDans l'onglet Type de couche, sélectionnez Personnaliser.
5. Dans les champs Name et Short name, saisissez le nom long et le nom court de la couche, puis choisissez Add layer.
6. Dans l'onglet Recettes, sous Custom Chef Recipes, plusieurs rubriques (Configuration, Configuration, Déploiement, Annulation du déploiement et Arrêt) correspondent aux OpsWorks événements du cycle de vie. OpsWorks déclenche ces événements à ces moments clés du cycle de vie de l'instance, qui exécute les recettes associées.

    > **Note**
    >
    > Si les en-têtes ci-dessus ne sont pas visibles, sous Custom Chef Recipes, choisissez edit.

7. Entrez logs::config, logs::install à côté de Setup, choisissez + pour l'ajouter à la liste, puis cliquez sur Save.

    OpsWorks exécute cette recette sur chacune des nouvelles instances de cette couche, juste après le démarrage de l'instance.

Étape 5 : Ajouter une instance

La couche contrôle uniquement la façon de configurer des instances. Vous devez maintenant ajouter des instances à la couche et les démarrer.

1. Ouvrez la OpsWorks console à l'adresse https://console.aws.amazon.com/opsworks/.
2. Dans le volet de navigation, choisissez Instances, puis sous votre couche, cliquez sur + Instance.
3. Acceptez les paramètres par défaut et choisissez Add Instance pour ajouter l'instance à la couche.
4. Dans de la colonne Actions de la ligne, cliquez sur start pour démarrer l'instance.

   OpsWorks lance une nouvelle EC2 instance et configure CloudWatch Logs. L'état de l'instance devient En ligne lorsqu'elle est prête.

Étape 6 : Afficher vos journaux

Vous devriez voir le groupe de journaux et le flux de journaux nouvellement créés dans la CloudWatch console après quelques instants d'exécution de l'agent.

Pour de plus amples informations, veuillez consulter Afficher les données du journal envoyées à CloudWatch Logs.

Signaler le statut de l'agent CloudWatch Logs

Utilisez la procédure suivante pour signaler l'état de l'agent CloudWatch Logs sur votre EC2 instance.

Pour signaler l'état de l'agent

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

   Pour plus d'informations sur les problèmes de connexion, consultez la section Résolution des problèmes liés à la connexion à votre instance dans le guide de EC2 l'utilisateur Amazon
2. À partir d'une invite de commande, saisissez la commande suivante :

   ```
   sudo service awslogs status
   ```

   Si vous utilisez Amazon Linux 2, saisissez la commande suivante :

```
sudo service awslogsd status
```

3. Vérifiez le fichier/var/log/awslogs.log pour détecter toute erreur, tout avertissement ou tout problème lié à l'agent CloudWatch Logs.

Démarrez l'agent CloudWatch Logs

Si l'agent CloudWatch Logs de votre EC2 instance n'a pas démarré automatiquement après l'installation, ou si vous l'avez arrêté, vous pouvez utiliser la procédure suivante pour démarrer l'agent.

Pour démarrer l'agent

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

   Pour plus d'informations sur les problèmes de connexion, consultez la section Résolution des problèmes de connexion à votre instance dans le guide de EC2 l'utilisateur Amazon.
2. À partir d'une invite de commande, saisissez la commande suivante :

   ```
   sudo service awslogs start
   ```

   Si vous utilisez Amazon Linux 2, saisissez la commande suivante :

   ```
   sudo service awslogsd start
   ```

Arrêter l'agent CloudWatch Logs

Utilisez la procédure suivante pour arrêter l'agent CloudWatch Logs sur votre EC2 instance.

Pour arrêter l'agent

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

   Pour plus d'informations sur les problèmes de connexion, consultez la section Résolution des problèmes de connexion à votre instance dans le guide de EC2 l'utilisateur Amazon.

2. À partir d'une invite de commande, saisissez la commande suivante :

```
sudo service awslogs stop
```

Si vous utilisez Amazon Linux 2, saisissez la commande suivante :

```
sudo service awslogsd stop
```

Démarrage rapide : AWS CloudFormation à utiliser pour commencer à utiliser les CloudWatch journaux

AWS CloudFormation vous permet de décrire et de provisionner vos AWS ressources au format JSON. Les avantages de cette méthode incluent la possibilité de gérer un ensemble de AWS ressources en tant qu'unité unique et de répliquer facilement vos AWS ressources entre les régions.

Lorsque vous provisionnez AWS en utilisant AWS CloudFormation, vous créez des modèles qui décrivent les AWS ressources à utiliser. L'exemple suivant est un extrait de modèle qui crée un groupe de journaux et un filtre de métrique qui compte 404 occurrences et envoie ce nombre dans le groupe de journaux.

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code =
 404, size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
```

```
                "MetricName": "test404Count"
            }
        ]
    }
}
```

Il s'agit d'un exemple basique. Vous pouvez configurer des déploiements de CloudWatch journaux beaucoup plus riches à l'aide AWS CloudFormation de. Pour plus d'informations sur les exemples de modèles, consultez les extraits de modèles Amazon CloudWatch Logs dans le guide de l'AWS CloudFormation utilisateur. Pour plus d'informations sur la mise en route, consultez Getting started with AWS CloudFormation (Mise en route avec) dans le Guide de l'utilisateur AWS CloudFormation .

Utilisation CloudWatch des journaux avec un AWS SDK

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS SDK pour C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK pour Go	AWS SDK pour Go exemples de code
AWS SDK pour Java	AWS SDK pour Java exemples de code
AWS SDK pour JavaScript	AWS SDK pour JavaScript exemples de code
AWS SDK pour Kotlin	AWS SDK pour Kotlin exemples de code
AWS SDK pour .NET	AWS SDK pour .NET exemples de code
AWS SDK pour PHP	AWS SDK pour PHP exemples de code
Outils AWS pour PowerShell	Outils AWS pour PowerShell exemples de code
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour 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

Pour des exemples spécifiques aux CloudWatch journaux, voirExemples de code pour CloudWatch les journaux utilisant AWS SDKs.

Exemple de disponibilité

Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Faire un commentaire en bas de cette page.

Analyse des données des CloudWatch journaux avec Logs Insights

Avec CloudWatch Logs Insights, vous pouvez rechercher et analyser de manière interactive les données de vos CloudWatch journaux dans Amazon Logs. Vous pouvez exécuter des requêtes pour répondre plus rapidement et plus efficacement aux problèmes opérationnels. En cas de problème, vous pouvez utiliser CloudWatch Logs Insights pour identifier les causes potentielles et valider les correctifs déployés.

CloudWatch Logs Insights prend en charge trois langages de requête que vous pouvez utiliser pour vos requêtes :

- Un langage de requête Logs Insights (Logs Insights QL) spécialement conçu avec quelques commandes simples mais puissantes.
- OpenSearch Langage de traitement par tuyaux de service (PPL). OpenSearch PPL vous permet d'analyser vos journaux à l'aide d'un ensemble de commandes délimitées par des canaux (|).

  Avec OpenSearch PPL, vous pouvez récupérer, interroger et analyser des données à l'aide de commandes regroupées, ce qui facilite la compréhension et la composition de requêtes complexes. La syntaxe permet d'enchaîner les commandes pour transformer et traiter les données. Avec PPL, vous pouvez filtrer et agréger des données, et utiliser un ensemble complet de fonctions mathématiques, de chaîne, de date, conditionnelles et autres pour l'analyse.
- OpenSearch Langage de requête structuré de service (SQL). Les requêtes OpenSearch SQL vous permettent d'analyser vos journaux de manière déclarative. Vous pouvez utiliser des commandes telles que SELECT, FROM, WHERE, GROUP BY, HAVING, ainsi que diverses autres commandes et fonctions disponibles dans SQL. Vous pouvez exécuter des tâches JOINs dans différents groupes de journaux, corréler les données entre les journaux à l'aide de sous-requêtes et utiliser le riche ensemble de fonctions JSON, mathématiques, chaînes, conditionnelles et autres fonctions SQL pour effectuer des analyses approfondies sur les journaux.

  Lorsque vous utilisez des commandes SQL ou PPL, assurez-vous de placer les champs avec des caractères spéciaux (non alphabétiques et non numériques) dans des backticks pour les interroger correctement. Par exemple, joignez `@messageOperation.Export`, et entre `Test::Field` backticks. Il n'est pas nécessaire de placer les champs avec des noms purement alphabétiques en backticks.

> **Important**
>
> Pour renforcer la sécurité des requêtes CloudWatch Logs Insights, à compter du 31 juillet 2025, les utilisateurs devront être connectés avec les `logs:GetQueryResults` autorisations `logs:StartQuery` et pour pouvoir exécuter des requêtes dans la CloudWatch console. Après cette date, les utilisateurs qui ne disposent pas de ces deux autorisations ne pourront pas consulter les résultats des requêtes dans la console et verront plutôt un message de bannière lorsqu'ils essaieront de consulter les résultats des requêtes dans la console.
>
> Après la modification, les autorisations d'expérience de console requises seront cohérentes avec les autorisations actuellement requises pour le SDK et AWS CLI les utilisateurs qui utilisent le StartQuery et GetQueryResults APIs.
>
> Pour plus d'informations sur la façon de créer des politiques IAM ou d'ajouter des autorisations aux politiques existantes, voir Définir des autorisations IAM personnalisées avec des politiques gérées par le client et Modifier les politiques IAM dans le Guide de l'utilisateur IAM.

CloudWatch Logs Insights propose les fonctionnalités suivantes qui peuvent être utilisées avec tous les langages de requête.

- Découverte automatique des champs de journal dans les journaux provenant de AWS services tels qu'Amazon Route 53,, AWS Lambda AWS CloudTrail, et Amazon VPC, ainsi que dans toute application ou journal personnalisé qui émet des événements de journal au format JSON.
- Création d'index de champs pour réduire les coûts et accélérer les résultats, en particulier pour les requêtes portant sur un grand nombre de groupes de journaux ou d'événements de journal. Après avoir créé des index de champs courants dans vos événements de journal, vous pouvez les utiliser dans une requête. La requête ignore les événements du journal de traitement connus pour ne pas inclure le champ indexé et traite moins de données.

  > **Note**
  >
  > La `filterIndex` commande n'est disponible que dans Logs Insights QL.

- Détection et analyse de modèles dans vos événements de journal. Un modèle est une structure de texte partagée récurrente dans vos champs de journal. Lorsque vous consultez les résultats d'une

requête, vous pouvez choisir l'onglet Modèles pour voir les modèles trouvés par CloudWatch Logs sur la base d'un échantillon de vos résultats.

- Enregistrer des requêtes, consulter l'historique de vos requêtes et réexécuter des requêtes enregistrées. Vous pouvez ainsi exécuter des requêtes complexes lorsque vous en avez besoin, sans avoir à les recréer chaque fois que vous voulez les exécuter.
- Ajouter des requêtes aux tableaux de bord.
- Chiffrer les résultats des requêtes avec AWS Key Management Service.

Les fonctionnalités suivantes de CloudWatch Logs Insights ne sont prises en charge que lorsque vous utilisez Logs Insights QL.

- Génération de requêtes en langage naturel.
- Interrogation des journaux dans la classe de journaux Infrequent Access.
- Requêtes de comparaison qui comparent les événements du journal d'un groupe de journaux aux événements du journal d'une période précédente.
- La `filterIndex`commande, qui force la requête à tenter d'analyser uniquement les événements du journal contenant un index de champ que vous spécifiez.

> **Important**
>
> CloudWatch Logs Insights ne peut pas accéder aux événements du journal dont l'horodatage est antérieur à l'heure de création du groupe de journaux.

Si vous êtes connecté à un compte configuré en tant que compte de surveillance dans le cadre de l'observabilité CloudWatch entre comptes, vous pouvez exécuter des requêtes CloudWatch Logs Insights sur les groupes de journaux des comptes sources liés à ce compte de surveillance. Vous pouvez exécuter une requête qui interroge plusieurs groupes de journaux situés dans différents comptes. Pour plus d'informations, consultez la section Observabilité inter-comptes de CloudWatch (français non garanti).

Lorsque vous créez des requêtes à l'aide de Logs Insights QL, vous pouvez également utiliser le langage naturel pour créer des requêtes CloudWatch Logs Insights. Pour ce faire, posez des questions ou décrivez les données que vous recherchez. Cette fonctionnalité assistée par l'IA génère une requête en fonction de votre demande et fournit une line-by-line explication du fonctionnement

de la requête. Pour plus d'informations, voir Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Logs Insights.

Les requêtes utilisant l'un des langages de requête pris en charge expirent au bout de 60 minutes, si elles ne sont pas terminées. Les résultats des requêtes sont disponibles pendant sept jours.

CloudWatch Les requêtes Logs Insights sont facturées en fonction de la quantité de données demandées, quelle que soit la langue de requête. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

Vous pouvez utiliser CloudWatch Logs Insights pour rechercher les données des CloudWatch journaux qui ont été envoyées à Logs le 5 novembre 2018 ou plus tard.

> **Important**
>
> Si votre équipe de sécurité réseau n'autorise pas l'utilisation de sockets Web, vous ne pouvez actuellement pas accéder à la section CloudWatch Logs Insights de la CloudWatch console. Vous pouvez utiliser les fonctionnalités de requête de CloudWatch Logs Insights à l'aide de APIs. Pour plus d'informations, consultez StartQueryle manuel Amazon CloudWatch Logs API Reference.

Table des matières

- Langages de requête pris en charge
- Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Logs Insights
- Journaux pris en charge et champs découverts
- Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation
- Analyse de modèles
- Enregistrez et réexécutez les requêtes CloudWatch Logs Insights
- Ajouter une requête au tableau de bord ou exporter les résultats de la requête
- Afficher les requêtes en cours d'exécution ou l'historique des requêtes
- Chiffrer les résultats de requête avec AWS Key Management Service
- Génération d'un résumé en langage naturel à partir des résultats des requêtes de CloudWatch Logs Insights

Langages de requête pris en charge

Les sections suivantes répertorient les commandes prises en charge dans chaque langage de requête. Ils décrivent également le format de syntaxe et fournissent des exemples de requêtes.

Rubriques

- CloudWatch Langage de requête Logs Insights (Logs Insights QL)
- OpenSearch Langage de traitement canalisé (PPL)
- OpenSearch Langage de requête structuré (SQL)

CloudWatch Langage de requête Logs Insights (Logs Insights QL)

Cette section inclut la documentation complète des commandes et fonctions de Logs Insights QL. Il inclut également des exemples de requêtes pour cette langue.

Pour plus d'informations sur les autres langages de requête que vous pouvez utiliser, consultez OpenSearch Service PPL, OpenSearch Service SQL et CloudWatch Metrics Insights.

Rubriques

- CloudWatch Syntaxe des requêtes en langage Logs Insights
- Commencez avec Logs Insights QL : tutoriels sur les requêtes
- Exemples de requêtes
- Comparer (diff) avec les plages temporelles précédentes
- Visualisation des données du journal dans des graphiques

CloudWatch Syntaxe des requêtes en langage Logs Insights

Cette section fournit des détails sur le Logs Insights QL. La syntaxe de la requête prend en charge différentes fonctions et opérations, y compris, mais sans se limiter aux fonctions générales, les opérations arithmétiques et de comparaison et les expressions régulières.

> **Important**
>
> Pour éviter d'encourir des frais excessifs en exécutant des requêtes volumineuses, gardez à l'esprit les meilleures pratiques suivantes :
>
> - Sélectionnez uniquement les groupes de journaux nécessaires pour chaque requête.

- Spécifiez toujours la plage de temps la plus courte possible pour vos requêtes.
- Lorsque vous utilisez la console pour exécuter des requêtes, annulez toutes vos requêtes avant de fermer la page de console CloudWatch Logs Insights. Dans le cas contraire, les requêtes continuent de s'exécuter jusqu'à ce qu'elles soient terminées.
- Lorsque vous ajoutez un widget CloudWatch Logs Insights à un tableau de bord, assurez-vous que le tableau de bord n'est pas actualisé à une fréquence élevée, car chaque actualisation lance une nouvelle requête.

Pour créer des requêtes contenant plusieurs commandes, séparez ces dernières par le caractère de barre verticale (|).

Pour créer des requêtes contenant des commentaires, définissez ceux-ci à l'aide du caractère dièse (#).

> **Note**
>
> CloudWatch Logs Insights découvre automatiquement les champs correspondant aux différents types de journaux et génère des champs commençant par le caractère @. Pour plus d'informations sur ces champs, consultez la section Journaux pris en charge et champs découverts dans le guide de CloudWatch l'utilisateur Amazon.

Le tableau suivant décrit brièvement chaque colonne. Il est suivi d'une description plus complète de chaque commande, avec des exemples.

> **Note**
>
> Toutes les commandes de requête Logs Insights QL sont prises en charge sur les groupes de journaux de la classe de journaux standard. Les groupes de journaux de la classe de journaux Infrequent Access prennent en charge toutes les commandes de requête Logs Insights QL`pattern`, à l'exception de`diff`, et. `unmask`

`display`	Affiche un ou plusieurs champs spécifiques dans les résultats de la requête.

fields	Affiche des champs spécifiques dans les résultats de la requête et prend en charge les fonctions et les opérations que vous pouvez utiliser pour modifier les valeurs des champs et créer des champs à utiliser dans votre requête.
filter	Filtre la requête pour ne renvoyer que les événements du journal qui satisfont à une ou plusieurs conditions.
filterIndex	Force une requête à tenter de scanner uniquement les groupes de journaux qui sont à la fois indexés sur le champ mentionné dans un index de champ et contenant également une valeur pour cet index de champ. Cela réduit le volume numérisé en essayant de scanner uniquement les événements de journal de ces groupes de journaux qui contiennent la valeur spécifiée dans la requête pour cet index de champ. Cette commande n'est pas prise en charge pour les groupes de journaux de la classe de journaux Infrequent Access.
pattern	Regroupe automatiquement les données de vos journaux dans des modèles. Un modèle est une structure de texte partagée récurrente dans vos champs de journal. CloudWatch Logs Insights vous permet d'analyse r les modèles trouvés dans vos événements de journal. Pour de plus amples informations, veuillez consulter Analyse de modèles.
diff	Compare les événements du journal trouvés au cours de la période demandée avec les événements du journal d'une période précédente de même durée, afin que vous puissiez rechercher des tendances et savoir si certains événements de journal sont nouveaux.
parse	Extrait les données d'un champ de journal afin de créer un champ extrait que vous pouvez traiter dans votre requête. **parse** prend en charge à la fois le mode glob utilisant des caractères génériques et des expressions régulières.
sort	Affiche les événements du journal renvoyés dans l'ordre croissant (`asc`) ou décroissant (`desc`).

SOURCE	L'inclusion SOURCE dans une requête est un moyen utile de spécifier un grand nombre de groupes de journaux en fonction du préfixe du nom du groupe de journaux, des identifiants de compte et de la classe de groupe de journaux à inclure dans une requête. Cette commande n'est prise en charge que lorsque vous créez une requête dans AWS CLI ou par programmation, et non dans la CloudWatch console.
stats	Calcule les statistiques agrégées en utilisant les valeurs des champs du journal.
limit	Spécifie le nombre maximum d'événements du journal que votre requête doit renvoyer. Utile avec **sort** pour renvoyer les « 20 meilleurs » résultats ou les « 20 plus récents ».
dedup	Supprime les résultats dupliqués en fonction de valeurs spécifiques dans les champs que vous indiquez.
unmask	Affiche tout le contenu d'un événement du journal dont certains contenus ont été masqués en raison d'une politique de protection des données. Pour plus d'informations sur la protection des données dans les groupes de journaux, consultez Aider à protéger les données sensibles des journaux grâce au masquage (français non garanti).
unnest	Aplatit une liste prise en entrée pour produire plusieurs enregistrements avec un seul enregistrement pour chaque élément de la liste.
Autres opérations et fonctions	CloudWatch Logs Insights prend également en charge de nombreuse s fonctions de comparaison, d'arithmétique, de date/heure, de données numériques, de chaînes, d'adresses IP, ainsi que des fonctions et opérations générales.

Les sections suivantes fournissent plus de détails sur les commandes de requête CloudWatch Logs Insights.

Rubriques

- Commandes Logs Insights QL prises en charge dans les classes de journaux
- display

- fields
- filtre
- Index des filtres
- SOURCE
- pattern
- diff
- parse
- sort
- stats
- limite
- dedup
- unmask
- unnest
- Fonctions booléennes, de comparaison, numériques, de date/heure et autres
- Champs contenant des caractères spéciaux
- Utilisation d'alias et de commentaires dans les requêtes

Commandes Logs Insights QL prises en charge dans les classes de journaux

Toutes les commandes de requête Logs Insights QL sont prises en charge sur les groupes de journaux de la classe de journaux standard. Les groupes de journaux de la classe de journaux Infrequent Access prennent en charge toutes les commandes de requête à l'exception de`pattern`, `difffilterIndex`, et. `unmask`

display

Utilisez la commande `display` pour afficher un ou plusieurs champ(s) spécifiques dans les résultats de la requête.

La commande `display` n'affiche que les champs que vous spécifiez. Si la requête contient plusieurs commandes `display`, les résultats de la requête affichent uniquement le ou les champs que vous avez spécifiés dans la commande `display` finale.

Exemple : Afficher un champ

L'extrait de code montre un exemple de requête qui utilise la commande parse pour extraire des données de `@message` pour créer les champs extraits `loggingType` et `loggingMessage`. La requête renvoie les événements du journal dans lesquels les valeurs pour `loggingType` sont ERROR (ERREUR). La commande `display` affiche uniquement les valeurs pour `loggingMessage` dans les résultats de la requête.

```
fields @message
| parse @message "[*] *" as loggingType, loggingMessage
| filter loggingType = "ERROR"
| display loggingMessage
```

> **Tip**
>
> Utiliser la commande `display` une seule fois dans une requête. Si vous utilisez la commande `display` plusieurs fois, les résultats de la requête affichent le champ spécifié dans la dernière occurrence de l'utilisation de la commande `display`.

fields

Utilisez la commande `fields` pour afficher des champs spécifiques dans les résultats de la requête.

Si les requêtes contiennent plusieurs commandes `fields` sans inclure une commande `display`, les résultats affichent tous les champs spécifiés dans les commandes `fields`.

Exemple : Afficher des champs spécifiques

L'exemple suivant présente une requête qui renvoie 20 événements du journal et les affiche par ordre décroissant. Les valeurs pour `@timestamp` et `@message` sont affichés dans les résultats de la requête.

```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
```

À utiliser `fields` plutôt que `display` lorsque vous souhaitez utiliser les différentes fonctions et opérations prises en charge par `fields` pour modifier les valeurs des champs et créer de nouveaux champs pouvant être utilisés dans les requêtes.

Vous pouvez utiliser la commande `fields` avec le mot clé as pour créer des champs extraits qui utilisent des champs et des fonctions dans vos événements de journal. Par exemple, `fields ispresent as isRes` crée un champ extrait nommé `isRes` pouvant être utilisé dans le reste de votre requête.

filtre

Utilisez la commande `filter` pour obtenir des événements du journal qui répondent à une ou plusieurs conditions.

Exemple : Filtrage des événements du journal à l'aide d'une condition

L'extrait de code montre un exemple de requête qui renvoie tous les événements du journal où la valeur de `range` est supérieur à 3 000. La requête limite les résultats à 20 événements du journal et trie les événements du journal par `@timestamp` et par ordre décroissant.

```
fields @timestamp, @message
| filter (range>3000)
| sort @timestamp desc
| limit 20
```

Exemple : Filtrage des événements du journal à l'aide de plusieurs conditions

Vous pouvez utiliser les mots-clés `and` et `or` pour combiner plusieurs conditions.

L'extrait de code montre un exemple de requête qui renvoie les événements du journal où la valeur de `range` est supérieure à 3 000 et la valeur de `accountId` est égale à 123 456 789 012. La requête limite les résultats à 20 événements du journal et trie les événements du journal par `@timestamp` et par ordre décroissant.

```
fields @timestamp, @message
| filter (range>3000 and accountId=123456789012)
| sort @timestamp desc
| limit 20
```

Champs indexés et commande de filtrage

Si vous avez créé des index de champs pour un groupe de journaux, vous pouvez exploiter ces index de champs pour améliorer l'efficacité de vos `filter` requêtes et réduire le volume numérisé. Supposons, par exemple, que vous ayez créé un index de champ pour`requestId`. Ensuite, toute

requête CloudWatch Logs Insights sur ce groupe de journaux qui inclut `filter requestId = value` ou `filter requestId IN [value, value, ...]` tentera d'ignorer les événements du journal de traitement dont on sait qu'ils n'incluent pas le champ indexé. En essayant d'analyser uniquement les événements du journal connus pour contenir ce champ indexé, le volume de numérisation peut être réduit et la requête est plus rapide.

Pour plus d'informations sur les index de champs et sur la façon de les créer, consultez Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation.

> **Important**
>
> Seules les requêtes avec `filter fieldName =...` et `filter fieldName IN...` bénéficieront des améliorations de l'index des champs. Les requêtes `filter fieldName like` n'utilisent pas d'index et analysent toujours tous les événements du journal dans les groupes de journaux sélectionnés.

Exemple : rechercher les événements du journal liés à un certain ID de demande, à l'aide d'index

Cet exemple suppose que vous avez créé un index de champ sur `requestId`. Pour les groupes de journaux qui utilisent cet index de champs, la requête utilisera les index de champs pour tenter de scanner le moins possible d'événements de journal afin de trouver des événements `requestId` ayant une valeur de `123456`

```
fields @timestamp, @message
| filter requestId = "1234656"
| limit 20
```

Correspondances et expressions régulières dans la commande filter

La commande de filtre prend en charge l'utilisation d'expressions régulières. Vous pouvez utiliser les opérateurs de comparaison suivants (=, `!=`, <, <=, >, >=) et les opérateurs booléens (`and`, `or` et `not`).

Vous pouvez utiliser le mot-clé `in` pour tester des membres d'un ensemble et vérifier les éléments d'une matrice. Pour vérifier les éléments dans une matrice, insérez la matrice après `in`. Vous pouvez utiliser l'opérateur booléen `not` avec `in`. Vous pouvez créer des requêtes utilisant `in` pour renvoyer les événements du journal où les champs sont des correspondances de chaînes. Les champs

doivent être des chaînes complètes. Par exemple, l'extrait de code suivant montre une requête qui utilise `in` pour renvoyer les événements du journal où le champ `logGroup` est le `example_group` de chaîne complète.

```
fields @timestamp, @message
| filter logGroup in ["example_group"]
```

Vous pouvez utiliser les phrases avec des mots-clés `like` et `not like` pour faire correspondre des sous-chaînes. Vous pouvez utiliser l'opérateur d'expression régulière =~ pour faire correspondre des sous-chaînes. Pour faire correspondre une sous-chaîne avec `like` et `not like`, placez la sous-chaîne à faire correspondre entre guillemets simples ou doubles. Vous pouvez utiliser des modèles d'expression régulière avec `like` et `not like`. Pour faire correspondre une sous-chaîne avec l'opérateur d'expression régulière, placez la sous-chaîne que vous souhaitez faire correspondre entre des barres obliques. Les exemples suivants contiennent des extraits de code qui montrent comment faire correspondre des sous-chaînes à l'aide de la commande `filter`.

Exemples : faire correspondre des sous-chaînes

Les exemples suivants renvoient des événements du journal où `f1` contient le mot Exception. Les trois exemples sont sensibles à la casse.

Le premier exemple fait correspondre une sous-chaîne avec `like`.

```
fields f1, f2, f3
| filter f1 like "Exception"
```

Le deuxième exemple fait correspondre une sous-chaîne avec `like` et un modèle d'expression régulière.

```
fields f1, f2, f3
| filter f1 like /Exception/
```

Le troisième exemple fait correspondre une sous-chaîne avec une expression régulière.

```
fields f1, f2, f3
| filter f1 =~ /Exception/
```

Exemple : faire correspondre des sous-chaînes avec des caractères de remplacement

Vous pouvez utiliser le point (`.`) comme caractère de remplacement dans les expressions régulières pour faire correspondre des sous-chaînes. Dans l'exemple suivant, la requête renvoie des correspondances lorsque la valeur de `f1` commence par la chaîne `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog./
```

Vous pouvez placer l'astérisque après le point (`.*`) pour créer un quantificateur gourmand qui renvoie autant de correspondances que possible. Par exemple, la requête suivante renvoie les résultats où la valeur de `f1` ne commence par la chaîne `ServiceLog` et inclut également la chaîne `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog.*/
```

Les correspondances possibles peuvent être mises en forme comme suit :

- `ServiceLogSampleApiLogGroup`
- `SampleApiLogGroupServiceLog`

Exemple : exclure des sous-chaines des correspondances

L'exemple suivant montre une requête qui renvoie des événements du journal où `f1` ne contient pas le mot Exception. L'exemple est sensible au cas par cas.

```
fields f1, f2, f3
| filter f1 not like "Exception"
```

Exemple : faire correspondre des sous-chaînes avec des modèles insensibles à la casse

Vous pouvez faire correspondre des sous-chaînes insensibles à la casse avec `like` et des expressions régulières. Placez le paramètre suivant (?i) avant la sous-chaîne à faire correspondre. L'exemple suivant montre une requête qui renvoie des événements du journal où `f1` contient pas le mot Exception ou exception.

```
fields f1, f2, f3
| filter f1 like /(?i)Exception/
```

Index des filtres

`filterIndex`À utiliser pour renvoyer uniquement des données indexées, en forçant une requête à analyser uniquement les groupes de journaux indexés sur un champ que vous spécifiez dans la requête. Pour les groupes de journaux indexés sur ce champ, cela optimise davantage la requête en ignorant les groupes de journaux qui ne contiennent aucun événement de journal contenant le champ spécifié dans la requête pour le champ indexé. Il réduit encore le volume numérisé en essayant de scanner uniquement les événements de journal de ces groupes de journaux qui correspondent à la valeur spécifiée dans la requête pour cet index de champ. Pour plus d'informations sur les index de champs et sur la façon de les créer, consultez[Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation](#).

L'utilisation de champs `filterIndex` indexés peut vous aider à interroger efficacement les groupes de journaux contenant des pétaoctets de données de journal en limitant l'espace de recherche réel aux groupes de journaux et aux événements de journal contenant des index de champs.

Supposons, par exemple, que vous ayez créé un index `IPaddress` de champs pour certains groupes de journaux de votre compte. Vous pouvez ensuite créer la requête suivante et choisir d'interroger tous les groupes de journaux du compte pour trouver les événements de journal qui incluent la valeur `198.51.100.0` dans le `IPaddress` champ.

```
fields @timestamp, @message
| filterIndex IPaddress = "198.51.100.0"
| limit 20
```

La `filterIndex` commande fait en sorte que cette requête tente d'ignorer tous les groupes de journaux qui ne sont pas indexés`IPaddress`. En outre, au sein des groupes de journaux indexés, la requête ignore les événements de journal dont la valeur de ce `IPaddress` champ n'a pas été observée`198.51.100.0`.

Utilisez l'`IN`opérateur pour étendre les résultats à l'une des multiples valeurs des champs indexés. L'exemple suivant recherche les événements de journal qui incluent soit la valeur, `198.51.100.0` soit `198.51.100.1` dans le `IPaddress` champ.

```
fields @timestamp, @message
| filterIndex IPaddress in ["198.51.100.0", "198.51.100.1"]
| limit 20
```

CloudWatch Logs fournit des index de champs par défaut pour tous les groupes de journaux de la classe de journaux standard. Les index de champs par défaut sont automatiquement disponibles pour les champs suivants :

- `@aws.region`
- `@aws.account`
- `@source.log`
- `traceId`

Les index de champs par défaut s'ajoutent aux index de champs personnalisés que vous définissez dans le cadre de votre politique. Les index de champs par défaut ne sont pas pris en compte dans votre quota d'index de champs.

FilterIndex comparé au filtre

Pour illustrer la différence entre `filterIndex` et`filter`, considérez les exemples de requêtes suivants. Supposons que vous ayez créé un index de champs pour quatre de vos groupes de journaux, mais pas pour un cinquième groupe de journaux. `IPaddress` L'utilisation de la requête suivante `filterIndex` ignorera l'analyse du groupe de journaux dont le champ n'est pas indexé. Pour chaque groupe de journaux indexé, il tente d'analyser uniquement les événements du journal contenant le champ indexé, et il ne renvoie également que les résultats obtenus après la création de l'index du champ.

```
fields @timestamp, @message
| filterIndex IPaddress = "198.51.100.0"
| limit 20
```

En revanche, si vous utilisez `filter` plutôt que `filterIndex` pour une requête portant sur les cinq mêmes groupes de journaux, la requête tentera d'analyser non seulement les événements de journal contenant la valeur dans les groupes de journaux indexés, mais également le cinquième groupe de journaux qui n'est pas indexé, et elle analysera chaque événement de journal de ce cinquième groupe de journaux.

```
fields @timestamp, @message
| filter IPaddress = "198.51.100.0"
| limit 20
```

SOURCE

L'inclusion `SOURCE` dans une requête est un moyen utile de spécifier les groupes de journaux à inclure dans une requête lorsque vous utilisez l'API AWS CLI or pour créer une requête. La `SOURCE` commande est prise en charge uniquement dans l'API AWS CLI and, pas dans la CloudWatch console. Lorsque vous utilisez la CloudWatch console pour démarrer une requête, vous utilisez l'interface de console pour spécifier les groupes de journaux.

`SOURCE`Pour spécifier les groupes de journaux à interroger, vous pouvez utiliser les mots clés suivants :

- `namePrefix`exécute la requête sur des groupes de journaux dont le nom commence par la chaîne que vous spécifiez. Si vous omettez cette option, tous les groupes de journaux sont interrogés.

  Vous pouvez inclure jusqu'à cinq préfixes dans la liste.
- `accountIdentifiers`exécute la requête sur les groupes de journaux du AWS compte spécifié. Cela ne fonctionne que lorsque vous exécutez la requête dans un compte de surveillance. Si vous omettez cette option, la requête par défaut consiste à interroger tous les comptes source liés ainsi que le compte de surveillance actuel. Pour plus d'informations sur l'observabilité entre comptes, consultez la section Observabilité CloudWatch entre comptes.

  Vous pouvez inclure jusqu'à 20 identifiants de compte dans la liste.
- `logGroupClass`exécute la requête sur des groupes de journaux appartenant à la classe de journaux spécifiée, qu'il s'agisse d'un accès standard ou peu fréquent. Si vous omettez cela, la classe de log standard par défaut est utilisée. Pour plus d'informations sur les classes de log, consultezClasses de log.

Comme vous pouvez spécifier un grand nombre de groupes de journaux à interroger de cette manière, nous vous recommandons de `SOURCE` ne les utiliser que dans les requêtes qui exploitent les index de champs que vous avez créés. Pour plus d'informations sur l'indexation des champs dans les groupes de journaux, voir Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation

L'exemple suivant sélectionne tous les groupes de journaux du compte. S'il s'agit d'un compte de surveillance, les groupes de journaux relatifs à la surveillance et à tous les comptes sources seront sélectionnés. Si le nombre total de groupes de journaux dépasse 10 000, un message d'erreur vous invite à réduire le nombre de groupes de journaux en utilisant une autre méthode de sélection des groupes de journaux.

```
SOURCE logGroups()
```

L'exemple suivant sélectionne les groupes de journaux dans le compte `111122223333` source. Si vous lancez une requête dans un compte de surveillance dans le cadre de l'observabilité CloudWatch entre comptes, les groupes de journaux de tous les comptes sources et du compte de surveillance sont sélectionnés par défaut.

```
SOURCE logGroups(accountIdentifiers:['111122223333'])
```

L'exemple suivant sélectionne les groupes de journaux en fonction des préfixes de noms.

```
SOURCE logGroups(namePrefix: ['namePrefix1', 'namePrefix2'])
```

L'exemple suivant sélectionne tous les groupes de journaux de la classe de journaux Infrequent Access. Si vous n'incluez pas l'`class`identifiant, la requête s'applique uniquement aux groupes de journaux de la classe de journaux standard, qui est la valeur par défaut.

```
SOURCE logGroups(class: ['INFREQUENT_ACCESS'])
```

L'exemple suivant sélectionne les groupes de journaux du compte 111122223333 qui commencent par des préfixes de nom spécifiques et appartiennent à la classe de journal standard. La classe n'est pas mentionnée dans la commande car Standard est la valeur de classe de journal par défaut.

```
SOURCE logGroups(accountIdentifiers:['111122223333'], namePrefix: ['namePrefix1',
 'namePrefix2']
```

Le dernier exemple montre comment utiliser la `SOURCE` commande avec la `start-query` AWS CLI commande.

```
aws logs start-query
--region us-east-1
--start-time 1729728200
--end-time 1729728215
--query-string "SOURCE logGroups(namePrefix: ['Query']) | fields @message | limit 5"
```

pattern

Utilisez `pattern` pour regrouper automatiquement les données de vos journaux dans des modèles.

Un modèle est une structure de texte partagée récurrente dans les champs de vos journaux. Vous pouvez l'utiliser `pattern` pour identifier les tendances émergentes, surveiller les erreurs connues et identifier les lignes de journal fréquentes ou coûteuses. CloudWatch Logs Insights fournit également une expérience de console que vous pouvez utiliser pour identifier et analyser de manière plus approfondie les modèles dans vos événements de journal. Pour de plus amples informations, veuillez consulter Analyse de modèles.

Comme la `pattern` commande identifie automatiquement les modèles courants, vous pouvez l'utiliser comme point de départ pour rechercher et analyser vos journaux. Vous pouvez également combiner la commande `pattern` avec les commandes `filter`, `parse` ou `sort` pour identifier des modèles dans des requêtes plus précises.

Entrée de commande Pattern

La commande `pattern` attend l'une des entrées suivantes : le champ `@message`, un champ extrait créé à l'aide de la commande `parse`, ou une chaîne manipulée à l'aide d'une ou plusieurs fonctions de chaîne.

Si CloudWatch Logs ne peut pas déduire le type de données représenté par un jeton dynamique, il l'affiche sous la forme <Token- *number* > et *number* indique où ce jeton apparaît dans le schéma, par rapport aux autres jetons dynamiques.

Les exemples courants de jetons dynamiques incluent les codes d'erreur, les adresses IP, les horodatages et les demandes. IDs

Sortie de commande Pattern

La commande `pattern` produit la sortie suivante :

- `@pattern` : structure de texte partagée récurrente dans les champs d'événements de vos journaux. Les champs qui varient au sein d'un modèle, tels qu'un ID de demande ou un horodatage, sont représentés par des jetons. Si CloudWatch Logs peut déterminer le type de données représenté par un jeton dynamique, il affiche le jeton sous la forme<*string*-*number*>. *string*Il s'agit d'une description du type de données que le jeton représente. *number*Indique où ce jeton apparaît dans le modèle, par rapport aux autres jetons dynamiques.

  CloudWatch Logs attribue la partie chaîne du nom en fonction de l'analyse du contenu des événements du journal qui le contiennent.

Si CloudWatch Logs ne peut pas déduire le type de données représenté par un jeton dynamique, il l'affiche sous la forme <Token- *number* > et *number* indique où ce jeton apparaît dans le schéma, par rapport aux autres jetons dynamiques.

Par exemple, `[INFO] Request time: <Time-1> ms` est une sortie potentielle pour le message de journal `[INFO] Request time: 327 ms`.

- `@ratio` : ratio d'événements de journal d'une période et de groupes de journaux spécifiés correspondant à un modèle identifié. Par exemple, si la moitié des événements de journal des groupes de journaux et de la période sélectionnés correspondent au modèle, `@ratio` renvoie `0.50`
- `@sampleCount` : nombre d'événements de journal d'une période et de groupes de journaux spécifiés correspondant à un modèle identifié.
- `@severityLabel` : gravité ou niveau de journal, indiquant le type d'informations contenues dans celui-ci. Par exemple, `Error`, `Warning` ou `Info` ou `Debug`.

Exemples

La commande suivante identifie les journaux présentant des structures similaires dans le(s) groupe(s) de journaux spécifié(s) sur la période sélectionnée, en les regroupant par modèle et par nombre

```
pattern @message
```

La commande `pattern` peut être utilisée en combinaison avec la commande `filter`

```
filter @message like /ERROR/
| pattern @message
```

La commande `pattern` peut être utilisée avec les commandes `parse` et `sort`

```
filter @message like /ERROR/
| parse @message 'Failed to do: *' as cause
| pattern cause
| sort @sampleCount asc
```

diff

Compare les événements du journal trouvés au cours de la période demandée avec les événements du journal d'une période précédente de même durée. De cette façon, vous pouvez rechercher des tendances et déterminer si des événements spécifiques du journal sont nouveaux.

Ajoutez un modificateur à la `diff` commande pour spécifier la période à laquelle vous souhaitez comparer :

- `diff`compare les événements du journal dans la plage de temps actuellement sélectionnée aux événements du journal de la plage de temps immédiatement précédente.
- `diff previousDay`compare les événements du journal dans la plage horaire actuellement sélectionnée aux événements du journal survenus à la même heure le jour précédent.
- `diff previousWeek`compare les événements du journal dans la plage horaire actuellement sélectionnée aux événements du journal survenus à la même période de la semaine précédente.
- `diff previousMonth`compare les événements du journal dans la plage de temps actuellement sélectionnée aux événements du journal survenus à la même époque le mois précédent.

Pour de plus amples informations, veuillez consulter Comparer (diff) avec les plages temporelles précédentes.

parse

Utilisez la commande `parse` pour extraire les données d'un champ de journal et créer un champ extrait que vous pouvez traiter dans votre requête. **`parse`** prend en charge à la fois le mode glob utilisant des caractères génériques et des expressions régulières. Pour plus d'informations sur la syntaxe des expressions régulières, consultezSyntaxe des expressions régulières (regex) prise en charge.

Vous pouvez analyser les champs JSON imbriqués avec une expression régulière.

Exemple : Analyse d'un champ JSON imbriqué

L'extrait de code montre comment analyser un événement du journal au format JSON qui a été aplati lors de l'ingestion.

```
{'fieldsA': 'logs', 'fieldsB': [{'fA': 'a1'}, {'fA': 'a2'}]}
```

L'extrait de code montre une requête avec une expression régulière qui extrait les valeurs de `fieldsA` et `fieldsB` pour créer les champs extraits `fld` et `array`.

```
parse @message "'fieldsA': '*', 'fieldsB': ['*']" as fld, array
```

Groupes de capture nommés

Lorsque vous utilisez **`parse`** avec une expression régulière, vous pouvez utiliser des groupes de capture nommés pour capturer un modèle dans un champ. La syntaxe est `parse @message (?<Name>pattern)`

L'exemple suivant utilise un groupe de capture sur un journal de flux VPC pour extraire l'ENI dans un champ nommé `NetworkInterface`.

```
parse @message /(?<NetworkInterface>eni-.*?) / | display NetworkInterface, @message
```

> **Note**
>
> Les événements du journal au format JSON sont aplatis lors de l'ingestion. Actuellement, l'analyse de champs JSON imbriqués avec une expression globale n'est pas prise en charge. Vous ne pouvez analyser que les événements du journal au format JSON qui ne contiennent pas plus de 200 champs d'événements de journal. Lorsque vous analysez des champs JSON imbriqués, vous devez mettre en forme l'expression régulière de votre requête pour qu'elle corresponde à votre événement du journal JSON.

Exemples de la commande d'analyse.

Utilisez une expression glob pour extraire les champs **`@user`**, **`@method`** et **`@latency`** à partir du champ de journal **`@message`** et renvoyer la latence moyenne pour chaque combinaison unique de **`@method`** et **`@user`**.

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Utilisez une expression régulière pour extraire les champs **`@user2`**, **`@method2`** et **`@latency2`** à partir du champ du journal **`@message`** et renvoyer la latence moyenne pour chaque combinaison unique de **`@method2`** et **`@user2`**.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Extrait les champs **`loggingTime`**, **`loggingType`** et **`loggingMessage`**, filtre vers le bas pour journaliser les événements qui contiennent les chaînes **`ERROR`** ou **`INFO`**, puis affiche uniquement les champs **`loggingMessage`** et **`loggingType`** pour les événements qui contiennent une chaîne **`ERROR`**.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

sort

Utilisez la commande `sort` pour afficher les événements du journal dans l'ordre croissant (`asc`) ou décroissant (`desc`) en fonction d'un champ spécifié. Vous pouvez l'utiliser avec la commande `limit` pour créer des requêtes de type « N premiers » ou « N derniers ».

L'algorithme de tri est une version actualisée du tri naturel. Si vous triez par ordre croissant, la logique suivante est utilisée.

- Toutes les valeurs non numériques précèdent toutes les valeurs numériques. Les valeurs numériques sont des valeurs qui incluent uniquement des chiffres, et non un mélange de chiffres et d'autres caractères.
- Pour les valeurs non numériques, l'algorithme regroupe les caractères numériques consécutifs et les caractères alphabétiques consécutifs en segments distincts à des fins de comparaison. Il classe les parties non numériques en fonction de leurs valeurs Unicode, et les parties numériques en fonction de leur longueur, puis en fonction de leur valeur numérique.

Pour plus d'informations sur l'ordre Unicode, voir Liste des caractères Unicode.

Par exemple, ce qui suit est le résultat d'un tri par ordre croissant.

```
!: >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> sorted by unicode order
#
*%04
0# >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Alphanumeric starting with numbers
```

```
5A
111A   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  Starts with more digits than 5A, so it sorted
 to be later than 5A
2345_
@ >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> 2345 is compared with @ in the unicode order,
@_
A >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Values starting with letters
A9876fghj
a12345hfh
0 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Number values
01
1
2
3
```

Si vous triez par ordre décroissant, les résultats du tri sont inverses.

Par exemple, la requête suivante pour les journaux de flux Amazon VPC permet de trouver les 15 premiers transferts de paquets entre les hôtes.

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

stats

Utilisez `stats` pour créer des visualisations les données de vos journaux, telles que des diagrammes à barres, des graphiques linéaires et des diagrammes à aires empilées. Cela vous permet d'identifier plus efficacement les modèles dans vos données de journal. CloudWatch Logs Insights génère des visualisations pour les requêtes qui utilisent la `stats` fonction et une ou plusieurs fonctions d'agrégation.

Par exemple, la requête suivante dans un groupe de journaux Route 53 renvoie des visualisations présentant la distribution des enregistrements Route 53 par heure, par type de requête.

```
stats count(*) by queryType, bin(1h)
```

Toutes ces requêtes peuvent produire des graphiques à barres. En outre, si votre requête utilise la fonction `bin()` pour regrouper les données en fonction d'un seul champ au fil du temps, vous pouvez également voir des graphiques linéaires et des graphiques en aires empilées.

les unités de temps et les abréviations suivantes sont prises en charge par la fonction `bin`. Pour toutes les unités et abréviations qui incluent plus d'un caractère, l'ajout de s au pluriel est pris en charge. Donc, les deux `hr` et `hrs` travaillent pour spécifier les heures.

- `millisecond` `ms` `msec`
- `second` `s` `sec`
- `minute` `m` `min`
- `hour` `h` `hr`
- `day` `d`
- `week` `w`
- `month` `mo` `mon`
- `quarter` `q` `qtr`
- `year` `y` `yr`

Rubriques

- Visualisation des données de séries temporelles
- Visualisation des données de journal regroupées par champs
- Utiliser plusieurs commandes stats dans une seule requête
- Fonctions à utiliser avec stats

Visualisation des données de séries temporelles

Les visualisations chronologiques fonctionnent pour les requêtes présentant les caractéristiques suivantes :

- La requête contient une ou plusieurs fonctions d'agrégation. Pour plus d'informations, consultez Aggregation Functions in the Stats Command.
- La requête utilise la fonction `bin()` pour regrouper les données en un champ.

Ces requêtes peuvent produire des graphiques linéaires, des graphiques en aires empilées et des graphiques à barres et des graphiques circulaires.

Exemples

Pour un didacticiel complet, consultez the section called "Didacticiel : Exécuter une requête qui produit des séries temporelles".

Voici d'autres exemples de requêtes qui fonctionnent pour la visualisation chronologiques.

La requête suivante génère une visualisation des valeurs moyennes du champ `myfield1`, avec un point de données créé toutes les cinq minutes. Chaque point de données correspond à l'agrégation des moyennes des valeurs `myfield1` des journaux des cinq minutes précédentes.

```
stats avg(myfield1) by bin(5m)
```

La requête suivante génère une visualisation des trois valeurs basées sur différents champs, avec un point de données créé toutes les cinq minutes. La visualisation est générée, car la requête contient des fonctions d'agrégation et utilise `bin()` comme champ de regroupement.

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

Restrictions pour les graphiques linéaires et les graphiques en aires empilées

Les requêtes qui regroupent les informations d'entrée de journal sans utiliser la fonction `bin()` peuvent générer des graphiques à barres. Toutefois, les requêtes ne peuvent pas générer de graphique linéaire ou de graphique en aires empilées. Pour plus d'informations sur ces types de requête, consultez the section called "Visualisation des données de journal regroupées par champs".

Visualisation des données de journal regroupées par champs

Vous pouvez produire des graphiques à barres pour les requêtes qui utilisent la fonction `stats` et une ou plusieurs fonctions d'agrégation. Pour plus d'informations, consultez Aggregation Functions in the Stats Command.

Pour afficher la visualisation, exécutez votre requête. Choisissez ensuite l'onglet Visualization (Visualisation), sélectionnez la flèche en regard de Line (Ligne), puis choisissez Bar (Barre). Les visualisations sont limitées à 100 barres dans le graphique à barres.

Exemples

Pour un didacticiel complet, consultez the section called "Didacticiel : Exécuter une requête qui génère une visualisation groupée par champs de journal". Les paragraphes suivants incluent d'autres exemples de requête pour la visualisation par champs.

La requête de journal de flux VPC suivante recherche le nombre moyen d'octets transférés par session pour chaque adresse de destination.

```
stats avg(bytes) by dstAddr
```

Vous pouvez également produire un graphique qui comprend plus d'une barre pour chaque valeur résultante. Par exemple, la requête de journal de flux VPC suivante recherche le nombre moyen et le nombre maximal d'octets transférés par session pour chaque adresse de destination.

```
stats avg(bytes), max(bytes) by dstAddr
```

La requête suivante recherche le nombre de journaux de requêtes Amazon Route 53 pour chaque type de requête.

```
stats count(*) by queryType
```

Utiliser plusieurs commandes stats dans une seule requête

Vous pouvez utiliser jusqu'à deux commandes `stats` dans une seule requête. Cela vous permet d'effectuer une agrégation supplémentaire sur la sortie de la première agrégation.

Exemple : requête avec deux commandes **`stats`**

Par exemple, la requête suivante trouve d'abord le volume de trafic total par tranches de 5 minutes, puis calcule le volume de trafic le plus élevé, le plus bas et le volume de trafic moyen parmi ces tranches de 5 minutes.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length)/1024/1024 as logs_mb BY bin(5m)
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb
```

Exemple : combiner plusieurs commandes stats avec d'autres fonctions telles que **`filter`**, **`fields`**, **`bin`**

Vous pouvez combiner deux commandes `stats` avec d'autres commandes, telles que `filter` et `fields` dans une seule requête. Par exemple, la requête suivante trouve le nombre d'adresses IP distinctes dans les sessions et le nombre de sessions par plateforme client, filtre ces adresses IP et enfin trouve la moyenne des demandes de session par plateforme client.

```
STATS count_distinct(client_ip) AS session_ips,
      count(*) AS requests BY session_id, client_platform
| FILTER session_ips > 1
| STATS count(*) AS multiple_ip_sessions,
        sum(requests) / count(*) AS avg_session_requests BY client_platform
```

Vous pouvez utiliser les fonctions `bin` et `dateceil` dans des requêtes comportant plusieurs commandes `stats`. Par exemple, la requête suivante combine d'abord les messages en blocs de 5 minutes, puis agrège ces blocs de 5 minutes en blocs de 10 minutes et calcule les volumes de trafic les plus élevés, les plus bas et la moyenne dans chaque bloc de 10 minutes.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length) / 1024 / 1024 AS logs_mb BY BIN(5m) as @t
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb BY dateceil(@t, 10m)
```

Remarques et limitations

Une requête peut avoir un maximum de deux commandes `stats`. Ce quota ne peut pas être modifié.

Si vous utilisez une commande `sort` ou `limit`, elle doit apparaître après la deuxième commande `stats`. Si elle est antérieure à la deuxième commande `stats`, la requête n'est pas valide.

Lorsqu'une requête comporte deux commandes `stats`, les résultats partiels de la requête ne commencent pas à s'afficher tant que la première agrégation `stats` n'est pas terminée.

Dans la deuxième commande `stats` d'une seule requête, vous ne pouvez faire référence qu'aux champs définis dans la première commande `stats`. Par exemple, la requête suivante n'est pas valide, car le champ `@message` ne sera plus disponible après la première agrégation `stats`.

```
FIELDS @message
| STATS SUM(Fault) by Operation
# You can only reference `SUM(Fault)` or Operation at this point
| STATS MAX(strlen(@message)) AS MaxMessageSize # Invalid reference to @message
```

Tous les champs auxquels vous faites référence après la première commande `stats` doivent être définis dans cette première commande `stats`.

```
STATS sum(x) as sum_x by y, z
```

```
| STATS max(sum_x) as max_x by z
# You can only reference `max(sum_x)`, max_x or z at this point
```

> **Important**
>
> La fonction `bin` utilise toujours implicitement le champ `@timestamp`. Cela signifie que vous ne pouvez pas utiliser `bin` dans la deuxième commande `stats` sans utiliser la première commande `stats` pour propager le champ `timestamp`. Par exemple, la requête suivante n'est pas valide.
>
> ```
> FIELDS strlen(@message) AS message_length
> | STATS sum(message_length) AS ingested_bytes BY @logStream
> | STATS avg(ingested_bytes) BY bin(5m) # Invalid reference to @timestamp field
> ```
>
> Définissez plutôt le champ `@timestamp` dans la première commande `stats`, puis utilisez-le avec `dateceil` dans la deuxième commande `stats`, comme dans l'exemple suivant.
>
> ```
> FIELDS strlen(@message) AS message_length
> | STATS sum(message_length) AS ingested_bytes, max(@timestamp) as @t BY
> @logStream
> | STATS avg(ingested_bytes) BY dateceil(@t, 5m)
> ```

Fonctions à utiliser avec stats

CloudWatch Logs Insights prend en charge à la fois les fonctions d'agrégation des statistiques et les fonctions de non-agrégation des statistiques.

Utilisez des fonctions statistiques d'agrégation dans la commande `stats` et en tant qu'arguments pour d'autres fonctions.

Fonction	Type de résultat	Description
`avg(fieldName: NumericLogField)`	nombre	Moyenne des valeurs dans le champ spécifié.
`count()`	nombre	Compte les événements du journal. `count()`(ou `count(*)`) compte tous les

Fonction	Type de résultat	Description
`count(fieldName: LogField)`		événements renvoyés par la requête, tandis que `count(fieldName)` compte tous les enregistrements qui incluent le nom du champ spécifié.
`count_distinct(fieldName: LogField)`	nombre	Renvoie le nombre de valeurs uniques pour le champ. Si le champ a une cardinalité très élevée (contient de nombreuses valeurs uniques), la valeur renvoyée par `count_distinct` n'est qu'une approximation.
`max(fieldName: LogField)`	LogFieldValue	Valeur maximale des valeurs pour ce journal dans le champ interrogé.
`min(fieldName: LogField)`	LogFieldValue	Valeur minimale des valeurs pour ce journal dans le champ interrogé.
`pct(fieldName: LogFieldValue, percent: number)`	LogFieldValue	Un centile indique la position relative d'une valeur dans un ensemble de données. Par exemple, `pct(@duration, 95)` renvoie la valeur `@duration` à laquelle 95 % des valeurs de `@duration` sont inférieures à cette valeur et 5 % des valeurs lui sont supérieures.
`stddev(fieldName: NumericLogField)`	nombre	Déviation standard des valeurs dans le champ spécifié.
`sum(fieldName: NumericLogField)`	nombre	Somme des valeurs dans le champ spécifié.

Fonctions statiques de non-agrégation

Utilisez des fonctions de non-agrégation dans la commande `stats` et en tant qu'arguments d'autres fonctions.

Fonction	Type de résultat	Description
`earliest(fieldName: LogField)`	LogField	Renvoie la valeur de `fieldName` à partir de l'événement de journal qui a l'horodatage le plus récent dans les journaux interrogés.
`latest(fieldName: LogField)`	LogField	Renvoie la valeur de `fieldName` à partir de l'événement de journal qui a l'horodatage le plus ancien dans les journaux interrogés.
`sortsFirst(fieldName: LogField)`	LogField	Renvoie la valeur de `fieldName` triée la première dans les journaux interrogés.
`sortsLast(fieldName: LogField)`	LogField	Renvoie la valeur de `fieldName` triée la dernière dans les journaux interrogés.

limite

Utilisez la commande `limit` pour spécifier le nombre d'événements du journal que vous voulez que votre requête retourne. Si vous omettez`limit`, la requête renverra jusqu'à 10 000 événements de journal dans les résultats.

Par exemple, l'exemple suivant renvoie uniquement les 25 derniers événements du journal

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

dedup

Utilisez `dedup` pour supprimer les résultats dupliqués en fonction de valeurs spécifiques dans les champs que vous indiquez. Vous pouvez l'utiliser `dedup` avec un ou plusieurs champs. Si vous spécifiez un champ avec `dedup`, un seul événement du journal est renvoyé pour chaque valeur unique de ce champ. Si vous spécifiez plusieurs champs, un événement du journal est renvoyé pour chaque combinaison unique de valeurs pour ces champs.

Les doublons sont supprimés en fonction de l'ordre de tri, seul le premier résultat selon l'ordre de tri étant conservé. Nous vous recommandons de trier vos résultats avant de les soumettre à la

commande `dedup`. Si les résultats ne sont pas triés avant d'être soumis à `dedup`, l'ordre de tri décroissant par défaut utilisant `@timestamp` est appliqué.

Les valeurs nulles ne sont pas considérées comme des doublons pour l'évaluation. Les événements du journal dont les valeurs sont nulles pour l'un des champs spécifiés sont conservés. Pour éliminer les champs contenant des valeurs nulles, utilisez **`filter`** en appliquant la fonction `isPresent(field)`.

La seule commande de requête que vous pouvez utiliser dans une requête après la commande `dedup` est `limit`.

Exemple : n'afficher que l'événement du journal le plus récent pour chaque valeur unique du champ nommé **`server`**

L'exemple suivant affiche les champs `timestamp`, `server`, `severity`, et `message` correspondant uniquement à l'événement le plus récent pour chaque valeur unique de `server`.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server
```

Pour d'autres exemples de requêtes CloudWatch Logs Insights, consultezRequêtes générales.

unmask

Utilisez la commande `unmask` pour afficher tout le contenu d'un événement du journal dont certains contenus ont été masqués en raison d'une politique de protection des données. Pour exécuter cette commande, vous devez disposer de l'autorisation `logs:Unmask`.

Pour plus d'informations sur la protection des données dans les groupes de journaux, consultez Aider à protéger les données sensibles des journaux grâce au masquage (français non garanti).

unnest

`unnest`À utiliser pour aplatir une liste prise en entrée afin de produire plusieurs enregistrements avec un seul enregistrement pour chaque élément de la liste. En fonction du nombre d'éléments contenus dans un champ, cette commande supprime l'enregistrement actuel et génère de nouveaux enregistrements. Chaque enregistrement inclut le`unnested_field`, qui représente un élément. Tous les autres champs proviennent de l'enregistrement d'origine.

L'entrée pour `unnest LIST` is provient de la `jsonParse` fonction. Pour plus d'informations, consultez la section Types de structures. Tous les autres types, tels que `MAPnumbers`, `String` et, sont traités comme une liste contenant un élément`unnest`.

Structure de commande

L'exemple suivant décrit le format de cette commande.

```
unnest field into unnested_field
```

Exemple de requête

L'exemple suivant analyse une chaîne d'objet JSON et développe une liste d'événements de champ.

```
fields jsonParse(@message) as json_message
| unnest json_message.events into event
| display event.name
```

L'événement de journal pour cet exemple de requête peut être une chaîne JSON comme suit :

```
{
   "events": [
        {
            "name": "exception"
        },
        {
            "name": "user action"
        }
   ]
}
```

Dans ce cas, l'exemple de requête produit deux enregistrements dans le résultat de la requête, l'un avec « `event.name` as » `exception` et l'autre avec « `event.name` as user action ».

Exemple de requête

L'exemple suivant aplatit une liste, puis filtre les éléments.

```
fields jsonParse(@message) as js
| unnest js.accounts into account
| filter account.type = "internal"
```

Exemple de requête

L'exemple suivant aplatit une liste à des fins d'agrégation.

```
fields jsonParse(trimmedData) as accounts
| unnest accounts into account
| stats sum(account.droppedSpans) as n by account.accountId
| sort n desc
| limit 10
```

Fonctions booléennes, de comparaison, numériques, de date/heure et autres

CloudWatch Logs Insights prend en charge de nombreuses autres opérations et fonctions dans les requêtes, comme expliqué dans les sections suivantes.

Rubriques

- Opérateurs arithmétiques
- Opérateurs booléens
- Opérateurs de comparaison
- Opérateurs numériques
- Types de structures
- Fonctions Datetime
- Fonctions générales
- Fonctions JSON
- Fonctions de chaîne d'adresse IP
- Fonctions de chaîne

Opérateurs arithmétiques

Les opérateurs arithmétiques acceptent les types de données numériques en tant qu'arguments et renvoient des résultats numériques. Utilisez des opérateurs arithmétiques dans les commandes `filter` et `fields` et en tant qu'arguments pour d'autres fonctions.

Opération	Description
`a + b`	Addition

Opération	Description
`a - b`	Soustraction
`a * b`	Multiplication
`a / b`	Division
`a ^ b`	Élévation à la puissance (`2 ^ 3` renvoie 8)
`a % b`	Valeurs restantes ou module (`10 % 3` renvoie 1)

Opérateurs booléens

Utilisez les opérateurs booléens **`and`**, **`or`** et **`not`**.

> **Note**
>
> Utilisez les opérateurs booléens uniquement dans les fonctions qui renvoient une valeur de TRUE (VRAI) ou FALSE (FAUX).

Opérateurs de comparaison

Les opérateurs de comparaison acceptent tous les types de données en tant qu'arguments et renvoient un résultat booléen. Utilisez des opérations de comparaison dans la commande `filter` et en tant qu'arguments pour d'autres fonctions.

Opérateur	Description
`=`	Égal à
`!=`	Non égal à
`<`	Inférieur à
`>`	Supérieure à

Opérateur	Description
`<=`	Inférieur ou égal à
`>=`	Supérieur ou égal à

Opérateurs numériques

Les opérations numériques acceptent les types de données numériques en tant qu'arguments numériques et renvoient des résultats numériques. Utilisez des opérations numériques dans les commandes `filter` et `fields` et en tant qu'arguments pour d'autres fonctions.

Opération	Type de résultat	Description
`abs(a: number)`	nombre	Valeur absolue
`ceil(a: number)`	nombre	Arrondir jusqu'au nombre entier supérieur suivant (le plus petit nombre entier supérieur à la valeur de a)
`floor(a: number)`	nombre	Arrondir jusqu'au nombre entier inférieur suivant (le plus grand nombre entier inférieur à la valeur de a)
`greatest(a: number, ...numbers: number[])`	nombre	Renvoie la valeur la plus grande
`least(a: number, ...numbers: number[])`	nombre	Renvoie la valeur la plus petite
`log(a: number)`	nombre	Journal naturel
`sqrt(a: number)`	nombre	Racine carrée

Types de structures

Une carte ou une liste est un type de structure dans CloudWatch Logs Insights qui vous permet d'accéder aux attributs et de les utiliser pour les requêtes.

Exemple : pour obtenir une carte ou une liste

`jsonParse`À utiliser pour analyser un champ qui est une chaîne JSON dans une carte ou une liste.

```
fields jsonParse(@message) as json_message
```

Exemple : pour accéder aux attributs

Utilisez l'opérateur d'accès par points (map.attribute) pour accéder aux éléments d'une carte. Si un attribut d'une carte contient des caractères spéciaux, utilisez des backticks pour placer le nom de l'attribut (map.attributes). `special.char`).

```
fields jsonParse(@message) as json_message
| stats count() by json_message.status_code
```

Utilisez l'opérateur d'accès entre crochets (list [index]) pour récupérer un élément à une position spécifique dans la liste.

```
fields jsonParse(@message) as json_message
| filter json_message.users[1].action = "PutData"
```

Enveloppez les caractères spéciaux en backticks (``) lorsque des caractères spéciaux sont présents dans le nom de la clé.

```
fields jsonParse(@message) as json_message
| filter json_message.`user.id` = "123"
```

Exemple : résultats vides

Les cartes et les listes sont traitées comme nulles pour les fonctions de chaîne, de nombre et de date.

```
fields jsonParse(@message) as json_message
| display toupper(json_message)
```

La comparaison de la carte et de la liste avec n'importe quel autre champ aboutit à`false`.

> **Note**
>
> L'utilisation de la carte et de la liste dans `deduppattern`,`sort`, et `stats` n'est pas prise en charge.

Fonctions Datetime

Fonctions Datetime

Utilisez les fonctions de date et heure dans les commandes `fields` et `filter` et en tant qu'arguments pour d'autres fonctions. Utilisez ces fonctions pour créer des compartiments de temps pour les requêtes avec des fonctions de regroupement. Utilisez des périodes composées d'un nombre et de l'un des éléments suivants :

- `ms`pendant des millisecondes
- `s`pendant quelques secondes
- `m`pendant quelques minutes
- `h`pendant des heures

Par exemple, `10m` correspond à 10 minutes et `1h` correspond à une heure.

> **Note**
>
> Utilisez l'unité de temps la plus appropriée pour votre fonction datetime. CloudWatch Logs plafonne votre demande en fonction de l'unité de temps que vous avez choisie. Par exemple, il plafonne à 60 la valeur maximale pour toute demande utilisants. Ainsi, si vous le spécifiez`bin(300s)`, CloudWatch Logs l'implémente en fait sous la forme de 60 secondes, car 60 est le nombre de secondes dans une minute. CloudWatch Logs n'utilisera donc pas un nombre supérieur à 60 avecs. Pour créer un bucket de 5 minutes, utilisez `bin(5m)` plutôt. Le plafond pour `ms` est de 1000, les capuchons pour `s` et `m` sont de 60, et le plafond pour `h` est de 24.

Le tableau suivant contient une liste des différentes fonctions de date et heure que vous pouvez utiliser dans des commandes de requête. Le tableau répertorie le type de résultat de chaque fonction et contient une description de chaque fonction.

> **Tip**
>
> Lorsque vous créez une commande de requête, vous pouvez utiliser le sélecteur d'intervalle pour sélectionner une période de temps à interroger. Par exemple, vous pouvez définir une période de temps entre des intervalles de 5 à 30 minutes, des intervalles de 1, 3 et 12 heures ou une période personnalisée. Vous pouvez également définir des périodes entre des dates spécifiques.

Fonction	Type de résultat	Description
`bin(period: Period)`	Horodatage	Arrondit la valeur de `@timestamp` à la période donnée, puis la tronque. Par exemple, `bin(5m)` arrondit la valeur de `@timestamp` aux 5 minutes les plus proches. Vous pouvez l'utiliser pour regrouper plusieurs entrées de journal dans une requête. L'exemple suivant indique le nombre d'exceptions par heure : `filter @message like /Exception/` `	stats count(*) as exceptionCount` ` by bin(1h)` `	sort exceptionCount desc`  les unités de temps et les abréviations suivantes sont prises en charge par la fonction `bin`. Pour toutes les unités et abréviations qui incluent plus d'un caractère, l'ajout de `s` au pluriel est pris en charge. Donc, les deux `hr` et `hrs` travaillent pour spécifier les heures.  • `millisecond` `ms` `msec` • `second` `s` `sec` • `minute` `m` `min`

Fonction	Type de résultat	Description
		• `hour` `h` `hr` • `day` `d` • `week` `w` • `month` `mo` `mon` • `quarter` `q` `qtr` • `year` `y` `yr`
`datefloor(timestamp: Timestamp, period: Period)`	Horodatage	Tronque l'horodatage pour la période donnée. Par exemple, `datefloor(@timestamp, 1h)` tronque toutes les valeurs de `@timestamp` vers la valeur la plus basse de l'heure.
`dateceil(timestamp : Timestamp, period: Period)`	Horodatage	Arrondit l'horodatage pour la période donnée, puis la tronque. Par exemple, `dateceil( @timestamp, 1h)` tronque toutes les valeurs de `@timestamp` vers la valeur la plus élevée de l'heure.
`fromMillis(fieldName: number)`	Horodatage	Interprète le champ en entrée comme le nombre de millisecondes depuis l'époque Unix et le convertit en horodatage.
`toMillis(fieldName: Timestamp)`	nombre	Convertit l'horodatage trouvé dans le champ nommé en un nombre représentant les millisecondes depuis l'époque Unix. Par exemple, `toMillis(@timestamp)` convertit l'horodatage `2022-01-14T13:18:031.000-08:00` à `1642195111000` .

Fonction	Type de résultat	Description

`now()`	nombre	Renvoie l'heure à laquelle le traitement de la requête a été lancé, en secondes d'époque. Cette fonction ne prend aucun argument.
		
		Vous pouvez l'utiliser pour filtrer les résultats de votre requête en fonction de l'heure actuelle.
		
		Par exemple, la requête suivante renvoie toutes les erreurs 4xx des deux dernières heures :
		
		`parse @message "Status Code: *;" as statusCode\n`
		`| filter statusCode >= 400 and statusCode <= 499  \n`
		`| filter toMillis(@timestamp) >= (now() * 1000 - 7200000)`
		
		L'exemple suivant renvoie toutes les entrées du journal des cinq dernières heures contenant le mot `error` ou `failure`
		
		`fields @timestamp, @message`
		`| filter @message like /(?i)(error|failure)/`
		`| filter toMillis(@timestamp) >= (now() * 1000 - 18000000)`

> **Note**
>
> À l'heure actuelle, CloudWatch Logs Insights ne prend pas en charge le filtrage des journaux avec des horodatages lisibles par l'homme.

Fonctions générales

Fonctions générales

Utilisez des fonctions générales dans les commandes `fields` et `filter` et en tant qu'arguments pour d'autres fonctions.

Fonction	Type de résultat	Description
`ispresent(fieldName: LogField)`	Booléen	Renvoie `true` si le champ existe
`coalesce(fieldName: LogField, ...fieldNames: LogField[ ])`	LogField	Renvoie la première valeur non nulle de la liste

Fonctions JSON

Fonctions JSON

Utilisez les fonctions JSON dans les `filter` commandes `fields` et comme arguments pour d'autres fonctions.

Fonction	Type de résultat	Description
`jsonParse(fieldName: string)`	Carte | Liste | Vide	Renvoie une carte ou une liste lorsque l'entrée est une représentation sous forme de chaîne d'un objet JSON ou d'un tableau JSON. Renvoie une valeur vide, si l'entrée ne fait pas partie de la représent ation.

Fonction	Type de résultat	Description
`jsonStringify(fieldName: Map | List)`	Chaîne	Renvoie une chaîne JSON à partir des données d'une carte ou d'une liste.

Fonctions de chaîne d'adresse IP

Fonctions de chaîne d'adresse IP

Utilisez les fonctions de chaîne d'adresse IP dans les commandes `filter` et `fields` et en tant qu'arguments pour d'autres fonctions.

Fonction	Type de résultat	Description
`isValidIp(fieldName: string)`	booléen	Renvoie `true` si le champ est une IPv6 adresse IPv4 ou une adresse valide.
`isValidIpV4(fieldName: string)`	boolean	Renvoie `true` si le champ est une IPv4 adresse valide.
`isValidIpV6(fieldName: string)`	boolean	Renvoie `true` si le champ est une IPv6 adresse valide.
`isIpInSubnet(fieldName: string, subnet: string)`	boolean	Renvoie `true` si le champ est une IPv6 adresse valide IPv4 dans le sous-réseau v4 ou v6 spécifié. Lorsque vous spécifiez le sous-réseau, utilisez la notation CIDR telle que `192.0.2.0/24` ou `2001:db8::/32` , où `192.0.2.0` ou `2001:db8::` est le début du bloc d'adresse CIDR.
`isIpv4InSubnet(fieldName: string, subnet: string)`	boolean	Renvoie `true` si le champ est une IPv4 adresse valide dans le sous-réseau v4 spécifié. Lorsque vous spécifiez le sous-réseau, utilisez la notation CIDR telle que `192.0.2.0/24` où

Fonction	Type de résultat	Description
		`192.0.2.0` est le début du bloc d'adresse CIDR.
`isIpv6InSubnet(fieldName: string, subnet: string)`	boolean	Renvoie `true` si le champ est une IPv6 adresse valide dans le sous-réseau v6 spécifié. Lorsque vous spécifiez le sous-réseau, utilisez la notation CIDR telle que `2001:db8::/32` où `2001:db8::` est le début du bloc d'adresse CIDR.

Fonctions de chaîne

Fonctions de chaîne

Utilisez des fonctions de chaîne dans les commandes `fields` et `filter` et en tant qu'arguments pour d'autres fonctions.

Fonction	Type de résultat	Description
`isempty(fieldName: string)`	Nombre	Renvoie 1 si le champ est manquant ou est une chaîne vide.
`isblank(fieldName: string)`	Nombre	Renvoie 1 si le champ est manquant, est une chaîne vide ou contient uniquement un espace.
`concat(str: string, ...strings: string[])`	chaîne	Concatène les chaînes.
`ltrim(str: string)` `ltrim(str: string, trimChars: string)`	chaîne	Si la fonction ne possède pas de deuxième argument, elle supprime les espaces blancs

Fonction	Type de résultat	Description
		à gauche de la chaîne. Si la fonction possède un deuxième argument de chaîne, elle ne supprime pas l'espace blanc. Au lieu de cela, elle supprime les caractères de `trimChars` à gauche de `str`. Par exemple, `ltrim("xy ZxyfooxyZ","xyZ")` renvoie `"fooxyZ"`.

`rtrim(str: string)`
 	
`rtrim(str: string, trimChars: string)`
	chaîne	Si la fonction ne possède pas de deuxième argument, elle supprime les espaces blancs à droite de la chaîne. Si la fonction possède un deuxième argument de chaîne, elle ne supprime pas l'espace blanc. Au lieu de cela, elle supprime les caractères de `trimChars` à droite de `str`. Par exemple, `rtrim("xy ZfooxyxyZ","xyZ")` renvoie `"xyZfoo"`.

Fonction	Type de résultat	Description

`trim(str: string)`
 	
`trim(str: string, trimChars: string)`
	chaîne	Si la fonction ne possède pas de deuxième argument, elle supprime les espaces blancs aux deux extrémités de la chaîne. Si la fonction possède un deuxième argument de chaîne, elle ne supprime pas l'espace blanc. Au lieu de cela, il supprime les caractère s de `trimChars` des deux côtés de `str`. Par exemple, `trim("xyZxyfooxyxyZ","xyZ")` renvoie `"foo"`.

`strlen(str: string)`	nombre	Renvoie la longueur de la chaîne en points de code Unicode.
`toupper(str: string)`	chaîne	Convertit la chaîne en majuscules.
`tolower(str: string)`	chaîne	Convertit la chaîne en minuscules.

Fonction	Type de résultat	Description

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	chaîne	Renvoie une sous-chaîne à partir de l'index spécifié par l'argument de nombre à la fin de la chaîne. Si la fonction comporte un second argument de nombre, elle comporte la longueur de la sous-chaîne à récupérer. Par exemple, `substr("xyZfooxyZ",3, 3)` renvoie `"foo"`.

`replace(fieldName: string, searchValue: string, replaceValue: string)`	chaîne	Remplace toutes les instances de `searchValue` dans `fieldName: string` par `replaceValue` . Par exemple, la fonction `replace(logGroup,"smoke_test","Smoke")` recherche les événements du journal où le champ `logGroup` contient la valeur de chaîne `smoke_test` et remplace la valeur par la chaîne `Smoke`.
`strcontains(str: string, searchValue: string)`	nombre	Renvoie 1 si `str` contient `searchValue` et 0 dans le cas contraire.

Champs contenant des caractères spéciaux

Si un champ contient des caractères non alphanumériques autres que le `@` symbole ou le point (`.`), vous devez l'entourer de caractères anticochés (). ` Par exemple, le champ de journal `foo-bar` doit

être placé entre accents graves (`foo-bar`), car il contient un caractère non alphanumérique, le trait d'union (-).

Utilisation d'alias et de commentaires dans les requêtes

Créez des requêtes contenant des alias. Utilisez des alias pour renommer des champs de journal ou lors de l'extraction de valeurs dans des champs. Utilisez le mot-clé `as` pour attribuer un champ de journal ou obtenir un alias. Vous pouvez utiliser plusieurs alias dans une requête. Vous pouvez utiliser des alias dans les commandes suivantes :

- `fields`
- `parse`
- `sort`
- `stats`

Les exemples suivants montrent comment créer des requêtes contenant des alias.

Exemple

La requête contient un alias dans la commande `fields`.

```
fields @timestamp, @message, accountId as ID
| sort @timestamp desc
| limit 20
```

La requête renvoie les valeurs des champs `@timestamp`, `@message` et `accountId`. Les résultats sont triés dans l'ordre décroissant et limités à 20. Les valeurs pour `accountId` sont répertoriés sous l'alias `ID`.

Exemple

La requête contient des alias dans les commandes `sort` et `stats`.

```
stats count(*) by duration as time
| sort time desc
```

La requête compte le nombre de fois que le champ `duration` apparaît dans le groupe de journaux et trie les résultats dans l'ordre décroissant. Les valeurs pour `duration` sont répertoriés sous l'alias `time`.

Utilisation de commentaires

CloudWatch Logs Insights prend en charge les commentaires dans les requêtes. Utilisez le caractère dièse (#) pour afficher les commentaires. Vous pouvez utiliser des commentaires pour ignorer les lignes des requêtes ou des requêtes de documents.

Exemple : requête

Lorsque la requête suivante est exécutée, la deuxième ligne est ignorée.

```
fields @timestamp, @message, accountId
# | filter accountId not like "7983124201998"
| sort @timestamp desc
| limit 20
```

Commencez avec Logs Insights QL : tutoriels sur les requêtes

Les sections suivantes incluent des exemples de didacticiels sur les requêtes pour vous aider à démarrer avec Logs Insights QL.

Rubriques

- Didacticiel : Exécution et modification d'un exemple de requête
- Didacticiel : Exécuter une requête avec une fonction d'agrégation
- Didacticiel : Exécuter une requête qui génère une visualisation groupée par champs de journal
- Didacticiel : Exécuter une requête qui produit des séries temporelles

Didacticiel : Exécution et modification d'un exemple de requête

Le didacticiel suivant vous aide à démarrer avec CloudWatch Logs Insights. Vous exécutez un exemple de requête dans Logs Insights QL, puis vous découvrez comment le modifier et le réexécuter.

Pour exécuter une requête, les journaux doivent déjà être stockés dans CloudWatch Logs. Si vous utilisez déjà les CloudWatch journaux et que vous avez configuré des groupes de journaux et des flux de journaux, vous êtes prêt à commencer. Il se peut également que vous disposiez déjà de journaux si vous utilisez des services tels qu' AWS CloudTrail Amazon Route 53 ou Amazon VPC et que vous avez configuré les journaux de ces services pour qu'ils soient accessibles dans Logs. CloudWatch

Pour plus d'informations sur l'envoi de CloudWatch journaux à Logs, consultezCommencer à utiliser CloudWatch Logs.

Les requêtes dans CloudWatch Logs Insights renvoient soit un ensemble de champs provenant des événements du journal, soit le résultat d'une agrégation mathématique ou d'une autre opération effectuée sur les événements du journal. Ce didacticiel illustre une requête qui renvoie une liste d'événements du journal.

Exécution d'un exemple de requête

Pour exécuter un exemple de requête CloudWatch Logs Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.

   Sur la page Logs Insights, l'éditeur de requêtes contient une requête par défaut dans Logs Insights QL qui renvoie les 20 événements de journal les plus récents.
3. Dans la liste déroulante Sélectionner les groupes de journaux, choisissez un ou plusieurs groupes de journaux à interroger.

   S'il s'agit d'un compte de surveillance dans le CloudWatch domaine de l'observabilité entre comptes, vous pouvez sélectionner des groupes de journaux dans les comptes sources ainsi que dans le compte de surveillance. Une seule requête peut interroger les journaux de différents comptes à la fois.

   Vous pouvez filtrer les groupes de journaux par nom de groupe de journaux, ID de compte ou étiquette de compte.

   Lorsque vous sélectionnez un groupe de journaux dans la classe de CloudWatch journaux standard, Logs Insights détecte automatiquement les champs de données du groupe. Pour consulter les champs découverts, sélectionnez le menu Champs en haut à droite de la page.

   > **Note**
   >
   > Les champs découverts ne sont pris en charge que pour les groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

4. (Facultatif) Utilisez le sélecteur d'intervalle de temps pour sélectionner une période à interroger.

Vous pouvez choisir entre des intervalles de 5 à 30 minutes, des intervalles de 1, 3 et 12 heures ou une période personnalisée.

5. Choisissez Run (Exécuter) pour afficher les résultats.

   Pour ce didacticiel, les résultats incluent les 20 événements du journal les plus récents.

   CloudWatch Logs affiche un graphique à barres des événements du groupe de journaux au fil du temps. Ce graphique à barres montre non seulement les événements dans le tableau, mais également la distribution des événements dans le groupe de journaux qui correspondent à la requête et à la période.

6. Pour consulter tous les champs d'un événement du journal renvoyé, choisissez l'icône de liste déroulante triangulaire à gauche de l'événement numéroté.

Modification de l'exemple de requête

Dans ce didacticiel, vous modifiez l'exemple de requête pour afficher les 50 événements du journal les plus récents.

Si vous n'avez pas encore exécuté le didacticiel précédent, faites-le maintenant. Ce didacticiel commence au moment où le didacticiel précédent se termine.

> **Note**
>
> Certains exemples de requêtes fournis avec CloudWatch Logs Insights utilisent des `tail` commandes `head` ou à la place `delimit`. Ces commandes sont considérées comme obsolètes et ont été remplacées par `limit`. Utilisez `limit` au lieu de `head` ou `tail` dans toutes les requêtes que vous écrivez.

Pour modifier l'exemple de requête CloudWatch Logs Insights

1. Dans l'éditeur de requêtes, remplacez 20 par 50, puis choisissez Exécuter.

   Les résultats de la nouvelle requête s'affichent. Si le groupe de journaux dispose de suffisamment de données dans la plage de temps par défaut, 50 journaux d'évènements sont désormais répertoriés.

2. (Facultatif) Vous pouvez enregistrer les requêtes que vous avez créées. Pour enregistrer cette requête, choisissez Enregistrer. Pour plus d'informations, consultez Enregistrez et réexécutez les requêtes CloudWatch Logs Insights.

Ajout d'une commande de filtre à l'exemple de requête

Ce didacticiel explique comment apporter une modification plus puissante à la requête dans l'éditeur de requête. Dans ce didacticiel, vous filtrez les résultats de la requête précédente basée sur un champ dans les événements du journal récupérés.

Si vous n'avez pas encore exécuté les didacticiels précédents, faites-le maintenant. Ce didacticiel commence au moment où le didacticiel précédent se termine.

Pour ajouter une commande de filtre à la requête précédente

1. Choisissez un champ à filtrer. Pour voir les champs les plus courants détectés par CloudWatch Logs dans les événements du journal contenus dans les groupes de journaux sélectionnés au cours des 15 dernières minutes, ainsi que le pourcentage de ces événements de journal dans lesquels chaque champ apparaît, sélectionnez Champs sur le côté droit de la page.

   Pour afficher les champs contenus dans un événement de journal spécifique, choisissez l'icône située à gauche de cette ligne.

   Le champ awsRegion peut s'afficher dans votre évènement de journal en fonction de la nature des événements qui se trouvent dans vos journaux. Pour le reste de ce didacticiel, nous utiliserons awsRegion en tant que champ de filtre, mais vous pouvez utiliser un autre champ si celui-ci n'est pas disponible.
2. Dans la zone de l'éditeur de requête, placez votre curseur après 50 et appuyez sur Entrée.
3. Sur la nouvelle ligne, commencez par entrer une barre verticale (« | ») et un espace. Les commandes d'une requête CloudWatch Logs Insights doivent être séparées par le caractère pipe.
4. Saisissez **`filter awsRegion="us-east-1"`**.
5. Cliquez sur Exécuter.

   La requête s'exécute à nouveau, et affiche désormais les 50 résultats les plus récents qui correspondent au nouveau filtre.

Si vous avez filtré sur un autre champ et obtenu un résultat d'erreur, il se peut que vous ayez besoin d'utiliser le nom du champ. Si le nom du champ inclut des caractères non alphanumériques, vous devez placer des guillemet inversés (`) avant et après le nom du champ (par exemple, **`error-code`="102"**).

Vous devez utiliser des guillemets inversés pour les noms de champs qui contiennent des caractères non alphanumériques, mais pas pour les valeurs. Les valeurs sont toujours entre guillemets (").

Logs Insights QL inclut de puissantes fonctionnalités de requête, notamment plusieurs commandes et la prise en charge des expressions régulières, des opérations mathématiques et statistiques. Pour de plus amples informations, veuillez consulter CloudWatch Syntaxe des requêtes en langage Logs Insights.

Didacticiel : Exécuter une requête avec une fonction d'agrégation

Vous pouvez utiliser des fonctions d'agrégation avec la commande `stats` et en tant qu'arguments pour d'autres fonctions. Dans ce didacticiel, vous exécutez une commande de requête qui compte le nombre d'événements du journal contenant un champ spécifié. La commande de requête renvoie un décompte total regroupé en fonction de la(des) valeur(s) du champ spécifié. Pour plus d'informations sur les fonctions d'agrégation, consultez la section Opérations et fonctions prises en charge dans le guide de l'utilisateur Amazon CloudWatch Logs.

Exécution d'une requête avec une fonction d'agrégation

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Vérifiez que l'onglet Logs Insights QL est sélectionné.
4. Dans la liste déroulante Sélectionner les groupes de journaux, choisissez un ou plusieurs groupes de journaux à interroger.

   S'il s'agit d'un compte de surveillance dans le CloudWatch domaine de l'observabilité entre comptes, vous pouvez sélectionner des groupes de journaux dans les comptes sources ainsi que dans le compte de surveillance. Une seule requête peut interroger les journaux de différents comptes à la fois.

Vous pouvez filtrer les groupes de journaux par nom de groupe de journaux, ID de compte ou étiquette de compte.

Lorsque vous sélectionnez un groupe de CloudWatch journaux, Logs Insights détecte automatiquement les champs de données du groupe de journaux s'il s'agit d'un groupe de journaux de classe Standard. Pour consulter les champs découverts, sélectionnez le menu Champs en haut à droite de la page.

5. Supprimez la requête par défaut dans l'éditeur de requête et saisissez la commande suivante :

```
stats count(*) by fieldName
```

6. *fieldName*Remplacez-le par un champ découvert dans le menu Champs.

   Le menu Champs se trouve en haut à droite de la page et affiche tous les champs découverts détectés par CloudWatch Logs Insights dans votre groupe de journaux.

7. Choisissez Run (Exécuter) pour consulter les résultats de requête.

   Les résultats de requête indiquent le nombre de registres dans votre groupe de journaux qui correspondent à la commande de requête et le décompte total regroupé en fonction de la(des) valeur(s) du champ spécifié.

Didacticiel : Exécuter une requête qui génère une visualisation groupée par champs de journal

Lorsque vous exécutez une requête qui utilise la fonction `stats` pour grouper les résultats renvoyés par les valeurs d'un ou de plusieurs champs dans les entrées du journal, vous pouvez afficher les résultats sous la forme d'un graphique à barres, d'un graphique circulaire, d'un graphique linéaire ou d'un graphique en aires empilées. Cela vous permet de visualiser plus efficacement les tendances dans vos journaux.

Pour exécuter une requête pour générer une visualisation

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Dans la liste déroulante Select log group(s) (Sélectionner groupe(s) de journaux), choisissez un ou plusieurs groupes de journaux à interroger.

   S'il s'agit d'un compte de surveillance dans le CloudWatch domaine de l'observabilité entre comptes, vous pouvez sélectionner des groupes de journaux dans les comptes sources ainsi

que dans le compte de surveillance. Une seule requête peut interroger les journaux de différents comptes à la fois.

Vous pouvez filtrer les groupes de journaux par nom de groupe de journaux, ID de compte ou étiquette de compte.

4. Dans l'éditeur de requête, supprimez le contenu actuel, saisissez la fonction `stats` suivante, puis choisissez Exécuter la requête.

```
stats count(*) by @logStream
    | limit 100
```

Les résultats indiquent le nombre d'événements de journal dans le groupe de journaux pour chaque flux de journaux. Les résultats sont limités à seulement 100 lignes.

5. Choisissez l'onglet Visualisation.
6. Sélectionnez la flèche en regard de Ligne, puis choisissez Barre.

Le graphique à barres apparaît, affichant une barre pour chaque flux de journaux du groupe de journaux.

Didacticiel : Exécuter une requête qui produit des séries temporelles

Lorsque vous exécutez une requête qui utilise la fonction `bin()` pour regrouper les résultats renvoyés selon une période de temps, vous pouvez afficher les résultats sous forme de graphique linéaire, de graphique en aires empilées, de graphique circulaire ou de diagramme à barres. Cela vous aide à visualiser plus efficacement les tendances des événements du journal au fil du temps.

Pour exécuter une requête pour générer une visualisation

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Vérifiez que l'onglet Logs Insights QL est sélectionné.
4. Dans la liste déroulante Sélectionner les groupes de journaux, choisissez un ou plusieurs groupes de journaux à interroger.

S'il s'agit d'un compte de surveillance dans le CloudWatch domaine de l'observabilité entre comptes, vous pouvez sélectionner des groupes de journaux dans les comptes sources ainsi

que dans le compte de surveillance. Une seule requête peut interroger les journaux de différents comptes à la fois.

Vous pouvez filtrer les groupes de journaux par nom de groupe de journaux, ID de compte ou étiquette de compte.

5. Dans l'éditeur de requête, supprimez le contenu actuel, saisissez la fonction `stats` suivante, puis choisissez Exécuter la requête.

```
stats count(*) by bin(30s)
```

Les résultats indiquent le nombre d'événements du groupe de journaux reçus par CloudWatch Logs pour chaque période de 30 secondes.

6. Choisissez l'onglet Visualisation.

Les résultats sont affichés sous forme de graphique linéaire. Pour basculer vers un graphique à barres, un graphique circulaire ou un graphique en aires empilées, choisissez la flèche en regard de Ligne en haut à gauche du graphique.

Exemples de requêtes

Cette section contient une liste de commandes de requête générales et utiles que vous pouvez exécuter dans la CloudWatchconsole. Pour plus d'informations sur l'exécution d'une commande de requête, consultez Tutoriel : Exécuter et modifier un exemple de requête dans le guide de l'utilisateur Amazon CloudWatch Logs.

Pour plus d'informations sur la syntaxe des requêtes, consultezCloudWatch Syntaxe des requêtes en langage Logs Insights.

Rubriques

- Requêtes générales
- Requêtes pour les journaux Lambda
- Requêtes pour les journaux de flux Amazon VPC
- Requêtes pour les journaux Route 53
- Requêtes pour les CloudTrail journaux
- Requêtes pour Amazon API Gateway
- Requêtes pour passerelle NAT

- Requêtes pour les journaux du serveur Apache
- Requêtes pour Amazon EventBridge
- Exemples de la commande d'analyse.

Requêtes générales

Rechercher les 25 derniers événements ajoutés au journal.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

Obtenir une liste du nombre d'exceptions par heure.

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

Obtenir une liste des événements de journal qui ne sont pas des exceptions.

```
fields @message | filter @message not like /Exception/
```

Obtenir l'événement du journal le plus récent pour chaque valeur unique du champ **`server`**.

```
fields @timestamp, server, severity, message
| sort @timestamp asc
| dedup server
```

Obtenir l'événement du journal le plus récent pour chaque valeur unique du champ **`server`** pour chaque type de **`severity`**.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server, severity
```

Requêtes pour les journaux Lambda

Déterminer la quantité de mémoire surallouée.

```
filter @type = "REPORT"
    | stats max(@memorySize / 1000 / 1000) as provisonedMemoryMB,
```

```
        min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

Créer un rapport de latence.

```
filter @type = "REPORT" |
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

Rechercher les invocations de fonctions lentes et éliminer les requêtes dupliquées qui peuvent résulter de nouvelles tentatives ou d'un code côté client. Dans cette requête, la valeur **`@duration`** est exprimée en millisecondes.

```
fields @timestamp, @requestId, @message, @logStream
| filter @type = "REPORT" and @duration > 1000
| sort @timestamp desc
| dedup @requestId
| limit 20
```

Requêtes pour les journaux de flux Amazon VPC

Rechercher les 15 premiers transferts de paquets entre les hôtes :

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

Trouver les 15 premiers transferts d'octets pour des hôtes sur un sous-réseau donné.

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

Rechercher les adresses IP qui utilisent UDP comme protocole de transfert de données.

```
filter protocol=17 | stats count(*) by srcAddr
```

Rechercher les adresses IP pour lesquelles des enregistrements de flux ont été ignorés durant la fenêtre de capture.

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

Trouver un enregistrement unique pour chaque connexion, afin de résoudre les problèmes de connectivité réseau.

```
fields @timestamp, srcAddr, dstAddr, srcPort, dstPort, protocol, bytes
| filter logStream = 'vpc-flow-logs' and interfaceId = 'eni-0123456789abcdef0'
| sort @timestamp desc
| dedup srcAddr, dstAddr, srcPort, dstPort, protocol
| limit 20
```

Requêtes pour les journaux Route 53

Rechercher la distribution d'enregistrements par heure par type de requête.

```
stats count(*) by queryType, bin(1h)
```

Rechercher les 10 premiers résolveurs DNS avec le plus grand nombre de requêtes.

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

Rechercher le nombre d'enregistrements par domaine et sous-domaine où le serveur n'a pas pu résoudre la requête DNS.

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

Requêtes pour les CloudTrail journaux

Trouvez le nombre d'entrées de journal pour chaque service, type d'événement et AWS région.

```
stats count(*) by eventSource, eventName, awsRegion
```

Trouvez les EC2 hôtes Amazon qui ont été démarrés ou arrêtés dans une AWS région donnée.

```
filter (eventName="StartInstances" or eventName="StopInstances") and awsRegion="us-
east-2"
```

Trouvez les AWS régions, les noms d'utilisateur et les ARNs utilisateurs IAM nouvellement créés.

```
filter eventName="CreateUser"
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

Rechercher le nombre d'enregistrements où une exception s'est produite lors de l'appel de l'API **`UpdateTrail`**.

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

Rechercher les entrées de journal dans lesquelles TLS 1.0 ou 1.1 a été utilisé

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by userIdentity.accountId, recipientAccountId,
 eventSource, eventName, awsRegion, tlsDetails.tlsVersion, tlsDetails.cipherSuite,
 userAgent
| sort eventSource, eventName, awsRegion, tlsDetails.tlsVersion
```

Rechercher le nombre d'appels par service qui ont utilisé les versions TLS 1.0 ou 1.1

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by eventSource
| sort numOutdatedTlsCalls desc
```

Requêtes pour Amazon API Gateway

Trouver les 10 dernières erreurs 4XX

```
fields @timestamp, status, ip, path, httpMethod
| filter status>=400 and status<=499
| sort @timestamp desc
```

```
| limit 10
```

Identifiez les 10 Amazon API Gateway demandes les plus anciennes de votre groupe de journaux Amazon API Gateway d'accès

```
fields @timestamp, status, ip, path, httpMethod, responseLatency
| sort responseLatency desc
| limit 10
```

Renvoyer la liste des chemins d'API les plus populaires de votre groupe de journaux d'accès Amazon API Gateway

```
stats count(*) as requestCount by path
| sort requestCount desc
| limit 10
```

Créer un rapport de latence d'intégration pour votre groupe de journaux d'accès Amazon API Gateway

```
filter status=200
| stats avg(integrationLatency), max(integrationLatency),
min(integrationLatency) by bin(1m)
```

Requêtes pour passerelle NAT

Si vous constatez des coûts plus élevés que d'habitude sur votre AWS facture, vous pouvez utiliser CloudWatch Logs Insights pour trouver les meilleurs contributeurs. Pour plus d'informations sur les commandes de requête suivantes, consultez Comment puis-je trouver les principaux contributeurs au trafic via la passerelle NAT de mon VPC ? sur la page d'assistance AWS premium.

> **Note**
>
> Dans les commandes de requête suivantes, remplacez « x.x.x.x » par l'adresse IP privée de votre passerelle NAT et remplacez « y.y » par les deux premiers octets de votre plage CIDR VPC.

Rechercher les instances qui envoient le plus de trafic via votre passerelle NAT.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.')
```

```
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Déterminer le trafic qui va vers et en provenance des instances de vos passerelles NAT.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.') or (srcAddr like 'xxx.xx.xx.xx'
 and dstAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Déterminer les destinations Internet avec lesquelles les instances de votre VPC communiquent le plus souvent pour les chargements et les téléchargements.

For uploads (Pour chargements)

```
filter (srcAddr like 'x.x.x.x' and dstAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Pour les téléchargements

```
filter (dstAddr like 'x.x.x.x' and srcAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Requêtes pour les journaux du serveur Apache

Vous pouvez utiliser CloudWatch Logs Insights pour interroger les journaux du serveur Apache. Pour plus d'informations sur les requêtes suivantes, consultez Simplifier les journaux du serveur Apache avec CloudWatch Logs Insights sur le blog AWS Cloud Operations & Migrations.

Trouver les champs les plus pertinents, afin de pouvoir consulter vos journaux d'accès et vérifier le trafic dans le chemin /admin de votre application.

```
fields @timestamp, remoteIP, request, status, filename| sort @timestamp desc
| filter filename="/var/www/html/admin"
| limit 20
```

Trouver le nombre de requêtes GET uniques qui ont accédé à votre page principale avec le code d'état « 200 » (succès).

```
fields @timestamp, remoteIP, method, status
| filter status="200" and referrer= http://34.250.27.141/ and method= "GET"
| stats count_distinct(remoteIP) as UniqueVisits
| limit 10
```

Rechercher le nombre de fois où votre service Apache a redémarré.

```
fields @timestamp, function, process, message
| filter message like "resuming normal operations"
| sort @timestamp desc
| limit 20
```

Requêtes pour Amazon EventBridge

Obtenez le nombre d' EventBridge événements regroupés par type de détail de l'événement

```
fields @timestamp, @message
| stats count(*) as numberOfEvents by `detail-type`
| sort numberOfEvents desc
```

Exemples de la commande d'analyse.

Utilisez une expression glob pour extraire les champs **`@user`**, **`@method`** et **`@latency`** à partir du champ de journal **`@message`** et renvoyer la latence moyenne pour chaque combinaison unique de **`@method`** et **`@user`**.

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Utilisez une expression régulière pour extraire les champs **`@user2`**, **`@method2`** et **`@latency2`** à partir du champ du journal **`@message`** et renvoyer la latence moyenne pour chaque combinaison unique de **`@method2`** et **`@user2`**.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Extrait les champs **loggingTime**, **loggingType** et **loggingMessage**, filtre vers le bas pour journaliser les événements qui contiennent les chaînes **ERROR** ou **INFO**, puis affiche uniquement les champs **loggingMessage** et **loggingType** pour les événements qui contiennent une chaîne **ERROR**.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

Comparer (diff) avec les plages temporelles précédentes

Vous pouvez utiliser CloudWatch Logs Insights avec Logs Insights QL pour comparer l'évolution des événements de vos journaux au fil du temps. Vous pouvez comparer les événements du journal ingérés au cours d'une période récente avec les journaux de la période immédiatement précédente. Vous pouvez également effectuer une comparaison avec des périodes antérieures similaires. Cela peut vous aider à déterminer si une erreur dans vos journaux a été introduite récemment ou s'est déjà produite, et peut vous aider à identifier d'autres tendances.

Les requêtes de comparaison renvoient uniquement des modèles dans les résultats, et non des événements de journal bruts. Les modèles renvoyés vous aideront à voir rapidement les tendances et l'évolution des événements du journal au fil du temps. Après avoir exécuté une requête de comparaison et obtenu les résultats des modèles, vous pouvez consulter des exemples d'événements de log bruts correspondant aux modèles qui vous intéressent. Pour plus d'informations sur les modèles de journalisation, consultezAnalyse de modèles.

Lorsque vous exécutez une requête de comparaison, celle-ci est analysée par rapport à deux périodes différentes : la période de requête initiale que vous avez sélectionnée et la période de comparaison. La période de comparaison est toujours d'une durée égale à celle de votre période de requête initiale. Les intervalles de temps par défaut pour les comparaisons sont les suivants.

- Période précédente : compare avec la période immédiatement antérieure à la période de votre requête.
- Jour précédent : compare avec la période du jour précédant la période de votre requête.
- Semaine précédente : comparaison avec la période d'une semaine précédant la période de votre requête.
- Mois précédent : comparaison avec la période d'un mois précédant la période de votre requête.

> **Note**
>
> Les requêtes utilisant des comparaisons entraînent des frais similaires à ceux de l'exécution d'une seule requête CloudWatch Logs Insights sur une période combinée. Pour en savoir plus, consultez Tarification Amazon CloudWatch.

Pour exécuter une requête de comparaison

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Logs, Logs Insights.

   Une requête par défaut apparaît dans la zone de requête.
3. Vérifiez que l'onglet Logs Insights QL est sélectionné.
4. Conservez la requête par défaut ou entrez une autre requête.
5. Dans le menu déroulant Sélectionner un ou plusieurs groupes de journaux, choisissez un ou plusieurs groupes de journaux à interroger.
6. (Facultatif) Utilisez le sélecteur d'intervalle de temps pour sélectionner une période à interroger. La requête par défaut porte sur les données du journal de l'heure précédente.
7. Dans le sélecteur de plage horaire, choisissez Comparer. Choisissez ensuite la période précédente à laquelle vous souhaitez comparer les journaux d'origine, puis choisissez Appliquer.
8. Choisissez Exécuter la requête.

   Pour que la requête récupère les données de la période de comparaison, la `diff` commande est ajoutée à votre requête.
9. Cliquez sur l'onglet Motifs pour voir les résultats.

   Le tableau affiche les informations suivantes :

   - Chaque modèle, avec des parties variables du modèle remplacées par le symbole de jeton dynamique<*string*-*number*>. *string*Il s'agit d'une description du type de données que le jeton représente. *number*Indique où ce jeton apparaît dans le modèle, par rapport aux autres jetons dynamiques. Pour de plus amples informations, veuillez consulter Analyse de modèles.
   - Le nombre d'événements est le nombre d'événements du journal présentant ce modèle au cours de la période initiale plus récente.

- Le nombre d'événements différentiels est la différence entre le nombre d'événements de journal correspondants au cours de la période en cours et celui de la période de comparaison. Une différence positive signifie qu'il y a davantage d'événements de ce type au cours de la période actuelle.
- La description de la différence résume brièvement l'évolution de ce schéma entre la période en cours et la période de comparaison.
- Le type de gravité est la gravité probable des événements du journal présentant ce modèle, sur la base de mots trouvés dans les événements du journal`FATAL`, tels que`ERROR`, et`WARN`.

10. Pour examiner plus en détail l'un des modèles de la liste, choisissez l'icône de l'un des modèles dans la colonne Inspecter.

    Le volet Pattern inspect apparaît et affiche les informations suivantes :

    - Le motif. Sélectionnez un jeton dans le modèle pour analyser les valeurs de ce jeton.
    - Un histogramme indiquant le nombre d'occurrences du modèle sur la plage de temps demandée. Cela peut vous aider à identifier des tendances intéressantes, telles qu'une augmentation soudaine de l'occurrence d'un modèle.
    - L'onglet Échantillons de journal affiche quelques-uns des événements du journal correspondant au modèle sélectionné.
    - L'onglet Valeurs du jeton affiche les valeurs du jeton dynamique sélectionné, si vous en avez sélectionné un.

    > **Note**
    >
    > Un maximum de 10 valeurs de jeton est capturée pour chaque jeton. Le nombre de jetons peut ne pas être précis. CloudWatch Logs utilise un compteur probabiliste pour générer le nombre de jetons, et non la valeur absolue.

    - L'onglet Modèles associés affiche d'autres modèles qui se sont produits fréquemment à peu près au même moment que le modèle que vous inspectez. Par exemple, si le modèle d'un `ERROR` message était généralement accompagné d'un autre événement de journal marqué comme `INFO` contenant des détails supplémentaires, ce modèle est affiché ici.

Visualisation des données du journal dans des graphiques

Vous pouvez utiliser des visualisations telles que des diagrammes à barres, des graphiques linéaires et des graphiques à aires empilées pour identifier plus efficacement les modèles dans vos données de journal. CloudWatch Logs Insights génère des visualisations pour les requêtes qui utilisent la `stats` fonction et une ou plusieurs fonctions d'agrégation. Pour plus d'informations, veuillez consulter stats.

OpenSearch Langage de traitement canalisé (PPL)

Cette section contient une introduction de base à l'interrogation des CloudWatch journaux à l'aide de OpenSearch PPL. Avec PPL, vous pouvez récupérer, interroger et analyser des données à l'aide de commandes groupées, ce qui facilite la compréhension et la composition de requêtes complexes. Sa syntaxe est basée sur les canaux Unix et permet d'enchaîner les commandes pour transformer et traiter les données. Avec PPL, vous pouvez filtrer et agréger des données, et utiliser un ensemble complet de fonctions mathématiques, de chaîne, de date, conditionnelles et autres pour l'analyse.

Vous ne pouvez utiliser OpenSearch PPL que pour les requêtes de groupes de journaux dans la classe de journal standard.

> **Note**
>
> Pour plus d'informations sur toutes les commandes de requête OpenSearch PPL prises en charge dans les CloudWatch journaux et des informations détaillées sur la syntaxe et les restrictions, consultez la section Commandes PPL prises en charge dans le Guide du développeur OpenSearch de services.
> Pour plus d'informations sur les autres langages de requête, vous pouvez utiliser CloudWatch Logs Insights, OpenSearch Service SQL et CloudWatch Metrics Insights

Commande ou fonction	Exemple de requête	Description
fields	`fields field1, field2`	Affiche un ensemble de champs qui doivent être projetés.

Commande ou fonction	Exemple de requête	Description
join	`LEFT JOIN left=l, right=r on l.id = r.id` `join_right_lg` `| fields l.field_1, r.field_2`	Joint deux ensembles de données.
où	`where field1="success" | where field2 ! = "i-023fe0a90929d8822" | fields field3, field4, field5,field6 | head 1000`	Filtre les données en fonction des conditions que vous spécifiez.
stats	`stats count(), count(field1), min(field 1), max(field1), avg(field1) by field2 | head 1000`	Effectue des agrégations et des calculs
parse	`parse field1 ".*/(?<field2>[^/]+$)" | where field2 = "requestId" | fields field1, field2 | head 1000`	Extrait un modèle d'expression régulière (regex) d'une chaîne et affiche le modèle extrait. Le modèle extrait peut également être utilisé pour créer de nouveaux champs ou filtrer des données.

Commande ou fonction	Exemple de requête	Description
sort	`stats count(), count(field1), min(field1) as field1Alias, max(`field1`), avg(`field1`) by field2 | sort -field1Alias | head 1000`	Triez les résultats affichés par nom de champ. Utilisez le tri - FieldName pour trier par ordre décroissant.
eval	`eval field2 = field1 * 2 | fields field1, field2 | head 20`	Modifie ou traite la valeur d'un champ et la stocke dans un autre champ. Cela est utile pour modifier mathématiquement une colonne, appliquer des fonctions de chaîne à une colonne ou appliquer des fonctions de date à une colonne.
renommer	`rename field2 as field1 | fields field1;`	Renomme un ou plusieurs champs dans les résultats de recherche.

Commande ou fonction	Exemple de requête	Description
head	`fields `@message` | head 20`	Limite les résultats de requête affichés aux N premières lignes.
top	`top 2 field1 by field2`	Recherche les valeurs les plus fréquentes pour un champ.
dedup	`dedup field1 | fields field1, field2, field3`	Supprime les entrées dupliquées en fonction des champs que vous spécifiez.
rare	`rare field1 by field2`	Recherche les valeurs les moins fréquentes de tous les champs de la liste de champs.
sous-requête	`where field_1 IN [ search source= `subquery_lg` | fields field_2 ] | fields id, field_1`	Exécute des requêtes complexes et imbriquées dans vos instructions PPL.

Commande ou fonction	Exemple de requête	Description
ligne de tendance	`trendline sma(2, field1) as field1Alias`	Calcule les moyennes mobiles des champs.
Statistiques de l'événement	`eventstats sum(field1) by field2`	Enrichissez les données de vos événement s grâce à des statistiques récapitulatives calculées. Il analyse les champs spécifiés au sein de vos événements, calcule diverses mesures statistiq ues, puis ajoute ces résultats à chaque événement d'origine sous forme de nouveaux champs.

Commande ou fonction	Exemple de requête	Description
développer	`eval tags_array_string = json_extr act(`@message`, '$.tags')\| eval tags_array = json_array(json_ex tract(tags_string, '$[0]'), json_extr act(tags_string, '$[1]'))| expand tags_array as color_tags`	Décompose un champ contenant plusieurs valeurs en lignes distinctes, en créant une nouvelle ligne pour chaque valeur du champ spécifié.
fillnull	`fields `@timestamp`, error_code, status_code | fillnull using status_code = "UNKNOWN", error_code = "UNKNOWN"`	Remplit les champs nuls avec la valeur que vous fournissez. Il peut être utilisé dans un ou plusieurs domaines.
flatten	`eval metadata_struct = json_obje ct('size', json_extract(metadata_strin g, '$.size'), 'color', json_extr act(metadata_string, '$.color')) | flatten metadata_struct as (meta_size, meta_color)`	Aplatit un champ. Le champ doit être de ce type : `struct<?,?>` ou `array<struct<?,?>>` .
cidrmatch	`where cidrmatch(ip, '2003:db8::/32') | fields ip`	Vérifie si l'adresse IP spécifiée se situe dans la plage CIDR donnée.

Commande ou fonction	Exemple de requête	Description
résumé du champ	`where field1 != 200 | fieldsummary includefields= field1 nulls=true`	Calcule les statistiques de base pour chaque champ (nombre, nombre distinct, min, max, avg, stddev et moyenne).
grok	`grok email '.+@%{HOSTNAME:host}' | fields email, host`	Analyse un champ de texte à l'aide d'un motif grok et ajoute les résultats au résultat de recherche.
Fonctions de chaîne	`eval field1Len = LENGTH(field1) | fields field1Len`	Fonctions intégrées dans PPL qui peuvent manipuler et transformer des chaînes et des données de texte dans les requêtes PPL. Par exemple, convertir des majuscules, combiner des chaînes, extraire des parties et nettoyer du texte.

Commande ou fonction	Exemple de requête	Description
Fonctions date-heure	`eval newDate = ADDDATE(DATE('2020-08-26'), 1) | fields newDate`	Fonctions intégrées pour gérer et transformer les données de date et d'horodatage dans les requêtes PPL. Par exemple, date_add, date_format, datediff, date-sub, timestamp add, timestamp diff, current_timezone, utc_timestamp et current_date.
Fonctions de condition	`eval field2 = isnull(field1) | fields field2, field1, field3`	Fonctions intégrées qui vérifient les conditions de champ spécifiques et évaluent les expressions de manière conditionnelle. Par exemple, si le champ 1 est nul, renvoie le champ 2.

Commande ou fonction	Exemple de requête	Description
Fonctions mathématiques	`eval field2 = ACOS(field1) | fields field1`	Fonctions intégrées pour effectuer des calculs mathématiques et des transformations dans les requêtes PPL. Par exemple, abs (valeur absolue), round (arrondit les nombres), sqrt (racine carrée), pow (calcul de puissance) et ceil (arrondit au nombre entier le plus proche).
CryptoGraphic fonctions	`eval crypto = MD5(field)| head 1000`	Pour calculer le hachage d'un champ donné

Commande ou fonction	Exemple de requête	Description
Fonctions JSON	`eval valid_json = json('[1,2,3,{"f1":1,"f2":[5,6]},4]') | fields valid_json`	Fonctions intégrées pour gérer le JSON, notamment les tableaux, l'extraction et la validation. Par exemple, json_object, json_array, to_json_string, json_array_length, json_extract, json_keys et json_valid.

OpenSearch Langage de requête structuré (SQL)

Cette section contient une introduction de base à l'interrogation des CloudWatch journaux à l'aide de OpenSearch SQL. Il s'agit d'une option familière si vous avez l'habitude de travailler avec des bases de données relationnelles. OpenSearch SQL offre un sous-ensemble de fonctionnalités SQL, ce qui en fait un bon choix pour effectuer des requêtes ad hoc et des tâches d'analyse de données. Avec OpenSearch SQL, vous pouvez utiliser des commandes telles que SELECT, FROM, WHERE, GROUP BY, HAVING, ainsi que diverses autres commandes et fonctions SQL. Vous pouvez exécuter des tâches JOINs dans différents groupes de journaux, corréler les données entre les groupes de journaux à l'aide de sous-requêtes et utiliser le riche ensemble de fonctions JSON, mathématiques, de chaîne, conditionnelles et autres fonctions SQL pour effectuer une analyse approfondie des données de journal et de sécurité.

Vous ne pouvez utiliser OpenSearch SQL que pour les requêtes de groupes de journaux dans la classe de journal standard.

> **Note**
>
> Le tableau suivant répertorie les commandes et fonctions SQL prises en charge dans les CloudWatch journaux. Pour plus d'informations sur toutes les commandes OpenSearch SQL, y compris la syntaxe, consultez la section Commandes SQL prises en charge dans le Guide du développeur de OpenSearch services.
> Pour plus d'informations sur les autres langages de requête que vous pouvez utiliser, consultez CloudWatch Logs Insights, OpenSearch Service PPL et CloudWatch Metrics Insights.

Commandes SQL prises en charge

> **Note**
>
> Dans la colonne d'exemple de requête, remplacez le cas `<logGroup>` échéant en fonction de la source de données que vous interrogez.

Commande ou fonction	Exemple de requête	Description
SELECT	`SELECT \`@message\`, Operation FROM \`LogGroupA\` `	Affiche les valeurs projetées .
FROM	`SELECT \`@message\`, Operation FROM \`LogGroupA\` `	Clause intégrée qui spécifie les tables ou les vues sources à partir desquelles récupérer les données, prenant en charge différents types de

Commande ou fonction	Exemple de requête	Description
		jointures et de sous-requêtes.
WHERE	`SELECT * FROM `LogGroupA` WHERE Operation = 'x'`	Les filtres enregistrent les événement s en fonction des critères de champ fournis.
GROUP BY	`SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream``	Les groupes enregistrent les événements par catégorie et trouvent la moyenne en fonction des statistiques.
HAVING	`SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream` HAVING log_count > 100`	Filtre les résultats en fonction des conditions de regroupement.
ORDER BY	`SELECT * FROM `LogGroupA` ORDER BY `@timestamp` DESC`	Trie les résultats en fonction des champs de la clause ORDER BY. Vous pouvez trier par ordre décroissant ou croissant.

Commande ou fonction	Exemple de requête	Description
JOIN	`SELECT A.`@message`, B.`@timestamp`FROM `LogGroupA` as A INNER JOIN `LogGroupB` as B ON A.`requestId` = B.`requestId``	Joint les résultats de deux tables sur la base de champs communs. La jointure interne ou la jointure externe gauche doivent être spécifiées
LIMIT	`Select * from `LogGroupA` limit 10`	Limite les résultats de requête affichés aux N premières lignes.
Fonctions de chaîne	`SELECT upper(Operation) , lower(Operation), Operation FROM `LogGroupA``	Fonctions intégrées dans SQL permettant de manipuler et de transformer des chaînes et des données de texte dans des requêtes SQL. Par exemple, convertir des majuscules, combiner des chaînes, extraire des parties et nettoyer du texte.

Commande ou fonction	Exemple de requête	Description
Fonctions de date	`SELECT current_date() as today, date_add( current_date(), 30) as thirty_da ys_later, last_day(current_date()) as month_end FROM `LogGroupA` `	Fonctions intégrées pour gérer et transformer les données de date et d'horodat age dans les requêtes SQL. Par exemple, date_add, date_form at, datediff et current_date.
Fonctions conditionnelles	`SELECT Operation, IF(Error > 0, 'High', 'Low') as error_category FROM `LogGroup A`; `	Fonctions intégrées qui exécutent des actions en fonction de conditions spécifiées ou qui évaluent les expressio ns de manière conditionnelle. Par exemple, CASE et IF.

Commande ou fonction	Exemple de requête	Description
Fonctions d'agrégation	`SELECT AVG(bytes) as bytesWritten FROM `LogGroupA` `	Fonctions intégrées qui effectuent des calculs sur plusieurs lignes pour produire une seule valeur résumée. Par exemple, SUM, COUNT, AVG, MAX et MIN.
Fonctions JSON	`SELECT get_json_object(json_column, '$.name') as name FROM `LogGroupA` `	Fonctions intégrées pour analyser, extraire, modifier et interroger des données au format JSON dans des requêtes SQL (par exemple, from_json , to_json, get_json_object, json_tuple) permettant de manipuler les structures JSON dans les ensembles de données.

Commande ou fonction	Exemple de requête	Description
Fonctions de tableau	`SELECT scores, size(scores) as length, array_contains(scores, 90) as has_90 FROM `LogGroupA`;`	Fonctions intégrées permettant de travaille r avec des colonnes de type tableau dans les requêtes SQL, permettant des opérations telles que l'accès, la modificat ion et l'analyse de données de tableau (par exemple, size, explode, array_contains).

Commande ou fonction	Exemple de requête	Description
Fonctions de fenêtrage	`SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) as field2Rank FROM `LogGroupA`;`	Fonctions intégrées qui effectuent des calculs sur un ensemble spécifique de lignes liées à la ligne actuelle (fenêtre), permettant des opération s telles que le classement, les totaux cumulés et les moyennes mobiles. Par exemple, ROW_NUMBER, RANK, LAG et LEAD

Commande ou fonction	Exemple de requête	Description
Fonctions de conversion	`SELECT CAST('123' AS INT) as converted_number, CAST(123 AS STRING) as converted_string FROM `LogGroupA` `	Fonctions intégrées pour convertir les données d'un type à un autre dans les requêtes SQL, permettant des transformations de type de données et des conversions de format. Par exemple, CAST, TO_DATE, TO_TIMESTAMP et BINARY.

Commande ou fonction	Exemple de requête	Description
Fonctions de prédicat	`SELECT scores, size(scores) as length, array_contains(scores, 90) as has_90 FROM `LogGroupA`;`	Fonctions intégrées qui évaluent les conditions et renvoient des valeurs booléennes (vrai/faux) en fonction de critères ou de modèles spécifiés. Par exemple, IN, LIKE, BETWEEN, IS NULL et EXISTS.
Sélection nez plusieurs groupes de journaux	`SELECT lg1.field1, lg1.field2 from `logGroups(logGroupIdentifier: ['LogGroup1', 'LogGroup2'])` as lg1 where lg1.field3= "Success"`	Vous permet de spécifier plusieurs groupes de journaux dans une instruction SELECT

SQL pris en charge pour les multi-log-group requêtes

Pour faciliter l'interrogation de plusieurs groupes de journaux dans SQL, vous pouvez utiliser la `logGroups` commande. À l'aide de cette syntaxe, vous pouvez interroger plusieurs groupes de journaux en les spécifiant dans la commande FROM.

Syntaxe :

```
`logGroups(
```

```
    logGroupIdentifier: ['LogGroup1','LogGroup2', ...'LogGroupn']
)
```

Dans cette syntaxe, vous pouvez spécifier jusqu'à 50 groupes de journaux dans le `logGroupIdentifier` paramètre. Pour référencer des groupes de journaux dans un compte de surveillance, utilisez ARNs plutôt que des `LogGroup` noms.

Exemple de requête :

```
SELECT LG1.Column1, LG1.Column2 from `logGroups(
    logGroupIdentifier: ['LogGroup1', 'LogGroup2']
)` as LG1 WHERE LG1.Column1 = 'ABC'
```

La syntaxe suivante impliquant plusieurs groupes de journaux après l'`FROM`instruction n'est PAS prise en charge lors de l'interrogation CloudWatch des journaux.

```
SELECT Column1, Column2 FROM 'LogGroup1', 'LogGroup2', ...'LogGroupn'
WHERE Column1 = 'ABC'
```

Restrictions

Les restrictions suivantes s'appliquent lorsque vous utilisez OpenSearch SQL pour effectuer des requêtes dans CloudWatch Logs Insights.

- Vous ne pouvez inclure qu'un seul JOIN dans une instruction SELECT.
- Un seul niveau de sous-requêtes imbriquées est pris en charge.
- Les requêtes d'instruction multiples séparées par des points-virgules (;) ne sont pas prises en charge.
- Les requêtes contenant des noms de champs identiques mais différents uniquement au cas où (par exemple field1 et FIELD1) ne sont pas prises en charge.

  Par exemple, la requête suivante n'est pas prise en charge :

  ```
  Select AWSAccountId, AwsAccountId from LogGroup
  ```

  Cependant, la requête suivante est prise en charge car le nom du champ (`@logStream`) est identique dans les deux groupes de journaux :

```
Select a.`@logStream`, b.`@logStream` from Table A INNER Join Table B on a.id = b.id
```

- Les fonctions et expressions doivent agir sur les noms de champs et faire partie d'une instruction SELECT avec un groupe de journaux spécifié dans la clause FROM.

  Par exemple, cette requête n'est pas prise en charge :

  ```
  SELECT cos(10) FROM LogGroup
  ```

  Cette requête est prise en charge :

  ```
  SELECT cos(field1) FROM LogGroup
  ```

- Lorsque vous utilisez des commandes SQL ou PPL, entourez certains champs de backticks pour les interroger correctement. Les champs contenant des caractères spéciaux (non alphabétiques et non numériques) doivent être cochés. Par exemple, joignez `@messageOperation.Export`, et entre `Test::Field` backticks. Il n'est pas nécessaire de placer les champs avec des noms purement alphabétiques en backticks.

  Exemple de requête avec des champs simples :

  ```
  SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
  LIMIT 1000;
  ```

  Requête similaire avec backticks ajoutés :

  ```
  SELECT `@SessionToken`, `@Operation`, `@StartTime`  FROM `LogGroup-A` LIMIT 1000;
  ```

Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Logs Insights

CloudWatch Logs prend en charge une fonctionnalité de requête en langage naturel pour vous aider à générer et à mettre à jour des requêtes pour CloudWatch Logs Insights, OpenSearch Service PPL, OpenSearch Service SQL et CloudWatch Metrics Insights.

Grâce à cette fonctionnalité, vous pouvez poser des questions ou décrire les données des CloudWatch journaux que vous recherchez dans un langage clair. La fonctionnalité de langage

naturel génère une requête en fonction d'une invite que vous entrez et fournit une line-by-line explication du fonctionnement de la requête. Vous pouvez également mettre à jour votre requête pour examiner plus en détail vos données.

En fonction de votre environnement, vous pouvez saisir des messages tels que « Quelles sont les 100 principales adresses IP sources en octets transférés ? » et « Trouvez les 10 requêtes de fonction Lambda les plus lentes ».

> **Note**
>
> La fonction de requête en langage naturel est un service régional. Pour certaines régions, la fonctionnalité effectue des appels interrégionaux vers des régions des États-Unis d'Amérique afin de traiter les demandes de renseignements. Pour plus d'informations, consultez Amazon CloudWatch étend la prise en charge des régions pour le résumé des résultats de requêtes en langage naturel et la génération de requêtes.

Pour générer une requête CloudWatch Logs Insights avec cette fonctionnalité, ouvrez l'éditeur de requêtes CloudWatch Logs Insights, sélectionnez le groupe de journaux que vous souhaitez interroger, puis choisissez Generate query.

> **Important**
>
> Pour utiliser la fonctionnalité de requête en langage naturel, vous devez être connecté avec les politiques CloudWatchLogsFullAccess,, CloudWatchLogsReadOnlyAccessAdministratorAccess, ou ReadOnlyAccessIAM, ou vous devez disposer de l'`cloudwatch:GenerateQuery`autorisation.

Exemples de requêtes

Les exemples de cette section décrivent comment générer et mettre à jour des requêtes à l'aide de la fonctionnalité de langage naturel.

> **Note**
>
> Pour plus d'informations sur l'éditeur de requêtes et la syntaxe de CloudWatch Logs Insights, voir Syntaxe de requête CloudWatch Logs Insights.

Exemples : générer une requête en langage naturel

Pour générer une requête en langage naturel, saisissez une invite et choisissez Générer une nouvelle requête. Cet exemple montre les requêtes qui effectuent une recherche de base.

Invite

Voici un exemple d'invite qui indique à la fonctionnalité de rechercher les 10 appels de fonction Lambda les plus lents.

```
Find the 10 slowest requests
```

Requête

Voici la requête utilisant le langage de requête CloudWatch Logs Insights généré par la fonctionnalité de langage naturel en fonction de l'invite. Remarquez comment l'invite apparaît dans un commentaire avant la requête. Après la requête, vous pouvez lire une explication qui décrit son fonctionnement.

```
# Find the 10 slowest requests
fields @timestamp, @message, @duration
| sort @duration desc
| limit 10
# This query retrieves the timestamp, message and duration fields from the logs and
 sorts them in descending order by duration to find the 10 slowest requests.
```

> **Note**
>
> Pour désactiver l'affichage de votre invite et de l'explication du fonctionnement de la requête, utilisez l'icône en forme de roue dentée dans votre éditeur.

Invite

Pour générer une requête OpenSearch SQL, sélectionnez l'onglet OpenSearch SQL, puis ouvrez la boîte de dialogue du générateur de requêtes pour saisir votre invite en langage naturel. Voici un exemple d'invite qui utilise la fonctionnalité du langage naturel pour générer une requête OpenSearch SQL.

```
Give me the number of errors and exceptions per hour
```

Requête

Voici la requête SQL générée par cette invite que vous pouvez utiliser pour trouver le nombre d'erreurs et d'exceptions agrégées par heure :

```
SELECT DATE_FORMAT(`@timestamp`, 'yyyy-MM-dd HH') AS hour,
       COUNT(*) AS error_count
FROM `/aws/lambda/CloudWatchOdysseyQueryGen`
WHERE `@message` LIKE '%error%'
  OR `@message` LIKE '%exception%'
GROUP BY DATE_FORMAT(`@timestamp`, 'yyyy-MM-dd HH')
ORDER BY hour
```

Invite

Pour générer une requête OpenSearch PPL, sélectionnez l'onglet OpenSearch PPL, puis ouvrez la boîte de dialogue du générateur de requêtes pour saisir votre invite en langage naturel. Voici un exemple d'invite qui utilise la fonctionnalité du langage naturel pour générer une requête OpenSearch PPL.

```
Give me all unique exception messages
```

Requête

Voici la requête PPL générée par cette invite que vous pouvez utiliser pour rechercher les messages d'exception uniques dans vos journaux :

```
dedup @message
| fields @message
```

Exemple : mettre à jour une requête en langage naturel

Vous pouvez mettre à jour une requête en modifiant l'invite initiale, puis en choisissant Mettre à jour la requête.

Invite mise à jour

L'exemple suivant présente une version mise à jour de la requête précédente. Au lieu d'une invite qui recherche les 10 appels de fonction Lambda les plus lents, cette invite indique désormais à la

fonctionnalité de rechercher les 20 appels de fonction Lambda les plus lents et d'inclure une autre colonne pour les événements de journal supplémentaires.

```
Show top 20 slowest requests instead and display requestId as a column
```

Requête mise à jour

Voici un exemple de requête mise à jour à l'aide du langage de requête CloudWatch Logs Insights. Remarquez comment l'invite mise à jour apparaît dans un commentaire avant la requête mise à jour. Après la requête, vous pouvez lire une explication qui décrit comment la requête d'origine a été mise à jour.

```
# Show top 20 slowest requests instead and display requestId as a column
fields @timestamp, @message, @requestId, @duration
| sort @duration desc
| limit 20
# This query modifies the original query by replacing the @message field with the
 @requestId field and changing the limit from 10 to 20 to return the top 20 log events
 by duration instead of the top 10.
```

Refus d'utiliser vos données pour améliorer le service

Les données d'invite en langage naturel que vous fournissez pour entraîner le modèle d'IA et générer des requêtes pertinentes ne sont utilisées que pour fournir et maintenir votre service. Ces données peuvent être utilisées pour améliorer la qualité de CloudWatch Logs Insights. Votre confiance, la confidentialité et la sécurité de votre contenu constituent nos priorités N° 1. Pour plus d'informations, veuillez consulter les rubriques Conditions de service AWS et AWS responsible AI policy.

Vous pouvez refuser que votre contenu soit utilisé pour développer ou améliorer la qualité des requêtes en langage naturel en créant une politique de désinscription des services d'IA. Pour désactiver la collecte de données pour toutes les fonctionnalités de CloudWatch Logs AI, y compris la fonctionnalité de génération de requêtes, vous devez créer une politique de désinscription pour CloudWatch Logs. Pour plus d'informations, veuillez consulter la rubrique Politiques de désactivation des services IA dans le Guide de l'utilisateur AWS Organizations .

Journaux pris en charge et champs découverts

CloudWatch Logs Insights prend en charge différents types de journaux. Pour chaque journal envoyé à un groupe de journaux de classe Standard dans Amazon CloudWatch Logs, CloudWatch Logs Insights génère automatiquement cinq champs système :

- `@message` contient l'événement de journal non analysé brut. C'est l'équivalent du `message` champ dans InputLogevent.
- `@timestamp` contient l'horodatage de l'événement dans le champ `timestamp` de l'événement du journal. C'est l'équivalent du `timestamp` champ dans InputLogevent.
- `@ingestionTime`contient l'heure à laquelle CloudWatch Logs a reçu l'événement du journal.
- `@logStream` contient le nom du flux de journaux auquel l'événement de journal a été ajouté. Les flux de journaux regroupent les journaux à travers le même processus qui les a générés.
- `@log` est un identificateur de groupe de journaux sous la forme de *`account-id`*`:`*`log-group-name`*. Lors de la requête de plusieurs groupes de journaux, cela peut être utile d'identifier le groupe de journaux auquel appartient un événement particulier.
- `@entity`contient du JSON aplati lié aux entités pour la fonctionnalité de télémétrie associée à Explore.

  Par exemple, ce JSON peut représenter une entité.

  ```
  {
    "Entity": {
      "KeyAttributes": {
        "Type": "Service",
        "Name": "PetClinic"
      },
      "Attributes": {
        "PlatformType": "AWS::EC2",
        "EC2.InstanceId": "i-1234567890123"
      }
    }
  }
  ```

  Pour cette entité, les champs système extraits seraient les suivants :

  ```
  @entity.KeyAttributes.Type = Service
  @entity.KeyAttributes.Name = PetClinic
  ```

```
@entity.Attributes.PlatformType = AWS::EC2
@entity.Attributes.EC2.InstanceId = i-1234567890123
```

> **Note**
>
> La découverte de champs n'est prise en charge que pour les groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

CloudWatch Logs Insights insère le symbole @ au début des champs qu'il génère.

Pour de nombreux types de CloudWatch journaux, Logs découvre également automatiquement les champs de journal contenus dans les journaux. Ces champs de détection automatique figurent dans le tableau suivant.

Pour les autres types de journaux contenant des champs que CloudWatch Logs Insights ne découvre pas automatiquement, vous pouvez utiliser la `parse` commande pour extraire et créer des champs extraits à utiliser dans cette requête. Pour de plus amples informations, veuillez consulter CloudWatch Syntaxe des requêtes en langage Logs Insights.

Si le nom d'un champ de journal découvert commence par le `@` caractère, CloudWatch Logs Insights l'affiche avec un ajout `@` ajouté au début. Par exemple, si un nom de champ de journal est `@example.com`, ce nom de champ s'affiche sous la forme de `@@example.com`.

> **Note**
>
> À l'exception de `@message``@timestamp`, ou`@log`, vous pouvez créer des index de champs pour les champs découverts. Pour plus d'informations sur les index de champs, consultezCréez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation.

Type de journal	Champs de journal détectés
Journaux de flux Amazon VPC	`@timestamp` , `@logStream` , `@message`, `accountId` , `endTime`, `interfaceId` , `logStatus` , `startTime` , `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`

Type de journal	Champs de journal détectés
Journaux Route 53	`@timestamp` , `@logStream` , `@message`, `edgeLocation` , `ednsClientSubnet` , `hostZoneId` , `protocol`, `queryName` , `queryTimestamp` , `queryType` , `resolverIp` , `responseCode` , `version`
Journaux Lambda	`@timestamp` , `@logStream` , `@message`, `@requestId` , `@duration`, `@billedDuration` , `@type`, `@maxMemoryUsed` , `@memorySize` Si une ligne de journal Lambda contient un ID de suivi X-Ray, elle inclut également les champs suivants : `@xrayTraceId` et `@xraySegmentId` . CloudWatch Logs Insights découvre automatiquement les champs de journal dans les journaux Lambda, mais uniquement pour le premier fragment JSON intégré à chaque événement de journal. Si un événement de journal Lambda contient plusieurs fragments JSON, vous pouvez analyser et extraire les champs de journal à l'aide de la commande **`parse`**. Pour de plus amples informations, veuillez consulter Champs des journaux JSON.
CloudTrail journaux Journaux au format JSON	Pour plus d'informations, consultez Champs des journaux JSON.
Autres types de journaux	`@timestamp` , `@ingestionTime` , `@logStream` , `@message`, `@log`.

Champs des journaux JSON

Avec CloudWatch Logs Insights, vous utilisez la notation par points pour représenter les champs JSON. Cette section contient un exemple d'événement JSON et un extrait de code qui montre comment vous pouvez accéder aux champs JSON à l'aide de la notation par points.

Exemple : événement JSON

```
{
    "eventVersion": "1.0",
```

```
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
}
```

L'exemple d'événement JSON contient un objet nommé `userIdentity`. `userIdentity` contient un champ nommé `type`. Pour représenter la valeur de `type` à l'aide de la notation par points, vous utilisez `userIdentity.type`.

L'exemple d'événement JSON contient des matrices qui s'aplatissent en listes de noms et de valeurs de champs imbriqués. Pour représenter la valeur de `instanceId` pour le premier élément de `requestParameters.instancesSet`, vous utilisez `requestParameters.instancesSet.items.0.instanceId`. Le numéro `0` placé avant le champ `instanceID` fait référence à la position des valeurs pour le champ `items`. L'exemple suivant contient un extrait de code qui montre comment vous pouvez accéder aux champs JSON imbriqués dans un événement du journal JSON.

Exemple : requête

```
fields @timestamp, @message
| filter requestParameters.instancesSet.items.0.instanceId="i-abcde123"
| sort @timestamp desc
```

L'extrait de code affiche une requête qui utilise la notation par points avec la commande `filter` pour accéder à la valeur du champ JSON imbriqué `instanceId`. La requête filtre les messages dont la valeur de `instanceId` équivaut à `"i-abcde123"` et renvoie tous les événements du journal contenant la valeur spécifiée.

> **Note**
>
> CloudWatch Logs Insights peut extraire un maximum de 200 champs d'événements de journal à partir d'un journal JSON. Pour les champs supplémentaires non extraits, vous pouvez utiliser la commande `parse` pour extraire ces champs à partir de l'événement du journal non analysé brut dans le champ de message. Pour plus d'informations sur la `parse` commande, consultez la section Syntaxe des requêtes dans le guide de CloudWatch l'utilisateur Amazon.

Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation

Vous pouvez créer des index des champs de votre journal des événements pour des recherches efficaces basées sur l'égalité. Lorsque vous utilisez ensuite un index de champ dans une requête

CloudWatch Logs Insights, la requête tente d'ignorer le traitement des événements du journal dont on sait qu'ils n'incluent pas le champ indexé. Cela réduit le volume de numérisation de vos requêtes qui utilisent des index de champs, ce qui permet de renvoyer des résultats plus rapidement. Cela peut vous aider à rechercher rapidement des pétaoctets de journaux au total dans des milliers de groupes de journaux et à affiner les journaux pertinents plus rapidement. Les bons champs à indexer sont ceux que vous devez souvent rechercher. Les champs dont la cardinalité des valeurs est élevée sont également de bons candidats pour les index de champs, car une requête utilisant ces index de champs se terminera plus rapidement car cela limite les événements du journal qui sont mis en correspondance avec la valeur cible.

Supposons, par exemple, que vous ayez créé un index de champ pour`requestId`. Ensuite, toute requête CloudWatch Logs Insights sur ce groupe de journaux qui inclut `requestId = value` ou `requestId IN [value, value, ...]` tentera de traiter uniquement les événements du journal dont on sait qu'ils contiennent ce champ indexé et la valeur demandée, et que CloudWatch Logs a détecté une valeur pour ce champ dans le passé.

Vous pouvez également tirer parti de vos index de champs pour créer des requêtes efficaces portant sur un plus grand nombre de groupes de journaux. Lorsque vous utilisez la `filterIndex` commande dans votre requête au lieu de la `filter` commande, celle-ci est exécutée sur des groupes de journaux sélectionnés sur des événements de journal comportant des index de champs. Ces requêtes peuvent analyser jusqu'à 10 000 groupes de journaux, que vous choisissez en spécifiant jusqu'à cinq préfixes de nom de groupe de journaux. S'il s'agit d'un compte de surveillance dans CloudWatch le domaine de l'observabilité entre comptes, vous pouvez choisir tous les comptes sources ou spécifier des comptes sources individuels pour sélectionner les « groupes de journaux ».

Les champs indexés font la distinction majuscules/majuscules. Par exemple, un index de champ de `RequestId` ne correspondra pas à un événement de journal contenant`requestId`.

Les index de champs ne sont pris en charge que pour les formats de journaux structurés JSON et les journaux de service.

CloudWatch Logs fournit des index de champs par défaut pour tous les groupes de journaux de la classe de journaux standard. Les index de champs par défaut sont automatiquement disponibles pour les champs suivants :

- `@aws.region`
- `@aws.account`
- `@source.log`

- `traceId`

Les index de champs par défaut s'ajoutent aux index de champs personnalisés que vous définissez dans le cadre de votre politique. Les index de champs par défaut ne sont pas pris en compte dans votre quota d'index de champs.

CloudWatch Consigne uniquement les événements du journal ingérés après la création d'une politique d'index. Il n'indexe pas les événements du journal ingérés avant la création de la politique. Une fois que vous avez créé un index de champ, chaque événement de journal correspondant reste indexé pendant 30 jours à compter de la date d'ingestion de l'événement de journal.

> **Note**
>
> Si vous créez une politique d'index de champs dans un compte de surveillance, cette politique n'est pas utilisée pour les groupes de journaux des comptes source liés. Une politique d'indexation des champs s'applique uniquement au compte sur lequel elle a été créée.

Les autres rubriques de cette section expliquent comment créer des index de champs. Pour plus d'informations sur la référence aux index de champs dans vos requêtes, reportez-vous aux sections Index des filtres etfiltre.

Rubriques

- Syntaxe et quotas d'index de champs
- Création d'une politique d'indexation des champs au niveau du compte
- Création d'une politique d'index de champs au niveau du groupe de journaux
- Options de sélection de groupes de journaux lors de la création d'une requête
- Effets de la suppression d'une politique d'index de champs

Syntaxe et quotas d'index de champs

Vous créez des index de champs en créant des politiques d'index de champs. Vous pouvez créer des politiques d'index au niveau du compte qui s'appliquent à l'ensemble de votre compte, et vous pouvez également créer des politiques qui ne s'appliquent qu'à un seul groupe de journaux. Pour les politiques d'indexation applicables à l'ensemble du compte, vous pouvez en définir une qui s'applique

à tous les groupes de journaux du compte. Vous pouvez également créer des politiques d'index au niveau du compte qui s'appliquent à un sous-ensemble de groupes de journaux du compte, sélectionnés par les préfixes de leurs noms de groupes de journaux. Si vous avez plusieurs politiques au niveau du compte dans le même compte, les préfixes des noms de groupes de journaux associés à ces politiques ne peuvent pas se chevaucher.

Les politiques d'index des champs au niveau du groupe de journaux remplacent les politiques d'index des champs au niveau du compte : si vous créez une politique d'index au niveau du groupe de journaux, ce groupe de journaux utilise uniquement cette politique et ignore les politiques au niveau du compte.

Les correspondances entre les événements du journal et les noms des index de champs distinguent les majuscules et minuscules. Par exemple, un index de champ de `RequestId` ne correspondra pas à un événement de journal contenant`requestId`.

Vous pouvez avoir jusqu'à 20 politiques d'indexation au niveau du compte. Si plusieurs politiques d'index au niveau du compte sont filtrées en fonction des préfixes de nom de groupe de journaux, aucune d'entre elles ne peut utiliser des préfixes de nom de groupe de journaux identiques ou des préfixes de nom de groupe de journaux qui se chevauchent. Par exemple, si vous avez une politique filtrée pour enregistrer les groupes commençant par`my-log`, vous ne pouvez pas filtrer une autre politique d'index de champs sur `my-logpprod` ou`my-logging`.

Si vous disposez d'une politique d'index au niveau du compte qui ne comporte aucun préfixe de nom et s'applique à tous les groupes de journaux, aucune autre politique d'index au niveau du compte ne peut être créée.

Chaque politique d'indice comporte les quotas et restrictions suivants :

- Jusqu'à 20 champs peuvent être inclus dans la politique.
- Chaque nom de champ peut comporter jusqu'à 100 caractères.
- Pour créer un index d'un champ personnalisé dans vos groupes de journaux commençant par`@`, vous devez spécifier le champ avec un extra `@` au début du nom du champ. Par exemple, si les événements de votre journal incluent un champ nommé`@userId`, vous `@@userId` devez spécifier de créer un index pour ce champ.

Champs générés et champs réservés

CloudWatch Logs Insights génère automatiquement des champs système pour chaque événement de journal. Ces champs générés sont préfixés par. `@` Pour plus d'informations sur les champs générés, consultez Journaux pris en charge et champs découverts.

Parmi ces champs générés, les suivants sont pris en charge pour être utilisés comme index de champs :

- `@logStream`
- `@ingestionTime`
- `@requestId`
- `@type`
- `@initDuration`
- `@duration`
- `@billedDuration`
- `@memorySize`
- `@maxMemoryUsed`
- `@xrayTraceId`
- `@xraySetmentId`

Pour indexer ces champs générés, il n'est pas nécessaire d'en ajouter un `@` autre lorsque vous les spécifiez, comme vous devez le faire pour les champs personnalisés commençant par`@`. Par exemple, pour créer un index de champ pour`@logStream`, il suffit de le spécifier `@logStream` comme index de champ.

CloudWatch Logs fournit des index de champs par défaut pour tous les groupes de journaux de la classe de journaux standard. Les index de champs par défaut sont automatiquement disponibles pour les champs suivants :

- `@aws.region`
- `@aws.account`
- `@source.log`
- `traceId`

Les index de champs par défaut s'ajoutent aux index de champs personnalisés que vous définissez dans le cadre de votre politique. Les index de champs par défaut ne sont pas pris en compte dans votre quota d'index de champs.

Champs enfants et champs de tableau dans les journaux JSON

Vous pouvez indexer des champs qui sont des champs enfants ou des champs matriciels imbriqués dans les journaux JSON.

Par exemple, vous pouvez créer un index du champ `accessKeyId` enfant dans le `userIdentity` champ de ce journal :

```
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "11112222",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [{
                "instanceId": "i-abcde123",
                "currentState": {
                    "code": 0,
                    "name": "pending"
                },
                "previousState": {
                    "code": 80,
                    "name": "stopped"
                }
            }]
        }
    }
```

```
}
```

Pour créer ce champ, vous devez y faire référence à l'aide de la notation par points (`userIdentity.accessKeyId`) à la fois lors de la création de l'index du champ et lors de sa spécification dans une requête. La requête pourrait ressembler à ceci :

```
fields @timestamp, @message
| filterIndex userIdentity.accessKeyId = "11112222"
```

Dans l'exemple précédent, le `instanceId` champ se trouve dans un tableau dans `requestParameters.instancesSet.items` Pour représenter ce champ à la fois lors de la création de l'index du champ et lors de l'interrogation, appelez-le car `requestParameters.instancesSet.items.0.instanceId` le 0 fait référence à la place de ce champ dans le tableau.

Une requête pour ce champ peut alors être la suivante :

```
fields @timestamp, @message
| filterIndex requestParameters.instancesSet.items.0.instanceId="i-abcde123"
```

Création d'une politique d'indexation des champs au niveau du compte

Suivez les étapes décrites dans cette section pour créer une politique d'index de champs qui s'applique à tous les groupes de journaux du compte ou à plusieurs groupes de journaux dont les noms de groupes de journaux commencent par la même chaîne.

Pour créer une politique d'indexation des champs au niveau du compte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Paramètres, puis l'onglet Journaux.
3. Dans la section Politiques d'indexation au niveau du compte, choisissez Gérer.
4. Choisissez Créer une politique d'index.
5. Dans Nom de la politique, entrez le nom de votre nouvelle politique.
6. Pour Sélectionner des groupes de journaux, effectuez l'une des opérations suivantes :
   - Choisissez Tous les groupes de journaux standard pour que la politique d'indexation s'applique à tous les groupes de journaux de classe standard du compte.

- choisissez Sélectionner le ou les groupes de journaux par correspondance de préfixe pour appliquer la politique à un sous-ensemble de groupes de journaux dont les noms commencent tous par la même chaîne. Entrez ensuite le préfixe de ces groupes de journaux dans Entrez un nom de préfixe.

  Après avoir saisi votre préfixe, vous pouvez choisir Prévisualiser les groupes de journaux correspondant au préfixe pour confirmer que votre préfixe correspond aux groupes de journaux que vous attendiez.

7. Pour la configuration d'un champ d'index personnalisé, choisissez Ajouter un chemin de champ pour saisir le premier champ à indexer.

   Entrez ensuite la chaîne à utiliser comme valeur du nom du champ. Cela doit correspondre exactement à ce qui apparaît dans le journal des événements. Par exemple, si les événements de votre journal incluent`requestId`, vous devez entrer `requestId` ici. `RequestIdrequestID`, et `request Id` ne correspondrait pas.

   Si vous souhaitez indexer un champ de journal personnalisé qui commence par le `@` caractère, vous devez inclure un `@` caractère supplémentaire lorsque vous entrez la chaîne d'index. Par exemple, si vous avez un champ de journal personnalisé`@emailname`, saisissez-le `@@emailname` dans la zone Ajouter un chemin de champ.

   Vous pouvez également créer des index pour les `@logStream` champs `@ingestionTime` et générés automatiquement par CloudWatch Logs. Si c'est le cas, il n'est pas nécessaire d'en ajouter un supplémentaire `@` lorsque vous les spécifiez.
8. Répétez l'étape précédente pour ajouter jusqu'à 20 index de champs.
9. Lorsque vous avez terminé, choisissez Créer.

Création d'une politique d'index de champs au niveau du groupe de journaux

Suivez les étapes décrites dans cette section pour créer une politique d'index de champs qui s'applique à un seul groupe de journaux.

Pour créer une politique d'indexation des champs au niveau du groupe de journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), Log groups (Groupes de journaux).
3. Choisissez le nom du groupe de journaux.
4. Choisissez l'onglet Index des champs.
5. Choisissez Gérer les index de champs pour ce groupe de journaux
6. Pour Gérer les index de champs au niveau des groupes de journaux, choisissez Ajouter un chemin de champ pour saisir le premier champ à indexer.

   Entrez ensuite la chaîne à utiliser comme valeur du nom du champ. Cela doit correspondre exactement à ce qui apparaît dans le journal des événements. Par exemple, si les événements de votre journal incluent`requestId`, vous devez entrer `requestId` ici. `RequestIdrequestID`, et ne `request Id` correspondrait pas.

   Si vous souhaitez indexer un champ de journal personnalisé qui commence par le `@` caractère, vous devez inclure un `@` caractère supplémentaire lorsque vous entrez la chaîne d'index. Par exemple, si vous avez un champ de journal personnalisé`@emailname`, saisissez-le `@@emailname` dans la zone Ajouter un chemin de champ.

   Vous pouvez également créer des index pour les `@logStream` champs `@ingestionTime` et générés automatiquement par CloudWatch Logs. Si c'est le cas, il n'est pas nécessaire d'en ajouter un supplémentaire `@` lorsque vous les spécifiez.
7. Répétez l'étape précédente pour ajouter jusqu'à 20 index de champs.
8. Lorsque vous avez terminé, choisissez Save.

Options de sélection de groupes de journaux lors de la création d'une requête

Cette section explique les différentes manières de sélectionner les groupes de journaux à inclure dans une requête.

Pour sélectionner des groupes de journaux pour une requête dans la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Logs, Logs Insights.
3. Sélectionnez la langue de requête que vous souhaitez utiliser pour cette requête. Vous pouvez choisir entre Logs Insights QL, OpenSearchPPL ou OpenSearch SQL.

4. Il existe trois méthodes pour sélectionner des groupes de journaux pour la requête :
   - Utilisez le champ Nom du groupe de journaux. Il s'agit de la méthode de sélection par défaut. Vous pouvez saisir jusqu'à 50 noms de groupes de journaux avec cette méthode. S'il s'agit d'un compte de surveillance dans le CloudWatch domaine de l'observabilité entre comptes, vous pouvez sélectionner des groupes de journaux dans les comptes sources ainsi que dans le compte de surveillance. Une seule requête peut interroger les journaux de différents comptes à la fois.
   - Utilisez la section Critères du groupe de journaux. Dans cette section, vous pouvez choisir des groupes de journaux en fonction du préfixe des noms de groupes de journaux. Vous pouvez inclure jusqu'à cinq préfixes dans une requête. Les groupes de journaux dont le nom contient ces préfixes seront sélectionnés. Sinon, l'option Tous les groupes de journaux sélectionne tous les groupes de journaux du compte.
   - S'il s'agit d'un compte de surveillance dans CloudWatch le cadre de l'observabilité entre comptes, vous pouvez sélectionner Tous les comptes dans le menu déroulant des comptes pour sélectionner les groupes de journaux parmi tous les comptes associés. Vous pouvez également sélectionner individuellement les comptes à inclure pour cette requête.

   Si vos choix correspondent à plus de 10 000 groupes de journaux, un message d'erreur vous invitant à affiner votre sélection s'affichera.
5. La classe de journal par défaut pour une requête est Standard. Vous pouvez utiliser la classe Log pour la remplacer par Accès peu fréquent.

À l'aide du AWS CLI

Pour effectuer ces types de sélections lorsque vous lancez une requête depuis la ligne de commande, vous pouvez utiliser la `source` commande dans votre requête. Pour plus d'informations et d'exemples, consultez SOURCE.

Effets de la suppression d'une politique d'index de champs

Si vous supprimez une politique d'index de champs en vigueur depuis un certain temps, les événements suivants se produisent :

- Jusqu'à 30 jours après la suppression de la politique, les requêtes peuvent toujours bénéficier des événements du journal indexé.

- Si vous supprimez une politique d'index au niveau du groupe de journaux et qu'une politique au niveau du compte est déjà en place qui s'appliquerait à ce groupe de journaux, la politique au niveau du compte finira par s'appliquer à ce groupe de journaux.

Analyse de modèles

CloudWatch Logs Insights utilise des algorithmes d'apprentissage automatique pour identifier des modèles lorsque vous interrogez vos journaux. Un modèle est une structure de texte partagée récurrente dans vos champs de journal. Lorsque vous consultez les résultats d'une requête, vous pouvez choisir l'onglet Modèles pour voir les modèles trouvés par CloudWatch Logs sur la base d'un échantillon de vos résultats. Vous pouvez également ajouter la `pattern` commande à votre requête pour analyser les modèles de l'ensemble des événements du journal correspondants.

Les modèles sont utiles pour analyser de grands ensembles de journaux, car un grand nombre d'événements de journal peuvent souvent être compressés en plusieurs modèles.

Examinez l'exemple suivant de trois événements de journal.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

Dans l'exemple précédent, les trois événements du journal suivent le même schéma :

```
<Time-1> [INFO] Calling DynamoDB to store for resource id <ID-2>
```

Les champs d'un modèle sont appelés jetons. Les champs qui varient au sein d'un modèle, tels qu'un ID de demande ou un horodatage, sont des jetons dynamiques. Chaque jeton dynamique est représenté par<*`string`*-*`number`*>. *`string`*Il s'agit d'une description du type de données que le jeton représente. *`number`*Indique où ce jeton apparaît dans le modèle, par rapport aux autres jetons dynamiques.

Les exemples courants de jetons dynamiques incluent les codes d'erreur, les horodatages et les demandes. IDs Une valeur de jeton représente une valeur particulière d'un jeton dynamique. Par exemple, si un jeton dynamique représente un code d'erreur HTTP, une valeur de jeton peut l'être`501`.

La détection de modèles est également utilisée dans le détecteur d'anomalies CloudWatch Logs et les fonctionnalités de comparaison. Pour plus d'informations, consultez Détection des anomalies du journal et Comparer (diff) avec les plages temporelles précédentes.

Commencer à utiliser l'analyse de modèles

La détection des modèles est automatiquement effectuée dans toutes CloudWatch les requêtes Logs Insights. Les requêtes qui n'incluent pas la `pattern` commande enregistrent à la fois les événements et les modèles dans les résultats.

Si vous incluez la `pattern` commande dans votre requête, l'analyse des modèles est effectuée sur l'ensemble des événements du journal correspondants. Cela permet d'obtenir des résultats de modèle plus précis, mais les événements du journal bruts ne sont pas renvoyés lorsque vous utilisez la `pattern` commande. Lorsqu'une requête n'inclut pas`pattern`, les résultats du modèle sont basés soit sur les 1 000 premiers événements de journal renvoyés, soit sur la valeur limite que vous avez utilisée dans votre requête. Si vous l'incluez `pattern` dans la requête, les résultats affichés dans l'onglet Modèles sont dérivés de tous les événements du journal correspondant à la requête.

Pour commencer à utiliser l'analyse de modèles dans CloudWatch Logs Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Logs, Logs Insights.

   Sur la page Logs Insights, l'éditeur de requête contient une requête par défaut qui renvoie les 20 événements du journal les plus récents.
3. Supprimez la `| limit 20` ligne dans la zone de requête afin que la requête ressemble à ce qui suit :

   ```
   fields @timestamp, @message, @logStream, @log
   | sort @timestamp desc
   ```
4. Dans le menu déroulant Sélectionner un ou plusieurs groupes de journaux, choisissez un ou plusieurs groupes de journaux à interroger.
5. (Facultatif) Utilisez le sélecteur d'intervalle de temps pour sélectionner une période à interroger.

   Vous pouvez choisir entre des intervalles de 5 minutes et 30 minutes, des intervalles de 1 heure, 3 heures et 12 heures, ou un intervalle de temps personnalisé.
6. Choisissez Exécuter la requête pour démarrer la requête.

Lorsque l'exécution de la requête est terminée, l'onglet Journaux affiche un tableau des événements du journal renvoyés par la requête. Au-dessus du tableau se trouve un message indiquant le nombre d'enregistrements correspondant à la requête, similaire à l'affichage de 10 000 enregistrements sur 71 101 correspondants.

7. Choisissez l'onglet Motifs.
8. Le tableau affiche désormais les modèles trouvés dans la requête. Comme la requête n'incluait pas la `pattern` commande, cet onglet affiche uniquement les modèles découverts parmi les 10 000 événements de journal indiqués dans le tableau de l'onglet Journaux.

   Pour chaque modèle, les informations suivantes sont affichées :

   - Le modèle, avec chaque jeton dynamique affiché sous la forme<*`string`*-*`number`*>. *`string`*Il s'agit d'une description du type de données que le jeton représente. *`number`*Indique où ce jeton apparaît dans le modèle, par rapport aux autres jetons dynamiques.
   - Le nombre d'événements, qui est le nombre de fois que le modèle est apparu dans le journal des événements demandé. Choisissez l'en-tête de la colonne Nombre d'événements pour trier les modèles par fréquence.
   - Le ratio d'événements, qui est le pourcentage des événements du journal interrogés qui contiennent ce modèle.
   - Le type de gravité, qui sera l'un des suivants :
     - ERREUR si le modèle contient le mot Error.
     - AVERTIR si le modèle contient le mot Warn mais ne contient pas Error.
     - INFO si le modèle ne contient ni avertissement ni erreur.

   Choisissez l'en-tête de la colonne Informations sur la gravité pour trier les modèles par gravité.
9. Modifiez maintenant la requête. Remplacez la `| sort @timestamp desc` ligne de la requête par`| pattern @message`, de sorte que la requête complète soit la suivante :

```
fields @timestamp, @message, @logStream, @log
| pattern @message
```

10. Choisissez Exécuter la requête.

    Lorsque la requête est terminée, aucun résultat ne s'affiche dans l'onglet Logs. Cependant, l'onglet Modèles contient probablement un plus grand nombre de modèles répertoriés, en fonction du nombre total d'événements de journal interrogés.

11. Que vous l'ayez inclus ou non `pattern` dans votre requête, vous pouvez examiner plus en détail les modèles renvoyés par la requête. Pour ce faire, choisissez l'icône de l'un des modèles dans la colonne Inspecter.

    Le volet Pattern inspect apparaît et affiche les informations suivantes :

    - Le motif. Sélectionnez un jeton dans le modèle pour analyser les valeurs de ce jeton.
    - Un histogramme indiquant le nombre d'occurrences du modèle sur la plage de temps demandée. Cela peut vous aider à identifier des tendances intéressantes, telles qu'une augmentation soudaine de l'occurrence d'un modèle.
    - L'onglet Échantillons de journal affiche quelques-uns des événements du journal correspondant au modèle sélectionné.
    - L'onglet Valeurs du jeton affiche les valeurs du jeton dynamique sélectionné, si vous en avez sélectionné un.

    > **Note**
    >
    > Un maximum de 10 valeurs de jeton est capturée pour chaque jeton. Le nombre de jetons peut ne pas être précis. CloudWatch Logs utilise un compteur probabiliste pour générer le nombre de jetons, et non la valeur absolue.

    - L'onglet Modèles associés affiche d'autres modèles qui se sont produits fréquemment à peu près au même moment que le modèle que vous inspectez. Par exemple, si le modèle d'un `ERROR` message était généralement accompagné d'un autre événement de journal marqué comme `INFO` contenant des détails supplémentaires, ce modèle est affiché ici.

Détails sur la commande pattern

Cette section contient plus de détails sur la `pattern` commande et ses utilisations.

- Dans le didacticiel précédent, nous avons supprimé la `sort` commande lors de son ajout `pattern` car une requête n'est pas valide si elle inclut une `pattern` commande après une `sort` commande. Il est valide d'avoir un `pattern` avant un`sort`.

  Pour plus de détails sur `pattern` la syntaxe, consultez[pattern](#).

- Lorsque vous l'utilisez `pattern` dans une requête, ce `@message` doit être l'un des champs sélectionnés dans la `pattern` commande.

- Vous pouvez inclure la `filter` commande avant une `pattern` commande pour que seul l'ensemble filtré d'événements du journal soit utilisé comme entrée pour l'analyse des modèles.
- Pour voir les résultats de modèles pour un champ particulier, tel qu'un champ dérivé de la `parse` commande, utilisez`pattern @fieldname`.
- Les requêtes dont la sortie n'est pas un journal, telles que les requêtes avec la `stats` commande, ne renvoient pas de résultats de modèle.

Enregistrez et réexécutez les requêtes CloudWatch Logs Insights

Après avoir créé une requête, vous pouvez l'enregistrer et la réexécuter ultérieurement. Les requêtes sont enregistrées dans une structure de dossiers afin que vous puissiez les organiser. Vous pouvez enregistrer jusqu'à 1 000 requêtes par région et par compte.

Les requêtes sont enregistrées à un niveau spécifique à la région, et non à un niveau spécifique à l'utilisateur. Si vous créez et enregistrez une requête, les autres utilisateurs ayant accès aux CloudWatch journaux de la même région peuvent voir toutes les requêtes enregistrées et leurs structures de dossiers dans la région.

Pour enregistrer une requête, vous devez être connecté à un rôle disposant de l'autorisation `logs:PutQueryDefinition`. Pour afficher une liste des requêtes enregistrées, vous devez être connecté à un rôle disposant de l'autorisation `logs:DescribeQueryDefinitions`.

Pour enregistrer une requête

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Dans l'éditeur de requête, créez une requête.
4. Choisissez Enregistrer.

   Si aucun bouton Enregistrer ne s'affiche, vous devez adopter le nouveau design de la console CloudWatch Logs. Pour ce faire :

   a. Dans le panneau de navigation, choisissez Groupes de journaux.
   b. Choisissez Try the new design (Tester la nouvelle interface).
   c. Dans le panneau de navigation, choisissez Insights, puis revenez à l'étape 3 de cette procédure.

5. Entrez un nom pour la requête.
6. (Facultatif) Choisissez un dossier dans lequel vous souhaitez enregistrer cette requête. Sélectionnez (Nouveau) pour créer un dossier. Si vous créez un dossier, vous pouvez utiliser des barres obliques (/) dans son nom pour définir une structure de dossier. Par exemple, le nom d'un nouveau dossier **`folder-level-1/folder-level-2`** crée un dossier de niveau supérieur appelé **`folder-level-1`**, avec un autre dossier appelé **`folder-level-2`** au sein de ce dossier. La requête est enregistrée dans **`folder-level-2`**.
7. (Facultatif) Modifiez les groupes de journaux ou le texte de la requête.
8. Choisissez Enregistrer.

> **Tip**
>
> Vous pouvez créer un dossier pour les requêtes enregistrées avec `PutQueryDefinition`. Pour créer un dossier pour vos requêtes enregistrées, utilisez une barre oblique (/) pour préfixer le nom de requête souhaité par le nom de dossier de votre choix : `<folder-name>/<query-name>`. Pour plus d'informations sur cette action, consultez PutQueryDefinition.

Pour exécuter une requête enregistrée

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Sur la droite, choisissez Queries (Requêtes).
4. Sélectionnez votre requête dans la liste Saved queries (Requêtes enregistrées). Celle-ci apparaît dans l'éditeur de requêtes.
5. Cliquez sur Exécuter.

Pour enregistrer une nouvelle version d'une requête enregistrée

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Sur la droite, choisissez Queries (Requêtes).
4. Sélectionnez votre requête dans la liste Saved queries (Requêtes enregistrées). Celle-ci apparaît dans l'éditeur de requêtes.

5. Modifiez la requête. Si vous devez l'exécuter pour vérifier le résultat de vos efforts, choisissez Run query (Exécuter la requête).
6. Lorsque vous êtes prêt à enregistrer la nouvelle version, choisissez Actions, puis Save as (Enregistrer sous).
7. Entrez un nom pour la requête.
8. (Facultatif) Choisissez un dossier dans lequel vous souhaitez enregistrer cette requête. Sélectionnez (Nouveau) pour créer un dossier. Si vous créez un dossier, vous pouvez utiliser des barres obliques (/) dans son nom pour définir une structure de dossier. Par exemple, le nom d'un nouveau dossier **`folder-level-1/folder-level-2`** crée un dossier de niveau supérieur appelé **`folder-level-1`**, avec un autre dossier appelé **`folder-level-2`** au sein de ce dossier. La requête est enregistrée dans **`folder-level-2`**.
9. (Facultatif) Modifiez les groupes de journaux ou le texte de la requête.
10. Choisissez Enregistrer.

Pour supprimer une requête, vous devez être connecté à un rôle disposant de l'autorisation `logs:DeleteQueryDefinition`.

Pour modifier ou supprimer une requête enregistrée

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Sur la droite, choisissez Queries (Requêtes).
4. Sélectionnez votre requête dans la liste Saved queries (Requêtes enregistrées). Celle-ci apparaît dans l'éditeur de requêtes.
5. Choisissez Actions et Edit (Modifier) ou Actions et Delete (Supprimer).

Ajouter une requête au tableau de bord ou exporter les résultats de la requête

Après avoir exécuté une requête, vous pouvez l'ajouter à un CloudWatch tableau de bord ou copier les résultats dans le presse-papiers.

Les requêtes ajoutées aux tableaux de bord s'exécutent automatiquement chaque fois que vous chargez le tableau de bord et chaque fois que le tableau de bord s'actualise. Ces requêtes sont prises en compte dans votre limite de 30 requêtes CloudWatch Logs Insights simultanées.

Pour ajouter les résultats d'une requête sur un tableau de bord

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Choisissez un ou plusieurs groupes de journaux et exécutez une requête.
4. Choisissez Add to dashboard (Ajouter au tableau de bord).
5. Sélectionnez le tableau de bord, ou choisissez Create (Nouveau) pour créer un tableau de bord pour les résultats de la requête.
6. Sélectionnez le type de widget à utiliser pour les résultats de la requête.
7. Entrez un nom pour ce widget.
8. Choisissez Add to dashboard (Ajouter au tableau de bord).

Pour copier les résultats de la requête dans le Presse-papiers ou les télécharger

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.
3. Choisissez un ou plusieurs groupes de journaux et exécutez une requête.
4. Choisissez Export results (Exporter les résultats), puis choisissez l'option souhaitée.

Afficher les requêtes en cours d'exécution ou l'historique des requêtes

Vous pouvez afficher les requêtes en cours, ainsi que votre historique des requêtes récentes.

Les requêtes en cours d'exécution incluent les requêtes que vous avez ajoutées à un tableau de bord. Vous êtes limité à 30 requêtes CloudWatch Logs Insights simultanées par compte, y compris les requêtes ajoutées aux tableaux de bord. Seules 15 de ces 30 requêtes peuvent utiliser le OpenSearch Service PPL ou le OpenSearch Service SQL.

Pour afficher l'historique de vos requêtes récentes

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Logs Insights.

3. Choisissez History si vous utilisez le nouveau design de la console CloudWatch Logs. Si vous utilisez l'ancienne interface, choisissez Actions, puis View query history for this account (Afficher l'historique des requêtes pour ce compte).

   La liste de vos requêtes récentes s'affiche. Pour les exécuter à nouveau, sélectionnez la requête et choisissez Run (Exécuter).

   Sous État, CloudWatch Logs affiche En cours pour toutes les requêtes en cours d'exécution.

Chiffrer les résultats de requête avec AWS Key Management Service

Par défaut, CloudWatch Logs chiffre les résultats enregistrés de vos requêtes CloudWatch Logs Insights en utilisant la méthode de chiffrement côté serveur CloudWatch Logs par défaut. Vous pouvez choisir d'utiliser une AWS KMS clé pour chiffrer ces résultats à la place. Si vous associez une AWS KMS clé à vos résultats de chiffrement, CloudWatch Logs utilise cette clé pour chiffrer les résultats enregistrés de toutes les requêtes du compte.

Si vous dissociez ultérieurement une clé des résultats de votre requête, CloudWatch Logs revient à la méthode de cryptage par défaut pour les requêtes ultérieures. Mais les requêtes exécutées alors que la clé était associée sont toujours chiffrées avec cette clé. CloudWatch Les journaux peuvent toujours renvoyer ces résultats une fois la clé KMS dissociée, car ils CloudWatch peuvent continuer à faire référence à la clé. Toutefois, si la clé est désactivée ultérieurement, CloudWatch Logs ne pourra pas lire les résultats de la requête chiffrés avec cette clé.

> **Important**
>
> CloudWatch Logs ne prend en charge que les clés KMS symétriques. N'utilisez pas de clé asymétrique pour chiffrer vos résultats de requête. Pour plus d'informations, consultez Utilisation des clés symétriques et asymétriques.

Limites

- Pour effectuer les étapes suivantes, vous devez disposer des autorisations suivantes : `kms:CreateKey`, `kms:GetKeyPolicy` et `kms:PutKeyPolicy`.

- L'association d'une clé à vos résultats de requête ou sa dissociation peut mettre jusqu'à cinq minutes à prendre effet.
- Si vous révoquez l'accès de CloudWatch Logs à une clé associée ou si vous supprimez une clé KMS associée, vos données chiffrées dans CloudWatch Logs ne peuvent plus être récupérées.
- Vous ne pouvez pas utiliser la CloudWatch console pour associer une clé, vous devez utiliser l'API AWS CLI or CloudWatch Logs.

Étape 1 : Création d'un AWS KMS key

Pour créer une clé KMS, utilisez la commande create-key suivante :

```
aws kms create-key
```

La sortie contient l'ID de clé et l'Amazon Resource Name (ARN) de la clé. Voici un exemple de sortie :

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Étape 2 : Définition des autorisations sur la clé KMS

Par défaut, toutes les clés KMS sont privées. Seul le propriétaire de la ressource peut l'utiliser pour chiffrer et déchiffrer des données. Cependant, le propriétaire de la ressource peut accorder à d'autres

utilisateurs et ressources des autorisations d'accès à la clé. Avec cette étape, vous autorisez le principal du service CloudWatch Logs à utiliser la clé. Ce principal de service doit se trouver dans la même AWS région que celle où la clé est stockée.

Il est recommandé de limiter l'utilisation de la clé aux seuls AWS comptes que vous spécifiez.

Tout d'abord, enregistrez la politique par défaut pour votre clé KMS à `policy.json` l'aide de la get-key-policycommande suivante :

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Ouvrez le fichier `policy.json` dans un éditeur de texte et ajoutez la section en gras à partir de l'une des instructions suivantes. Séparez l'instruction existante de la nouvelle instruction par une virgule. Ces instructions utilisent `Condition` des sections pour renforcer la sécurité de la AWS KMS clé. Pour de plus amples informations, veuillez consulter AWS KMS clés et contexte de chiffrement.

La `Condition` section de cet exemple limite l'utilisation de la AWS KMS clé pour les résultats de la requête CloudWatch Logs Insights dans le compte spécifié.

JSON

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
```

```
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                "aws:SourceArn": "arn:aws:logs:us-east-1:111122223333:query-
result:*"
                },
                "StringEquals": {
                "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Enfin, ajoutez la politique mise à jour à l'aide de la put-key-policycommande suivante :

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Étape 3 : Associer une clé KMS à vos résultats de requête

Pour associer la clé KMS aux résultats de la requête dans le compte

Utilisez la commande disassociate-kms-key comme suit :

```
aws logs associate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-
result:*" --kms-key-id "key-arn"
```

Étape 4 : Dissocier une clé des résultats de requête dans le compte

Pour dissocier la clé KMS associée aux résultats de la requête, utilisez la disassociate-kms-keycommande suivante :

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-
id:query-result:*"
```

Génération d'un résumé en langage naturel à partir des résultats des requêtes de CloudWatch Logs Insights

L'analyse des données des journaux est essentielle pour comprendre le comportement de vos applications, mais l'interprétation de gros volumes d'entrées de journal peut prendre beaucoup de temps. CloudWatch Logs Insights propose désormais une fonctionnalité de synthèse en langage naturel qui transforme les résultats de requêtes complexes en résumés clairs et concis. Cette fonctionnalité vous permet d'identifier rapidement les problèmes et d'obtenir des informations exploitables à partir des données de vos journaux.

Comment ça marche

CloudWatch Logs Insights peut générer un résumé lisible par l'homme à partir des résultats de vos requêtes à l'aide d'Amazon Bedrock. Cette fonctionnalité prend en charge tous les langages de requête de CloudWatch Logs Insights et fournit des informations claires et exploitables à partir des données de vos journaux.

Disponibilité régionale et traitement des données

> **Important**
>
> Lorsque vous utilisez cette fonctionnalité, les résultats de votre requête peuvent être traités différemment Région AWS. Par exemple, si vous exécutez une requête dans l'est des États-Unis (Virginie du Nord), le résumé peut avoir lieu dans l'ouest des États-Unis (Oregon).

Le tableau suivant répertorie les traitements possibles Région AWS pour les différentes zones géographiques dans lesquelles la fonctionnalité de résultats de requête est disponible :

Géographie des CloudWatch journaux pris en charge	Région de traitement possible
États-Unis	US East (N. Virginia) Region

Géographie des CloudWatch journaux pris en charge	Région de traitement possible
	US East (Ohio) Region US West (Oregon) Region
Europe	Région Europe (Frankfurt) Europe (Ireland) Region Région Europe (Paris) Région Europe (Stockholm) Région Europe (Londres)
Asie-Pacifique	US East (N. Virginia) Region US East (Ohio) Region US West (Oregon) Region
Amérique du Sud	US East (N. Virginia) Region US East (Ohio) Region US West (Oregon) Region

Premiers pas

Pour générer un résumé en langage naturel

1. Exécutez votre requête CloudWatch Logs Insights.
2. Une fois la requête terminée, sélectionnez Résumer les résultats.

Autorisations

Vous devez posséder l'un des éléments suivants :

- `CloudWatchLogsFullAccess` autorisation
- `CloudWatchLogsReadOnlyAccess` autorisation
- Politique IAM personnalisée, y compris les `cloudwatch:GenerateQueryResultsSummary` actions`logs:GetQueryResults`, `logs:DescribeQueries` et `logs:FilterLogEvents`

Confidentialité des données

Les résultats de vos requêtes sont traités de manière sécurisée et ne sont pas utilisés pour former ou améliorer CloudWatch Logs Insights ou Amazon Bedrock. Si vous choisissez de fournir des commentaires sur le résumé des résultats de la requête à l'aide des boutons de commentaires, vos commentaires indiquent votre niveau de satisfaction à l'égard de la fonctionnalité fournie dans CloudWatch Logs Insights.

Détection des anomalies du journal

Vous pouvez créer un détecteur d'anomalies de journal pour chaque groupe de journaux. Le détecteur d'anomalies analyse les événements du journal ingérés dans le groupe de journaux et détecte les anomalies dans les données du journal. La détection des anomalies utilise l'apprentissage automatique et la reconnaissance de formes pour établir des bases de référence pour le contenu typique des journaux.

Une fois que vous avez créé un détecteur d'anomalies pour un groupe de journaux, celui-ci s'entraîne en utilisant les événements des deux dernières semaines dans le groupe de journaux à des fins d'entraînement. La période d'entraînement peut durer jusqu'à 15 minutes. Une fois la formation terminée, elle commence à analyser les journaux entrants pour identifier les anomalies, qui sont affichées dans la console CloudWatch Logs pour que vous puissiez les examiner.

CloudWatch La reconnaissance des modèles de journaux extrait les modèles de journaux en identifiant le contenu statique et dynamique de vos journaux. Les modèles sont utiles pour analyser de grands ensembles de journaux, car un grand nombre d'événements de journal peuvent souvent être compressés en plusieurs modèles.

Par exemple, consultez l'exemple suivant de trois événements de journal.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for ResourceID: 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for ResourceID: 324892398123-1234R
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for ResourceID: 3ff231242342-12345
```

Dans l'exemple précédent, les trois événements du journal suivent le même schéma :

```
<Date-1> <Time-2> [INFO] Calling DynamoDB to store for resource id <ResourceID-3>
```

Les champs d'un modèle sont appelés jetons. Les champs qui varient au sein d'un modèle, tels qu'un ID de demande ou un horodatage, sont appelés jetons dynamiques. Chaque valeur différente trouvée pour un jeton dynamique est appelée valeur de jeton.

Si CloudWatch Logs peut déduire le type de données représenté par un jeton dynamique, il affiche le jeton sous <*string*-*number*> la forme. *string*Il s'agit d'une description du type de données que le jeton représente. *number*Indique où ce jeton apparaît dans le modèle, par rapport aux autres jetons dynamiques.

CloudWatch Logs attribue la partie chaîne du nom en fonction de l'analyse du contenu des événements du journal qui le contiennent.

Si CloudWatch Logs ne peut pas déduire le type de données représenté par un jeton dynamique, il affiche le jeton sous la forme <Token- *number* > et *number* indique où ce jeton apparaît dans le schéma, par rapport aux autres jetons dynamiques.

Les exemples courants de jetons dynamiques incluent les codes d'erreur, les adresses IP, les horodatages et les demandes. IDs

La détection des anomalies dans les journaux utilise ces modèles pour détecter les anomalies. Après la période de formation du modèle de détecteur d'anomalies, les journaux sont évalués par rapport aux tendances connues. Le détecteur d'anomalies signale les fluctuations importantes comme des anomalies.

Ce chapitre explique comment activer la détection des anomalies, visualiser les anomalies, créer des alarmes pour les détecteurs d'anomalies dans les journaux et les mesures publiées par les détecteurs d'anomalies dans les journaux. Il décrit également comment chiffrer le détecteur d'anomalies et ses résultats avec. AWS Key Management Service

La création de détecteurs d'anomalies du journal n'entraîne aucun frais.

Gravité et priorité des anomalies et des modèles

Une priorité est attribuée à chaque anomalie détectée par un détecteur d'anomalies logarithmiques. Une gravité est attribuée à chaque modèle détecté.

- La priorité est automatiquement calculée et est basée à la fois sur le niveau de gravité du modèle et sur le degré d'écart par rapport aux valeurs attendues. Par exemple, si la valeur d'un jeton augmente soudainement de 500 %, cette anomalie peut être désignée comme `HIGH` prioritaire, même si sa gravité l'est. `NONE`
- La sévérité est basée uniquement sur les mots clés trouvés dans les modèles tels que `FATAL``ERROR`, et`WARN`. Si aucun de ces mots clés n'est trouvé, la gravité d'un modèle est marquée comme`NONE`.

Durée de visibilité des anomalies

Lorsque vous créez un détecteur d'anomalies, vous spécifiez la période maximale de visibilité des anomalies pour celui-ci. Il s'agit du nombre de jours pendant lesquels l'anomalie s'affiche dans

la console et est renvoyée par l'opération ListAnomaliesd'API. Une fois ce délai écoulé pour une anomalie, si elle persiste, elle est automatiquement acceptée comme un comportement normal et le modèle de détecteur d'anomalie cesse de la signaler comme une anomalie.

Si vous ne réglez pas le temps de visibilité lorsque vous créez un détecteur d'anomalie, 21 jours sont utilisés par défaut.

Supprimer une anomalie

Une fois qu'une anomalie a été détectée, vous pouvez choisir de la supprimer temporairement ou définitivement. La suppression d'une anomalie empêche le détecteur d'anomalie de signaler cette occurrence comme une anomalie pendant la durée que vous spécifiez. Lorsque vous supprimez une anomalie, vous pouvez choisir de supprimer uniquement cette anomalie spécifique ou de supprimer toutes les anomalies liées au schéma dans lequel l'anomalie a été détectée.

Vous pouvez toujours consulter les anomalies supprimées dans la console. Vous pouvez également choisir de ne plus les supprimer.

Questions fréquentes (FAQ)

Est-ce que mes données sont AWS utilisées pour entraîner des algorithmes d'apprentissage automatique destinés à être AWS utilisés ou destinés à d'autres clients ?

Non. Le modèle de détection des anomalies créé par la formation est basé sur les événements du journal d'un groupe de journaux et n'est utilisé qu'au sein de ce groupe de journaux et de ce AWS compte.

Quels types d'événements de journal fonctionnent bien avec la détection des anomalies ?

La détection des anomalies dans les journaux convient parfaitement aux journaux d'applications et aux autres types de journaux dans lesquels la plupart des entrées de journal correspondent à des modèles classiques. Les groupes de journaux contenant des événements contenant un niveau de journalisation ou des mots clés de gravité tels que INFO, ERROR et DEBUG sont particulièrement adaptés à la détection des anomalies des journaux.

La détection des anomalies du journal n'est pas adaptée pour : enregistrez les événements avec des structures JSON extrêmement longues, tels que CloudTrail les journaux. L'analyse des modèles analyse uniquement les 1 500 premiers caractères d'une ligne de journal, de sorte que tous les caractères dépassant cette limite sont ignorés.

Les journaux d'audit ou d'accès, tels que les journaux de flux VPC, seront également moins efficaces en matière de détection des anomalies. La détection des anomalies est destinée à détecter les problèmes d'application. Elle peut donc ne pas être adaptée aux anomalies du réseau ou de l'accès.

Pour vous aider à déterminer si un détecteur d'anomalies convient à un certain groupe de journaux, utilisez l'analyse des modèles de CloudWatch journaux pour déterminer le nombre de modèles dans les événements de journal du groupe. Si le nombre de modèles n'est pas supérieur à environ 300, la détection des anomalies peut fonctionner correctement. Pour plus d'informations sur l'analyse des modèles, consultezAnalyse de modèles.

Qu'est-ce qui est considéré comme une anomalie ?

Les événements suivants peuvent entraîner le marquage d'un événement du journal comme une anomalie :

- Un événement de journal dont le schéma n'a jamais été observé auparavant dans le groupe de journaux.
- Variation significative par rapport à un schéma connu.
- Nouvelle valeur pour un jeton dynamique comportant un ensemble discret de valeurs habituelles.
- Modification importante du nombre d'occurrences d'une valeur pour un jeton dynamique.

Bien que tous les éléments précédents puissent être marqués comme des anomalies, ils ne signifient pas tous que l'application fonctionne mal. Par exemple, un higher-than-usual certain nombre de valeurs de `200` réussite peuvent être signalées comme des anomalies. Dans de tels cas, vous pouvez envisager de supprimer ces anomalies qui n'indiquent pas de problèmes.

Que se passe-t-il avec les données sensibles masquées ?

Les parties des événements du journal qui sont masquées comme des données sensibles ne sont pas analysées pour détecter toute anomalie. Pour plus d'informations sur le masquage des données sensibles, voir Aider à protéger les données de journal sensibles par le masquage.

Activer la détection des anomalies sur un groupe de journaux

Procédez comme suit pour utiliser la CloudWatch console afin de créer un détecteur d'anomalies de journal qui analyse un groupe de journaux à la recherche d'anomalies.

Vous pouvez également créer des détecteurs d'anomalies par programmation. Pour de plus amples informations, veuillez consulter CreateLogAnomalyDetector.

Pour créer un détecteur d'anomalies dans le journal

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Logs, Log Anomalies.
3. Choisissez Créer un détecteur d'anomalies.
4. Sélectionnez le groupe de journaux pour lequel créer ce détecteur d'anomalies.
5. Entrez le nom du détecteur dans Nom du détecteur d'anomalies.
6. (Facultatif) Modifiez la fréquence d'évaluation par rapport à la valeur par défaut de 5 minutes. Définissez cette valeur en fonction de la fréquence à laquelle le groupe de journaux reçoit de nouveaux journaux. Par exemple, si le groupe de journaux reçoit de nouveaux événements de journal par lots toutes les 10 minutes, il peut être approprié de définir la fréquence d'évaluation sur 15 minutes.
7. (Facultatif) Pour configurer le détecteur d'anomalies afin de rechercher des anomalies uniquement dans les événements du journal contenant certains mots ou chaînes, choisissez Filtrer les modèles.

   Entrez ensuite un modèle dans Modèle de filtre de détection des anomalies. Pour plus d'informations sur la syntaxe des modèles,Syntaxe des modèles de filtres pour les filtres de métriques, les filtres d'abonnements, les filtres d'événements du journal et Live Tail.

   (Facultatif) Pour tester votre modèle de filtre, entrez des messages de journal dans les messages d'événements du journal, puis choisissez Test Pattern.
8. (Facultatif) Pour modifier la période de visibilité des anomalies par rapport à la période par défaut ou pour associer une AWS KMS clé à ce détecteur d'anomalies, choisissez Configuration avancée.

   a. Pour modifier la période de visibilité des anomalies par rapport à la période par défaut, entrez une nouvelle valeur dans Période maximale de visibilité des anomalies (jours).
   b. Pour associer une AWS KMS clé à ce détecteur d'anomalie, entrez l'ARN dans l'ARN de la clé KMS. Si vous attribuez une clé, les informations d'anomalie détectées par ce détecteur sont cryptées au repos avec la clé. Les utilisateurs doivent disposer des autorisations nécessaires pour utiliser cette clé et permettre au détecteur d'anomalies de récupérer des informations sur les anomalies détectées.

      Vous devez également vous assurer que le principal du service CloudWatch Logs est autorisé à utiliser la clé. Pour de plus amples informations, veuillez consulter Chiffrez un détecteur d'anomalie et ses résultats avec AWS KMS.

9. Choisissez Activer la détection des anomalies.

   Le détecteur d'anomalies est créé et commence à entraîner son modèle, en fonction des événements du journal ingéré par le groupe de journaux. Après environ 15 minutes, la détection des anomalies est active et commence à détecter et à détecter les anomalies.

Afficher les anomalies détectées

Après avoir créé un ou plusieurs détecteurs d'anomalies dans le journal, vous pouvez utiliser la CloudWatch console pour visualiser les anomalies détectées.

Vous pouvez visualiser les anomalies par programmation. Pour de plus amples informations, veuillez consulter ListAnomalies.

Pour visualiser les anomalies détectées par tous vos détecteurs d'anomalies logarithmiques

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Logs, Log Anomalies.

   Le tableau des anomalies du journal s'affiche. Le chiffre en haut à côté de Log Anomalies indique le nombre d'anomalies log répertoriées dans le tableau. Chaque ligne du tableau affiche les informations suivantes :

   - La colonne Anomalie affiche un bref résumé de l'anomalie. Ces résumés sont générés par CloudWatch Logs.
   - La priorité de l'anomalie. La priorité est automatiquement calculée en fonction de l'ampleur des modifications apportées aux événements du journal, de mots clés tels que « `Exception` survenance dans un événement du journal », etc.
   - Le modèle Log sur lequel l'anomalie est basée. Pour plus d'informations sur les modèles, consultezDétection des anomalies du journal.
   - La tendance du journal des anomalies affiche un histogramme illustrant le volume de journaux correspondant au modèle.
   - L'heure de la dernière détection indique l'heure à laquelle cette anomalie a été détectée pour la dernière fois.
   - L'heure de première détection indique la date à laquelle cette anomalie a été détectée pour la première fois.

- Le détecteur d'anomalies affiche le nom du groupe de journaux contenant les événements de journal liés à cette anomalie. Vous pouvez choisir ce nom pour afficher la page de détails du groupe de journaux.

3. Pour examiner plus en détail une anomalie, cliquez sur le bouton radio dans la rangée correspondante.

   Le volet Pattern inspect apparaît et affiche les informations suivantes :

   - Le modèle sur lequel cette anomalie est basée. Sélectionnez un jeton dans le modèle pour analyser les valeurs de ce jeton.
   - Un histogramme indiquant le nombre d'occurrences de l'anomalie sur la plage de temps demandée.
   - L'onglet Échantillons de journal affiche quelques-uns des événements du journal qui font partie de l'anomalie.
   - L'onglet Valeurs du jeton affiche les valeurs du jeton dynamique sélectionné, si vous en avez sélectionné un.

   > **Note**
   >
   > Un maximum de 10 valeurs de jeton est capturée pour chaque jeton. Le nombre de jetons peut ne pas être précis. CloudWatch Logs utilise un compteur probabiliste pour générer le nombre de jetons, et non la valeur absolue.

4. Pour supprimer une anomalie, cliquez sur le bouton radio dans sa ligne, puis procédez comme suit :

   a. Choisissez Actions, Supprimer l'anomalie.
   b. Spécifiez ensuite la durée pendant laquelle vous souhaitez que l'anomalie soit supprimée.
   c. Pour supprimer toutes les anomalies liées à ce modèle, sélectionnez Supprimer le modèle.
   d. Choisissez Supprimer l'anomalie.

Pour afficher les anomalies détectées dans un seul groupe de logs

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Journaux, Groupes de journaux.
3. Choisissez le nom d'un groupe de journaux, puis cliquez sur l'onglet Détection des anomalies.

Le tableau de détection des anomalies apparaît. Le chiffre en haut à côté de Log Anomalies indique le nombre d'anomalies log répertoriées dans le tableau. Chaque ligne du tableau affiche les informations suivantes :

- La colonne Anomalie affiche un bref résumé de l'anomalie. Ces résumés sont générés par CloudWatch Logs.
- La priorité de l'anomalie. La priorité est automatiquement calculée en fonction de l'ampleur des modifications apportées aux événements du journal, de mots clés tels que « `Exception` survenance dans un événement du journal », etc.
- Le modèle Log sur lequel l'anomalie est basée. Pour plus d'informations sur les modèles, consultez Détection des anomalies du journal.
- La tendance du journal des anomalies affiche un histogramme illustrant le volume de journaux correspondant au modèle.
- L'heure de la dernière détection indique l'heure à laquelle cette anomalie a été détectée pour la dernière fois.
- L'heure de première détection indique la date à laquelle cette anomalie a été détectée pour la première fois.

4. Pour examiner plus en détail une anomalie, cliquez sur le bouton radio dans la rangée correspondante.

   Le volet Pattern inspect apparaît et affiche les informations suivantes :

   - Le modèle sur lequel cette anomalie est basée. Sélectionnez un jeton dans le modèle pour analyser les valeurs de ce jeton.
   - Un histogramme indiquant le nombre d'occurrences de l'anomalie sur la plage de temps demandée.
   - L'onglet Échantillons de journal affiche quelques-uns des événements du journal qui font partie de l'anomalie.
   - L'onglet Valeurs du jeton affiche les valeurs du jeton dynamique sélectionné, si vous en avez sélectionné un.

> **Note**
>
> Un maximum de 10 valeurs de jeton est capturée pour chaque jeton. Le nombre de jetons peut ne pas être précis. CloudWatch Logs utilise un compteur probabiliste pour générer le nombre de jetons, et non la valeur absolue.

5. Pour supprimer une anomalie, cliquez sur le bouton radio dans sa ligne, puis procédez comme suit :
   a. Choisissez Actions, Supprimer l'anomalie.
   b. Spécifiez ensuite la durée pendant laquelle vous souhaitez que l'anomalie soit supprimée.
   c. Pour supprimer toutes les anomalies liées à ce modèle, sélectionnez Supprimer le modèle.
   d. Choisissez Supprimer l'anomalie.

Créez des alarmes sur les détecteurs d'anomalies du journal

Vous pouvez créer une alarme pour un détecteur d'anomalie du journal dans un groupe de journaux. Vous pouvez définir l'`ALARM`état de l'alarme lorsqu'un certain nombre d'anomalies sont détectées dans le groupe de journaux pendant une période spécifiée. Vous pouvez également utiliser des filtres afin que seules les anomalies correspondant à des priorités spécifiées soient prises en compte par l'alarme.

Pour créer une alarme pour un détecteur d'anomalies du journal

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Logs, Log Anomalies.

   Le tableau des détecteurs d'anomalies logarithmiques apparaît.
3. Choisissez le bouton radio du détecteur d'anomalie pour lequel vous souhaitez régler l'alarme, puis choisissez Créer une alarme.

   L'assistant de création d' CloudWatch alarmes apparaît. Le LogAnomalyDetectorchamp affiche le nom du détecteur d'anomalie que vous avez choisi. Le champ Nom de la métrique s'affiche AnomalyCount.
4. (Facultatif) Pour filtrer cette alarme en fonction de la priorité des anomalies, effectuez l'une des opérations suivantes :

- Pour que l'alarme ne compte que les anomalies prioritaires, entrez **`HIGH`** pour LogAnomalyPriority.
- Pour que l'alarme ne compte que les anomalies de priorité élevée et moyenne, entrez **`MEDIUM`** pour LogAnomalyPriority.

Pour plus d'informations sur les niveaux de priorité, consultezGravité et priorité des anomalies et des modèles.

5. Choisissez d'utiliser un seuil de détection d'anomalie statique ou métrique pour l'alarme. Cette sélection détermine le mode de réglage du seuil d'alarme. Un seuil statique signifie que le seuil d'alarme est un nombre statique et constant que vous choisissez. Un seuil de détection d'anomalie signifie qu'il CloudWatch détermine une plage de valeurs habituelles, et l'alarme se déclenche si le nombre réel dépasse le seuil de cette bande. Il n'est pas nécessaire de choisir Détection des anomalies pour enregistrer une alarme de détection des anomalies. Pour plus d'informations sur la détection des anomalies métriques, consultez la section Utilisation de la détection des CloudWatch anomalies.
6. Pour Whenever, ***`your-metric-name`*** c'est... , choisissez Plus grand, Plus grand/égal, inférieur/égal ou inférieur. Pour à . . ., spécifiez un nombre pour votre valeur de seuil. L'alarme passe en `ALARM` état si le détecteur d'anomalies détecte un nombre d'alarmes supérieur à ce nombre pendant une période spécifiée par Period.
7. Sélectionnez Additional configuration (Configuration supplémentaire). Pour Datapoints to alarm (Points de données avant l'alerte), spécifiez le nombre de périodes d'évaluation (points de données) devant être à l'état `ALARM` pour déclencher l'alerte. Si les deux valeurs sont compatibles, vous créez une alerte qui passe à l'état `ALARM` lorsque le nombre de périodes consécutives dépasse ces valeurs.

   Pour créer une alerte M sur N, spécifiez un nombre pour la première valeur qui est inférieur à celui de la deuxième valeur. Pour plus d'informations, consultez la section Évaluation d'une alarme.
8. Pour Missing data treatment (Traitement des données manquantes), choisissez comment l'alerte doit se comporter lorsqu'il manque certains points de données. Pour plus d'informations, voir Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.
9. Choisissez Suivant.
10. Pour Notification, choisissez Ajouter une notification, puis spécifiez une rubrique Amazon SNS pour vous avertir lorsque votre alarme passe à l'état `ALARMOK`, ou`INSUFFICIENT_DATA`.

a. (Facultatif) Pour envoyer plusieurs notifications pour le même état d'alarme ou pour les différents états de l'alarme, sélectionnez Add notification (Ajouter une notification).

> **Note**
>
> Nous vous recommandons de configurer l'alarme pour qu'elle prenne des mesures lorsqu'elle passe en état Données insuffisantes, en plus de lorsqu'elle passe en état Alarme. En effet, de nombreux problèmes liés à la fonction Lambda qui se connecte à la source de données peuvent entraîner le passage de l'alarme à Données insuffisantes.

b. (Facultatif) Pour ne pas envoyer de notifications Amazon SNS, choisissez Supprimer.

11. (Facultatif) Si vous souhaitez que votre alarme exécute des actions pour Amazon EC2 Auto Scaling, Amazon EC2, des tickets AWS Systems Manager, ou que vous choisissiez le bouton approprié et spécifiez l'état et l'action de l'alarme.

> **Note**
>
> Votre alarme peut effectuer des actions Systems Manager uniquement lorsqu'elle est à l'état `ALARM`. Pour plus d'informations sur les actions de Systems Manager, consultez les CloudWatch sections Configuration pour créer OpsItems et Création d'incidents.

12. Choisissez Suivant.
13. Sous Add a description (Ajouter une description), saisissez un nom et une description pour l'alerte et choisissez Next (Suivant). Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

> **Tip**
>
> Le nom de l'alarme ne peut contenir que des caractères UTF-8. Il ne peut pas contenir de caractères de contrôle ASCII.

14. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont correctes, et choisissez Create alarm (Créer une alerte).

Métriques publiées par les détecteurs d'anomalies logarithmiques

CloudWatch Logs publie la AnomalyCountmétrique dans les CloudWatch métriques. Cette métrique est publiée dans l'espace de `AWS/Logs` noms.

La AnomalyCountmétrique est publiée avec les dimensions suivantes :

- LogAnomalyDetector— Le nom du détecteur d'anomalies
- LogAnomalyPriority— Le niveau de priorité de l'anomalie

Chiffrez un détecteur d'anomalie et ses résultats avec AWS KMS

Les données des détecteurs d'anomalies sont toujours cryptées dans les CloudWatch journaux. Par défaut, CloudWatch Logs utilise le chiffrement côté serveur pour les données au repos. Comme alternative, vous pouvez utiliser AWS Key Management Service ce cryptage. Si c'est le cas, le chiffrement est effectué à l'aide d'une AWS KMS clé. Le chiffrement à l'aide AWS KMS est activé au niveau du détecteur d'anomalies, en associant une clé KMS à un détecteur d'anomalie.

> **Important**
>
> CloudWatch Logs ne prend en charge que les clés KMS symétriques. N'utilisez pas de clé asymétrique pour chiffrer les données de vos groupes de journaux. Pour plus d'informations, consultez Utilisation des clés symétriques et asymétriques.

Limites

- Pour effectuer les étapes suivantes, vous devez disposer des autorisations suivantes : `kms:CreateKey`, `kms:GetKeyPolicy` et `kms:PutKeyPolicy`.
- Une fois que vous avez associé ou dissocié une clé d'un détecteur d'anomalie, l'opération peut prendre jusqu'à cinq minutes pour prendre effet.
- Si vous révoquez l'accès de CloudWatch Logs à une clé associée ou si vous supprimez une clé KMS associée, vos données chiffrées dans CloudWatch Logs ne peuvent plus être récupérées.

Étape 1 : Création d'une AWS KMS clé

Pour créer une clé KMS, utilisez la commande create-key suivante :

```
aws kms create-key
```

La sortie contient l'ID de clé et l'Amazon Resource Name (ARN) de la clé. Voici un exemple de sortie :

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "key-default-1",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/key-default-1",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Étape 2 : Définition des autorisations sur la clé KMS

Par défaut, toutes les AWS KMS clés sont privées. Seul le propriétaire de la ressource peut l'utiliser pour chiffrer et déchiffrer des données. Cependant, le propriétaire de la ressource peut accorder à d'autres utilisateurs et ressources des autorisations d'accès à la clé KMS. Avec cette étape, vous autorisez le principal du service CloudWatch Logs à utiliser la clé. Ce principal de service doit se trouver dans la même AWS région que celle où la clé KMS est stockée.

En tant que bonne pratique, nous vous recommandons de limiter l'utilisation de la clé KMS aux seuls AWS comptes ou détecteurs d'anomalies que vous spécifiez.

Tout d'abord, enregistrez la politique par défaut pour votre clé KMS à `policy.json` l'aide de la get-key-policycommande suivante :

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Ouvrez le fichier `policy.json` dans un éditeur de texte et ajoutez la section en gras à partir de l'une des instructions suivantes. Séparez l'instruction existante de la nouvelle instruction par une virgule. Ces instructions utilisent `Condition` des sections pour renforcer la sécurité de la AWS KMS clé. Pour de plus amples informations, veuillez consulter AWS KMS clés et contexte de chiffrement.

La `Condition` section de cet exemple limite l'utilisation de la AWS KMS clé au compte spécifié, mais elle peut être utilisée pour n'importe quel détecteur d'anomalie.

JSON

```
{
"Version": "2012-10-17",
"Id": "key-default-1",
"Statement": [
    {
        "Sid": "EnableIAMUserPermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
        },
        "Action": "kms:*",
        "Resource": "*"
    },
    {
        "Sid": "AllowCloudWatchLogsEncryption",
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
        },
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:GenerateDataKey*"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "123456789012"
            },
            "StringLike": {
```

```
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsDescribeKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsReEncryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
 "arn:aws:logs:us-east-1:123456789012:anomaly-detector:*"
                },
```

```
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsDescribeKeyForReEncryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

Enfin, ajoutez la politique mise à jour à l'aide de la put-key-policycommande suivante :

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Étape 3 : Associer une clé KMS à un détecteur d'anomalies

Vous pouvez associer une clé KMS à un détecteur d'anomalies lorsque vous la créez dans la console ou à l'aide du AWS CLI ou APIs.

Étape 4 : Dissocier la clé d'un détecteur d'anomalie

Une fois qu'une clé a été associée à un détecteur d'anomalie, vous ne pouvez pas la mettre à jour. La seule façon de retirer la clé est de supprimer le détecteur d'anomalie, puis de le recréer.

Résoudre les problèmes avec CloudWatch Logs Live Tail

CloudWatch Logs Live Tail vous aide à résoudre rapidement les incidents en visualisant la liste des nouveaux événements enregistrés au fur et à mesure de leur ingestion. Vous pouvez afficher, filtrer et mettre en évidence les journaux ingérés en temps quasi réel, ce qui vous permet de détecter et de résoudre rapidement les problèmes. Vous pouvez filtrer les journaux en fonction des termes que vous spécifiez et mettre en évidence les journaux qui contiennent les termes spécifiés pour vous aider à trouver rapidement ce que vous cherchez.

Le coût des sessions Live Tail est calculé en fonction du temps d'utilisation de la session, par minute. Pour plus d'informations sur les tarifs, consultez l'onglet Logs sur Amazon CloudWatch Pricing.

Live Tail n'est pris en charge que pour les groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

Les sections suivantes expliquent comment utiliser Live Tail dans la console et dans le AWS CLI. Vous pouvez également démarrer une session Live Tail par programmation. Pour de plus amples informations, veuillez consulter StartLiveTail. Pour des exemples de SDK, voir Démarrer une session Live Tail à l'aide d'un AWS SDK.

Vous pouvez également utiliser Live Tail dans le AWS Toolkit for Visual Studio Code. Pour démarrer une session Live Tail à partir de la palette de commandes VS Code, consultez la section Amazon CloudWatch Logs Live Tail du guide de l' AWS Toolkit for Visual Studio Code utilisateur.

La fonctionnalité Live Tail est disponible dans toutes les AWS régions commerciales. Il n'est pas disponible dans les régions de Chine ni dans les régions AWS GovCloud (États-Unis).

> **Note**
>
> L'`StartLiveTail`API achemine les demandes vers `streaming-logs.`*`Region`*`.amazonaws.com` l'aide de l'injection de préfixe hôte du SDK. La prise en charge des points de terminaison VPC n'est pas disponible pour cette API.

Démarrez une session Live Tail à l'aide du AWS CLI

La `start-live-tail` AWS CLI commande démarre une session de streaming Live Tail pour un ou plusieurs groupes de journaux dans un terminal. Une session Live Tail peut durer jusqu'à trois

heures. Si plus de 500 événements de journal par seconde correspondent au filtre, les événements de journal affichés sont un échantillon du nombre total d'événements de journal, afin de fournir une expérience de suivi en temps réel. Pour plus d'informations sur la `start-live-tail` commande, voir start-live-tail

Vous pouvez l'utiliser `start-live-tail` dans deux modes :

- impression uniquement : il s'agit du mode par défaut
- interactif

impression uniquement

En `print-only` mode, les événements du journal sont diffusés sur le terminal. De nouveaux événements sont ajoutés en bas de page chaque seconde, créant une expérience de suivi en temps quasi réel similaire à celle de `tail -f` Linux.

Pour démarrer une session Live Tail en mode impression uniquement, entrez la commande suivante.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs
```

Lorsque vous utilisez le mode d'impression uniquement, vous pouvez également le rediriger vers d'autres commandes Linux afin d'augmenter ses capacités analytiques. L'exemple suivant filtre les événements du journal à l'aide du `error` mot clé et imprime les deuxième et quatrième colonnes de ces événements pour vous aider à extraire des informations spécifiques.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs --mode print-only | grep "error" | awk '{print
 $2, $4}'
```

interactif

En `interactive` mode, vous pouvez surligner des termes et changer le format des événements du journal de sortie entre JSON et texte brut. Le mode interactif affiche également des informations sur la session Live Tail, telles que la durée de la session, si la session est échantillonnée, les termes actuellement surlignés et le nombre de fois qu'ils ont été rencontrés.

Pour démarrer une session Live Tail en mode interactif, entrez la commande suivante.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs --mode interactive
```

La session Live Tail commence. La vidéo suivante montre une partie d'une session d'exemple.

Pour surligner un terme dans les journaux de diffusion, appuyez sur h, puis saisissez-le. Ce qui suit montre l'écran une fois que le terme `latency` a été surligné.

Pour effacer un terme surligné, appuyez sur c, puis tapez le chiffre qui représente le terme que vous souhaitez arrêter de surligner.

Vous pouvez appuyer sur t pour passer du format d'affichage des événements entrants au format JSON au texte brut. Cette fonctionnalité de bascule est optimale et n'est disponible que si le format du journal des événements est compatible.

Vous pouvez utiliser les flèches vers le haut et vers le bas pour faire défiler la page, et utiliser `CTRL +u` et `CTRL+d` pour faire défiler la page plus rapidement.

L'image suivante montre le surlignage du `latency` terme lors d'une session Live Tail.

Démarrer une session Live Tail dans la console

Vous utilisez la CloudWatch console pour démarrer une session Live Tail. La procédure suivante explique comment démarrer une session Live Tail à l'aide de l'option Live Tail dans le volet de navigation de gauche. Vous pouvez également démarrer des sessions Live Tail depuis la page Log Groups ou CloudWatch Logs Insights.

Si vous utilisez des politiques de protection des données pour masquer des données sensibles dans un groupe de journaux que vous consultez avec Live Tail, les données sensibles apparaissent toujours masquées dans la session Live Tail. Pour plus d'informations sur la protection des données dans les groupes de journaux, veuillez consulter Aider à protéger les données sensibles des journaux grâce au masquage.

> **Important**
>
> Si votre équipe de sécurité réseau n'autorise pas l'utilisation de sockets Web, vous ne pouvez actuellement pas accéder à la partie Live Tail de la CloudWatch console. Vous pouvez

utiliser Live Tail avec le AWS CLI ou APIs. Pour plus d'informations, consultez Démarrez une session Live Tail à l'aide du AWS CLI et StartLiveTail.

Pour démarrer une session Live Tail

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Journaux, Live Tail.
3. Pour Sélectionner les groupes de journaux, sélectionnez les groupes de journaux dont vous souhaitez consulter les événements, dans la session Live Tail. Vous pouvez sélectionner jusqu'à 10 groupes de journaux.
4. (Facultatif) Si vous n'avez sélectionné qu'un seul groupe de journaux, vous pouvez filtrer davantage votre session Live Tail en sélectionnant un ou plusieurs flux de journaux à partir desquels consulter les événements du journal. Pour ce faire, sous Sélectionner les flux de journaux, sélectionnez les noms des flux de journaux dans la liste déroulante. Vous pouvez également utiliser la seconde zone située sous Sélectionner les flux de journaux pour saisir un préfixe de nom de flux de journaux, puis tous les flux de journaux dont le nom correspond au préfixe seront sélectionnés.
5. (Facultatif) Pour afficher uniquement les événements du journal contenant certains mots ou d'autres chaînes de caractères, saisissez le mot ou la chaîne de caractères dans `Add filter patterns`.

   Par exemple, pour afficher uniquement les événements du journal qui incluent le mot `Warning`, saisissez **`Warning`**. Le champ des filtres est sensible à la casse. Vous pouvez inclure plusieurs termes et opérateurs de modèles dans ce champ :

   - **`error 404`** n'affiche que les événements du journal qui incluent à la fois `error` et `404`
   - **`?Error ?error`** affiche les événements du journal qui incluent `Error` ou `error`
   - **`-INFO`** affiche tous les événements du journal qui incluent `INFO`
   - **`{ $.eventType = "UpdateTrail" }`** affiche tous les événements du journal JSON dont la valeur du champ de type d'événement est `UpdateTrail`

   Vous pouvez également utiliser une expression régulière (regex) pour filtrer :

   - **`%ERROR%`** utilise une expression régulière pour afficher tous les événements du journal contenant le mot clé ERROR

- **`{ $.names = %Steve% }`** utilise une expression régulière pour afficher les événements du journal JSON où Steve se trouve dans la propriété `"name"`
- **`[ w1 = %abc%, w2 ]`** utilise une expression régulière pour afficher les événements du journal délimités par des espaces, où le premier mot est abc

Pour plus d'informations sur la syntaxe des modèles, consultez Syntaxe des modèles de filtres.

6. (Facultatif) Pour mettre en évidence certains des événements du journal affichés, saisissez un terme à rechercher et à mettre en évidence sous Live Tail. Saisissez les termes à mettre en évidence un par un. Si vous ajoutez plusieurs termes à mettre en évidence, une couleur différente est attribuée pour représenter chaque terme. Un indicateur de mise en évidence s'affiche à gauche de tout événement du journal contenant le terme spécifié et apparaît également sous le terme lui-même lorsque vous agrandissez l'événement du journal dans la fenêtre principale pour afficher l'événement du journal complet.

   Vous pouvez utiliser le filtrage et la mise en évidence pour résoudre rapidement les problèmes. Par exemple, vous pouvez filtrer les événements pour n'afficher que les événements qui contiennent `Error`, puis également mettre en évidence les événements qui en contiennent `404`.

7. Pour démarrer la session, choisissez Appliquer les filtres

   Les événements du journal correspondants commencent à apparaître dans la fenêtre. Les informations suivantes sont également affichées :

   - Le chronomètre indique depuis combien de temps la session Live Tail est active.
   - events/sec affiche le nombre d'événements de journal ingérés par seconde qui correspondent aux filtres que vous avez définis.
   - Pour éviter que la session ne défile trop vite car de nombreux événements correspondent aux filtres, il est possible que les CloudWatch journaux n'affichent que certains événements correspondants. Dans ce cas, le pourcentage d'événements correspondants qui s'affichent à l'écran est indiqué en % affiché.

8. Pour suspendre le flux d'événements afin d'examiner ce qui est actuellement affiché, cliquez n'importe où dans la fenêtre des événements.

9. Au cours de la session, vous pouvez utiliser ce qui suit pour obtenir plus de détails sur chaque événement du journal.

   - Pour afficher le texte complet d'un événement du journal dans la fenêtre principale, cliquez sur la flèche en regard de cet événement du journal.

- Pour afficher le texte complet d'un événement du journal dans une fenêtre latérale, choisissez la loupe + en regard de cet événement du journal. Le flux d'événements s'interrompt et la fenêtre latérale s'affiche.

  L'affichage du texte d'un événement du journal dans la fenêtre latérale peut être utile pour comparer son texte aux autres événements de la fenêtre principale.

10. Pour arrêter la session Live Tail, choisissez Arrêter.
11. Pour redémarrer la session, utilisez éventuellement le volet Filtre pour modifier les critères de filtrage, puis choisissez Appliquer les filtres. Choisissez ensuite Start (Démarrer).

Utilisation des groupes de journaux et des flux de journaux

Un flux de journal est une séquence d'événements du journaux qui partagent la même source. Chaque source distincte de CloudWatch journaux dans Logs constitue un flux de journaux distinct.

Un groupe de journaux est un groupe de flux de journaux qui partagent les mêmes paramètres de conservation, de surveillance et de contrôle d'accès. Vous pouvez définir des groupes de journaux et spécifier les flux à placer dans chaque groupe. Le nombre de flux de journaux pouvant appartenir à un groupe de journaux est illimité.

Vous pouvez utiliser les procédures décrites dans cette section pour travailler avec des groupes de journaux et des flux de journaux.

Création d'un groupe de CloudWatch journaux dans Logs

Lorsque vous installez l'agent CloudWatch Logs sur une EC2 instance Amazon en suivant les étapes décrites dans les sections précédentes du guide de l'utilisateur Amazon CloudWatch Logs, le groupe de journaux est créé dans le cadre de ce processus. Vous pouvez également créer un groupe de journaux directement dans la CloudWatch console.

Pour créer un groupe de journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Choisissez Actions, puis Create Bucket (Créer un compartiment).
4. Tapez un nom pour le groupe de journaux, puis choisissez Créer un groupe de journaux.

> **Tip**
>
> Vous pouvez mettre en favoris les groupes de journaux, ainsi que les tableaux de bord et les alarmes, à partir du menu Favorites and recents (Favoris et récents) dans le volet de navigation. Sous la colonne Recetly visited (Visité récemment), survolez le groupe de journaux que vous souhaitez mettre en favoris et choisissez le symbole étoile à côté de celui-ci.

Envoi de journaux à un groupe de journaux

CloudWatch Logs reçoit automatiquement les événements du journal provenant de plusieurs AWS services. Vous pouvez également envoyer d'autres événements de journal à CloudWatch Logs en utilisant l'une des méthodes suivantes :

- CloudWatch agent — L' CloudWatch agent unifié peut envoyer à la fois des métriques et des CloudWatch journaux à Logs. Pour plus d'informations sur l'installation et l'utilisation de l' CloudWatch agent, consultez la section Collecte de métriques et de journaux à partir d' EC2 instances Amazon et de serveurs sur site avec l' CloudWatch agent dans le guide de l' CloudWatch utilisateur Amazon.
- AWS CLI: put-log-eventstélécharge des lots d'événements du journal dans Logs. CloudWatch
- Par programmation : l'PutLogEventsAPI vous permet de télécharger par programmation des lots d'événements de journal dans Logs. CloudWatch

Afficher les données du journal envoyées à CloudWatch Logs

Vous pouvez afficher et parcourir les données des journaux telles qu'elles sont envoyées à CloudWatch Logs par l'agent CloudWatch Logs. stream-by-stream Vous pouvez spécifier la plage de temps pour les données de journal à afficher.

Pour afficher les données des journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Pour Log Groups, choisissez le groupe de journaux pour afficher les flux.
4. Dans la liste des groupes de journaux, choisissez le nom du groupe de journaux que vous souhaitez afficher.
5. Dans la liste des flux de journaux, choisissez le nom du flux de journaux que vous souhaitez afficher.
6. Pour modifier la façon dont les données de journal sont affichées, effectuez l'une des actions suivantes :
   - Pour développer un événement de journal, sélectionnez la flèche en regard de celui-ci.
   - Pour développer tous les événements de journaux et les afficher sous forme de texte brut, au-dessus de la liste des événements de journaux, choisissez Text.

- Pour filtrer les événements de journaux, saisissez le filtre de recherche souhaité dans le champ de recherche. Pour plus d'informations, consultez Création de métriques à partir d'événements du journal à l'aide de filtres.
- Pour afficher les données de journal pour une plage de dates et d'heures spécifiée, sélectionnez la flèche en regard de la date et de l'heure, près du filtre de recherche. Pour spécifier une plage de dates et d'heures, choisissez Absolu. Pour choisir un nombre prédéfini de minutes, d'heures, de jours ou de semaines, choisissez Relatif. Vous pouvez également basculer entre UTC et le fuseau horaire local.

Recherche de données journal au moyen de modèles de filtres

Vous pouvez effectuer une recherche dans les données de vos journaux en utilisant la Syntaxe des modèles de filtres pour les filtres de métriques, les filtres d'abonnements, les filtres d'événements du journal et Live Tail. Vous pouvez rechercher tous les flux de journaux au sein d'un groupe de journaux ou, en utilisant le, AWS CLI vous pouvez également rechercher des flux de journaux spécifiques. A chaque exécution, la recherche renvoie la première page de données trouvées et un jeton pour récupérer la page de données suivante ou pour continuer la recherche. Si aucun résultat n'est obtenu, vous pouvez continuer la recherche.

Vous pouvez définir la plage de temps que vous souhaitez interroger pour limiter la portée de votre recherche. Vous pouvez commencer par une durée plus importante pour voir où se trouvent les lignes de journaux qui vous intéressent, puis la réduire afin de consulter des journaux spécifiques de cette période.

Vous pouvez également transiter directement des métriques extraites de vos journaux aux journaux correspondants.

Si vous êtes connecté à un compte configuré en tant que compte de surveillance dans l'observabilité CloudWatch entre comptes, vous pouvez rechercher et filtrer les événements du journal à partir des comptes sources liés à ce compte de surveillance. Pour plus d'informations, consultez la section Observabilité inter-comptes de CloudWatch (français non garanti).

Recherche d'entrées de journal à l'aide de la console

Vous pouvez rechercher des entrées de journal qui correspondent à des critères spécifiés à partir de la console.

Pour effectuer une recherche dans vos journaux à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Pour Log Groups, choisissez le nom du groupe de journaux contenant le flux de journal devant faire l'objet de la recherche.
4. Pour Log Streams, choisissez le nom du flux de journal devant faire l'objet de la recherche.
5. Sous Journal des événements, saisissez la syntaxe du filtre à utiliser.

Pour effectuer une recherche dans toutes les entrées de journal pour une plage de temps donnée à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Pour Log Groups, choisissez le nom du groupe de journaux contenant le flux de journal devant faire l'objet de la recherche.
4. Choisissez Rechercher un groupe de journaux.
5. Dans Événements de journaux, sélectionnez la plage de date et d'heure, puis saisissez la syntaxe du filtre.

Recherchez les entrées du journal à l'aide du AWS CLI

Vous pouvez rechercher des entrées de journal qui répondent à des critères spécifiques à l'aide du AWS CLI.

Pour rechercher des entrées de journal à l'aide du AWS CLI

À partir d'une invite de commande, exécutez la commande filter-log-events suivante. Utilisez `--filter-pattern` pour limiter les résultats au modèle de filtre spécifié et `--log-stream-names` pour limiter les résultats au flux de journaux spécifiés.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Pour rechercher des entrées de journal sur une période donnée à l'aide du AWS CLI

À l'invite de commande, exécutez la filter-log-eventscommande suivante :

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-
time 1482217558365] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Transition des métriques aux journaux

Vous pouvez accéder à des entrées de journal spécifiques à partir d'autres emplacements de la console.

Pour passer de widgets de tableau de bord à des journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Dashboards (Tableaux de bord).
3. Choisissez un tableau de bord.
4. Sur le widget, choisissez l'icône View logs, puis choisissez View logs in this time range. S'il existe plusieurs filtres de métriques, sélectionnez-en un dans la liste. Si le nombre de filtres de métriques est supérieur au nombre de filtres pouvant être affichés dans la liste, choisissez More metric filters, puis sélectionnez ou recherchez un filtre de métrique.

Pour passer des métriques aux journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Dans le champ de recherche situé sous l'onglet All metrics, saisissez le nom de la métrique et appuyez sur Entrée.
4. Sélectionnez une ou plusieurs métriques dans les résultats de la recherche.
5. Choisissez Actions, View logs. S'il existe plusieurs filtres de métriques, sélectionnez-en un dans la liste. Si le nombre de filtres de métriques est supérieur au nombre de filtres pouvant être affichés dans la liste, choisissez More metric filters, puis sélectionnez ou recherchez un filtre de métrique.

Résolution des problèmes

La recherche prend trop longtemps

Si vous avez beaucoup de données de journal, la recherche peut prendre beaucoup de temps. Afin d'accélérer le processus, vous pouvez procéder comme indiqué ci-dessous :

- Si vous utilisez le AWS CLI, vous pouvez limiter la recherche aux flux de journaux qui vous intéressent. Par exemple, si votre groupe de journaux compte 1 000 flux de journaux, mais que vous souhaitez uniquement voir trois flux de journaux dont vous savez qu'ils sont pertinents, vous pouvez utiliser le AWS CLI pour limiter votre recherche aux trois flux de journaux du groupe de journaux uniquement.
- Utilisez une période plus courte, à niveau de détail supérieur, ce qui réduit la quantité de données dans laquelle effectuer la recherche et accélère la requête.

Conservation des données du journal des modifications dans CloudWatch les journaux

Par défaut, les données du journal sont stockées dans les CloudWatch journaux indéfiniment. Vous pouvez néanmoins configurer la durée de stockage des données de journaux dans un groupe de journaux. Toutes les données antérieures au paramètre de conservation actuel sont supprimées. Vous pouvez modifier la durée de conservation des journaux pour chaque groupe de journaux à tout moment.

> **Note**
>
> CloudWatch Logs ne supprime pas immédiatement les événements du journal lorsqu'ils atteignent leur paramètre de rétention. Cela prend généralement jusqu'à 72 heures avant que les événements du journal ne soient supprimés, mais dans de rares cas, cela peut prendre plus de temps.
>
> Cela signifie que si vous modifiez un groupe de journaux pour qu'il dispose d'un paramètre de conservation plus long lorsqu'il contient des événements du journal qui ont dépassé la date d'expiration, mais qui n'ont pas été réellement supprimés, ces événements du journal prendront jusqu'à 72 heures pour être supprimés une fois la nouvelle date de conservation atteinte. Pour vous assurer que les données du journal sont supprimées définitivement, conservez un groupe de journaux à son paramètre de rétention inférieur jusqu'à ce que 72 heures se soient écoulées après la fin de la période de conservation précédente, ou que vous ayez confirmé que les anciens événements du journal sont supprimés.
>
> Lorsque les événements du journal atteignent leur paramètre de rétention, ils sont marqués pour suppression. Une fois qu'ils sont marqués pour suppression, ils n'augmentent plus vos

coûts de stockage d'archives, même s'ils ne sont supprimés que plus tard. Ces événements du journal marqués pour suppression ne sont pas non plus inclus lorsque vous utilisez une API pour récupérer la valeur `storedBytes` afin de connaître le nombre d'octets stockés par un groupe de journaux.

Pour modifier le paramètre de conservation des journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), Log groups (Groupes de journaux).
3. Recherchez le groupe de journaux à mettre à jour.
4. Dans la colonne Conservation de ce groupe de journaux, choisissez le paramètre de rétention actuel, tel que Never Expire.
5. Dans Paramètre de rétention, pour Expirer les événements après, choisissez une valeur de rétention du journal, puis sélectionnez Enregistrer.

Étiqueter les groupes de journaux dans Amazon CloudWatch Logs

Vous pouvez attribuer vos propres métadonnées aux groupes de journaux que vous créez dans Amazon CloudWatch Logs sous forme de balises. Une balise est une paire clé-valeur que vous définissez pour un groupe de journaux. 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.

> **Note**
>
> Vous pouvez utiliser des balises pour contrôler l'accès aux ressources CloudWatch des journaux, notamment aux groupes de journaux et aux destinations. L'accès aux flux de journaux est contrôlé au niveau du groupe de journaux, en raison de la relation hiérarchique entre les groupes de journaux et les flux de journaux. Pour plus d'informations sur l'utilisation d'identifications pour contrôler l'accès, consultez Contrôle de l'accès aux ressources Amazon Web Services à l'aide de balises.

Table des matières

- Principes de base des balises

- Suivi des coûts à l'aide d'étiquettes
- Restrictions liées aux étiquettes
- Marquage de groupes de journaux à l'aide du AWS CLI
- Marquage de groupes de journaux à l'aide de l'API CloudWatch Logs

Principes de base des balises

Vous utilisez AWS CloudFormation l' AWS CLI API ou CloudWatch Logs pour effectuer les tâches suivantes :

- Ajout de balises à un groupe de journaux au moment de sa création
- Ajout de balises à un groupe de journaux existant
- Affichage de la liste de balises d'un groupe de journaux
- Suppression de balises d'un groupe de journaux

Vous pouvez utiliser des balises pour classer vos groupes de journaux par catégories. Par exemple, vous pouvez les classer 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. Ainsi, vous pouvez définir un ensemble de balises vous permettant de suivre les groupes de journaux par propriétaire et application associée. Voici quelques exemples de balises :

- Projet : nom du projet
- Propriétaire : nom
- Objectif : test de la charge
- Application : nom de l'application
- Environnement : production

Suivi des coûts à l'aide d'étiquettes

Vous pouvez utiliser des balises pour classer et suivre vos AWS coûts. Lorsque vous appliquez des balises à vos AWS ressources, y compris à des groupes de journaux, votre rapport de répartition des AWS coûts inclut l'utilisation et les coûts agrégés par balises. Vous pouvez appliquer des balises associées à des catégories métier (telles que les centres de coûts, les noms d'applications ou les

propriétaires) pour organiser les coûts relatifs à divers services. Pour plus d'informations, consultez Utilisation des identifications de répartition des coûts pour les rapports de facturation personnalisés dans le Guide de l'utilisateur AWS Billing .

Restrictions liées aux étiquettes

Les restrictions suivantes s'appliquent aux balises :

Restrictions de base

- Le nombre maximal de balises par groupe de journaux est 50.
- Les clés et valeurs de balise sont sensibles à la casse.
- Vous ne pouvez pas changer ou modifier les balises d'un groupe de journaux 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 commencer une clé de balise par, `aws:` car ce préfixe est réservé à l'usage de AWS. AWS crée des balises qui commencent par ce préfixe en votre nom, mais vous ne pouvez ni les modifier ni les supprimer.
- Les clés de balise doivent comporter entre 1 et 128 caractères Unicode.
- Les clés de balise doivent comporter les caractères suivants : lettres Unicode, chiffres, espaces et les caractères spéciaux suivants :`_ . / = + - @`.

Restrictions relatives à la valeur de balise

- Les valeurs de balise doivent comporter entre 0 et 255 caractères Unicode.
- Les valeurs de balise peuvent être vides. Si tel n'est pas le cas, elles doivent être composées des caractères suivants : lettres Unicode, chiffres, espaces et les caractères spéciaux suivants :`_ . / = + - @`.

Marquage de groupes de journaux à l'aide du AWS CLI

Vous pouvez ajouter, répertorier et supprimer des balises à l'aide de l' AWS CLI. Pour obtenir des exemples, consultez la documentation suivante :

create-log-group

Crée un groupe de journaux. Vous pouvez éventuellement ajouter des balises au moment de créer le groupe de journaux.

tag-resource

Affecte une ou plusieurs balises (paires clé-valeur) à la ressource Logs spécifiée CloudWatch .

list-tags-for-resource

Affiche les balises associées à une ressource CloudWatch Logs.

untag-resource

Supprime une ou plusieurs balises de la ressource CloudWatch Logs spécifiée.

Marquage de groupes de journaux à l'aide de l'API CloudWatch Logs

Vous pouvez ajouter, répertorier et supprimer des balises à l'aide de l'API CloudWatch Logs. Pour obtenir des exemples, consultez la documentation suivante :

CreateLogGroup

Crée un groupe de journaux. Vous pouvez éventuellement ajouter des balises au moment de créer le groupe de journaux.

TagResource

Affecte une ou plusieurs balises (paires clé-valeur) à la ressource Logs spécifiée CloudWatch .

ListTagsForResource

Affiche les balises associées à une ressource CloudWatch Logs.

UntagResource

Supprime une ou plusieurs balises de la ressource CloudWatch Logs spécifiée.

Chiffrez les données du journal dans CloudWatch Logs à l'aide de AWS Key Management Service

Les données des groupes de journaux sont toujours cryptées dans CloudWatch Logs. Par défaut, CloudWatch Logs utilise le chiffrement côté serveur avec le mode AES-GCM (Advanced Encryption

Standard Galois/Counter Mode) 256 bits pour chiffrer les données des journaux au repos. En guise d'alternative, vous pouvez utiliser AWS Key Management Service pour ce chiffrement. Si c'est le cas, le chiffrement est effectué à l'aide d'une AWS KMS clé. L'utilisation du chiffrement AWS KMS est activée au niveau du groupe de journaux, en associant une clé KMS à un groupe de journaux, soit lorsque vous créez le groupe de journaux, soit après son existence.

> **Important**
>
> CloudWatch Les journaux prennent désormais en charge le contexte de chiffrement, en utilisant `kms:EncryptionContext:aws:logs:arn` comme clé et l'ARN du groupe de journaux comme valeur de cette clé. Si vous disposez de groupes de journaux que vous avez déjà chiffrés avec une clé KMS et que vous souhaitez restreindre l'utilisation de cette clé à un seul compte et à un seul groupe de journaux, vous devez affecter une nouvelle clé KMS incluant une condition dans la politique IAM. Pour de plus amples informations, veuillez consulter AWS KMS clés et contexte de chiffrement.

> **Important**
>
> CloudWatch Logs prend désormais en charge `kms:ViaService` ce qui permet aux journaux de passer AWS KMS des appels en votre nom. Vous devez l'ajouter à vos rôles qui appellent CloudWatch Logs soit dans votre Key Policy, soit dans IAM. Pour plus d'informations, voir kms : ViaService.

Dès lors que vous associez une clé KMS à un groupe de journaux, toutes les données nouvellement ingérées pour le groupe de journaux sont chiffrées à l'aide de cette clé. Ces données sont stockées sous forme cryptée pendant toute la durée de conservation. CloudWatch Logs déchiffre ces données chaque fois qu'elles sont demandées. CloudWatch Les journaux doivent disposer d'autorisations pour la clé KMS chaque fois que des données chiffrées sont demandées.

Si vous dissociez ultérieurement une clé KMS d'un groupe de CloudWatch journaux, Logs chiffre les données nouvellement ingérées à l'aide de la méthode de chiffrement par défaut de CloudWatch Logs. Toutes les données précédemment ingérées qui ont été chiffrées avec la clé KMS restent chiffrées avec la clé KMS. CloudWatch Les journaux peuvent toujours renvoyer ces données une fois la clé KMS dissociée, car CloudWatch les journaux peuvent continuer à faire référence à la clé. Toutefois, si la clé est désactivée ultérieurement, CloudWatch Logs ne pourra pas lire les journaux chiffrés avec cette clé.

> **Important**
>
> CloudWatch Logs ne prend en charge que les clés KMS symétriques. N'utilisez pas de clé asymétrique pour chiffrer les données de vos groupes de journaux. Pour plus d'informations, consultez Utilisation des clés symétriques et asymétriques.

Limites

- Pour effectuer les étapes suivantes, vous devez disposer des autorisations suivantes : `kms:CreateKey`, `kms:GetKeyPolicy` et `kms:PutKeyPolicy`.
- L'association et la dissociation d'une clé vis-à-vis d'un groupe de journaux peuvent mettre jusqu'à cinq minutes à être prises en compte.
- Si vous révoquez l'accès de CloudWatch Logs à une clé associée ou si vous supprimez une clé KMS associée, vos données chiffrées dans CloudWatch Logs ne peuvent plus être récupérées.
- Vous ne pouvez pas associer une clé KMS à un groupe de journaux existant à l'aide de la CloudWatch console.

Étape 1 : Création d'une AWS KMS clé

Pour créer une clé KMS, utilisez la commande create-key suivante :

```
aws kms create-key
```

La sortie contient l'ID de clé et l'Amazon Resource Name (ARN) de la clé. Voici un exemple de sortie :

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
```

```
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Étape 2 : Définition des autorisations sur la clé KMS

Par défaut, toutes les AWS KMS clés sont privées. Seul le propriétaire de la ressource peut l'utiliser pour chiffrer et déchiffrer des données. Cependant, le propriétaire de la ressource peut accorder à d'autres utilisateurs et ressources des autorisations d'accès à la clé KMS. Au cours de cette étape, vous autorisez le principal du service CloudWatch Logs et le rôle de l'appelant à utiliser la clé. Ce principal de service doit se trouver dans la même AWS région que celle où la clé KMS est stockée.

Il est recommandé de limiter l'utilisation de la clé KMS aux seuls AWS comptes ou groupes de journaux que vous spécifiez.

Tout d'abord, enregistrez la politique par défaut pour votre clé KMS à `policy.json` l'aide de la get-key-policycommande suivante :

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Ouvrez le fichier `policy.json` dans un éditeur de texte et ajoutez la section en gras à partir de l'une des instructions suivantes. Séparez l'instruction existante de la nouvelle instruction par une virgule. Ces instructions utilisent `Condition` des sections pour renforcer la sécurité de la AWS KMS clé. Pour de plus amples informations, veuillez consulter AWS KMS clés et contexte de chiffrement.

La section `Condition` de cet exemple restreint la clé à un seul ARN de groupe de journaux.

JSON

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
```

```
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333:log-group:log-group-name"
                }
            }
        }
    ]
}
```

La section `Condition` de cet exemple limite l'utilisation de la clé AWS KMS au compte spécifié, mais elle peut être utilisée pour n'importe quel groupe de journaux.

JSON

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
```

```
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:123456789012:*"
                }
            }
        }
    ]
}
```

Ajoutez ensuite des autorisations au rôle qui appellera les CloudWatch journaux. Vous pouvez le faire en ajoutant une déclaration supplémentaire à la politique AWS KMS clé ou via IAM sur le rôle lui-même. CloudWatch Enregistre `kms:ViaService` les utilisations pour passer AWS KMS des appels au nom du client. Pour plus d'informations, voir kms : ViaService.

Pour ajouter des autorisations dans la politique AWS KMS clé, ajoutez la déclaration supplémentaire suivante à votre politique clé. Si vous utilisez cette méthode, il est recommandé d'étendre la politique aux seuls rôles qui interagiront avec des groupes de journaux AWS KMS chiffrés.

```
{
  "Effect": "Allow",
```

```
  "Principal": {
    "AWS": "arn:aws:iam::account_id:role/role_name"
  },
  "Action": [
    "kms:Encrypt",
    "kms:ReEncrypt*",
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:Describe*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "logs.region.amazonaws.com"
      ]
    }
  }
}
```

Si vous souhaitez gérer les autorisations des rôles dans IAM, vous pouvez également ajouter des autorisations équivalentes par le biais de la politique suivante. Cela peut être ajouté à une politique de rôle existante ou attaché à un rôle en tant que stratégie distincte supplémentaire. Si vous utilisez cette méthode, il est recommandé d'étendre la politique aux seules AWS KMS clés qui seront utilisées pour le chiffrement des journaux. Pour plus d'informations, consultez la section Modifier les politiques IAM.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:ReEncrypt*",
                "kms:Decrypt",
                "kms:GenerateDataKey*",
                "kms:Describe*"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "logs.us-east-1.amazonaws.com"
                    ]
                }
            },
            "Resource": "arn:aws:kms:us-east-1:444455556666:key/key_id"
        }
    ]
}
```

Enfin, ajoutez la politique mise à jour à l'aide de la put-key-policycommande suivante :

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Étape 3 : Association d'une clé KMS à un groupe de journaux

Vous pouvez associer une clé KMS à un groupe de journaux lors de sa création, ou ultérieurement.

Pour savoir si une clé KMS est déjà associée à un groupe de journaux, utilisez la describe-log-groupscommande suivante :

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

Si la sortie inclut un champ `kmsKeyId`, le groupe de journaux est associé à la clé affichée pour la valeur de ce champ.

Pour associer la clé KMS à un groupe de journaux lors de sa création

Utilisez la commande create-log-group comme suit :

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

Pour associer la clé KMS à un groupe de journaux existant

Utilisez la commande associate-kms-key comme suit :

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

Étape 4 : Dissociation de la clé d'un groupe de journaux

Pour dissocier la clé KMS associée à un groupe de journaux, utilisez la disassociate-kms-keycommande suivante :

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

AWS KMS clés et contexte de chiffrement

Pour renforcer la sécurité de vos AWS Key Management Service clés et de vos groupes de CloudWatch journaux chiffrés, Logs intègre désormais le groupe ARNs de journaux dans le contexte de chiffrement utilisé pour chiffrer les données de vos journaux. Le contexte de chiffrement est un ensemble de paires clé-valeur qui sont utilisées comme données authentifiées supplémentaires. Le contexte de chiffrement vous permet d'utiliser les conditions de la politique IAM pour limiter l'accès à votre AWS KMS clé par AWS compte et par groupe de journaux. Pour plus d'informations, consultez Contexte de chiffrement et Éléments de politique JSON IAM : Condition.

Nous vous recommandons d'utiliser différentes clés KMS pour chacun de vos groupes de journaux chiffrés.

Si vous disposez d'un groupe de journaux déjà chiffré et que vous souhaitez le modifier afin d'utiliser une nouvelle clé KMS dédiée à celui-ci, procédez comme suit.

Pour convertir un groupe de journaux chiffré afin d'utiliser une clé KMS avec une politique la limitant à ce groupe de journaux

1. Saisissez la commande suivante pour trouver l'ARN de la clé actuelle du groupe de journaux :

   ```
   aws logs describe-log-groups
   ```

   La sortie comprend la ligne suivante. Prenez note de l'ARN. Vous devez l'utiliser à l'étape 7.

   ```
   ...
   "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
   cdef-0123-456789abcdef"
   ...
   ```

2. Saisissez la commande suivante pour créer une nouvelle clé KMS :

```
aws kms create-key
```

3. Entrez la commande suivante pour enregistrer la stratégie de la nouvelle clé dans un fichier `policy.json` :

```
aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
policy.json
```

4. Utilisez un éditeur de texte pour ouvrir `policy.json` et ajouter une expression `Condition` à la stratégie :

   JSON

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "ArnLike": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333:log-group:LOG-GROUP-NAME"
                }
            }
        }
    ]
}
```

5. Saisissez la commande suivante pour ajouter la politique mise à jour à la nouvelle clé KMS :

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. Entrez la commande suivante pour associer la stratégie à votre groupe de journaux :

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

CloudWatch Logs chiffre désormais toutes les nouvelles données à l'aide de la nouvelle clé.

7. Ensuite, révoquez toutes les autorisations à l'exception de `Decrypt` provenant de l'ancienne clé. Tout d'abord, saisissez la commande suivante pour récupérer l'ancienne politique :

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text
 > ./policy.json
```

8. Utilisez un éditeur de texte pour ouvrir `policy.json` et supprimer toutes les valeurs de la liste `Action`, à l'exception de `kms:Decrypt`

JSON

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

9. Saisissez la commande suivante pour ajouter la politique mise à jour à l'ancienne clé :

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

Aider à protéger les données sensibles des journaux grâce au masquage

Vous pouvez contribuer à protéger les données sensibles ingérées par CloudWatch Logs en utilisant les politiques de protection des données des groupes de journaux. Ces politiques vous permettent d'auditer et de masquer les données sensibles qui apparaissent dans les événements du journal ingérés par les groupes de journaux de votre compte.

Lorsque vous créez une politique de protection des données, les données sensibles correspondant aux identifiants de données que vous avez sélectionnés sont masquées par défaut à tous les points de sortie, y compris CloudWatch Logs Insights, les filtres métriques et les filtres d'abonnement. Seuls les utilisateurs disposant de l'autorisation IAM `logs:Unmask` peuvent consulter les données non masquées.

Vous pouvez créer une politique de protection des données pour tous les groupes de journaux de votre compte, et vous pouvez également créer une politique de protection des données pour des groupes de journaux individuels. Lorsque vous créez une politique pour l'ensemble de votre compte, elle s'applique à la fois aux groupes de journaux existants et aux groupes de journaux qui seront créés ultérieurement.

Si vous créez une politique de protection des données pour l'ensemble de votre compte et que vous créez également une politique pour un seul groupe de journaux, les deux politiques s'appliquent à ce groupe de journaux. Tous les identifiants de données gérés qui sont spécifiés dans l'une ou l'autre des politiques sont audités et masqués dans ce groupe de journaux.

> **Note**
>
> Le masquage des données sensibles n'est pris en charge que pour les groupes de journaux de la classe de journaux standard. Si vous créez une politique de protection des données pour tous les groupes de journaux de votre compte, elle s'applique uniquement aux groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

Chaque groupe de journaux ne peut avoir qu'une seule politique de protection des données au niveau du groupe de journaux, mais cette politique peut spécifier de nombreux identifiants de données gérés à auditer et à masquer. La limite d'une politique de protection des données est de 30 720 caractères.

> **Important**
>
> Les données sensibles sont détectées et masquées lorsqu'elles sont ingérées dans le groupe de journaux. Lorsque vous définissez une politique de protection des données, les événements du journal enregistrés dans le groupe de journaux avant cette date ne sont pas masqués.

CloudWatch Logs prend en charge de nombreux identifiants de données gérés, qui proposent des types de données préconfigurés que vous pouvez sélectionner pour protéger les données financières, les informations médicales personnelles (PHI) et les informations personnelles identifiables (PII). CloudWatch La protection des données des journaux vous permet de tirer parti de modèles de correspondance et de modèles d'apprentissage automatique pour détecter les données sensibles. Pour certains types d'identifiants de données gérés, la détection dépend également de la recherche de certains mots clés à proximité des données sensibles. Vous pouvez également utiliser des identifiants de données personnalisés pour créer des identifiants de données adaptés à votre cas d'utilisation spécifique.

CloudWatch Lorsque des données sensibles sont détectées, une métrique correspondant aux identifiants de données que vous sélectionnez est émise. Il s'agit de la LogEventsWithFindingsmétrique émise dans l'espace de noms AWS/Logs. Vous pouvez utiliser cette métrique pour créer des CloudWatch alarmes, et vous pouvez la visualiser sous forme de graphiques et de tableaux de bord. Les métriques émises par la protection des données sont des métriques vendues et sont gratuites. Pour plus d'informations sur les métriques auxquelles CloudWatch Logs envoie CloudWatch, consultezSurveillance à l'aide de CloudWatch métriques.

Chaque identifiant de données géré est conçu pour détecter un type spécifique de données sensibles, telles que les numéros de carte de crédit, les clés d'accès AWS secrètes ou les numéros de passeport d'un pays ou d'une région en particulier. Lorsque vous créez une politique de protection des données, vous pouvez la configurer pour qu'elle utilise ces identifiants afin d'analyser les journaux ingérés par le groupe de journaux, et prendre des mesures lorsqu'ils sont détectés.

CloudWatch La protection des données des journaux peut détecter les catégories suivantes de données sensibles à l'aide d'identifiants de données gérés :

- Informations d'identification, telles que les clés privées ou les clés d'accès AWS secrètes
- Les informations financières, telles que les numéros de carte de crédit
- Les données d'identification personnelles (PII), telles que les permis de conduire ou les numéros de sécurité sociale
- Les informations protégées sur la santé (PHI) telles que les numéros d'assurance maladie ou d'identification médicale
- Les identifiants d'appareil, tels que les adresses IP ou MAC

Pour plus de détails sur les types de données que vous pouvez protéger, consultez Types de données que vous pouvez protéger (français non garanti).

Table des matières

- Comprendre les politiques de protection des données
  - En quoi consistent les politiques de protection des données ?
  - Comment est structurée la politique de protection des données ?
    - Propriétés JSON pour la politique de protection des données
    - Propriétés JSON pour une instruction de politique
    - Propriétés JSON pour une opération d'instruction de politique
- Autorisations IAM requises pour créer ou utiliser une politique de protection des données

- Autorisations requises pour les politiques de protection des données au niveau du compte
- Autorisations requises pour les politiques de protection des données pour un seul groupe de journaux
- Exemple de politique de protection des données
- Création d'une politique de protection des données à l'échelle du compte
  - console
  - AWS CLI
    - Syntaxe de la politique de protection des données pour AWS CLI nos opérations d'API
- Création d'une politique de protection des données pour un seul groupe de journaux
  - console
  - AWS CLI
    - Syntaxe de la politique de protection des données pour AWS CLI nos opérations d'API
- Affichage de données non masquées
- Rapports de résultats d'audit
  - Politique clé requise pour envoyer les résultats de l'audit à un compartiment protégé par AWS KMS
- Types de données que vous pouvez protéger
  - CloudWatch Enregistre les identifiants de données gérés pour les types de données sensibles
    - Informations d'identification
      - Identifiant de données ARNs pour les types de données d'identification
    - Identifiants de l'appareil
      - Identifiant de données ARNs pour les types de données de l'appareil
    - Informations financières
      - Identifiant de données ARNs pour les types de données financières
    - Informations protégées sur la santé (PHI)
      - Identifiant de données ARNs pour les types de données d'informations de santé protégées (PHI)
    - données d'identification personnelle (PII)
      - Mots clés pour les numéros d'identification du permis de conduire
      - Mots clés pour les numéros d'identification nationaux
      - Mots clés pour les numéros de passeport

    - Mots clés pour les numéros d'identification et de référence des contribuables
    - Identifiant de données ARNs pour les informations personnelles identifiables (PII)
- Identificateurs des données personnalisés
    - En quoi consistent les identifiants de données personnalisés ?
    - Contraintes liées aux identifiants de données personnalisés
    - Utilisation d'identifiants de données personnalisés dans la console
    - Utilisation d'identifiants de données personnalisés dans votre politique de protection des données

Comprendre les politiques de protection des données

Rubriques

- En quoi consistent les politiques de protection des données ?
- Comment est structurée la politique de protection des données ?

En quoi consistent les politiques de protection des données ?

CloudWatch Logs utilise des politiques de protection des données pour sélectionner les données sensibles que vous souhaitez scanner et les mesures que vous souhaitez prendre pour protéger ces données. Pour sélectionner les données sensibles qui vous intéressent, vous utilisez des identifiants de données. CloudWatch La protection des données enregistre puis détecte les données sensibles à l'aide de l'apprentissage automatique et de la correspondance de modèles. Pour agir sur les identifiants de données détectés, vous pouvez définir des opérations d'audit et de désidentification. Ces opérations vous permettent de journaliser les données sensibles détectées et de masquer les données sensibles lorsque les événements du journal sont affichés.

Comment est structurée la politique de protection des données ?

Comme illustré dans la figure suivante, un document de politique de protection des données inclut les éléments suivants :

- Informations facultatives sur l'ensemble de la politique (en haut du document)
- Une instruction qui définit les actions d'audit et de désidentification

Une seule politique de protection des données peut être définie par groupe de CloudWatch journaux Logs. La politique de protection des données peut comporter une ou plusieurs instructions de refus ou d'anonymisation, mais seulement une instruction d'audit.

Propriétés JSON pour la politique de protection des données

Une politique de protection des données nécessite les informations de politique de base suivantes à des fins d'identification :

- Nom - Nom de la politique.
- Description (Facultatif) - Description de la politique.
- Version - Version du langage de la politique. La version actuelle est 2021-06-01.
- Instruction - Liste d'instructions qui spécifie les actions de la politique de protection des données.

```
{
  "Name": "CloudWatchLogs-PersonalInformation-Protection",
  "Description": "Protect basic types of sensitive data",
  "Version": "2021-06-01",
  "Statement": [
        ...
  ]
}
```

Propriétés JSON pour une instruction de politique

Une instruction de politique définit le contexte de détection pour l'opération de protection des données.

- Sid (facultatif) - L'identifiant de l'instruction.
- DataIdentifier— Les données sensibles que CloudWatch Logs doit scanner. Par exemple, nom, adresse ou numéro de téléphone.
- Opération — Les actions de suivi, qu'il s'agisse d'un audit ou d'une anonymisation. CloudWatch Logs exécute ces actions lorsqu'il trouve des données sensibles.

```
{
  ...
  "Statement": [
```

```
    {
      "Sid": "audit-policy",
      "DataIdentifier": [
        "arn:aws:dataprotection::aws:data-identifier/Address"
      ],
      "Operation": {
        "Audit": {
          "FindingsDestination": {}
        }
      }
    },
```

Propriétés JSON pour une opération d'instruction de politique

Une instruction de politique définit l'une des opérations de protection des données suivantes.

- Audit - Émet des métriques et de rapports de résultats sans interrompre la journalisation. Les chaînes qui correspondent incrémentent la LogEventsWithFindingsmétrique publiée par CloudWatch Logs dans l'espace de noms AWS/Logs. CloudWatch Vous pouvez utiliser ces métriques pour créer des alarmes.

  Pour obtenir un exemple de rapport de résultats, veuillez consulter Rapports de résultats d'audit.

  Pour plus d'informations sur les métriques auxquelles CloudWatch Logs envoie CloudWatch, consultezSurveillance à l'aide de CloudWatch métriques.
- Désidentification - Masquer les données sensibles sans interrompre la journalisation.

Autorisations IAM requises pour créer ou utiliser une politique de protection des données

Pour pouvoir utiliser les politiques de protection des données pour les groupes de journaux, vous devez disposer de certaines autorisations, comme indiqué dans les tableaux suivants. Les autorisations sont différentes pour les politiques de protection des données applicables à l'ensemble du compte et pour les politiques de protection des données qui s'appliquent à un seul groupe de journaux.

Autorisations requises pour les politiques de protection des données au niveau du compte

> **Note**
>
> Si vous effectuez l'une de ces opérations dans une fonction Lambda, le rôle d'exécution Lambda et la limite des autorisations doivent également inclure les autorisations suivantes.

Opération	Autorisation IAM requise	Ressource
Créer une politique de protection des données sans destination d'audit	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
Créez une politique de protection des données avec CloudWatch les journaux comme destination d'audit	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`logs:PutResourcePolicy`	*
	`logs:DescribeResourcePolicies`	*
	`logs:DescribeLogGroups`	*
Créez une politique de protection des données avec Firehose comme destination d'audit	`logs:PutAccountPolicy`	*

Opération	Autorisation IAM requise	Ressource
	`logs:PutDataProtectionPolicy`	`*`
	`logs:CreateLogDelivery`	`*`
	`firehose:TagDeliveryStream`	`arn:aws:logs:::deliverystream/` *YOUR_DELIVERY_STREAM*
Créer une politique de protection des données avec Amazon S3 comme destination d'audit	`logs:PutAccountPolicy`	`*`
	`logs:PutDataProtectionPolicy`	`*`
	`logs:CreateLogDelivery`	`*`
	`s3:GetBucketPolicy`	`arn:aws:s3:::` *YOUR_BUCKET*
	`s3:PutBucketPolicy`	`arn:aws:s3:::` *YOUR_BUCKET*
Démasquer les événements du journal masqués dans un groupe de journaux spécifié	`logs:Unmask`	`arn:aws:logs:::log-group:*`
Afficher une politique de protection des données existante	`logs:GetDataProtectionPolicy`	`*`
Supprimer une politique de protection des données	`logs:DeleteAccountPolicy`	`*`

Opération	Autorisation IAM requise	Ressource
	`logs:DeleteDataProtectionPolicy`	`*`

Si des journaux d'audit de protection des données sont déjà envoyés à une destination, les autres politiques qui envoient des journaux à la même destination n'ont besoin que des autorisations `logs:PutDataProtectionPolicy` et `logs:CreateLogDelivery`.

Autorisations requises pour les politiques de protection des données pour un seul groupe de journaux

> **Note**
>
> Si vous effectuez l'une de ces opérations dans une fonction Lambda, le rôle d'exécution Lambda et la limite des autorisations doivent également inclure les autorisations suivantes.

Opération	Autorisation IAM requise	Ressource
Créer une politique de protection des données sans destination d'audit	`logs:PutDataProtectionPolicy`	`arn:aws:logs:::log-group:` *`YOUR_LOG_GROUP`* `:*`

Créez une politique de protection des données avec CloudWatch les journaux comme destination d'audit	`logs:PutDataProtectionPolicy`
	`logs:CreateLogDelivery`
	`logs:PutResourcePolicy`
	`logs:DescribeResourcePolicies`
		`arn:aws:logs:::log-group:` *`YOUR_LOG_GROUP`* `:*`
		`*`
		`*`
		`*`
		`*`

Opération	Autorisation IAM requise	Ressource
	`logs:DescribeLogGroups`	

Créez une politique de protection des données avec Firehose comme destination d'audit	`logs:PutDataProtectionPolicy`
	
	`logs:CreateLogDelivery`
	
	`firehose:TagDeliveryStream`
		`arn:aws:logs:::log-group:`*YOUR_LOG_GROUP*`:*`
		
		`*`
		
		`arn:aws:logs:::deliverystream/`*YOUR_DELIVERY_STREAM*

Création d'une politique de protection des données avec Amazon S3 comme destination d'audit	`logs:PutDataProtectionPolicy`
	
	`logs:CreateLogDelivery`
	
	`s3:GetBucketPolicy`
	
	`s3:PutBucketPolicy`
		`arn:aws:logs:::log-group:`*YOUR_LOG_GROUP*`:*`
		
		`*`
		
		`arn:aws:s3:::`*YOUR_BUCKET*
		
		`arn:aws:s3:::`*YOUR_BUCKET*

Démasquage des événements du journal masqués	`logs:Unmask`	`arn:aws:logs:::log-group:`*YOUR_LOG_GROUP*`:*`
Affichage d'une politique de protection des données existante	`logs:GetDataProtectionPolicy`	`arn:aws:logs:::log-group:`*YOUR_LOG_GROUP*`:*`
Supprimer une politique de protection des données	`logs:DeleteDataProtectionPolicy`	`arn:aws:logs:::log-group:`*YOUR_LOG_GROUP*`:*`

Si des journaux d'audit de protection des données sont déjà envoyés à une destination, les autres politiques qui envoient des journaux à la même destination n'ont besoin que des autorisations `logs:PutDataProtectionPolicy` et `logs:CreateLogDelivery`.

Exemple de politique de protection des données

L'exemple de politique suivant permet à un utilisateur de créer, d'afficher et de supprimer des politiques de protection des données qui peuvent envoyer des résultats d'audit aux trois types de destinations d'audit. Il ne permet pas à l'utilisateur d'afficher les données démasquées.

JSON

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Sid": "AllowLogDeliveryConfiguration",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogDelivery",
            "logs:PutResourcePolicy",
            "logs:DescribeLogGroups",
            "logs:DescribeResourcePolicies"
        ],
        "Resource": "*"
    },
    {
        "Sid": "AllowDataProtectionAndBucketConfiguration",
        "Effect": "Allow",
        "Action": [
            "logs:GetDataProtectionPolicy",
            "logs:DeleteDataProtectionPolicy",
            "logs:PutDataProtectionPolicy",
            "s3:PutBucketPolicy",
            "firehose:TagDeliveryStream",
            "s3:GetBucketPolicy"
        ],
        "Resource": [
            "arn:aws:firehose:::deliverystream/delivery-stream-name",
            "arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "arn:aws:logs:::log-group:log-group-name:*"
```

```
            ]
        }
    ]
}
```

Création d'une politique de protection des données à l'échelle du compte

Vous pouvez utiliser la console CloudWatch Logs ou AWS CLI les commandes pour créer une politique de protection des données afin de masquer les données sensibles pour tous les groupes de journaux de votre compte. Cela s'applique à la fois les groupes de journaux actuels et les groupes de journaux que vous créerez à l'avenir.

> **Important**
>
> Les données sensibles sont détectées et masquées lorsqu'elles sont ingérées dans le groupe de journaux. Lorsque vous définissez une politique de protection des données, les événements du journal enregistrés dans le groupe de journaux avant cette date ne sont pas masqués.

Rubriques

- console
- AWS CLI

console

Pour utiliser la console afin de créer une politique de protection des données à l'échelle du compte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Il est situé vers le bas de la liste.
3. Sélectionnez l'onglet Logs (Journaux).
4. Choisissez Configurer.
5. Pour les identificateurs de données gérés, sélectionnez les types de données que vous souhaitez auditer et masquer pour tous vos groupes de journaux. Vous pouvez saisir dans la zone de sélection pour trouver les identifiants souhaités.

Nous vous recommandons de sélectionner uniquement les identifiants de données pertinents pour vos données de journal et votre activité. Le choix de plusieurs types de données peut entraîner des faux positifs.

Pour plus de détails sur les différents types de données que vous pouvez protéger, consultez Types de données que vous pouvez protéger (français non garanti).

6. (Facultatif) Si vous souhaitez auditer et masquer d'autres types de données à l'aide d'identifiants de données personnalisés, choisissez Ajouter un identifiant de données personnalisé. Entrez ensuite un nom pour le type de données et l'expression régulière à utiliser pour rechercher ce type de données dans le journal des événements. Pour de plus amples informations, veuillez consulter Identificateurs des données personnalisés.

   Une seule politique de protection des données peut inclure jusqu'à 10 identifiants de données personnalisés. Chaque expression régulière qui définit un identifiant de données personnalisé doit comporter 200 caractères ou moins.

7. (Facultatif) Choisissez un ou plusieurs services auxquels envoyer les résultats de l'audit. Même si vous choisissez de ne pas envoyer les résultats d'audit à l'un de ces services, les types de données sensibles que vous sélectionnez resteront masqués.

8. Choisissez Activate data protection (Activer la protection des données).

AWS CLI

Pour utiliser le AWS CLI pour créer une politique de protection des données

1. Utilisez un éditeur de texte pour créer un fichier de politique nommé `DataProtectionPolicy.json`. Pour plus d'informations sur la syntaxe des politiques, consultez la section suivante.

2. Entrez la commande suivante :

```
aws logs put-account-policy \
--policy-name TEST_POLICY --policy-type "DATA_PROTECTION_POLICY" \
--policy-document file://policy.json \
--scope "ALL" \
--region us-west-2
```

Syntaxe de la politique de protection des données pour AWS CLI nos opérations d'API

Lorsque vous créez une politique de protection des données JSON à utiliser dans le cadre d'une AWS CLI commande ou d'une opération d'API, la politique doit inclure deux blocs JSON :

- Le premier bloc doit comprendre à la fois un tableau `DataIdentifer` et une `Operation` propriété avec une `Audit` action. Le `DataIdentifer` tableau répertorie les types de données sensibles que vous souhaitez masquer. Pour obtenir des informations détaillées sur toutes les options disponibles, veuillez consulter Types de données que vous pouvez protéger.

  La `Operation` propriété associée à une `Audit` action est requise pour trouver les termes relatifs aux données sensibles. Cette action `Audit` doit contenir un objet `FindingsDestination`. Vous pouvez éventuellement utiliser cet objet `FindingsDestination` pour répertorier une ou plusieurs destinations vers lesquelles envoyer les rapports de résultats d'audit. Si vous spécifiez des destinations telles que des groupes de journaux, des flux Amazon Data Firehose et des compartiments S3, elles doivent déjà exister. Pour obtenir un exemple de rapport de résultats d'audit, veuillez consulter Rapports de résultats d'audit.
- Le deuxième bloc doit comprendre à la fois un tableau `DataIdentifer` et une propriété `Operation` avec une action `Deidentify`. Le tableau `DataIdentifer` doit correspondre exactement au `DataIdentifer` tableau du premier bloc de la politique.

  La propriété `Operation` associée à l'action `Deidentify` est ce qui masque réellement les données, et elle doit contenir l'objet `"MaskConfig": {}`. L'objet `"MaskConfig": {}` doit être vide.

Voici un exemple de politique de protection des données utilisant uniquement des identifiants de données gérés. Cette politique masque les adresses e-mail et les permis de conduire des États-Unis d'Amérique.

Pour plus d'informations sur les politiques qui spécifient des identifiants de données personnalisés, consultezUtilisation d'identifiants de données personnalisés dans votre politique de protection des données.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
```

```
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Création d'une politique de protection des données pour un seul groupe de journaux

Vous pouvez utiliser la console CloudWatch Logs ou AWS CLI les commandes pour créer une politique de protection des données afin de masquer les données sensibles.

Vous pouvez attribuer une politique de protection des données à chaque groupe de journaux. Chaque politique de protection des données peut être auditée pour plusieurs types d'informations. Chaque politique de protection des données peut inclure une déclaration d'audit.

Rubriques

- console
- AWS CLI

console

Utilisation de la console pour la création d'une politique de protection des données

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), Log groups (Groupes de journaux).
3. Choisissez le nom du groupe de journaux.
4. Choisissez Actions, puis Create data protection policy (Créer une politique de protection des données).
5. Pour les identificateurs de données gérées, sélectionnez les types de données que vous souhaitez auditer et masquer dans ce groupe de journaux. Vous pouvez saisir dans la zone de sélection pour trouver les identifiants souhaités.

   Nous vous recommandons de sélectionner uniquement les identifiants de données pertinents pour vos données de journal et votre activité. Le choix de plusieurs types de données peut entraîner des faux positifs.

   Pour plus de détails sur les types de données que vous pouvez protéger à l'aide d'identifiants de données gérés, consultezTypes de données que vous pouvez protéger.
6. (Facultatif) Si vous souhaitez auditer et masquer d'autres types de données à l'aide d'identifiants de données personnalisés, choisissez Ajouter un identifiant de données personnalisé. Entrez ensuite un nom pour le type de données et l'expression régulière à utiliser pour rechercher ce type de données dans le journal des événements. Pour de plus amples informations, veuillez consulter Identificateurs des données personnalisés.

   Une seule politique de protection des données peut inclure jusqu'à 10 identifiants de données personnalisés. Chaque expression régulière qui définit un identifiant de données personnalisé doit comporter 200 caractères ou moins.

7. (Facultatif) Choisissez un ou plusieurs services auxquels envoyer les résultats de l'audit. Même si vous choisissez de ne pas envoyer les résultats d'audit à l'un de ces services, les types de données sensibles que vous sélectionnez resteront masqués.
8. Choisissez Activate data protection (Activer la protection des données).

AWS CLI

Pour utiliser le AWS CLI pour créer une politique de protection des données

1. Utilisez un éditeur de texte pour créer un fichier de politique nommé `DataProtectionPolicy.json`. Pour plus d'informations sur la syntaxe des politiques, consultez la section suivante.
2. Entrez la commande suivante :

```
aws logs put-data-protection-policy --log-group-identifier "my-log-group" --policy-
document file:///Path/DataProtectionPolicy.json --region us-west-2
```

Syntaxe de la politique de protection des données pour AWS CLI nos opérations d'API

Lorsque vous créez une politique de protection des données JSON à utiliser dans le cadre d'une AWS CLI commande ou d'une opération d'API, la politique doit inclure deux blocs JSON :

- Le premier bloc doit comprendre à la fois un tableau `DataIdentifer` et une `Operation` propriété avec une `Audit` action. Le `DataIdentifer` tableau répertorie les types de données sensibles que vous souhaitez masquer. Pour obtenir des informations détaillées sur toutes les options disponibles, veuillez consulter Types de données que vous pouvez protéger.

  La `Operation` propriété associée à une `Audit` action est requise pour trouver les termes relatifs aux données sensibles. Cette action `Audit` doit contenir un objet `FindingsDestination`. Vous pouvez éventuellement utiliser cet objet `FindingsDestination` pour répertorier une ou plusieurs destinations vers lesquelles envoyer les rapports de résultats d'audit. Si vous spécifiez des destinations telles que des groupes de journaux, des flux Amazon Data Firehose et des compartiments S3, elles doivent déjà exister. Pour obtenir un exemple de rapport de résultats d'audit, veuillez consulter Rapports de résultats d'audit.
- Le deuxième bloc doit comprendre à la fois un tableau `DataIdentifer` et une propriété `Operation` avec une action `Deidentify`. Le tableau `DataIdentifer` doit correspondre exactement au `DataIdentifer` tableau du premier bloc de la politique.

La propriété `Operation` associée à l'action `Deidentify` est ce qui masque réellement les données, et elle doit contenir l'objet `"MaskConfig": {}`. L'objet `"MaskConfig": {}` doit être vide.

Voici un exemple de politique de protection des données qui masque les adresses électroniques et les permis de conduire américains.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
```

```
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Affichage de données non masquées

Pour afficher des données non masquées, un utilisateur doit disposer de l'autorisation `logs:Unmask`. Les utilisateurs qui disposent de cette autorisation peuvent accéder aux données non masquées des manières suivantes :

- Lorsque vous consultez les événements dans un flux de journal, choisissez Display (Afficher), Unmask (Non masquer).
- Utilisez une requête CloudWatch Logs Insights qui inclut la commande unmask (@message). L'exemple de requête suivant affiche les 20 événements de journal les plus récents du flux, non masqués :

  ```
  fields @timestamp, @message, unmask(@message)
  | sort @timestamp desc
  | limit 20
  ```

  Pour plus d'informations sur CloudWatch les commandes Logs Insights, consultez CloudWatch Syntaxe des requêtes en langage Logs Insights.
- Utilisez une FilterLogEvents opération GetLogEvents ou avec le `unmask` paramètre.

La CloudWatchLogsFullAccess politique inclut l'`logs:Unmask` autorisation. Pour accorder une autorisation `logs:Unmask` à un utilisateur qui n'en a pas CloudWatchLogsFullAccess, vous pouvez associer une politique IAM personnalisée à cet utilisateur. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur (console).

Rapports de résultats d'audit

Si vous configurez CloudWatch des politiques d'audit de protection des données de CloudWatch Logs pour rédiger des rapports d'audit pour Logs, Amazon S3 ou Firehose, ces rapports de résultats sont similaires à l'exemple suivant. CloudWatch Logs rédige un rapport de résultats pour chaque événement de journal contenant des données sensibles.

```
{
    "auditTimestamp": "2023-01-23T21:11:20Z",
    "resourceArn": "arn:aws:logs:us-west-2:111122223333:log-group:/aws/lambda/
MyLogGroup:*",
    "dataIdentifiers": [
        {
            "name": "EmailAddress",
            "count": 2,
            "detections": [
                {
                    "start": 13,
                    "end": 26
                },
{
                    "start": 30,
                    "end": 43
                }
            ]
        }
    ]
}
```

Les champs du rapport sont les suivants :

- Le champ `resourceArn` affiche le groupe de journaux dans lequel les données sensibles ont été trouvées.
- L'objet `dataIdentifiers` affiche des informations sur les résultats relatifs à un type de données sensibles que vous auditez.
- Le champ `name` identifie le type de données sensibles dont traite cette section.
- Le champ `count` indique le nombre de fois que ce type de données sensibles apparaît dans l'événement du journal.
- Les champs `start` et `end` indiquent où chaque occurrence des données sensibles apparaît dans l'événement du journal, par nombre de caractères.

L'exemple précédent montre un rapport indiquant la recherche de deux adresses e-mail dans un événement du journal. La première adresse e-mail commence au 13e caractère d'événement du journal et se termine au 26e caractère. La deuxième adresse e-mail est comprise entre le 30e et le 43e caractère. Même si cet événement du journal possède deux adresses e-mail, la valeur de la métrique `LogEventsWithFindings` n'est incrémentée que d'une unité, car cette métrique compte

le nombre d'événements du journal contenant des données sensibles et non le nombre d'occurrences de données sensibles.

Politique clé requise pour envoyer les résultats de l'audit à un compartiment protégé par AWS KMS

Vous pouvez protéger les données d'un compartiment Amazon S3 en activant le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) ou le chiffrement côté serveur avec des clés KMS (SSE-KMS). Pour plus d'informations, consultez Protection des données à l'aide du chiffrement côté serveur dans le Guide de l'utilisateur d'Amazon S3.

Si vous envoyez des résultats d'audit à un compartiment protégé par SSE-S3, aucune configuration supplémentaire n'est requise. Amazon S3 gère la clé de chiffrement.

Si vous envoyez des résultats d'audit à un compartiment protégé par SSE-KMS, vous devez mettre à jour la stratégie de clé pour votre clé KMS afin que le compte de diffusion du journal puisse écrire dans votre compartiment S3. Pour plus d'informations sur la politique clé requise pour une utilisation avec SSE-KMS, consultez Amazon S3 le guide de l'utilisateur Amazon CloudWatch Logs.

Types de données que vous pouvez protéger

Cette section contient des informations sur les types de données que vous pouvez protéger dans le cadre d'une politique de protection des données CloudWatch Logs. CloudWatch Les identifiants de données gérés par les journaux proposent des types de données préconfigurés pour protéger les données financières, les informations médicales personnelles (PHI) et les informations personnelles identifiables (PII). Vous pouvez également utiliser des identifiants de données personnalisés pour créer des identifiants de données adaptés à votre cas d'utilisation spécifique.

Table des matières

- CloudWatch Enregistre les identifiants de données gérés pour les types de données sensibles
  - Informations d'identification
    - Identifiant de données ARNs pour les types de données d'identification
  - Identifiants de l'appareil
    - Identifiant de données ARNs pour les types de données de l'appareil
  - Informations financières
    - Identifiant de données ARNs pour les types de données financières
  - Informations protégées sur la santé (PHI)

    - Identifiant de données ARNs pour les types de données d'informations de santé protégées (PHI)
  - données d'identification personnelle (PII)
    - Mots clés pour les numéros d'identification du permis de conduire
    - Mots clés pour les numéros d'identification nationaux
    - Mots clés pour les numéros de passeport
    - Mots clés pour les numéros d'identification et de référence des contribuables
    - Identifiant de données ARNs pour les informations personnelles identifiables (PII)
- Identificateurs des données personnalisés
  - En quoi consistent les identifiants de données personnalisés ?
  - Contraintes liées aux identifiants de données personnalisés
  - Utilisation d'identifiants de données personnalisés dans la console
  - Utilisation d'identifiants de données personnalisés dans votre politique de protection des données

CloudWatch Enregistre les identifiants de données gérés pour les types de données sensibles

Cette section contient des informations sur les types de données que vous pouvez protéger à l'aide d'identifiants de données gérés, ainsi que sur les pays et les régions concernés par chacun de ces types de données.

Pour certains types de données sensibles, la protection des données de CloudWatch Logs recherche les mots clés situés à proximité des données et ne trouve une correspondance que si elle trouve ce mot clé. Si un mot clé doit se trouver à proximité d'un type de données particulier, il doit généralement se trouver à moins de 30 caractères (inclus) des données.

Si un mot clé contient un espace, la protection CloudWatch des données de Logs fait automatiquement correspondre les variantes de mots clés qui ne contiennent pas d'espace ou qui contiennent un trait de soulignement (_) ou un trait d'union (-) à la place de l'espace. Dans certains cas, CloudWatch Logs développe ou abrège également un mot clé pour tenir compte des variations courantes du mot clé.

Les tableaux suivants répertorient les types d'informations d'identification, d'appareil, financières, médicales et de santé protégées (PHI) que CloudWatch Logs peut détecter à l'aide d'identifiants de

données gérés. Ces données s'ajoutent à certains types de données qui pourraient également être considérés comme des données d'identification personnelle (PII).

Identifiants pris en charge indépendants de la langue et de la région

Identifiant	Catégorie
`Address`	Personnel
`AwsSecretKey`	Informations d'identification
`CreditCardExpiration`	Services financiers
`CreditCardNumber`	Services financiers
`CreditCardSecurityCode`	Services financiers
`EmailAddress`	Personnel
`IpAddress`	Personnel
`LatLong`	Personnel
`Name`	Personnel
`OpenSshPrivateKey`	Informations d'identification
`PgpPrivateKey`	Informations d'identification
`PkcsPrivateKey`	Informations d'identification
`PuttyPrivateKey`	Informations d'identification
`VehicleIdentificationNumber`	Personnel

Les identifiants de données dépendants de la région doivent comprendre le nom de l'identifiant, puis un trait d'union, et enfin les codes à deux lettres (ISO 3166-1 alpha-2). Par exemple, `DriversLicense-US`.

Identifiants pris en charge devant comprendre un code de pays ou de région à deux lettres

Identifiant	Catégorie	Pays et langues
BankAccountNumber	Services financiers	DE, ES, FR, GB, IT, ÉTATS-UNIS
CepCode	Personnel	BR
Cnpj	Personnel	BR
CpfCode	Personnel	BR
DriversLicense	Personnel	AT, AU, BE, BG, CA, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IT, LT, LU, LV, MT, NL, PL, PT, RO, SE, SI, SK, US
DrugEnforcementAgencyNumber	Santé	ETATS-UNIS
ElectoralRollNumber	Personnel	Go
HealthInsuranceCardNumber	Santé	UE
HealthInsuranceClaimNumber	Santé	ETATS-UNIS
HealthInsuranceNumber	Santé	FR
HealthcareProcedureCode	Santé	ETATS-UNIS
IndividualTaxIdentificationNumber	Personnel	ETATS-UNIS
InseeCode	Personnel	FR
MedicareBeneficiaryNumber	Santé	ETATS-UNIS
NationalDrugCode	Santé	ETATS-UNIS
NationalIdentificationNumber	Personnel	DE, ES, IT

Identifiant	Catégorie	Pays et langues
NationalInsuranceNumber	Personnel	Go
NationalProviderId	Santé	ETATS-UNIS
NhsNumber	Santé	Go
NieNumber	Personnel	ES
NifNumber	Personnel	ES
PassportNumber	Personnel	CA, DE, ES, FR, GB, IT, US
PermanentResidenceNumber	Personnel	CA
PersonalHealthNumber	Santé	CA
PhoneNumber	Personnel	BR, DE, ES, FR, GB, IT, US
PostalCode	Personnel	CA
RgNumber	Personnel	BR
SocialInsuranceNumber	Personnel	CA
Ssn	Personnel	ES, US
TaxId	Personnel	DE, ES, FR, GB
ZipCode	Personnel	ETATS-UNIS

Informations d'identification

CloudWatch La protection des données des journaux peut trouver les types d'informations d'identification suivants.

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions
AWS clé d'accès secrète	`AwsSecretKey`	`aws_secret_access_key` , `credentials` , `secret access key`, `secret key`, `set-awscredential`	Tous
Clé privée OpenSSH	`OpenSSHPrivateKey`	Aucun	Tous
Clé privée PGP	`PgpPrivateKey`	Aucun	Tous
Clés privées PKCS	`PkcsPrivateKey`	Aucun	Tous
Clé privée PuTTY	`PuttyPrivateKey`	Aucun	Tous

Identifiant de données ARNs pour les types de données d'identification

Vous trouverez ci-dessous la liste des Amazon Resource Names (ARNs) pour les identifiants de données que vous pouvez ajouter à vos politiques de protection des données.

Identifiant des données d'identification ARNs
`arn:aws:dataprotection::aws:data-identifier/AwsSecretKey`
`arn:aws:dataprotection::aws:data-identifier/OpenSshPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PgpPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PkcsPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PuttyPrivateKey`

Identifiants de l'appareil

CloudWatch La protection des données des journaux permet de trouver les types d'identifiants d'appareils suivants.

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions
Adresse IP	`IpAddress`	Aucun	Tous

Identifiant de données ARNs pour les types de données de l'appareil

Vous trouverez ci-dessous la liste des Amazon Resource Names (ARNs) pour les identifiants de données que vous pouvez ajouter à vos politiques de protection des données.

ARN d'identifiant de données d'appareil
`arn:aws:dataprotection::aws:data-identifier/IpAddress`

Informations financières

CloudWatch La protection des données des journaux permet de trouver les types d'informations financières suivants.

Si vous définissez une politique de protection des données, CloudWatch Logs recherche les identifiants de données que vous spécifiez, quelle que soit la géolocalisation dans laquelle se trouve le groupe de journaux. Les informations de la colonne Countries and regions (Pays et régions) de ce tableau indiquent si des codes de pays à deux lettres doivent être ajoutés à l'identifiant de données pour détecter les mots clés appropriés pour ces pays et régions.

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Numéro de compte bancaire	`BankAccountNumber`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails,	France, Allemagr , Italie,	Comprend les numéros

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
		consultez le tableau Des mots clés pour les numéros de comptes bancaires plus loin dans cette section.	Espagne Royaume Uni, États-Unis d'Amériq e	de compte bancaire internati onaux (IBANs) composés d'un maximum de 34 caractère s alphanumé riques, y compris des éléments tels que les codes de pays.
Date d'expiration de carte de crédit	`CreditCar dExpirati on`	`exp d`, `exp m`, `exp y`, `expiration` , `expiry`	Tous	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Numéro de carte de crédit	`CreditCardNumber`	`account number`, `american express`, `amex`, `bank card`, `card`, `card number`, `card num`, `cc #`, `ccn`, `check card`, `credit`, `credit card#`, `dankort`, `debit`, `debit card`, `diners club`, `discover`, `electron`, `japanese card bureau`, `jcb`, `mastercard `, `mc`, `pan`, `payment account number`, `payment card number`, `pcn`, `union pay`, `visa`	Tous	La détection nécessite que les données soient une séquence de 13 à 19 chiffres conforme à la formule de vérification de Luhn et utilise un préfixe de numéro de carte standard pour les types de

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
				cartes de crédit suivants : American Express, Dankort, Diner's Club, Discover, Electron, Japanese Card Bureau (JCB), Mastercard et Visa. UnionPay
Code de vérification de carte de crédit	`CreditCardSecurityCode`	`card id`, `card identification code`, `card identification number` , `card security code`, `card validation code` , `card validation number` , `card verification data` , `card verification value`, `cvc`, `cvc2`, `cvv`, `cvv2`, `elo verification code`	Tous	

Mots clés pour les numéros de compte bancaire

Utilisez les mots clés suivants pour les numéros de compte bancaire. Cela inclut les numéros de compte bancaire internationaux (IBANs) composés d'un maximum de 34 caractères alphanumériques, y compris des éléments tels que les codes de pays.

Pays	Mots clés
France	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `code bancaire`, `compte bancaire`, `customer account id`, `customer account number`, `customer bank account id`, `iban`, `numéro de compte`
Allemagne	`account code`, `account number`, `accountno#` , `accountnumber#` , `bankleitzahl` , `bban`, `customer account id`, `customer account number`, `customer bank account id`, `geheimzahl` , `iban`, `kartennummer` , `kontonummer` , `kreditkartennummer` , `sepa`
Italie	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `codice bancario`, `conto bancario`, `customer account id`, `customer account number`, `customer bank account id`, `iban`, `numero di conto`
Espagne	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `código cuenta`, `código cuenta bancaria`, `cuenta cliente id`, `customer account ID`, `customer account number`, `customer bank account id`, `iban`, `número cuenta bancaria cliente`, `número cuenta cliente`
Royaume-Uni	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `customer account ID`, `customer account number`, `customer bank account id`, `iban`, `sepa`
États-Unis	`bank account`, `bank acct`, `checking account`, `checking acct`, `deposit account`, `deposit acct`, `savings account`, `savings acct`, `chequing account`, `chequing acct`

CloudWatch Les journaux ne signalent pas les occurrences des séquences suivantes, que les émetteurs de cartes de crédit ont réservées aux tests publics.

```
122000000000003, 2222405343248877, 2222990905257051, 2223007648726984,
 2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
 36148900647913,
36700102000000, 371449635398431, 378282246310005, 378734493671000, 38520000023237,
 4012888888881881,
4111111111111111, 4222222222222, 4444333322221111, 4462030000000000, 4484070000000000,
 4911830000000,
4917300800000000, 4917610000000000, 4917610000000000003, 5019717010103742,
 5105105105105100,
5111010030175156, 5185540810000019, 5200828282828210, 5204230080000017,
 5204740009900014, 5420923878724339,
5454545454545454, 5455330760000018, 5506900490000436, 5506900490000444,
 5506900510000234, 5506920809243667,
5506922400634930, 5506927427317625, 5553042241984105, 5555553753048194,
 5555555555554444, 5610591081018250,
6011000990139424, 6011000400000000, 6011111111111117, 630490017740292441,
 630495060000000000,
6331101999990016, 6759649826438453, 6799990100000000019, and 76009244561.
```

Identifiant de données ARNs pour les types de données financières

Vous trouverez ci-dessous la liste des Amazon Resource Names (ARNs) pour les identifiants de données que vous pouvez ajouter à vos politiques de protection des données.

Identifiant des données financières ARNs
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-DE`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-ES`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-FR`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-GB`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-IT`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-US`

Identifiant des données financières ARNs
`arn:aws:dataprotection::aws:data-identifier/CreditCardExpiration`
`arn:aws:dataprotection::aws:data-identifier/CreditCardNumber`
`arn:aws:dataprotection::aws:data-identifier/CreditCardSecurityCode`

Informations protégées sur la santé (PHI)

CloudWatch La protection des données des journaux permet de trouver les types d'informations de santé protégées (PHI) suivants.

Si vous définissez une politique de protection des données, CloudWatch Logs recherche les identifiants de données que vous spécifiez, quelle que soit la géolocalisation dans laquelle se trouve le groupe de journaux. Les informations de la colonne Countries and regions (Pays et régions) de ce tableau indiquent si des codes de pays à deux lettres doivent être ajoutés à l'identifiant de données pour détecter les mots clés appropriés pour ces pays et régions.

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions
Numéro d'enregistrement de DEA (Drug Enforcement Agency)	`DrugEnforcementAgencyNumber`	`dea number`, `dea registration`	États-Unis
Numéro de carte de sécurité sociale (EHIC)	`HealthInsuranceCardNumber`	`assicurazione sanitaria numero`, `carta assicurazione numero`, `carte d'assurance maladie` , `carte européenne d'assurance maladie` , `ceam`, `ehic`, `ehic#`, `finlandehicnumber#` , `gesundheitskarte` , `hälsokort` , `health card`, `health`	Union européenne

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions
		card number, health insurance card, health insurance number, insurance card number, krankenversicherungskarte , krankenversicherungsnummer , medical account number, numero conto medico, numéro d'assurance maladie , numéro de carte d'assurance , numéro de compte medical, número de cuenta médica, número de seguro de salud, número de tarjeta de seguro, sairaanhoitokortin , sairausvakuutuskortti , sairausvakuutusnumero , sjukförsäkringnummer, sjukförsäkringskort , suomi ehic-numero , tarjeta de salud, terveyskortti , tessera sanitaria assicurazione numero , versicherungsnummer	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions
Numéro de règlement de sécurité sociale (HICN)	`HealthInsuranceClaimNumber`	`health insurance claim number`, `hic no`, `hic no.`, `hic number`, `hic#`, `hicn`, `hicn#`, `hicno#`	États-Unis
Numéro d'assurance maladie ou d'identification médicale	`HealthInsuranceNumber`	`carte d'assuré social`, `carte vitale`, `insurance card`	France
Code du système de codage des procédures communes pour les soins de santé (HCPCS)	`HealthcareProcedureCode`	`current procedural terminology` , `hcpcs`, `healthcare common procedure coding system`	États-Unis
Numéro de bénéficiaire Medicare (MBN)	`MedicareBeneficiaryNumber`	`mbi`, `medicare beneficiary`	États-Unis
Code national des médicaments (NCD)	`NationalDrugCode`	`national drug code`, `ndc`	États-Unis
Identifiant de fournisseur national (NPI)	`NationalProviderId`	`hipaa`, `n.p.i.`, `national provider`, `npi`	États-Unis
Numéro du service national de santé (NHS)	`NhsNumber`	`national health service`, `NHS`	Grande-Bretagne
Numéro de santé personnel (PHN)	`PersonalHealthNumber`	`canada healthcare number`, `msp number`, `care number`, `phn`, `soins de santé`	Canada

Identifiant de données ARNs pour les types de données d'informations de santé protégées (PHI)

Vous trouverez ci-dessous la liste de l'identifiant de données Amazon Resource Names (ARNs) qui peut être utilisé dans le cadre des politiques de protection des données relatives à la santé (PHI).

Identifiant de données PHI ARNs
`arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US`
`arn:aws:dataprotection::aws:data-identifier/HealthcareProcedureCode-US`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceCardNumber-EU`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceClaimNumber-US`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceNumber-FR`
`arn:aws:dataprotection::aws:data-identifier/MedicareBeneficiaryNumber-US`
`arn:aws:dataprotection::aws:data-identifier/NationalDrugCode-US`
`arn:aws:dataprotection::aws:data-identifier/NationalInsuranceNumber-GB`
`arn:aws:dataprotection::aws:data-identifier/NationalProviderId-US`
`arn:aws:dataprotection::aws:data-identifier/NhsNumber-GB`
`arn:aws:dataprotection::aws:data-identifier/PersonalHealthNumber-CA`

données d'identification personnelle (PII)

CloudWatch La protection des données des journaux permet de trouver les types suivants d'informations personnelles identifiables (PII).

Si vous définissez une politique de protection des données, CloudWatch Logs recherche les identifiants de données que vous spécifiez, quelle que soit la géolocalisation dans laquelle se trouve le groupe de journaux. Les informations de la colonne Countries and regions (Pays et régions) de ce tableau indiquent si des codes de pays à deux lettres doivent être ajoutés à l'identifiant de données pour détecter les mots clés appropriés pour ces pays et régions.

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Date de naissance	`DateOfBirth`	`dob`, `date of birth`, `birthdate` , `birth date`, `birthday`, `b-day`, `bday`	N'importe quel compte	Cela inclut la plupart des formats de date, tels que les formats avec des chiffres ainsi que les combinaisons de chiffres et de noms

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
				de mois. Les composants de date peuvent être séparés par des espaces, des barres obliques (/) ou des traits d'union (-).
Código de Endereçamento Postal (CEP)	`CepCode`	`cep`, `código de endereçamento postal`, `codigo de endereçamento postal`	Brésil	
Cadastro Nacional da Pessoa Jurídica (CNPJ)	`Cnpj`	`cadastro nacional da pessoa jurídica`, `cadastro nacional da pessoa juridica`, `cnpj`	Brésil	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Cadastro de Pessoas Físicas (CPF)	`CpfCode`	`Cadastro de pessoas fisicas`, `cadastro de pessoas físicas`, `cadastro de pessoa física`, `cadastro de pessoa fisica`, `cpf`	Brésil	
Numéro d'identification du permis de conduire	`DriversLicense`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau des Numéros d'identification des permis de conduire plus loin dans cette section.	De nombreu pays. Pour plus de détails, consulte le tableau des Numéros d'identification des permis de conduire	
Numéro de liste électorale	`ElectoralRollNumber`	`electoral #`, `electoral number`, `electoral roll #`, `electoral roll no.`, `electoral roll number`, `electoral rollno`	Royaume Uni	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Identification individuelle de contribuable	`IndividualTaxIdentificationNumber`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau des Numéros d'identification des contribuables individuels plus loin dans cette section.	Brésil, France, Allemagne, Espagne, Royaume-Uni	
Institut national de la statistique et des études économiques (INSEE)	`InseeCode`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau Des mots clés pour les numéros d'identification nationaux plus loin dans cette section.	France	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Numéro d'identification nationale	`NationalIdentificationNumber`	Oui. Pour plus de détails, consultez le tableau Des mots clés pour les numéros d'identification nationaux plus loin dans cette section.	Allemagr , Italie, Espagne	Cela inclut les identifiants Documento Nacional de Identidad (DNI) (Espagne), les codes Codice fiscale (Italie) et les numéros de carte d'identité nationale (Allemagne).

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Numéro d'assurance nationale (NINO)	`NationalInsuranceNumber`	`insurance no.`, `insurance number`, `insurance#` , `national insurance number`, `nationalinsurance#` , `nationalinsurancenumber` , `nin`, `nino`	Royaume-Uni	–
Número de identidad de extranjero (NIE)	`NieNumber`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau des Numéros d'identification des contribuables individuels plus loin dans cette section.	Espagne	
Número de Identificación Fiscal (NIF)	`NifNumber`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau des Numéros d'identification des contribuables individuels plus loin dans cette section.	Espagne	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Numéro de passeport	`PassportNumber`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau Des mots clés pour les numéros de passeport plus loin dans cette section.	Canada, France, Allemagr , Italie, Espagne Royaum Uni, États-Unis	
Numéro de résidence permanente	`Permanent Residence Number`	`carte résident permanent` , `numéro carte résident permanent` , `numéro résident permanent` , `permanent resident card`, `permanent resident card number`, `permanent resident no`, `permanent resident no.`, `permanent resident number`, `pr no`, `pr no.`, `pr non`, `pr number`, `résident permanent no.`, `résident permanent non`	Canada	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Phone number (Numéro de téléphone)	`PhoneNumber`	Brésil : les mots clés incluent également : `cel`, `celular`, `fone`, `móvel`, `número residencial` , `numero residencial` , `telefone` Autres : `cell`, `contact`, `fax`, `fax number`, `mobile`, phone, phone `number`, `tel`, `telephone` , `telephone number`	Brésil, Canada, France, Allemagr , Italie, Espagne Royaum Uni, États-Unis	Cela inclut les numéros gratuits aux États-Unis et les numéros de fax. Si un mot clé se trouve à proximité des données, il n'est pas nécessair e que le numéro inclue un code de pays. Si un mot

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
				clé ne se trouve pas à proximité des données, il est nécessaire que le numéro inclue un code de pays.
Code postal	`PostalCode`	Aucun	Canada	
Registro Geral (RG)	`RgNumber`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau des Numéros d'identification des contribuables individuels plus loin dans cette section.	Brésil	
Numéro de sécurité sociale	`SocialInsuranceNumber`	`canadian id`, `numéro d'assurance sociale`, `social insurance number`, `sin`	Canada	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Numéro de sécurité sociale (SSN)	`Ssn`	Espagne : `número de la seguridad social`, `social security no.`, `social security no.` `número de la seguridad social`, `social security number`, `socialsecurityno#` , `ssn`, `ssn#` États-Unis : `social security`, `ss#`, `ssn`	Espagne États-Unis	
Numéro d'identification ou de référence du contribuable	`TaxId`	Oui. Différents mots clés s'appliquent à différents pays. Pour plus de détails, consultez le tableau des Numéros d'identification des contribuables individuels plus loin dans cette section. .	France, Allemagne, Espagne, Royaume-Uni	Cela inclut TIN (France) ; Steueridentifikationsnummer (Allemagne) ; CIF (Espagne) et TRN, UTR (Royaume-Uni).

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Code ZIP	`ZipCode`	`zip code`, `zip+4`	États-Unis	Code postal des États-Unis.
Adresse postale	`Address`	Aucun	Australie, Canada, France, Allemagr, Italie, Espagne Royaume Uni, États-Unis	Bien qu'aucun mot-clé ne soit requis, la détection nécessite que l'adresse comprenne le nom d'une ville ou d'un lieu ainsi qu'un code postal.
Adresse électronique	`EmailAddr ess`	Aucun	N'importe quel compte	

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Coordonnées du système de positionnement mondial (GPS)	`LatLong`	`coordinate` , `coordinates` , `lat long`, `latitude longitude` , `location`, `position`	N'importe quel compte	CloudWatch Les journaux peuvent détecter les coordonnées GPS si les coordonnées de latitude et de longitude sont stockées par paire et qu'elles sont au format décimal (DD), par exemple 41.948614 , -87.655311. La

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
				prise en charge n'inclut pas les coordonnées au format Degrés décimaux minutes (DDM), par exemple 41°56.9168'N 87°39.3187'W, ni au format Degrés, minutes, secondes (DMS), par exemple 41°56'55.0104"N 87°39'19.1196"W.

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Nom complet	`Name`	Aucun	N'importe quel compte	CloudWatch Les journaux ne peuvent détecter que les noms complets. Le support est limité aux jeux de caractères latins.

Type de données	ID de l'identifiant des données	Mot-clé requis	Pays et régions	Remarques
Numéro d'identification de véhicule (VIN)	`VehicleIdentificationNumber`	`Fahrgestellnummer` , `niv`, `numarul de identificare` , `numarul seriei de sasiu`, `serie sasiu`, `numer VIN`, `Número de Identificação do Veículo`, `Número de Identificación de Automóviles` , `numéro d'identification du véhicule`, `vehicle identification number`, `vin`, `VIN numeris`	N'importe quel compte	CloudWatch Les journaux peuvent détecter VINs une séquence de 17 caractères et respecter les normes ISO 3779 et 3780. Ces normes ont été conçues pour être utilisées dans le monde entier.

Mots clés pour les numéros d'identification du permis de conduire

Pour détecter différents types de numéros d'identification de permis de conduire, CloudWatch Logs nécessite qu'un mot clé se trouve à proximité des numéros. Le tableau suivant répertorie les mots clés reconnus par CloudWatch Logs pour des pays et des régions spécifiques.

Pays ou région	Mots clés
Australie	dl# dl:, dl :, dlno# driver licence, driver license, driver permit, drivers lic., drivers licence, driver's licence, drivers license, driver's license, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Autriche	führerschein, fuhrerschein, führerschein republik österreich, fuhrerschein republik osterreich
Belgique	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer, numéro permis conduire, permis de conduire, rijbewijs, rijbewijsnummer
Bulgarie	превозно средство, свидетелство за управление на моторно, свидетелство за управление на мпс, сумпс, шофьорска книжка
Canada	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit,

Pays ou région	Mots clés
	drivers permit number, driving licence, driving license, driving permit, permis de conduire
Croatie	vozačka dozvola
Chypre	άδεια οδήγησης
République tchèque	číslo licence, číslo licence řidiče, číslo řidičskéh o průkazu, ovladače lic., povolení k jízdě, povolení řidiče, řidiči povolení, řidičský prúkaz, řidičský průkaz
Danemark	kørekort, kørekortnummer
Estonie	juhi litsentsi number, juhiloa number, juhiluba, juhiluba number
Finlande	ajokortin numero, ajokortti, förare lic., körkort, körkort nummer, kuljettaja lic., permis de conduire
France	permis de conduire
Allemagne	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer
Grèce	δεια οδήγησης, adeia odigisis
Hongrie	illesztőprogramok lic, jogosítvány, jogsi, licencszám, vezető engedély, vezetői engedély
Irlande	ceadúnas tiomána
Italie	patente di guida, patente di guida numero, patente guida, patente guida numero

Pays ou région	Mots clés
Lettonie	autovadītāja apliecība, licences numurs, vadītāja apliecība, vadītāja apliecības numurs, vadītāja atļauja, vadītāja licences numurs, vadītāji lic.
Lituanie	vairuotojo pažymėjimas
Luxembourg	fahrerlaubnis, führerschäin
Malte	liċenzja tas-sewqan
Pays-Bas	permis de conduire, rijbewijs, rijbewijsnummer
Pologne	numer licencyjny, prawo jazdy, zezwolenie na prowadzenie
Portugal	carta de condução, carteira de habilitação, carteira de motorist, carteira habilitação, carteira motorist, licença condução, licença de condução, número de licença, número licença, permissão condução, permissão de condução
Roumanie	numărul permisului de conducere, permis de conducere
Slovaquie	číslo licencie, číslo vodičského preukazu, ovládače lic., povolenia vodičov, povolenie jazdu, povolenie na jazdu, povolenie vodiča, vodičský preukaz
Slovénie	vozniško dovoljenje

Pays ou région	Mots clés
Espagne	carnet conducer, el carnet de conducer, licencia conducer, licencia de manejo, número carnet conducer, número de carnet de conducer, número de permiso conducer, número de permiso de conducer, número licencia conducer, número permiso conducer, permiso conducción, permiso conducer, permiso de conducción
Suède	ajokortin numero, dlno# ajokortti, drivere lic., förare lic., körkort, körkort nummer, körkortsn ummer, kuljettajat lic.
Royaume-Uni	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
États-Unis	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit

Mots clés pour les numéros d'identification nationaux

Pour détecter différents types de numéros d'identification nationaux, CloudWatch Logs a besoin qu'un mot clé soit placé à proximité des numéros. Cela inclut les identifiants Documento Nacional de

Identidad (DNI) (Espagne), les codes de l'Institut national de la statistique et des études économiques (INSEE), les numéros de carte nationale d'identité allemande (Allemagne) et les numéros du Registro Geral (RG) (Brésil).

Le tableau suivant répertorie les mots clés reconnus par CloudWatch Logs pour des pays et des régions spécifiques.

Pays ou région	Mots clés
Brésil	registro geral, rg
France	assurance sociale, carte nationale d'identité, cni, code sécurité sociale, French social security number, fssn#, insee, insurance number, national id number, nationalid#, numéro d'assurance, sécurité sociale, sécurité sociale non., sécurité sociale numéro, social, social security, social security number, socialsecuritynumber, ss#, ssn, ssn#
Allemagne	ausweisnummer, id number, identification number, identity number, insurance number, personal id, personalausweis
Italie	codice fiscal, dati anagrafici, ehic, health card, health insurance card, p. iva, partita i.v.a., personal data, tax code, tessera sanitaria
Espagne	dni, dni#, dninúmero#, documento nacional de identidad, identidad único, identidadúnico#, insurance number, national identification number, national identity, nationalid#, nationaldno#, número nacional identidad, personal identification number, personal identity no, unique identity number, uniqueid#

Mots clés pour les numéros de passeport

Pour détecter différents types de numéros de passeport, CloudWatch Logs nécessite qu'un mot clé se trouve à proximité des numéros. Le tableau suivant répertorie les mots clés reconnus par CloudWatch Logs pour des pays et des régions spécifiques.

Pays ou région	Mots clés
Canada	passeport, passeport#, passport, passport#, passportno, passportno#
France	numéro de passeport, passeport, passeport #, passeport #, passeportn °, passeport n °, passeportNon, passeport non
Allemagne	ausstellungsdatum, ausstellungsort, geburtsdatum, passport, passports, reisepass, reisepass–nr, reisepassnummer
Italie	italian passport number, numéro passeport , numéro passeport italien, passaporto, passaporto italiana, passaporto numero, passport number, repubblica italiana passaporto
Espagne	españa pasaporte, libreta pasaporte, número pasaporte, pasaporte, passport, passport book, passport no, passport number, spain passport
Royaume-Uni	passeport #, passeport n °, passeportNon, passeport non, passeportn °, passport #, passport no, passport number, passport#, passportid
États-Unis	passport, travel document

Mots clés pour les numéros d'identification et de référence des contribuables

Pour détecter les différents types de numéros d'identification et de référence des contribuables, CloudWatch Logs a besoin qu'un mot clé se trouve à proximité des numéros. Le tableau suivant répertorie les mots clés reconnus par CloudWatch Logs pour des pays et des régions spécifiques.

Pays ou région	Mots clés
Brésil	cadastro de pessoa física, cadastro de pessoa fisica, cadastro de pessoas físicas, cadastro de pessoas fisicas, cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj, cpf
France	numéro d'identification fiscale, tax id, tax identification number, tax number, tin, tin#
Allemagne	identifikationsnummer, steuer id, steueride ntifikationsnummer, steuernummer, tax id, tax identification number, tax number
Espagne	cif, cif número, cifnúmero#, nie, nif, número de contribuyente, número de identidad de extranjero, número de identificación fiscal, número de impuesto corporativo, personal tax number, tax id, tax identification number, tax number, tin, tin#
Royaume-Uni	paye, tax id, tax id no., tax id number, tax identification, tax identification#, tax no., tax number, tax reference, tax#, taxid#, temporary reference number, tin, trn, unique tax reference, unique taxpayer reference, utr
États-Unis	numéro individuel d'identification de contribua ble, itin, i.t.i.n.

Identifiant de données ARNs pour les informations personnelles identifiables (PII)

Le tableau suivant répertorie les Amazon Resource Names (ARNs) pour les identifiants de données d'identification personnelle (PII) que vous pouvez ajouter à vos politiques de protection des données.

Identifiant de données PII ARNs
`arn:aws:dataprotection::aws:data-identifier/Address`
`arn:aws:dataprotection::aws:data-identifier/CepCode-BR`
`arn:aws:dataprotection::aws:data-identifier/Cnpj-BR`
`arn:aws:dataprotection::aws:data-identifier/CpfCode-BR`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-AT`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-AU`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-BE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-BG`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CA`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CY`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CZ`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-DE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-DK`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-EE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-ES`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-FI`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-FR`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-GB`

Identifiant de données PII ARNs
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LV
arn:aws:dataprotection::aws:data-identifier/DriversLicense-MT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-NL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-RO
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-US
arn:aws:dataprotection::aws:data-identifier/ElectoralRollNumber-GB
arn:aws:dataprotection::aws:data-identifier/EmailAddress

Identifiant de données PII ARNs
arn:aws:dataprotection::aws:data-identifier/IndividualTaxIdentificationNumber-US
arn:aws:dataprotection::aws:data-identifier/InseeCode-FR
arn:aws:dataprotection::aws:data-identifier/LatLong
arn:aws:dataprotection::aws:data-identifier/Name
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-DE
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-ES
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-IT
arn:aws:dataprotection::aws:data-identifier/NieNumber-ES
arn:aws:dataprotection::aws:data-identifier/NifNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-CA
arn:aws:dataprotection::aws:data-identifier/PassportNumber-DE
arn:aws:dataprotection::aws:data-identifier/PassportNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-FR
arn:aws:dataprotection::aws:data-identifier/PassportNumber-GB
arn:aws:dataprotection::aws:data-identifier/PassportNumber-IT
arn:aws:dataprotection::aws:data-identifier/PassportNumber-US
arn:aws:dataprotection::aws:data-identifier/PermanentResidenceNumber-CA

Identifiant de données PII ARNs
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-BR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-DE
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-ES
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-FR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-GB
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-IT
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-US
arn:aws:dataprotection::aws:data-identifier/PostalCode-CA
arn:aws:dataprotection::aws:data-identifier/RgNumber-BR
arn:aws:dataprotection::aws:data-identifier/SocialInsuranceNumber-CA
arn:aws:dataprotection::aws:data-identifier/Ssn-ES
arn:aws:dataprotection::aws:data-identifier/Ssn-US
arn:aws:dataprotection::aws:data-identifier/TaxId-DE
arn:aws:dataprotection::aws:data-identifier/TaxId-ES
arn:aws:dataprotection::aws:data-identifier/TaxId-FR
arn:aws:dataprotection::aws:data-identifier/TaxId-GB
arn:aws:dataprotection::aws:data-identifier/VehicleIdentificationNumber
arn:aws:dataprotection::aws:data-identifier/ZipCode-US

Identificateurs des données personnalisés

Rubriques

- En quoi consistent les identifiants de données personnalisés ?
- Contraintes liées aux identifiants de données personnalisés
- Utilisation d'identifiants de données personnalisés dans la console
- Utilisation d'identifiants de données personnalisés dans votre politique de protection des données

En quoi consistent les identifiants de données personnalisés ?

Les identificateurs de données personnalisés (CDIs) vous permettent de définir vos propres expressions régulières personnalisées qui peuvent être utilisées dans votre politique de protection des données. En utilisant des identifiants de données personnalisés, vous pouvez cibler des cas d'utilisation de données d'identification personnelle (PII) propres à un domaine d'activité, ce que les identifiants de données gérés ne peuvent pas offrir. Par exemple, vous pouvez utiliser un identifiant de données personnalisé pour rechercher un employé spécifique à l'entreprise. IDs Les identifiants de données personnalisés peuvent être utilisés conjointement avec des identifiants de données gérés.

Contraintes liées aux identifiants de données personnalisés

CloudWatch Les identifiants de données personnalisés des journaux présentent les limites suivantes :

- Le nombre d'identifiants de données personnalisés pris en charge pour chaque politique de protection des données est limité à 10.
- Les noms d'identificateurs de données personnalisés ont une longueur maximale de 128 caractères. Les caractères pris en charge sont les suivants :
  - Alphanumériques : (a-zA-Z0-9)
  - Symboles : ('_' | '-')
- La longueur maximale de RegEx est de 200 caractères. Les caractères pris en charge sont les suivants :
  - Alphanumériques : (a-zA-Z0-9)
  - Symboles : ('_' | '#' | '=' | '@' |'/' | ';' | ',' | '-' | ' ')
  - Caractères réservés pour RegEx : ('^' | '$' | '?' | '[' | ']' | '{' | '}' | '|' \ \ '|' * '|' * '|' + '|'. ')

- Les identifiants de données personnalisés ne peuvent pas avoir le même nom qu'un identifiant de données géré.
- Les identifiants de données personnalisés peuvent être spécifiés dans une politique de protection des données au niveau du compte ou dans des politiques de protection des données au niveau du groupe de journaux. À l'instar des identifiants de données gérés, les identifiants de données personnalisés définis dans le cadre d'une politique au niveau du compte fonctionnent en combinaison avec les identifiants de données personnalisés définis dans une politique au niveau du groupe de journaux.

Utilisation d'identifiants de données personnalisés dans la console

Lorsque vous utilisez la CloudWatch console pour créer ou modifier une politique de protection des données, pour spécifier un identifiant de données personnalisé, il vous suffit de saisir un nom et une expression régulière pour l'identifiant de données. Par exemple, vous pouvez saisir **`Employee_ID`** le **`EmployeeID-\d{9}`** nom et l'expression régulière. Cette expression régulière détectera et masquera les événements du journal suivis de neuf chiffres`EmployeeID-`. Par exemple, `EmployeeID-123456789`

Utilisation d'identifiants de données personnalisés dans votre politique de protection des données

Si vous utilisez l' AWS API AWS CLI or pour spécifier un identifiant de données personnalisé, vous devez inclure le nom de l'identifiant de données et l'expression régulière dans la politique JSON utilisée pour définir la politique de protection des données. La politique de protection des données suivante détecte et masque les événements enregistrés qui concernent des employés spécifiques à l'entreprise. IDs

1. Créez un bloc `Configuration` dans votre politique de protection des données.
2. Nommez votre identifiant de données personnalisé dans `Name`. Par exemple, **`EmployeeId`**.
3. Nommez votre identifiant de données personnalisé dans `Regex`. Par exemple, **`EmployeeID-\d{9}`**. Cette expression régulière correspondra aux événements du journal `EmployeeID-` contenant neuf chiffres après`EmployeeID-`. Par exemple, `EmployeeID-123456789`
4. Faites référence à l'identifiant de données personnalisé suivant dans une déclaration de politique.

```
{
    "Name": "example_data_protection_policy",
    "Description": "Example data protection policy with custom data identifiers",
    "Version": "2021-06-01",
```

```
    "Configuration": {
      "CustomDataIdentifier": [
        {"Name": "EmployeeId", "Regex": "EmployeeId-\\d{9}"}
      ]
    },
    "Statement": [
        {
            "Sid": "audit-policy",
            "DataIdentifier": [
                "EmployeeId"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
            "EmployeeId"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {
                    }
                }
            }
        }
    ]
}
```

5. (Facultatif) Continuez à ajouter des identificateurs de données personnalisés supplémentaires au bloc `Configuration`, si nécessaire. Les politiques de protection des données prennent actuellement en charge au maximum 10 identifiants de données personnalisés.

Transformez les journaux lors de l'ingestion

Grâce à la transformation et à l'enrichissement des journaux, vous pouvez normaliser tous vos journaux dans un format cohérent et riche en contexte au moment de leur ingestion dans CloudWatch Logs. Vous pouvez structurer vos journaux en utilisant des modèles préconfigurés pour les AWS services courants tels qu' AWS WAF Amazon Route 53, ou en créant des transformateurs personnalisés avec des analyseurs natifs tels que Grok. Vous pouvez également renommer les attributs existants et ajouter des métadonnées supplémentaires à vos journaux, telles que l'identifiant du compte et la région.

La transformation des journaux permet de simplifier et de raccourcir vos requêtes de journal dans l'ensemble de vos applications, ainsi que de simplifier la création d'alertes sur vos journaux. Cette fonctionnalité permet de transformer les types de journaux courants avec des modèles de out-of-the-box transformation pour les principales sources de AWS journaux telles que les journaux VPC Flow, Route 53 et. Amazon RDS for PostgreSQL Vous pouvez utiliser des modèles de transformation préconfigurés ou créer des transformateurs personnalisés en fonction de vos besoins.

La transformation des journaux vous permet de gérer les journaux émis par diverses sources dont le format et les noms d'attributs varient considérablement.

Après avoir créé un transformateur, les événements du journal ingéré sont convertis et stockés dans un format standard. Vous pouvez tirer parti de ces journaux transformés pour accélérer votre expérience d'analyse grâce aux fonctionnalités suivantes :

- Index des champs
- CloudWatch Champs découverts par Logs Insights
- Flexibilité des alarmes à l'aide de filtres métriques
- Transfert via des filtres d'abonnement
- Création de données métriques à partir des événements du journal avec Contributor Insights, où vous pouvez choisir que la règle Contributor Insights évalue les événements du journal avant ou après leur transformation.

Les transformations se produisent uniquement lors de l'ingestion du journal. Vous ne pouvez pas transformer les événements du journal qui ont déjà été ingérés. Les transformations ne sont pas réversibles. Les journaux originaux et transformés sont stockés dans CloudWatch des journaux avec la même politique de conservation. La capacité de transformation et d'enrichissement des logs est

incluse dans le prix d'ingestion de la classe de log standard existant. Les coûts de stockage des journaux seront basés sur la taille des journaux après transformation, qui peut dépasser le volume des journaux d'origine.

> **Important**
>
> Une fois les événements du journal transformés, vous devez utiliser CloudWatch les requêtes Logs Insights pour afficher les versions transformées des journaux. Les FilterLogEventsactions GetLogEventset renvoient uniquement les versions d'origine des événements du journal, avant leur transformation.

En plus de les transformer en différents formats, vous pouvez également enrichir vos journaux avec du contexte supplémentaire, tel que l'identifiant du compte, la région et le mot clé. Ils sont extraits du nom du groupe de journaux et de mots clés statiques.

La transformation des journaux vous aide à gérer les journaux émis par diverses sources dont le format et les noms d'attributs varient considérablement.

La transformation et l'enrichissement des journaux ne sont pris en charge que pour les groupes de journaux de la classe de journaux standard.

Vous pouvez créer des transformateurs pour des groupes de journaux individuels, et vous pouvez également créer des transformateurs au niveau du compte qui s'appliquent à tous les groupes de journaux ou à de nombreux groupes de journaux de votre compte. Si un groupe de journaux possède un transformateur au niveau du groupe de journaux, ce transformateur remplace tout transformateur au niveau du compte qui s'appliquerait autrement à ce groupe de journaux.

Rubriques

- Création et gestion de transformateurs de log
- Processeurs que vous pouvez utiliser
- Métriques de transformation et erreurs

Création et gestion de transformateurs de log

Un transformateur logarithmique comprend un ou plusieurs processeurs qui forment ensemble un pipeline logique. Chaque processeur est appliqué à un événement journal, l'un après l'autre, dans l'ordre dans lequel ils sont répertoriés dans la configuration du transformateur.

Certains processeurs sont du type analyseur syntaxique. Chaque transformateur doit avoir au moins un analyseur, et le premier processeur d'un transformateur doit être un analyseur syntaxique.

Certains des analyseurs sont des analyseurs intégrés configurés pour un certain type de journal vendu AWS .

Les autres types de processeurs sont les mutateurs de chaînes, les mutateurs JSON et les processeurs de données.

Vous pouvez créer des transformateurs pour des groupes de journaux individuels, et vous pouvez également créer des transformateurs au niveau du compte qui s'appliquent à tous les groupes de journaux ou à de nombreux groupes de journaux de votre compte. Si un groupe de journaux possède un transformateur au niveau du groupe de journaux, ce transformateur remplace tout transformateur au niveau du compte qui s'appliquerait autrement à ce groupe de journaux. Vous pouvez avoir jusqu'à 20 transformateurs au niveau du compte dans une région sur votre compte.

Vous devez suivre les instructions suivantes lorsque vous créez un transformateur :

- Si vous incluez un analyseur préconfiguré pour un type de journaux vendus, il doit AWS s'agir du premier processeur répertorié dans le transformateur. Vous ne pouvez inclure qu'un seul processeur de ce type dans un transformateur.
- Vous ne pouvez inclure qu'un seul `grok` processeur dans un transformateur.
- Vous devez disposer d'au moins un processeur de type analyseur dans un transformateur. Vous pouvez inclure jusqu'à cinq processeurs de type analyseur. Cette limite de cinq inclut à la fois les analyseurs intégrés et les analyseurs configurables.
- Un transformateur peut contenir jusqu'à 20 processeurs.
- Vous ne pouvez inclure qu'un seul processeur AddKeys dans un transformateur.
- Vous ne pouvez inclure qu'un seul processeur CopyValue dans un transformateur.
- Chaque transformateur peut extraire jusqu'à 200 champs d'un événement journal.

Pour plus d'informations sur tous les processeurs pris en charge et leur syntaxe, consultezProcesseurs que vous pouvez utiliser.

Rubriques

- Création d'une politique de transformation au niveau du compte
- Modifier ou supprimer une politique de transformation au niveau du compte

- Créez un transformateur de log-group-level bûches à partir de zéro
- Créez un log-group-level transformateur en copiant un transformateur existant
- Modifier un log-group-level transformateur
- Supprimer un log-group-level transformateur

Création d'une politique de transformation au niveau du compte

Suivez les étapes décrites dans cette section pour créer une politique de transformation qui s'applique à tous les groupes de journaux du compte ou à plusieurs groupes de journaux dont les noms de groupes de journaux commencent par la même chaîne (préfixe). Vous pouvez avoir jusqu'à 20 politiques de transformation au niveau du compte dans une région.

Vous ne pouvez pas créer deux politiques de transformation dans la même région qui utilisent le même préfixe ou dont un préfixe est contenu dans un autre. Par exemple, si vous créez une politique de transformation pour le préfixe de chaîne`/aws/lambda`, vous ne pouvez pas en créer une autre avec le préfixe`/aws`. Mais vous pourriez avoir un transformateur pour `/aws/lambda` et un autre pour `/aws/waf`

Pour créer une politique de transformation au niveau du compte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Paramètres, puis l'onglet Journaux.
3. Dans la section Politique de transformation pour le compte, choisissez Créer une politique de transformation.
4. Pour le nom de la politique du transformateur, entrez le nom de votre nouvelle politique.
5. Pour Sélectionner des groupes de journaux, effectuez l'une des opérations suivantes :
   - Choisissez Tous les groupes de journaux standard pour que la politique de transformation s'applique à tous les groupes de journaux de classe standard du compte.
   - choisissez Groupes de journaux par correspondance de préfixes pour appliquer la politique à un sous-ensemble de groupes de journaux dont les noms commencent tous par la même chaîne. Entrez ensuite le préfixe de ces groupes de journaux dans Critères de sélection.
6. Dans la zone Sélectionner les analyseurs, utilisez les analyseurs pour sélectionner un analyseur à inclure dans votre transformateur.

   S'il s'agit d'un analyseur préconfiguré pour un type de journal AWS vendu, il n'est pas nécessaire de spécifier de configuration pour celui-ci.

S'il s'agit d'un autre analyseur, vous devez spécifier sa configuration. Pour plus d'informations, consultez les informations relatives à ce processeur dansProcesseurs configurables de type analyseur.

7. Pour ajouter un autre processeur, choisissez Sélectionner un processeur. Sélectionnez ensuite le processeur souhaité dans le champ Processeur et renseignez les paramètres de configuration. Pour plus d'informations sur les paramètres de configuration, consultez la section relative à ce processeur dansProcesseurs que vous pouvez utiliser.

   N'oubliez pas que les processeurs agissent sur les événements du journal dans l'ordre dans lequel vous les ajoutez au transformateur.

8. (Facultatif) Pour ajouter des processeurs supplémentaires, choisissez + Processeur et répétez l'étape précédente.
9. (Facultatif) À tout moment, vous pouvez tester le transformateur que vous avez construit jusqu'à présent à l'aide d'un exemple de journal d'événements. Pour ce faire, effectuez l'une des opérations suivantes dans la section d'aperçu du transformateur :
   - Sélectionnez jusqu'à cinq groupes de journaux dans Sélectionner les groupes de journaux, puis choisissez Charger les derniers événements du journal. Choisissez ensuite le transformateur de test.
   - Copiez les événements du journal directement dans Sample log events, puis choisissez Test transformer.

   La version transformée du journal apparaît alors.

10. Lorsque vous avez terminé d'ajouter des processeurs et que vous êtes satisfait des tests effectués sur les exemples de journaux, choisissez Enregistrer.
11. Lorsque vous avez terminé, choisissez Créer.

Modifier ou supprimer une politique de transformation au niveau du compte

Suivez les étapes décrites dans cette section pour modifier ou supprimer une politique de transformation au niveau du compte.

Pour modifier ou supprimer une politique de transformation au niveau du compte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Paramètres, puis l'onglet Journaux.

3. Dans la section Politique de compte Transformer, choisissez Gérer.
4. Sélectionnez le bouton correspondant à la politique de transformation que vous souhaitez gérer, puis choisissez Modifier ou Supprimer.

   Si vous modifiez la politique, consultez les étapes 5 à 11 ci-dessous pour connaître les options qui s'offrent Processeurs configurables de type analyseur à vous.

Créez un transformateur de log-group-level bûches à partir de zéro

Suivez ces étapes pour créer un log-group-level transformateur à partir de zéro.

Pour utiliser la console afin de créer un transformateur de journaux pour un groupe de journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), Log groups (Groupes de journaux).
3. Choisissez le groupe de journaux pour lequel vous souhaitez créer le transformateur.
4. Choisissez l'onglet Transformer. Vous devrez peut-être faire défiler la liste des onglets vers la droite pour l'afficher.
5. Choisissez Créer un transformateur.
6. Dans la zone Choisissez un analyseur, sélectionnez un analyseur à inclure dans votre transformateur.

   S'il s'agit d'un analyseur préconfiguré pour un type de journal AWS vendu, il n'est pas nécessaire de spécifier de configuration pour celui-ci.

   S'il s'agit d'un autre analyseur, vous devez spécifier sa configuration. Pour plus d'informations, consultez les informations relatives à ce processeur dansProcesseurs configurables de type analyseur.
7. Pour ajouter un autre processeur, choisissez + Ajouter un processeur. Sélectionnez ensuite le processeur de votre choix dans la zone Choisir les processeurs, puis renseignez les paramètres de configuration. Pour plus d'informations sur les paramètres de configuration, consultez la section relative à ce processeur dansProcesseurs que vous pouvez utiliser.

   N'oubliez pas que les processeurs agissent sur les événements du journal dans l'ordre dans lequel vous les ajoutez au transformateur.

8. (Facultatif) À tout moment, vous pouvez tester le transformateur que vous avez construit jusqu'à présent à l'aide d'un exemple de journal d'événements. Pour ce faire, procédez comme suit :

   - Dans la section Aperçu de la transformation, choisissez Charger un exemple de journal pour charger un exemple d'événement de journal à partir du groupe de journaux auquel ce transformateur est destiné, ou collez un événement de journal dans la zone de texte.

     Choisissez Tester le transformateur. La version transformée du journal apparaît

9. Lorsque vous avez terminé d'ajouter des processeurs et que vous êtes satisfait des tests effectués sur les exemples de journaux, choisissez Enregistrer.

Pour utiliser le AWS CLI pour créer un transformateur de bûches à partir de zéro

- Utilisez la commande `aws logs put-transformer`. Lors de l'utilisation en `parseJSON` tant que premier processeur, vous devez analyser l'intégralité de l'événement du journal en utilisant `@message` comme champ source. Après l'analyse JSON initiale, vous pouvez ensuite manipuler des champs spécifiques dans les processeurs suivants. Voici un exemple de création d'un transformateur qui inclut les `addKeys` processeurs `parseJSON` et :

```
 aws logs put-transformer \
  --transformer-config '[{"parseJSON":{"source":"@message"}},{"addKeys":
{"entries":[{"key":"metadata.transformed_in","value":"CloudWatchLogs"},
{"key":"feature","value":"Transformation"}]}},{"trimString":{"withKeys":
["status"]}}]' \
  --log-group-identifier my-log-group-name
```

Créez un log-group-level transformateur en copiant un transformateur existant

Vous pouvez utiliser la console pour copier la configuration JSON d'un transformateur existant. Vous pouvez ensuite utiliser ce code pour créer un transformateur identique en utilisant le AWS CLI, ou vous pouvez d'abord modifier la configuration.

Pour créer un transformateur de log en copiant un transformateur existant

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), Log groups (Groupes de journaux).
3. Choisissez le groupe de journaux contenant le transformateur que vous souhaitez copier.
4. Choisissez l'onglet Transformations. Vous devrez peut-être faire défiler la liste des onglets vers la droite pour l'afficher.
5. Choisissez Gérer le transformateur.
6. Choisissez Copier le transformateur. Cela copie le JSON du transformateur dans votre presse-papiers.
7. Créez un fichier et collez-le dans la configuration du transformateur. Dans cet exemple, nous appellerons le fichier `CopiedTransformer.json`
8. Utilisez le AWS CLI pour créer un nouveau transformateur avec cette configuration.

```
aws logs put-transformer --log-group-identifier my-log-group-name \
--transformer-config file://CopiedTransformer.json
```

Modifier un log-group-level transformateur

Suivez ces étapes pour modifier un transformateur de log existant.

Pour modifier un Log Transformer

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), Log groups (Groupes de journaux).
3. Choisissez le groupe de journaux contenant le transformateur que vous souhaitez modifier.
4. Choisissez l'onglet Transformations. Vous devrez peut-être faire défiler la liste des onglets vers la droite pour l'afficher.
5. Choisissez Gérer le transformateur.
6. Dans les sections Analyseurs et Processeurs, apportez vos modifications.
7. Pour ajouter un autre processeur, choisissez + Ajouter un processeur. Sélectionnez ensuite le processeur souhaité dans le champ Processeur et renseignez les paramètres de configuration. Pour plus d'informations sur les paramètres de configuration, consultez la section relative à ce processeur dansProcesseurs que vous pouvez utiliser.

N'oubliez pas que les processeurs agissent sur les événements du journal dans l'ordre dans lequel vous les ajoutez au transformateur.

8. (Facultatif) À tout moment, vous pouvez tester le transformateur que vous avez construit jusqu'à présent à l'aide d'un exemple de journal d'événements. Pour ce faire, procédez comme suit :

   - Dans la section Aperçu de la transformation, choisissez Charger un exemple de journal pour charger un exemple d'événement de journal à partir du groupe de journaux auquel ce transformateur est destiné, ou collez un événement de journal dans la zone de texte.

     Choisissez Tester la transformation. La version transformée du journal apparaît

9. Lorsque vous avez terminé d'ajouter des processeurs et que vous êtes satisfait des tests effectués sur les exemples de journaux, choisissez Enregistrer.

Supprimer un log-group-level transformateur

Procédez comme suit pour supprimer un transformateur de log.

Pour supprimer un Log Transformer

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), Log groups (Groupes de journaux).
3. Choisissez le groupe de journaux contenant le transformateur que vous souhaitez modifier.
4. Choisissez l'onglet Transformations. Vous devrez peut-être faire défiler la liste des onglets vers la droite pour l'afficher.
5. Sélectionnez Delete (Supprimer).
6. Dans la boîte de confirmation, choisissez Supprimer la politique.

Processeurs que vous pouvez utiliser

Cette section contient des informations sur chaque processeur que vous pouvez utiliser dans un transformateur d'événements de journalisation. Les processeurs peuvent être classés en analyseurs, mutateurs de chaînes, mutateurs JSON et processeurs de date.

Table des matières

- Processeurs configurables de type analyseur
  - parseJSON
  - grok
    - Exemples de grok
      - Exemple 1 : utiliser grok pour extraire un champ à partir de journaux non structurés
      - Exemple 2 : utilisez grok en combinaison avec parseJSON pour extraire des champs d'un événement de journal JSON
      - Exemple 3 : modèle Grok avec annotation en pointillés dans FIELD_NAME
    - Motifs de grok pris en charge
      - Exemples de formats de journaux courants
        - Exemple de journal Apache
        - Exemple de journal NGINX
        - Exemple de journal du protocole Syslog (RFC 5424)
  - csv
  - parseKeyValue
- Processeurs intégrés pour les AWS journaux vendus
  - Analyser WAF
  - Analyser Postgres
  - ParseCloud Front
  - Par Route 53
  - Analyser VPC
- Processeurs String Mutate
  - lowerCaseString
  - upperCaseString
  - Chaîne divisée
  - Chaîne de remplacement
  - TrimString
- Processeurs de mutation JSON
  - Ajouter des clés
  - Supprimer les clés
  - Touches de déplacement

- Renommer les clés
- Copier la valeur
- listToMap
- Processeurs de conversion de types de données
  - Type de convertisseur
  - Convertisseur Date/Heure

Processeurs configurables de type analyseur

parseJSON

Le processeur ParseJSON analyse les événements du journal JSON et insère les paires clé-valeur JSON extraites sous la destination. Si vous ne spécifiez pas de destination, le processeur place la paire clé-valeur sous le nœud racine. Lors de l'utilisation en `parseJSON` tant que premier processeur, vous devez analyser l'intégralité de l'événement du journal en utilisant `@message` comme champ source. Après l'analyse JSON initiale, vous pouvez ensuite manipuler des champs spécifiques dans les processeurs suivants.

Le `@message` contenu original n'est pas modifié, les nouvelles clés sont ajoutées au message.

Champ	Description	Obligatoire ?	Par défaut	Limites
source	Chemin d'accès au champ du journal des événements qui sera analysé. Utilisez la notation par points pour accéder aux champs secondaires. Par exemple, `store.book`	Non	`@message`	Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
destination	Le champ de destination du JSON analysé	Non	`Parent JSON node`	Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3

Exemple

Supposons qu'un événement de journal ingéré ressemble à ceci :

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

Ensuite, si nous avons ce processeur ParseJSON :

```
[
   {
        "parseJSON": {
            "destination": "new_key"
        }
   }
]
```

L'événement du journal transformé serait le suivant.

```
{
    "new_key": {
        "outer_key": {
            "inner_key": "inner_value"
        }
    }
}
```

grok

Utilisez le processeur grok pour analyser et structurer des données non structurées à l'aide de modèles de correspondance. Ce processeur peut également extraire des champs des messages du journal.

Champ	Description	Obligatoire ?	Par défaut	Limites	Remarques
source	Chemin du champ sur lequel appliquer la correspondance Grok	Non	`@message`	Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3	
match	Le modèle de grok à comparer à l'événement du journal	Oui		Longueur maximale : 512 Motifs de rainure maximum : 20 Certains types de modèles de grok ont des limites d'utilisation individuelles. Toute combinaison des modèles suivants peut être utilisée jusqu'à cinq fois : {URI, URIPARAM, URIPATHPARAM, SPACE, DATA, GREEDYDATA, GREEDYDATA_MULTILINE} Les modèles Grok ne prennent pas en charge les conversions de type.	Voir tous les modèles Grok pris en charge

Champ	Description	Obligatoire ?	Par défaut	Limites	Remarques
				Pour les modèles de format de journal courants (APACHE_ACCESS_LOG, NGINX_ACCESS_LOG, SYSLOG5424), seuls les modèles DATA, GREEDYDATA ou GREEDYDATA_MULTILINE peuvent être inclus après le modèle de journal commun.	

Structure d'un motif Grok

Voici la structure de modèle de grok prise en charge :

```
%{PATTERN_NAME:FIELD_NAME}
```

- PATTERN_NAME : fait référence à une expression régulière prédéfinie pour faire correspondre un type de données spécifique. Seuls les modèles de groupe prédéfinis de la liste des modèles de groupe pris en charge sont pris en charge. La création de modèles personnalisés n'est pas autorisée.
- FIELD_NAME : Attribue un nom à la valeur extraite. `FIELD_NAME`est facultatif, mais si vous ne spécifiez pas cette valeur, les données extraites seront supprimées de l'événement du journal transformé. S'il `FIELD_NAME` utilise une notation en pointillés (par exemple, « parent.child »), il est considéré comme un chemin JSON.
- Conversion de type : les conversions de type explicites ne sont pas prises en charge. Utilisez le TypeConverter processeur pour convertir le type de données de toute valeur extraite par grok.

Pour créer des expressions correspondantes plus complexes, vous pouvez combiner plusieurs modèles de grok. Jusqu'à 20 modèles de grok peuvent être combinés pour correspondre à un événement de journal. Par exemple, cette combinaison de modèles `%{NUMBER:timestamp} [%{NUMBER:db} %{IP:client_ip}:%{NUMBER:client_port}] %{GREEDYDATA:data}` peut être utilisée pour extraire des champs d'une entrée de journal lent de Redis comme suit :

```
1629860738.123456 [0 127.0.0.1:6379] "SET" "key1" "value1"
```

Exemples de grok

Exemple 1 : utiliser grok pour extraire un champ à partir de journaux non structurés

Exemple de journal :

```
293750 server-01.internal-network.local OK "[Thread-000] token generated"
```

Transformateur utilisé :

```
[
     {
         "grok": {
             "match": "%{NUMBER:version} %{HOSTNAME:hostname} %{NOTSPACE:status}
 %{QUOTEDSTRING:logMsg}"
         }
    }
]
```

Sortie :

```
{
  "version": "293750",
  "hostname": "server-01.internal-network.local",
  "status": "OK",
  "logMsg": "[Thread-000] token generated"
}
```

Exemple de journal :

```
23/Nov/2024:10:25:15 -0900 172.16.0.1 200
```

Transformateur utilisé :

```
[
    {
        "grok": {
            "match": "%{HTTPDATE:timestamp} %{IPORHOST:clientip}
 %{NUMBER:response_status}"
        }
    }
]
```

Sortie :

```
{
  "timestamp": "23/Nov/2024:10:25:15 -0900",
  "clientip": "172.16.0.1",
  "response_status": "200"
}
```

Exemple 2 : utilisez grok en combinaison avec parseJSON pour extraire des champs d'un événement de journal JSON

Exemple de journal :

```
{
    "timestamp": "2024-11-23T16:03:12Z",
    "level": "ERROR",
    "logMsg": "GET /page.html HTTP/1.1"
}
```

Transformateur utilisé :

```
[
     {
        "parseJSON": {}
    },
    {
         "grok": {
            "source": "logMsg",
             "match": "%{WORD:http_method} %{NOTSPACE:request} HTTP/
%{NUMBER:http_version}"
         }
    }
```

```
]
```

Sortie :

```
{
  "timestamp": "2024-11-23T16:03:12Z",
  "level": "ERROR",
  "logMsg": "GET /page.html HTTP/1.1",
  "http_method": "GET",
  "request": "/page.html",
  "http_version": "1.1"
}
```

Exemple 3 : modèle Grok avec annotation en pointillés dans FIELD_NAME

Exemple de journal :

```
192.168.1.1 GET /index.html?param=value 200 1234
```

Transformateur utilisé :

```
[
    {
        "grok": {
            "match": "%{IP:client.ip} %{WORD:method} %{URIPATHPARAM:request.uri}
 %{NUMBER:response.status} %{NUMBER:response.bytes}"
        }
    }
]
```

Sortie :

```
{
  "client": {
    "ip": "192.168.1.1"
  },
  "method": "GET",
  "request": {
    "uri": "/index.html?param=value"
  },
  "response": {
    "status": "200",
```

```
    "bytes": "1234"
  }
}
```

Motifs de grok pris en charge

Les tableaux suivants répertorient les modèles pris en charge par le `grok` processeur.

Motifs généraux de grok

Motif Grok	Description	Limite maximale de motifs	exemple
NOM D'UTILISATEUR ou UTILISATEUR	Correspond à un ou plusieurs caractères pouvant inclure des lettres minuscules (a-z), des lettres majuscules (A-Z), des chiffres (0-9), des points (.), des traits de soulignement (_) ou des traits d'union (-)	20	Entrée : `user123.name-TEST` Modèle : `%{USERNAME:name}` Sortie : `{"name": "user123.name-TEST"}`
INT	Correspond à un signe plus ou moins facultatif suivi d'un ou de plusieurs chiffres.	20	Entrée : `-456` Modèle : `%{INT:num}` Sortie : `{"num": "-456"}`
BASE10NUM	Correspond à un entier ou à un nombre à virgule flottante avec un signe et une virgule décimale facultatifs	20	Entrée : `-0.67` Modèle : `%{BASE10NUM:num}` Sortie : `{"num": "-0.67"}`
BASE16NUM	Fait correspondre les nombres décimaux et hexadécimaux avec un signe optionnel (+ ou -) et un préfixe 0x facultatif	20	Entrée : `+0xA1B2` Modèle : `%{BASE16NUM:num}`

Motif Grok	Description	Limite maximale de motifs	exemple
			Sortie : `{"num": "+0xA1B2"}`
POINT	Correspond à des entiers entiers positifs sans zéros en tête, composés d'un ou de plusieurs chiffres (1 à 9 suivis de 0 à 9)	20	Entrée : `123` Modèle : `%{POSINT:num}` Sortie : `{"num": "123"}`
NONNEGINT	Correspond à tous les nombres entiers (composés d'un ou plusieurs chiffres de 0 à 9), y compris zéro et les nombres avec des zéros en tête.	20	Entrée : `007` Modèle : `%{NONNEGINT:num}` Sortie : `{"num": "007"}`
MOT	Correspond à des mots entiers composés d'un ou de plusieurs caractères (\w), y compris des lettres, des chiffres et des traits de soulignement	20	Entrée : `user_123` Modèle : `%{WORD:user}` Sortie : `{"user": "user_123"}`
PAS D'ESPACE	Correspond à un ou plusieurs caractères autres que des espaces blancs.	5	Entrée : `hello_world123` Modèle : `%{NOTSPACE:msg}` Sortie : `{"msg": "hello_world123"}`

Motif Grok	Description	Limite maximale de motifs	exemple
SPACE	Correspond à zéro ou plusieurs caractères d'espacement.	5	Entrée : `" "` Modèle : `%{SPACE:extra}` Sortie : `{"extra": " "}`
DATA	Correspond à n'importe quel caractère (sauf la nouvelle ligne) zéro fois ou plus, ce n'est pas gourmand.	5	Entrée : `abc def ghi` Modèle : `%{DATA:x} %{DATA:y}` Sortie : `{"x": "abc", "y": "def ghi"}`
DONNÉES GOURMANDES	Correspond à n'importe quel caractère (sauf la nouvelle ligne) zéro fois ou plus, gourmand.	5	Entrée : `abc def ghi` Modèle : `%{GREEDYDATA:x} %{GREEDYDATA:y}` Sortie : `{"x": "abc def", "y": "ghi"}`

GREEDYDATA_MULTILINE	Correspond à n'importe quel personnage (y compris une nouvelle ligne) zéro fois ou plus, gourmand.	1	Entrée :
			
			`abc`
			
			`def`
			
			`ghi`
			
			Modèle : `%{GREEDYDATA_MULTILINE:data}`
			
			Sortie : `{"data": "abc \ndef\nghi"}`

Motif Grok	Description	Limite maximale de motifs	exemple
CHAÎNE ENTRE GUILLEMETS	Fait correspondre les chaînes entre guillemets (guillemets simples ou doubles) aux caractères échappés.	20	Entrée : `"Hello, world!"` Modèle : `%{QUOTEDSTRING:msg}` Sortie : `{"msg": "Hello, world!"}`
UUID	Correspond à un format UUID standard : 8 caractères hexadécimaux, suivis de trois groupes de 4 caractères hexadécimaux et se terminant par 12 caractères hexadécimaux, tous séparés par des tirets.	20	Entrée : `550e8400-e29b-41d4-a716-446655440000` Modèle : `%{UUID:id}` Sortie : `{"id": "550e8400-e29b-41d4-a716-446655440000"}`
URNE	Correspond à la syntaxe URN (Uniform Resource Name).	20	Entrée : `urn:isbn:0451450523` Modèle : `%{URN:urn}` Sortie : `{"urn": "urn:isbn:0451450523"}`

AWS motifs de grok

Modèle	Description	Limite maximale de motifs	exemple
ARN	Correspond aux noms de ressources AWS Amazon	5	Entrée : `arn:aws:iam:us-east-1:12345`

Modèle	Description	Limite maximale de motifs	exemple
	(ARNs), en capturant la partition (`awsaws-cn`, ou`aws-us-gov`), le service, la région, l'ID de compte et jusqu'à 5 identifiants de ressources hiérarchiques séparés par des barres obliques. Cela ne correspon dra pas ARNs aux informati ons manquantes entre deux points.		`6789012:user/johnd oe` Modèle : `%{ARN:arn}` Sortie : `{"arn": "arn:aws: iam:us-east-1:1234 56789012:user/john doe"}`

Modèles de réseaux

Motif Grok	Description	Limite maximale de motifs	exemple
CISCOMAC	Correspond à une adresse MAC au format hexadécimal 4-4-4.	20	Entrée : `0123.4567.89AB` Modèle : `%{CISCOMA C:MacAddress}` Sortie : `{"MacAddress": "0123.4567.89AB"}`
WINDOWSMA C	Correspond à une adresse MAC au format hexadécimal avec des tirets	20	Entrée : `01-23-45-67-89-AB` Modèle : `%{WINDOWS MAC:MacAddress}`

Motif Grok	Description	Limite maximale de motifs	exemple
			Sortie : `{"MacAddress": "01-23-45-67-89-AB"}`
MAC COMMUN	Correspond à une adresse MAC au format hexadécimal avec deux points.	20	Entrée : `01:23:45:67:89:AB` Modèle : `%{COMMONMAC:MacAddress}` Sortie : `{"MacAddress": "01:23:45:67:89:AB"}`
MAC	Correspond à l'un des modèles de grappes CISCOMAC, WINDOWSMAC ou COMMONMAC	20	Entrée : `01:23:45:67:89:AB` Modèle : `%{MAC:m1}` Sortie : `{"m1":"01:23:45:67:89:AB"}`
IPV6	Correspond IPv6 aux adresses, y compris aux formulaires compressés et aux adresses IPv4 mappées IPv6 .	5	Entrée : `2001:db8:3333:4444:5555:6666:7777:8888` Modèle : `%{IPV6:ip}` Sortie : `{"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}`
IPV4	Correspond à une IPv4 adresse.	20	Entrée : `192.168.0.1` Modèle : `%{IPV4:ip}` Sortie : `{"ip": "192.168.0.1"}`

Motif Grok	Description	Limite maximale de motifs	exemple
IP	Correspond soit IPv6 aux adresses prises en charge par % {IPv6}, soit IPv4 aux adresses prises en charge par % {IPv4}	5	Entrée : `192.168.0.1` Modèle : `%{IP:ip}` Sortie : `{"ip": "192.168.0.1"}`
HOSTNAME ou HOST	Correspond aux noms de domaine, y compris les sous-domaines	5	Entrée : `server-01.internal-network.local` Modèle : `%{HOST:host}` Sortie : `{"host": "server-01.internal-network.local"}`
IP OU HÔTE	Correspond à un nom d'hôte ou à une adresse IP	5	Entrée : `2001:db8:3333:4444:5555:6666:7777:8888` Modèle : `%{IPORHOST:ip}` Sortie : `{"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}`
PORT HÔTE	Correspond à une adresse IP ou à un nom d'hôte tel que pris en charge par le modèle % {IPORHOST} suivi de deux points et d'un numéro de port, capturant le port sous la forme « PORT » dans la sortie.	5	Entrée : `192.168.1.1:8080` Modèle : `%{HOSTPORT:ip}` Sortie : `{"ip":"192.168.1.1:8080","PORT":"8080"}`

Motif Grok	Description	Limite maximale de motifs	exemple
URI-HOST	Correspond à une adresse IP ou à un nom d'hôte tel que pris en charge par le modèle %{IPORHOST}, éventuellement suivi de deux points et d'un numéro de port, capturant le port en tant que « port » s'il est présent.	5	Entrée : `example.com:443 10.0.0.1` Modèle : `%{URIHOST:host} %{URIHOST:ip}` Sortie : `{"host":"example.com:443","port":"443","ip":"10.0.0.1"}`

Motifs de bandes de chemins

Motif Grok	Description	Limite maximale de motifs	exemple
UNIXPATH	Correspond aux chemins d'URL, y compris éventuellement les paramètres de requête.	20	Entrée : `/search?q=regex` Modèle : `%{UNIXPATH:path}` Sortie : `{"path":"/search?q=regex"}`
WINPATH	Correspond aux chemins de fichiers Windows.	5	Entrée : `C:\Users\John\Documents\file.txt` Modèle : `%{WINPATH:path}` Sortie : `{"path": "C:\\Users\\John\\`

Motif Grok	Description	Limite maximale de motifs	exemple
			`Documents\\file.tx t"}`
PATH	Correspond à une URL ou à un chemin de fichier Windows	5	Entrée : `/search?q=regex` Modèle : `%{PATH:path}` Sortie : `{"path":"/ search?q=regex"}`
TTY	Correspond aux chemins des périphériques Unix pour les terminaux et les pseudo-te rminaux.	20	Entrée : `/dev/tty1` Modèle : `%{TTY:path}` Sortie : `{"path":"/dev/ tty1"}`
URIPROTO	Correspond aux lettres, éventuellement suivies d'un caractère plus (+) et de lettres supplémentaires ou de caractères plus (+)	20	Entrée : `web+transformer` Modèle : `%{URIPROT O:protocol}` Sortie : `{"protocol":"web +transformer"}`
URIPATH	Correspond au composant de chemin d'un URI	20	Entrée : `/category/ sub-category/prod uct_name` Modèle : `%{URIPATH :path}` Sortie : `{"path":"/ category/sub-cate gory/product_name"}`

Motif Grok	Description	Limite maximale de motifs	exemple
URIPARAM	Correspond aux paramètres de requête d'URL	5	Entrée : `?param1=value1&param2=value2` Modèle : `%{URIPARAM:url}` Sortie : `{"url":"?param1=value1&param2=value2"}`
PARAMÈTRE URIPATH	Correspond à un chemin d'URI éventuellement suivi de paramètres de requête	5	Entrée : `/category/sub-category/product?id=12345&color=red` Modèle : `%{URIPATHPARAM:path}` Sortie : `{"path":"/category/sub-category/product?id=12345&color=red"}`

Motif Grok	Description	Limite maximale de motifs	exemple
URI	Correspond à un URI complet	5	Entrée : `https://user:password@example.com/path/to/resource?param1=value1&param2=value2` Modèle : `%{URI:uri}` Sortie : `{"path":"https://user:password@example.com/path/to/resource?param1=value1&param2=value2"}`

Schémas de plage de dates et d'heures

Motif Grok	Description	Limite maximale de motifs	exemple
MONTH	Correspond aux noms de mois anglais complets ou abrégés sous forme de mots entiers	20	Entrée : `Jan` Modèle : `%{MONTH:month}` Sortie : `{"month": "Jan"}` Entrée : `January` Modèle : `%{MONTH:month}` Sortie : `{"month": "January"}`

Motif Grok	Description	Limite maximale de motifs	exemple
MONTHNUM	Correspond aux numéros des mois compris entre 1 et 12, avec un zéro en tête optionnel pour les mois à un chiffre.	20	Entrée : `5` Modèle : `%{MONTHNUM:month}` Sortie : `{"month": "5"}` Entrée : `05` Modèle : `%{MONTHNUM:month}` Sortie : `{"month": "05"}`
MONTHNUM	Correspond aux numéros de mois à deux chiffres compris entre 01 et 12.	20	Entrée : `05` Modèle : `%{MONTHNUM2:month}` Sortie : `{"month": "05"}`
JOUR DU MOIS	Correspond au jour du mois compris entre 1 et 31, avec un zéro en tête optionnel.	20	Entrée : `31` Modèle : `%{MONTHDAY:monthDay}` Sortie : `{"monthDay":"31"}`

Motif Grok	Description	Limite maximale de motifs	exemple
YEAR	Correspond à l'année en deux ou quatre chiffres	20	Entrée : `2024` Modèle : `%{YEAR:year}` Sortie : `{"year":"2024"}` Entrée : `24` Modèle : `%{YEAR:year}` Sortie : `{"year":"24"}`
DAY	Correspond aux noms de jour complets ou abrégés.	20	Entrée : `Tuesday` Modèle : `%{DAY:day}` Sortie : `{"day":"Tuesday"}`
HOUR	Correspond à l'heure au format 24 heures avec un zéro (0) 0-23 en tête en option.	20	Entrée : `22` Modèle : `%{HOUR:hour}` Sortie : `{"hour":"22"}`
MINUTE	Correspond aux minutes (00-59).	20	Entrée : `59` Modèle : `%{MINUTE:min}` Sortie : `{"min":"59"}`

Motif Grok	Description	Limite maximale de motifs	exemple
SECOND	Correspond à un nombre représentant les secondes (0) 0 à 60, éventuellement suivi d'un point décimal ou de deux points et d'un ou plusieurs chiffres pour des fractions de minutes	20	Entrée : `3` Modèle : `%{SECOND:second}` Sortie : `{"second":"3"}` Entrée : `30.5` Modèle : `%{SECOND:minSec}` Sortie : `{"minSec":"30.5"}` Entrée : `30:5` Modèle : `%{SECOND:minSec}` Sortie : `{"minSec":"30:5"}`
TIME	Correspond à un format horaire avec des heures, des minutes et des secondes au format (H) H:mm : (s) s. Les secondes incluent les secondes intercalaires (0) de 0 à 60.	20	Entrée : `09:45:32` Modèle : `%{TIME:time}` Sortie : `{"time":"09:45:32"}`

Motif Grok	Description	Limite maximale de motifs	exemple
DATE_US	Correspond à une date au format (M) M/ (d) d/ (yy) yy ou (M) M- (d) d- (yy) yy.	20	Entrée : `11/23/2024` Modèle : `%{DATE_US:date}` Sortie : `{"date":"11/23/2024"}` Entrée : `1-01-24` Modèle : `%{DATE_US:date}` Sortie : `{"date":"1-01-24"}`
DATE_EU	Correspond à la date au format (d) d/ (M) M/ (yy) yy, (d) d- (M) M- (yy) yy, ou (d) d. (M) M. (yy) yy.	20	Entrée : `23/11/2024` Modèle : `%{DATE_EU:date}` Sortie : `{"date":"23/11/2024"}` Entrée : `1.01.24` Modèle : `%{DATE_EU:date}` Sortie : `{"date":"1.01.24"}`

Motif Grok	Description	Limite maximale de motifs	exemple
ISO8601_FUSEAU HORAIRE	Correspond au décalage UTC « Z » ou au décalage de fuseau horaire avec deux points optionnels au format [+-] (H) H (:) mm.	20	Entrée : `+05:30` Modèle : `%{ISO8601_TIMEZONE:tz}` Sortie : `{"tz":"+05:30"}` Entrée : `-530` Modèle : `%{ISO8601_TIMEZONE:tz}` Sortie : `{"tz":"-530"}` Entrée : `Z` Modèle : `%{ISO8601_TIMEZONE:tz}` Sortie : `{"tz":"Z"}`
ISO8601_SECONDE	Correspond à un nombre représentant les secondes (0) de 0 à 60, éventuellement suivi d'un point décimal ou de deux points et d'un ou plusieurs chiffres pour des fractions de secondes	20	Entrée : `60` Modèle : `%{ISO8601_SECOND:second}` Sortie : `{"second":"60"}`

Motif Grok	Description	Limite maximale de motifs	exemple
HORODATAGE_ 01 ISO86	Correspond au format date/heure ISO86 01 (yy) yy- (M) M- (d) dT (H) H:mm : ((s) s) (Z|[+-] (H) H:mm) avec les secondes et le fuseau horaire facultatifs.	20	Entrée : `2023-05-15T14:30:00+05:30` Modèle : `%{TIMESTAMP_ISO8601:timestamp}` Sortie : `{"timestamp":"2023-05-15T14:30:00+05:30"}` Entrée : `23-5-1T1:25+5:30` Modèle : `%{TIMESTAMP_ISO8601:timestamp}` Sortie : `{"timestamp":"23-5-1T1:25+5:30"}` Entrée : `23-5-1T1:25Z` Modèle : `%{TIMESTAMP_ISO8601:timestamp}` Sortie : `{"timestamp":"23-5-1T1:25Z"}`

Motif Grok	Description	Limite maximale de motifs	exemple
DATE	Correspond à une date au format américain en utilisant % {DATE_US} ou au format européen en utilisant % {DATE_EU}	20	Entrée : `11/29/2024` Modèle : `%{DATE:date}` Sortie : `{"date":"11/29/2024"}` Entrée : `29.11.2024` Modèle : `%{DATE:date}` Sortie : `{"date":"29.11.2024"}`
HORODATAGE	Correspond à % {DATE} suivi du modèle % {TIME}, séparé par un espace ou un trait d'union.	20	Entrée : `29-11-2024 14:30:00` Modèle : `%{DATESTAMP:dateTime}` Sortie : `{"dateTime":"29-11-2024 14:30:00"}`
TZ	Correspond aux abréviations de fuseau horaire courantes (PST, PDT, MST, MDT, CST CDT, EST, EDT, UTC).	20	Entrée : `PDT` Modèle : `%{TZ:tz}` Sortie : `{"tz":"PDT"}`

Motif Grok	Description	Limite maximale de motifs	exemple
DATESTAMP_RFC822	Correspond à la date et à l'heure au format : Jour MonthName (D) D (YY) YY (H) H:mm :(s) s Fuseau horaire	20	Entrée : `Monday Jan 5 23 1:30:00 CDT` Modèle : `%{DATESTAMP_RFC822:dateTime}` Sortie : `{"dateTime":"Monday Jan 5 23 1:30:00 CDT"}` Entrée : `Mon January 15 2023 14:30:00 PST` Modèle : `%{DATESTAMP_RFC822:dateTime}` Sortie : `{"dateTime":"Mon January 15 2023 14:30:00 PST"}`

Motif Grok	Description	Limite maximale de motifs	exemple
DATESTAMP_RFC2822	Correspond au format RFC2822 date-heure : Jour, (d) d MonthName (yy) yy (H) H:mm :(s) s Z|[+-] (H) H:mm	20	Entrée : `Mon, 15 May 2023 14:30:00 +0530` Modèle : `%{DATESTAMP_RFC2822:dateTime}` Sortie : `{"dateTime":"Mon, 15 May 2023 14:30:00 +0530"}` Entrée : `Monday, 15 Jan 23 14:30:00 Z` Modèle : `%{DATESTAMP_RFC2822:dateTime}` Sortie : `{"dateTime":"Monday, 15 Jan 23 14:30:00 Z"}`
DATESTAMP_AUTRE	Correspond à la date et à l'heure au format : Jour MonthName (d) d (H) H:mm :(s) s Fuseau horaire (yy) yy	20	Entrée : `Mon May 15 14:30:00 PST 2023` Modèle : `%{DATESTAMP_OTHER:dateTime}` Sortie : `{"dateTime":"Mon May 15 14:30:00 PST 2023"}`

Motif Grok	Description	Limite maximale de motifs	exemple
DATESTAMP_EVENTLOG	Correspond au format date/heure compact sans séparateurs : (yy) YYmm (d) d (H) Hmm (s) s	20	Entrée : `20230515143000` Modèle : `%{DATESTAMP_EVENTLOG:dateTime}` Sortie : `{"dateTime":"20230515143000"}`

Motifs en bois

Motif Grok	Description	Limite maximale de motifs	exemple
NIVEAU DU JOURNAL	Correspond aux niveaux de log standard dans différentes majuscules et abréviations, notamment les suivantes :`Alert/ALERT` ,`Trace/TRACE` ,`Debug/DEBUG` ,`Notice/NOTICE` ,,`Info/INFO` ,`Warn/Warning/WARN/WARNING` ,`Err/Error/ERR/ERROR` , `Crit/Critical/CRIT/CRITICAL` `Fatal/FATAL` `Severe/SEVERE` `Emerg/Emergency/EMERG/EMERGENCY`	20	Entrée : `INFO` Modèle : `%{LOGLEVEL:logLevel}` Sortie : `{"logLevel":"INFO"}`

Motif Grok	Description	Limite maximale de motifs	exemple
HTTPDATE	Correspond au format de date et d'heure souvent utilisé dans les fichiers journaux. Format : (d) MonthName d/ (yy) yy :(H) H:mm :(s) s Fuseau horaire : correspond aux noms de mois anglais complets ou abrégés (exemple MonthName : « Jan » ou « January ») Fuseau horaire : Correspond au modèle % {INT} grok	20	Entrée : `23/Nov/2024:14:30:00 +0640` Modèle : `%{HTTPDATE:date}` Sortie : `{"date":"23/Nov/2024:14:30:00 +0640"}`
HORODATAGE SYSLOG	Correspond au format de date avec MonthName (d) d (H) H:mm :(s) s MonthName : correspond aux noms de mois anglais complets ou abrégés (exemple : « Jan » ou « January »)	20	Entrée : `Nov 29 14:30:00` Modèle : `%{SYSLOGTIMESTAMP:dateTime}` Sortie : `{"dateTime":"Nov 29 14:30:00"}`
PROG	Correspond à un nom de programme composé d'une chaîne de lettres, de chiffres, de points, d'un trait de soulignement, d'une barre oblique, d'un signe de pourcentage et de tirets.	20	Entrée : `user.profile/settings-page` Modèle : `%{PROG:program}` Sortie : `{"program":"user.profile/settings-page"}`

Motif Grok	Description	Limite maximale de motifs	exemple
SYSLOGPROG	Correspond au modèle PROG grok éventuellement suivi d'un identifiant de processus entre crochets.	20	Entrée : `user.profile/settings-page[1234]` Modèle : `%{SYSLOGPROG:programWithId}` Sortie : `{"programWithId":"user.profile/settings-page[1234]","program":"user.profile/settings-page","pid":"1234"}`
HÔTE SYSLOG	Correspond à un modèle % {HOST} ou % {IP}	5	Entrée : `2001:db8:3333:4444:5555:6666:7777:8888` Modèle : `%{SYSLOGHOST:ip}` Sortie : `{"ip":"2001:db8:3333:4444:5555:6666:7777:8888"}`

Motif Grok	Description	Limite maximale de motifs	exemple
INSTALLATION SYSLOG	Correspond à la priorité Syslog au format décimal. La valeur doit être placée entre crochets (<>).	20	Entrée : `<13.6>` Modèle : `%{SYSLOGFACILITY:syslog}` Sortie : `{"syslog":"<13.6>","facility":"13","priority":"6"}`

Motifs courants de rainurage en rondins

Vous pouvez utiliser des modèles de grok personnalisés prédéfinis pour correspondre aux formats de journal Apache, NGINX et Syslog Protocol (RFC 5424). Lorsque vous utilisez ces modèles spécifiques, ils doivent être les premiers modèles de votre configuration correspondante, et aucun autre modèle ne peut les précéder. De plus, vous ne pouvez les suivre qu'avec une seule donnée. Modèle GREEDYDATA ou GREEDYDATA_MULTILINE.

Modèle grok	Description	Limite maximale de motifs
JOURNAL D'ACCÈS APACHE	Correspond aux journaux d'accès Apache	1
JOURNAL D'ACCÈS NGINX	Correspond aux journaux d'accès NGINX	1
SYSLOG5424	Correspond aux journaux du protocole Syslog (RFC 5424)	1

Vous trouverez ci-dessous des exemples valides et non valides d'utilisation de ces modèles de format de journal courants.

```
"%{NGINX_ACCESS_LOG} %{DATA}" // Valid
"%{SYSLOG5424}%{DATA:logMsg}" // Valid
"%{APACHE_ACCESS_LOG} %{GREEDYDATA:logMsg}" // Valid
"%{APACHE_ACCESS_LOG} %{SYSLOG5424}" // Invalid (multiple common log patterns used)
"%{NGINX_ACCESS_LOG} %{NUMBER:num}" // Invalid (Only GREEDYDATA and DATA patterns are
 supported with common log patterns)
"%{GREEDYDATA:logMsg} %{SYSLOG5424}" // Invalid (GREEDYDATA and DATA patterns are
 supported only after common log patterns)
```

Exemples de formats de journaux courants

Exemple de journal Apache

Exemple de journal :

```
127.0.0.1 - - [03/Aug/2023:12:34:56 +0000] "GET /page.html HTTP/1.1" 200 1234
```

Transformateur :

```
[
     {
        "grok": {
            "match": "%{APACHE_ACCESS_LOG}"
        }
    }
]
```

Sortie :

```
{
    "request": "/page.html",
    "http_method": "GET",
    "status_code": 200,
    "http_version": "1.1",
    "response_size": 1234,
    "remote_host": "127.0.0.1",
    "timestamp": "2023-08-03T12:34:56Z"
}
```

Exemple de journal NGINX

Exemple de journal :

```
192.168.1.100 - Foo [03/Aug/2023:12:34:56 +0000] "GET /account/login.html HTTP/1.1"
 200 42 "https://www.amazon.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36"
```

Transformateur :

```
[
     {
        "grok": {
            "match": "%{NGINX_ACCESS_LOG}"
        }
    }
]
```

Sortie :

```
{
    "request": "/account/login.html",
    "referrer": "https://www.amazon.com/",
    "agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/92.0.4515.131 Safari/537.36",
    "http_method": "GET",
    "status_code": 200,
    "auth_user": "Foo",
    "http_version": "1.1",
    "response_size": 42,
    "remote_host": "192.168.1.100",
    "timestamp": "2023-08-03T12:34:56Z"
}
```

Exemple de journal du protocole Syslog (RFC 5424)

Exemple de journal :

```
<165>1 2003-10-11T22:14:15.003Z mymachine.example.com evntslog - ID47
 [exampleSDID@32473 iut="3" eventSource= "Application" eventID="1011"]
[examplePriority@32473 class="high"]
```

Transformateur :

```
[
     {
        "grok": {
            "match": "%{SYSLOG5424}"
        }
    }
]
```

Sortie :

```
{
  "pri": 165,
  "version": 1,
  "timestamp": "2003-10-11T22:14:15.003Z",
  "hostname": "mymachine.example.com",
  "app": "evntslog",
  "msg_id": "ID47",
  "structured_data": "exampleSDID@32473 iut=\"3\" eventSource= \"Application\" eventID=
\"1011\"",
  "message": "[examplePriority@32473 class=\"high\"]"
}
```

csv

Le processeur csv analyse les valeurs séparées par des virgules (CSV) des événements du journal en colonnes.

Champ	Description	Obligatoire ?	Par défaut	Limites
source	Chemin d'accès au champ du journal des événements qui sera analysé	Non	`@message`	Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3

Champ	Description	Obligatoire ?	Par défaut	Limites
delimiter	Le caractère utilisé pour séparer chaque colonne dans le journal des valeurs d'origine séparées par des virgules	Non	`,`	Longueur maximale : 1 sauf si la valeur est `\t` ou `\s`
Personnage de citation	Caractère utilisé comme qualificatif de texte pour une seule colonne de données	Non	`"`	Longueur maximale : 1
columns	Liste des noms à utiliser pour les colonnes de l'événement du journal transformé.	Non	`[column_1, column_2`	Nombre maximum de colonnes CSV : 100 Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3

Le réglage `delimiter` sur « » `\t` séparera chaque colonne sur un caractère de tabulation et `\t` séparera chaque colonne sur un seul caractère d'espace.

Exemple

Supposons qu'une partie d'un événement de journal ingéré ressemble à ceci :

```
'Akua Mansa':28:'New York: USA'
```

Supposons que nous n'utilisions que le processeur CSV :

```
[
     {
        "csv": {
            "delimiter": ":",
            "quoteCharacter": "'"
        }
```

```
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "column_1": "Akua Mansa",
  "column_2": "28",
  "column_3": "New York: USA"
}
```

parseKeyValue

Utilisez le parseKeyValueprocesseur pour analyser un champ spécifié en paires clé-valeur. Vous pouvez personnaliser le processeur pour analyser les informations des champs à l'aide des options suivantes.

Champ	Description	Obligatoire ?	Par défaut	Limites
source	Chemin d'accès au champ du journal des événements qui sera analysé	Non	`@message`	Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
destination	Le champ de destination dans lequel placer les paires clé-valeur extraites	Non		Longueur maximale : 128
Délimiteur de champs	Chaîne de délimitation de champs utilisée entre les paires clé-valeur dans les événements du journal d'origine	Non	&	Longueur maximale : 128
keyValueDelimiter	Chaîne de délimitation à utiliser entre la clé et la valeur dans	Non	=	Longueur maximale : 128

Champ	Description	Obligatoire ?	Par défaut	Limites
	chaque paire dans l'événement du journal transformé			
nonMatchValue	Une valeur à insérer dans le champ de valeur du résultat, lorsqu'une paire clé-valeur n'est pas correctement divisée.	Non		Longueur maximale : 128
keyPrefix	Si vous souhaitez ajouter un préfixe à toutes les clés transformées, spécifiez-le ici.	Non		Longueur maximale : 128
overwriteIfExists	S'il faut remplacer la valeur si la clé de destination existe déjà	Non	`false`	

Exemple

Prenons l'exemple d'événement de journal suivant :

```
key1:value1!key2:value2!key3:value3!key4
```

Supposons que nous utilisions la configuration de processeur suivante :

```
[
    {
        "parseKeyValue": {
            "destination": "new_key",
            "fieldDelimiter": "!",
            "keyValueDelimiter": ":",
            "nonMatchValue": "defaultValue",
            "keyPrefix": "parsed_"
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "new_key": {
    "parsed_key1": "value1",
    "parsed_key2": "value2",
    "parsed_key3": "value3",
    "parsed_key4": "defaultValue"
  }
}
```

Processeurs intégrés pour les AWS journaux vendus

Analyser WAF

Utilisez ce processeur pour analyser AWS WAF les journaux vendus. Il prend le contenu de chaque nom d'en-tête `httpRequest.headers` et crée des clés JSON à partir de celui-ci, avec la valeur correspondante. Il en va de même pour`labels`. Ces transformations peuvent faciliter considérablement l'interrogation AWS WAF des journaux. Pour plus d'informations sur le format des AWS WAF journaux, voir Exemples de journaux pour le trafic ACL Web.

Ce processeur n'accepte `@message` que comme entrée.

> **Important**
>
> Si vous utilisez ce processeur, il doit s'agir du premier processeur de votre transformateur.

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
  "timestamp": 1576280412771,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
  "terminatingRuleId": "STMTest_SQLi_XSS",
  "terminatingRuleType": "REGULAR",
  "action": "BLOCK",
  "terminatingRuleMatchDetails": [
    {
```

```
      "conditionType": "SQL_INJECTION",
      "sensitivityLevel": "HIGH",
      "location": "HEADER",
      "matchedData": ["10", "AND", "1"]
    }
  ],
  "httpSourceName": "-",
  "httpSourceId": "-",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": [],
  "httpRequest": {
    "clientIp": "1.1.1.1",
    "country": "AU",
    "headers": [
      { "name": "Host", "value": "localhost:1989" },
      { "name": "User-Agent", "value": "curl/7.61.1" },
      { "name": "Accept", "value": "*/*" },
      { "name": "x-stm-test", "value": "10 AND 1=1" }
    ],
    "uri": "/myUri",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "rid"
  },
  "labels": [{ "name": "value" }]
}
```

La configuration du processeur est la suivante :

```
[
    {
        "parseWAF": {}
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "httpRequest": {
    "headers": {
      "Host": "localhost:1989",
```

```
      "User-Agent": "curl/7.61.1",
      "Accept": "*/*",
      "x-stm-test": "10 AND 1=1"
    },
    "clientIp": "1.1.1.1",
    "country": "AU",
    "uri": "/myUri",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "rid"
  },
  "labels": { "name": "value" },
  "timestamp": 1576280412771,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
  "terminatingRuleId": "STMTest_SQLi_XSS",
  "terminatingRuleType": "REGULAR",
  "action": "BLOCK",
  "terminatingRuleMatchDetails": [
    {
      "conditionType": "SQL_INJECTION",
      "sensitivityLevel": "HIGH",
      "location": "HEADER",
      "matchedData": ["10", "AND", "1"]
    }
  ],
  "httpSourceName": "-",
  "httpSourceId": "-",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": []
}
```

Analyser Postgres

Utilisez ce processeur pour analyser les journaux Amazon RDS for PostgreSQL vendus, extraire des champs et les convertir au format JSON. Pour plus d'informations sur le format de journal RDS pour PostgreSQL, consultez la section Fichiers journaux de base de données RDS pour PostgreSQL.

Ce processeur n'accepte `@message` que comme entrée.

> **Important**
>
> Si vous utilisez ce processeur, il doit s'agir du premier processeur de votre transformateur.

Exemple

Prenons l'exemple d'événement de journal suivant :

```
2019-03-10 03:54:59 UTC:10.0.0.123(52834):postgres@logtestdb:[20175]:ERROR: column
 "wrong_column_name" does not exist at character 8
```

La configuration du processeur est la suivante :

```
[
    {
        "parsePostgres": {}
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "logTime": "2019-03-10 03:54:59 UTC",
  "srcIp": "10.0.0.123(52834)",
  "userName": "postgres",
  "dbName": "logtestdb",
  "processId": "20175",
  "logLevel": "ERROR"
}
```

ParseCloud Front

Utilisez ce processeur pour analyser les journaux Amazon CloudFront vendus, extraire des champs et les convertir au format JSON. Les valeurs de champ codées sont décodées. Les valeurs qui sont des nombres entiers et des doubles sont traitées comme telles. Pour plus d'informations sur le format des Amazon CloudFront journaux, voir Configurer et utiliser des journaux standard (journaux d'accès).

Ce processeur n'accepte `@message` que comme entrée.

> **Important**
>
> Si vous utilisez ce processeur, il doit s'agir du premier processeur de votre transformateur.

Exemple

Prenons l'exemple d'événement de journal suivant :

```
2019-12-04  21:02:31   LAX1   392    192.0.2.24    GET
 d111111abcdef8.cloudfront.net  /index.html    200    -  Mozilla/5.0%20(Windows
%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36  -  -  Hit
    SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ==
 d111111abcdef8.cloudfront.net  https  23 0.001  -  TLSv1.2    ECDHE-RSA-AES128-GCM-
SHA256    Hit    HTTP/2.0   -  -  11040  0.001  Hit    text/html  78 -  -
```

La configuration du processeur est la suivante :

```
[
    {
        "parseCloudfront": {}
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "date": "2019-12-04",
  "time": "21:02:31",
  "x-edge-location": "LAX1",
  "sc-bytes": 392,
  "c-ip": "192.0.2.24",
  "cs-method": "GET",
  "cs(Host)": "d111111abcdef8.cloudfront.net",
  "cs-uri-stem": "/index.html",
  "sc-status": 200,
  "cs(Referer)": "-",
  "cs(User-Agent)": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36",
  "cs-uri-query": "-",
```

```
  "cs(Cookie)": "-",
  "x-edge-result-type": "Hit",
  "x-edge-request-id": "SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ==",
  "x-host-header": "d111111abcdef8.cloudfront.net",
  "cs-protocol": "https",
  "cs-bytes": 23,
  "time-taken": 0.001,
  "x-forwarded-for": "-",
  "ssl-protocol": "TLSv1.2",
  "ssl-cipher": "ECDHE-RSA-AES128-GCM-SHA256",
  "x-edge-response-result-type": "Hit",
  "cs-protocol-version": "HTTP/2.0",
  "fle-status": "-",
  "fle-encrypted-fields": "-",
  "c-port": 11040,
  "time-to-first-byte": 0.001,
  "x-edge-detailed-result-type": "Hit",
  "sc-content-type": "text/html",
  "sc-content-len": 78,
  "sc-range-start": "-",
  "sc-range-end": "-"
}
```

Par Route 53

Utilisez ce processeur pour analyser les journaux Amazon Route 53 Public Data Plane vendus, extraire des champs et les convertir au format JSON. Les valeurs de champ codées sont décodées. Ce processeur ne prend pas en charge Amazon Route 53 Resolver les journaux.

Ce processeur n'accepte `@message` que comme entrée.

> **Important**
>
> Si vous utilisez ce processeur, il doit s'agir du premier processeur de votre transformateur.

Exemple

Prenons l'exemple d'événement de journal suivant :

```
1.0 2017-12-13T08:15:50.235Z Z123412341234 example.com AAAA NOERROR TCP IAD12 192.0.2.0
 198.51.100.0/24
```

La configuration du processeur est la suivante :

```
[
    {
        "parseRoute53": {}
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "version": 1.0,
  "queryTimestamp": "2017-12-13T08:15:50.235Z",
  "hostZoneId": "Z123412341234",
  "queryName": "example.com",
  "queryType": "AAAA",
  "responseCode": "NOERROR",
  "protocol": "TCP",
  "edgeLocation": "IAD12",
  "resolverIp": "192.0.2.0",
  "ednsClientSubnet": "198.51.100.0/24"
}
```

Analyser VPC

Utilisez ce processeur pour analyser les journaux Amazon VPC vended, extraire des champs et les convertir au format JSON. Les valeurs de champ codées sont décodées.

Ce processeur n'accepte `@message` que comme entrée.

> **Important**
>
> Si vous utilisez ce processeur, il doit s'agir du premier processeur de votre transformateur.

Exemple

Prenons l'exemple d'événement de journal suivant :

```
2 123456789010 eni-abc123de 192.0.2.0 192.0.2.24 20641 22 6 20 4249 1418530010
 1418530070 ACCEPT OK
```

La configuration du processeur est la suivante :

```
[
    {
        "parseVPC": {}
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "version": 2,
  "accountId": "123456789010",
  "interfaceId": "eni-abc123de",
  "srcAddr": "192.0.2.0",
  "dstAddr": "192.0.2.24",
  "srcPort": 20641,
  "dstPort": 22,
  "protocol": 6,
  "packets": 20,
  "bytes": 4249,
  "start": 1418530010,
  "end": 1418530070,
  "action": "ACCEPT",
  "logStatus": "OK"
}
```

Processeurs String Mutate

lowerCaseString

Le `lowerCaseString` processeur convertit une chaîne en sa version minuscule.

Champ	Description	Obligatoire ?	Par défaut	Limites
Avec clés	Une liste de clés à convertir en minuscules	Oui		Nombre maximum d'entrées : 10

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "INNER_VALUE"
    }
}
```

La configuration du transformateur est la suivante, `lowerCaseString` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "lowerCaseString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  }
}
```

upperCaseString

Le `upperCaseString` processeur convertit une chaîne en sa version majuscule.

Champ	Description	Obligatoire ?	Par défaut	Limites
Avec clés	Une liste de clés à convertir en majuscules	Oui		Nombre maximum d'entrées : 10

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuration du transformateur est la suivante, `upperCaseString` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "upperCaseString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key": "INNER_VALUE"
  }
}
```

Chaîne divisée

Le `splitString` processeur est un type de processeur à mutation de chaîne qui divise un champ en un tableau à l'aide d'un caractère de délimitation.

Champ	Description	Obligatoire ?	Par défaut	Limites
entries	Tableau d'entrées. Chaque élément du tableau doit contenir	Oui		Nombre maximum d'entrées : 10

Champ	Description	Obligatoire ?	Par défaut	Limites
	`source` des `delimiter` champs.			
source	La clé de la valeur du champ à diviser	Oui		Longueur maximale : 128
delimiter	La chaîne de délimitation sur laquelle diviser la valeur du champ	Oui		Longueur maximale : 128

Exemple 1

Prenons l'exemple d'événement de journal suivant :

```
[
    {
        "parseJSON": {}
    },
    {
        "splitString": {
            "entries": [
                {
                    "source": "outer_key.inner_key",
                    "delimiter": "_"
                }
            ]
        }
    }
]
```

La configuration du transformateur est la suivante, `splitString` en utilisant `parseJSON` :

```
[
     {
        "parseJSON": {}
    },
    {
         "splitString": {
            "entries": [
```

```
                {
                    "source": "outer_key.inner_key",
                    "delimiter": "_"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key": [
      "inner",
      "value"
    ]
  }
}
```

Exemple 2

Le délimiteur sur lequel la chaîne doit être divisée peut comporter plusieurs caractères.

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "item1, item2, item3"
    }
}
```

La configuration du transformateur est la suivante :

```
[
     {
        "parseJSON": {}
    },
    {
         "splitString": {
            "entries": [
                {
```

```
                    "source": "outer_key.inner_key",
                    "delimiter": ", "
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key": [
      "item1",
      "item2",
      "item3"
    ]
  }
}
```

Chaîne de remplacement

Le `substituteString` processeur est un type de processeur de mutation de chaîne qui fait correspondre la valeur d'une clé à une expression régulière et remplace toutes les correspondances par une chaîne de remplacement.

Champ	Description	Obligatoire ?	Par défaut	Limites
entries	Tableau d'entrées. Chaque élément du tableau doit contenir des `to` champs `sourcefrom`, et.	Oui		Nombre maximum d'entrées : 10
source	La clé du champ à modifier	Oui		Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3

Champ	Description	Obligatoire ?	Par défaut	Limites
from	Chaîne d'expression régulière à remplacer. Les caractères réguliers spéciaux tels que [et] doivent être évités en utilisant \ \ lorsque vous utilisez des guillemets doubles et avec \ lorsque vous utilisez des guillemets simples ou lorsqu'ils sont configurés à partir du AWS Management Console. Pour plus d'informations, consultez Class Pattern sur le site Web d'Oracle. Vous pouvez intégrer un motif `(...)` pour créer un groupe de capture numéroté et créer des groupes de capture `(?P<group_name>...)` nommés qui peuvent être référencés `to` sur le terrain.	Oui		Longueur maximale : 128
to	La chaîne à substituer pour chaque correspondance de `from` références arrières à des groupes de capture peut être utilisée. Utilisez le formulaire $n pour les groupes numérotés tels que `$1` et utilisez-le `${group_name}` pour les groupes nommés tels que `${my_group}` . >	Oui		Longueur maximale : 128 Nombre maximum de références arrières : 10 Nombre maximum de références arrières dupliquées : 2

Exemple 1

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key1": "[]",
        "inner_key2": "123-345-567",
        "inner_key3": "A cat takes a catnap."
    }
}
```

La configuration du transformateur est la suivante, `substituteString` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "substituteString": {
            "entries": [
                {
                    "source": "outer_key.inner_key1",
                    "from": "\\[\\]",
                    "to": "value1"
                },
                {
                    "source": "outer_key.inner_key2",
                    "from": "[0-9]{3}-[0-9]{3}-[0-9]{3}",
                    "to": "xxx-xxx-xxx"
                },
                {
                    "source": "outer_key.inner_key3",
                    "from": "cat",
                    "to": "dog"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
```

```
    "inner_key1": "value1",
    "inner_key2": "xxx-xxx-xxx",
    "inner_key3": "A dog takes a dognap."
  }
}
```

Exemple 2

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key1": "Tom, Dick, and Harry",
        "inner_key2": "arn:aws:sts::123456789012:assumed-role/MyImportantRole/
MySession"
    }
}
```

La configuration du transformateur est la suivante, `substituteString` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "substituteString": {
            "entries": [
                {
                    "source": "outer_key.inner_key1",
                    "from": "(\w+), (\w+), and (\w+)",
                    "to": "$1 and $3"
                },
                {
                    "source": "outer_key.inner_key2",
                    "from": "^arn:aws:sts::(?P<account_id>\\d{12}):assumed-role/(?
P<role_name>[\\w+=,.@-]+)/(?P<role_session_name>[\\w+=,.@-]+)$",
                    "to": "${account_id}:${role_name}:${role_session_name}"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key1": "Tom and Harry",
    "inner_key2": "123456789012:MyImportantRole:MySession"
  }
}
```

TrimString

Le `trimString` processeur supprime les espaces entre le début et la fin d'une touche.

Champ	Description	Obligatoire ?	Par défaut	Limites
Avec clés	Une liste de clés à découper	Oui		Nombre maximum d'entrées : 10

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "   inner_value  "
    }
}
```

La configuration du transformateur est la suivante, `trimString` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "trimString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
```

```
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  }
}
```

Processeurs de mutation JSON

Ajouter des clés

Utilisez le `addKeys` processeur pour ajouter de nouvelles paires clé-valeur à l'événement du journal.

Champ	Description	Obligatoire ?	Par défaut	Limites
entries	Tableau d'entrées. Chaque élément du tableau peut contenir des `overwriteIfExists` champs `keyvalue`, et.	Oui		Nombre maximum d'entrées : 5
clé	La clé de la nouvelle entrée à ajouter	Oui		Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
value	La valeur de la nouvelle entrée à ajouter	Oui		Longueur maximale : 256
overwriteIfExists	Si vous définissez ce paramètre sur`true`, la valeur existante est remplacée si elle existe `key` déjà dans l'événement. La valeur par défaut est `false`.	Non	false	Aucune limite

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuration du transformateur est la suivante, `addKeys` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "addKeys": {
            "entries": [
                {
                    "source": "outer_key.new_key",
                    "value": "new_value"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key": "inner_value",
    "new_key": "new_value"
  }
}
```

Supprimer les clés

Utilisez le `deleteKeys` processeur pour supprimer des champs d'un événement du journal. Ces champs peuvent inclure des paires clé-valeur.

Champ	Description	Obligatoire ?	Par défaut	Limites
Avec clés	La liste des clés à supprimer.	Oui	Aucune limite	Nombre maximum d'entrées : 5

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuration du transformateur est la suivante, `deleteKeys` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "deleteKeys": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {}
}
```

Touches de déplacement

Utilisez le `moveKeys` processeur pour déplacer une touche d'un champ à l'autre.

Champ	Description	Obligatoire ?	Par défaut	Limites
entries	Tableau d'entrées. Chaque élément du tableau peut contenir des `overwriteIfExists` champs `sourcetarget`, et.	Oui		Nombre maximum d'entrées : 5
source	La clé pour bouger	Oui		Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
cible	La clé pour passer à	Oui		Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
overwriteIfExists	Si vous définissez ce paramètre sur`true`, la valeur existante est remplacée si elle existe `key` déjà dans l'événement. La valeur par défaut est `false`.	Non	false	Aucune limite

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key1": {
        "inner_key1": "inner_value1"
    },
    "outer_key2": {
        "inner_key2": "inner_value2"
    }
```

```
}
```

La configuration du transformateur est la suivante, `moveKeys` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "moveKeys": {
            "entries": [
                {
                    "source": "outer_key1.inner_key1",
                    "target": "outer_key2"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key1": {},
  "outer_key2": {
    "inner_key2": "inner_value2",
    "inner_key1": "inner_value1"
  }
}
```

Renommer les clés

Utilisez le `renameKeys` processeur pour renommer les clés dans un événement de journal.

Champ	Description	Obligatoire ?	Par défaut	Limites
entries	Tableau d'entrées. Chaque élément du tableau peut contenir des `overwriteIfExists` champs `keytarget`, et.	Oui	Aucune limite	Nombre maximum d'entrées : 5

Champ	Description	Obligatoire ?	Par défaut	Limites
clé	La clé pour renommer	Oui	Aucune limite	Longueur maximale : 128
cible	Le nouveau nom de clé	Oui	Aucune limite	Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
overwriteIfExists	Si vous définissez ce paramètre sur`true`, la valeur existante est remplacée si elle existe `key` déjà dans l'événement. La valeur par défaut est `false`.	Non	false	Aucune limite

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuration du transformateur est la suivante, `renameKeys` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "renameKeys": {
            "entries": [
                {
                    "key": "outer_key",
```

```
                    "target": "new_key"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "new_key": {
    "inner_key": "inner_value"
  }
}
```

Copier la valeur

Utilisez le `copyValue` processeur pour copier des valeurs dans un journal d'événements. Vous pouvez également utiliser ce processeur pour ajouter des métadonnées aux événements du journal, en copiant les valeurs des clés de métadonnées suivantes dans le journal des événements :`@logGroupName`,`@logGroupStream`,`@accountId`,`@regionName`. Ceci est illustré dans l'exemple suivant.

Champ	Description	Obligatoire ?	Par défaut	Limites
entries	Tableau d'entrées. Chaque élément du tableau peut contenir des `overwriteIfExists` champs `sourcetarget`, et.	Oui		Nombre maximum d'entrées : 5
source	La clé à copier	Oui		Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
cible	La clé vers laquelle copier la valeur	Oui	Aucune limite	Longueur maximale : 128

Champ	Description	Obligatoire ?	Par défaut	Limites
				Profondeur maximale des touches imbriquées : 3
overwriteIfExists	Si vous définissez ce paramètre sur`true`, la valeur existante est remplacée si elle existe `key` déjà dans l'événement. La valeur par défaut est `false`.	Non	false	Aucune limite

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuration du transformateur est la suivante, `copyValue` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "copyValue": {
            "entries": [
                {
                    "source": "outer_key.new_key",
                    "target": "new_key"
                },
                {
                    "source": "@logGroupName",
                    "target": "log_group_name"
                },
```

```
                {
                    "source": "@logGroupStream",
                    "target": "log_group_stream"
                },
                {
                    "source": "@accountId",
                    "target": "account_id"
                },
                {
                    "source": "@regionName",
                    "target": "region_name"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  },
  "new_key": "inner_value",
  "log_group_name": "myLogGroupName",
  "log_group_stream": "myLogStreamName",
  "account_id": "012345678912",
  "region_name": "us-east-1"
}
```

listToMap

Le `listToMap` processeur prend une liste d'objets contenant des champs clés et les convertit en une carte de clés cibles.

Champ	Description	Obligatoire ?	Par défaut	Limites
source	La clé ProcessingEvent avec une liste d'objets qui seront convertis en carte	Oui		Longueur maximale : 128

Champ	Description	Obligatoire ?	Par défaut	Limites
				Profondeur maximale des touches imbriquées : 3
clé	La clé des champs à extraire sous forme de clés dans la carte générée	Oui		Longueur maximale : 128
Clé de valeur	Si cela est spécifié, les valeurs que vous spécifiez dans ce paramètre seront extraites des `source` objets et insérées dans les valeurs de la carte générée. Dans le cas contraire, les objets originaux de la liste source seront placés dans les valeurs de la carte générée.	Non		Longueur maximale : 128
cible	La clé du champ qui contiendra la carte générée	Non	Nœud racine	Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3

Champ	Description	Obligatoire ?	Par défaut	Limites
flatten	Valeur booléenne indiquant si la liste sera aplatie en éléments individuels ou si les valeurs de la carte générée seront des listes. Par défaut, les valeurs des clés correspondantes seront représentées dans un tableau. Définissez `flatten` sur `true` pour convertir le tableau en une valeur unique basée sur la valeur de`flattenedElement` .	Non	false	
Élément aplati	Si vous définissez `flatten` sur`true`, utilisez `flattenedElement` pour spécifier quel élément, `first` ou`last`, à conserver.	Obligatoire lorsque `flatten` le paramètre est défini sur `true`		La valeur ne peut être que `first` ou `last`

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
```

```
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ]
}
```

Transformateur pour le cas d'utilisation 1 : `flatten` est `false`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": false
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
```

```
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": [
        "val-a"
    ],
    "b": [
        "val-b1",
        "val-b2"
    ],
    "c": [
        "val-c"
    ]
}
```

Transformateur pour le cas d'utilisation 2 : `flatten` est `true` et `flattenedElement` est `first`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": true,
            "flattenedElement": "first"
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
```

```
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": "val-a",
    "b": "val-b1",
    "c": "val-c"
}
```

Transformateur pour le cas d'utilisation 3 : `flatten` est `true` et `flattenedElement` est `last`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": true,
            "flattenedElement": "last"
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
```

```
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": "val-a",
    "b": "val-b2",
    "c": "val-c"
}
```

Processeurs de conversion de types de données

Type de convertisseur

Utilisez le `typeConverter` processeur pour convertir un type de valeur associé à la clé spécifiée en type spécifié. Il s'agit d'un processeur de casting qui modifie les types des champs spécifiés. Les valeurs peuvent être converties dans l'un des types de données suivants :`integer`,`double`, `string` et. `boolean`

Champ	Description	Obligatoire ?	Par défaut	Limites
entries	Tableau d'entrées. Chaque élément du tableau doit contenir `key` des `type` champs.	Oui		Nombre maximum d'entrées : 10
clé	La clé dont la valeur doit être convertie en un autre type	Oui		Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3

Champ	Description	Obligatoire ?	Par défaut	Limites
type	Le type vers lequel effectuer la conversion. Les valeurs valides sont `integer``double`, `string` et`boolean`.	Oui		

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{
    "name": "value",
    "status": "200"
}
```

La configuration du transformateur est la suivante, `typeConverter` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "typeConverter": {
            "entries": [
                {
                    "key": "status",
                    "type": "integer"
                }
            ]
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
    "name": "value",
    "status": 200
```

```
}
```

Convertisseur Date/Heure

Utilisez le `datetimeConverter` processeur pour convertir une chaîne de date/heure dans un format que vous spécifiez.

Champ	Description	Obligatoire ?	Par défaut	Limites
source	La clé à laquelle appliquer la conversion de date.	Oui		Nombre maximum d'entrées : 10
Motifs de match	Une liste de modèles à comparer au `source` champ	Oui		Nombre maximum d'entrées : 5
cible	Le champ JSON dans lequel stocker le résultat.	Oui		Longueur maximale : 128 Profondeur maximale des touches imbriquées : 3
Format cible	Format de date/heure à utiliser pour les données converties dans le champ cible.	Non	`yyyy-MM-dd'T'HH:mm:ss.SSS'Z`	Longueur maximale : 64
Fuseau horaire de la source	Fuseau horaire du champ source. Pour obtenir la liste des valeurs possibles, consultez la section Identifiants et décalages de zone pris en charge par Java.	Non	UTC	Longueur minimale : 1
Fuseau horaire cible	Fuseau horaire du champ cible.	Non	UTC	Longueur minimale : 1

Champ	Description	Obligatoire ?	Par défaut	Limites
	Pour obtenir la liste des valeurs possibles, consultez la section Identifiants et décalages de zone pris en charge par Java.			
locale	Les paramètres régionaux du champ source. Pour une liste des valeurs possibles, consultez la méthode locale getAvailableLocales () en Java avec des exemples.	Oui		Longueur minimale : 1

Exemple

Prenons l'exemple d'événement de journal suivant :

```
{"german_datetime": "Samstag 05. Dezember 1998 11:00:00"}
```

La configuration du transformateur est la suivante, `dateTimeConverter` en utilisant `parseJSON` :

```
[
    {
        "parseJSON": {}
    },
    {
        "dateTimeConverter": {
            "source": "german_datetime",
            "target": "target_1",
            "locale": "de",
            "matchPatterns": ["EEEE dd. MMMM yyyy HH:mm:ss"],
            "sourceTimezone": "Europe/Berlin",
            "targetTimezone": "America/New_York",
            "targetFormat": "yyyy-MM-dd'T'HH:mm:ss z"
        }
    }
]
```

L'événement du journal transformé serait le suivant.

```
{
    "german_datetime": "Samstag 05. Dezember 1998 11:00:00",
    "target_1": "1998-12-05T17:00:00 MEZ"
}
```

Métriques de transformation et erreurs

CloudWatch Logs publie les indicateurs de transformation sur CloudWatch. Ces indicateurs incluent `TransformedLogEventsTransformedBytes`, et`TransformationErrors`. Pour de plus amples informations, veuillez consulter Métriques et dimensions des transformateurs de log.

Chaque fois que CloudWatch Logs essaie et échoue de transformer un événement de journal, il ajoute un champ `@transformationError` système à cet événement de journal. Lorsque vous exécutez une requête CloudWatch Logs Insights, ce champ apparaît dans tous les événements du journal qui ont connu un échec de transformation. Vous pouvez rechercher ce champ à l'aide d'une requête permettant `filter ispresent(@transformationError)` de rechercher tous les événements de transformation ayant échoué.

Analyser avec Amazon OpenSearch Service

CloudWatch Logs s'intègre Amazon OpenSearch Service pour vous permettre de créer des tableaux de bord automatiques qui affichent les indicateurs clés que le OpenSearch Service obtient à partir des journaux vendus par les services. AWS Les tableaux de bord suivants sont disponibles :

- Un tableau de bord des journaux de flux Amazon VPC capture les données de flux réseau pour Amazon VPC. Il vous aide à analyser le trafic réseau, à détecter les modèles inhabituels et à surveiller l'utilisation des ressources. Les indicateurs clés affichés sont les suivants :
  - Flux totaux, acceptation et rejet de ces flux
  - Schémas du trafic au fil du temps
  - Un diagramme de Sankey illustrant le flux de données entre la source et la destination IPs (top talkers)
  - Nombre IPs d'octets et de paquets transférés

  > **Note**
  >
  > Actuellement, seul le format de champs VPC version 2 est pris en charge.

- Un tableau de bord des AWS WAF journaux fournit des informations sur le trafic Web surveillé par AWS WAF. Ce tableau de bord vous aide à identifier les modèles de trafic, les demandes bloquées et les menaces potentielles provenant de régions ou de sites spécifiques IPs. Les indicateurs clés affichés sont les suivants :
  - Nombre total de demandes, y compris selon le nombre de demandes « AUTORISER » et « BLOQUER ».
  - Historique des demandes au fil du temps, avec affichage des demandes autorisées et bloquées.
  - Répartition des demandes par nom d'ACL Web, demandes bloquées par règle de terminaison et source. IPs
  - Répartition géographique des origines des demandes.
  - Principaux clients IPs et règles de terminaison en fonction du nombre de demandes.
- Un tableau de bord CloudTrail des journaux fournit une vue d'ensemble de l'activité des API au sein de votre AWS environnement à l'aide de CloudTrail journaux. Il est utile pour surveiller l'activité des API, auditer les actions et identifier les problèmes potentiels de sécurité ou de conformité. Les indicateurs clés affichés sont les suivants :

  - Nombre total d'événements et historique des événements au fil du temps
  - Répartition des événements par compte IDs, catégories et régions.
  - Top APIs, services et source IPs impliqués dans la génération d'événements.
  - Tableau des principaux utilisateurs qui génèrent des événements, détaillant les informations relatives aux comptes utilisateurs et le nombre d'événements.
- Un AWS Network Firewalltableau de bord améliore la visibilité du trafic réseau et fournit des informations précieuses pour la surveillance et l'analyse de la sécurité. Ce tableau de bord offre une vue complète des différents paramètres et modèles du réseau, afin d'identifier rapidement les problèmes de sécurité potentiels et d'optimiser les configurations réseau. Les indicateurs clés affichés sont les suivants :
  - Les meilleurs orateurs et protocoles
  - Informations sur les PrivateLink terminaux
  - Trafic d'indication de nom de serveur TLS autorisé et bloqué

Les statistiques affichées dans ces tableaux de bord organisés sont dérivées d' Amazon OpenSearch Service analyses.

Avant de pouvoir consulter ces tableaux de bord, vous devez créer un rôle IAM et effectuer une intégration unique des CloudWatch journaux avec. Amazon OpenSearch Service Cette intégration ponctuelle configure les Amazon OpenSearch Service ressources nécessaires pour créer et afficher le tableau de bord. Les OpenSearch services utilisés vous seront facturés. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

Vous pouvez créer ces tableaux de bord organisés uniquement pour les groupes de journaux de la classe de journaux standard.

> **Important**
>
> N'utilisez pas de transformateurs de journaux pour les groupes de journaux pour lesquels vous souhaitez créer des tableaux de bord de journaux vendus. La transformation des événements du journal entraînera l'affichage de données vides dans les tableaux de bord.

Rubriques

- Étape 1 : Création de l'intégration avec le OpenSearch service
- Étape 2 : Création de tableaux de bord de journaux vendus

- Afficher, modifier ou supprimer les tableaux de bord des journaux vendus
- Politiques IAM pour les utilisateurs
- Autorisations nécessaires à l'intégration

Étape 1 : Création de l'intégration avec le OpenSearch service

La première étape consiste à créer l'intégration avec OpenSearch Service, ce que vous ne devez faire qu'une seule fois. La création de l'intégration créera les ressources suivantes dans votre compte.

- Une collection de séries OpenSearch Service chronologiques sans haute disponibilité.

  Une collection est un ensemble d'index de OpenSearch service qui fonctionnent ensemble pour prendre en charge une charge de travail.
- Deux politiques de sécurité pour la collection. L'un définit le type de chiffrement, qui utilise soit une AWS KMS clé gérée par le client, soit une clé appartenant au service. L'autre politique définit l'accès au réseau, permettant à l'application OpenSearch de service d'accéder à la collection. Pour plus d'informations, consultez la section Chiffrement des données au repos pour Amazon OpenSearch Service.
- Une politique d'accès aux données de OpenSearch service qui définit qui peut accéder aux données de la collection.
- Une source de données de requête directe de type OpenSearch Service dont CloudWatch les journaux sont définis comme source.
- Une application OpenSearch de service portant le nom`aws-analytics`. L'application sera configurée pour permettre la création d'un espace de travail. Si une application nommée existe `aws-analytics` déjà, elle sera mise à jour pour ajouter cette collection en tant que source de données.
- Un espace OpenSearch de travail de service qui hébergera les tableaux de bord et permettra à tous ceux qui y ont accès de lire depuis l'espace de travail.

Rubriques

- Autorisations requises
- Création de l'intégration

Autorisations requises

Pour créer l'intégration, vous devez être connecté à un compte doté de la politique IAM CloudWatchOpenSearchDashboardsFullAccessgérée ou d'autorisations équivalentes, comme indiqué ici. Vous devez également disposer de ces autorisations pour supprimer l'intégration, créer, modifier et supprimer des tableaux de bord, et pour actualiser le tableau de bord manuellement.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "CloudWatchOpenSearchDashboardsIntegration",
            "Effect": "Allow",
            "Action": [
                "logs:ListIntegrations",
                "logs:GetIntegration",
                "logs:DeleteIntegration",
                "logs:PutIntegration",
                "logs:DescribeLogGroups",
                "opensearch:ApplicationAccessAll",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLogsOpensearchReadAPIs",
            "Effect": "Allow",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "es:ListApplications"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
```

```
            "Sid": "CloudWatchLogsOpensearchCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsObservabilityCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
observability.aoss.amazonaws.com/AWSServiceRoleForAmazonOpenSearchServerless",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "observability.aoss.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionRequestAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateCollection"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
```

```
                "aws:TagKeys": "CloudWatchOpenSearchIntegration"
            }
        }
    },
    {
        "Sid": "CloudWatchLogsApplicationRequestAccess",
        "Effect": "Allow",
        "Action": [
            "es:CreateApplication"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:CalledViaFirst": "logs.amazonaws.com",
                "aws:RequestTag/OpenSearchIntegration": [
                    "Dashboards"
                ]
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "OpenSearchIntegration"
            }
        }
    },
    {
        "Sid": "CloudWatchLogsCollectionResourceAccess",
        "Effect": "Allow",
        "Action": [
            "aoss:DeleteCollection"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:CalledViaFirst": "logs.amazonaws.com",
                "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                    "Dashboards"
                ]
            }
        }
    },
    {
        "Sid": "CloudWatchLogsApplicationResourceAccess",
        "Effect": "Allow",
        "Action": [
            "es:UpdateApplication",
```

```
                "es:GetApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:DeleteSecurityPolicy",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAPIAccessAll",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*"
                }
            }
        },
```

```
        {
            "Sid": "CloudWatchLogsIndexPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:GetAccessPolicy",
                "aoss:CreateLifecyclePolicy",
                "aoss:DeleteLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:index": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsDQSRequestQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsStartDirectQueryAccess",
            "Effect": "Allow",
            "Action": [
                "opensearch:StartDirectQuery",
                "opensearch:GetDirectQuery"
            ],
```

```
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*"
        },
        {
            "Sid": "CloudWatchLogsDQSResourceQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:GetDirectQueryDataSource",
                "es:DeleteDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsPassRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService":
 "directquery.opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAossTagsAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:TagResource"
            ],
            "Resource": "arn:aws:aoss:*:*:collection/*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
```

```
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsApplicationTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
            ],
            "Resource": "arn:aws:opensearch:*:*:application/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ],
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "OpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsDataSourceTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ],
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
```

```
                }
            }
        }
    ]
}
```

Création de l'intégration

Suivez ces étapes pour créer l'intégration.

Pour intégrer les CloudWatch journaux à Amazon OpenSearch Service

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Logs Insights, puis sélectionnez l' OpenSearchonglet Analyser avec.
3. Choisissez Créer une intégration.
4. Dans Nom de l'intégration, saisissez un nom pour l'intégration.
5. (Facultatif) Pour chiffrer les données écrites sur OpenSearch Service Serverless, entrez l'ARN de la AWS KMS clé que vous souhaitez utiliser dans l'ARN de la clé KMS. Pour plus d'informations, consultez Encryption at rest dans le manuel Amazon OpenSearch Service Developer Guide.
6. Pour la conservation des données, entrez la durée pendant laquelle vous souhaitez que les index de données du OpenSearch service soient conservés. Cela définit également la période maximale pendant laquelle vous pouvez afficher les données dans les tableaux de bord. Le choix d'une période de conservation des données plus longue entraînera des coûts de recherche et d'indexation supplémentaires. Pour plus d'informations, consultez la section Tarification des OpenSearch services sans serveur.

   La durée maximale de conservation est de 30 jours.

   La durée de conservation des données sera également utilisée pour créer la politique de cycle de vie de la collecte des OpenSearch services.
7. Pour le rôle IAM pour écrire dans la OpenSearch collection, créez un nouveau rôle IAM ou sélectionnez un rôle IAM existant à utiliser pour écrire dans la OpenSearch collection de services.

La création d'un nouveau rôle est la méthode la plus simple, et le rôle sera créé avec les autorisations nécessaires.

> **Note**
>
> Si vous créez un rôle, il sera autorisé à lire tous les groupes de journaux du compte.

Si vous souhaitez sélectionner un rôle existant, les autorisations doivent être répertoriées dansAutorisations nécessaires à l'intégration. Vous pouvez également choisir Utiliser un rôle existant, puis dans la section Vérifier les autorisations d'accès du rôle sélectionné, vous pouvez choisir Créer un rôle. De cette façon, vous pouvez utiliser les autorisations Autorisations nécessaires à l'intégration répertoriées dans un modèle et les modifier. Par exemple, si vous souhaitez spécifier un contrôle plus fin des groupes de journaux.

8. Pour les rôles IAM et les utilisateurs qui peuvent consulter les tableaux de bord, vous sélectionnez la manière dont vous souhaitez accorder l'accès aux rôles IAM et aux utilisateurs IAM pour l'accès au tableau de bord des journaux vendus :
   - Pour limiter l'accès au tableau de bord à certains utilisateurs, choisissez Sélectionner les rôles IAM et les utilisateurs autorisés à consulter les tableaux de bord, puis recherchez et sélectionnez dans la zone de texte les rôles IAM et les utilisateurs IAM auxquels vous souhaitez accorder l'accès.
   - Pour accorder l'accès au tableau de bord à tous les utilisateurs, choisissez Autoriser tous les rôles et utilisateurs de ce compte à consulter les tableaux de bord.

   > **Important**
   >
   > La sélection de rôles ou d'utilisateurs, ou le choix de tous les utilisateurs, ne fait que les ajouter à la politique d'accès aux données nécessaire pour accéder à la collection de OpenSearch services qui stocke les données du tableau de bord. Pour qu'ils puissent consulter les tableaux de bord des journaux vendus, vous devez également accorder à ces rôles et à ces utilisateurs la politique IAM CloudWatchOpenSearchDashboardAccess gérée.

9. Choisissez Créer une intégration

   La création de l'intégration prendra quelques minutes.

Étape 2 : Création de tableaux de bord de journaux vendus

Après avoir créé l'intégration, vous pouvez créer des tableaux de bord. Des tableaux de bord sont disponibles pour les journaux de flux CloudTrail , les journaux et les journaux Amazon VPC. AWS WAF

Pour créer un tableau de bord de journaux automatiques avec des mesures dérivées par Service OpenSearch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Logs Insights, puis sélectionnez l' OpenSearchonglet Analyser avec.
3. Choisissez Create dashboard (Créer un tableau de bord).
4. Choisissez le type de journaux pour lequel vous souhaitez créer le tableau de bord AWS WAF, les journaux de flux Amazon VPC ou. CloudTrail AWS Network Firewall
5. Entrez un nom pour le tableau de bord, et éventuellement une description.
6. Pour la fréquence de synchronisation des données, entrez la fréquence à laquelle vous souhaitez que le OpenSearch Service interroge CloudWatch afin que les métriques et les index créés dans OpenSearch Service puissent être synchronisés et mis à jour avec les nouvelles données. OpenSearch Le service crée des métriques et des index sur vos journaux pour afficher le tableau de bord.

   Le choix d'une durée plus courte permet de maintenir les données plus à jour et entraîne des coûts plus élevés.
7. Sélectionnez les groupes de journaux à partir desquels collecter les données pour ce tableau de bord. Assurez-vous de sélectionner des groupes de journaux correspondant au type de tableau de bord que vous créez.

   Vous pouvez utiliser le bouton Parcourir les groupes de journaux et l'option Afficher les échantillons de journaux des groupes de journaux sélectionnés lorsque vous effectuez ces choix, afin de vous assurer d'obtenir les groupes de journaux souhaités.
8. Choisissez Create dashboard (Créer un tableau de bord).

   Dans un premier temps, le tableau de bord apparaît sans aucune donnée. Après quelques minutes, les données apparaîtront dans le tableau de bord. Lorsque les données apparaissent pour la première fois, elles concernent les 15 dernières minutes d'entrées dans le journal.

Afficher, modifier ou supprimer les tableaux de bord des journaux vendus

Afficher les tableaux de bord des journaux vendus dans CloudWatch Logs or Service OpenSearch

Pour pouvoir consulter les tableaux de bord, vous devez être connecté à un principal IAM doté de la politique CloudWatchOpenSearchDashboardAccessIAM.

Pour consulter les tableaux de bord des journaux vendus

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Logs Insights, puis sélectionnez l' OpenSearchonglet Analyser avec.
3. Choisissez le tableau de bord dans la zone OpenSearch Tableaux de bord.
4. (Facultatif) Dans le coin supérieur droit, choisissez Afficher dans OpenSearch.

   La console OpenSearch de service s'ouvre et vous y voyez le même tableau de bord. Dans la console de OpenSearch service, vous pouvez apporter des modifications au tableau de bord et à ses widgets. Ces modifications seront également visibles lorsque vous consulterez le tableau de bord dans CloudWatch Logs.

Accorder l'accès à des rôles IAM ou à des utilisateurs IAM supplémentaires pour consulter le tableau de bord

Pour accorder l'accès à des principaux IAM supplémentaires après avoir créé l'intégration, procédez comme suit.

Pour accorder l'accès au tableau de bord du journal automatique à des rôles ou utilisateurs IAM supplémentaires

1. Modifiez la politique d'accès aux données pour la collection afin d'ajouter ces rôles ou utilisateurs. Pour plus d'informations, consultez la section Contrôle d'accès aux données pour Amazon OpenSearch Service Serverless dans le Guide du développeur OpenSearch de services.

2. CloudWatchOpenSearchDashboardAccessAccordez-les à ces utilisateurs. Pour plus d'informations sur le contenu de cette politique, consultezCloudWatchOpenSearchDashboardAccess.

Modifier la configuration du tableau de bord

Vous pouvez modifier le nom, la description et la fréquence de synchronisation des tableaux de bord des journaux de vente existants.

Pour modifier le tableau de bord d'un journal automatique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Logs Insights, puis sélectionnez l' OpenSearchonglet Analyser avec.
3. Choisissez le tableau de bord dans la zone OpenSearch Tableaux de bord.
4. Choisissez Actions, puis Modifier les détails du tableau de bord.
5. Apportez vos modifications, puis choisissez Confirmer les modifications.

Supprimer un tableau de bord de vended log

Vous pouvez supprimer le tableau de bord d'un journal de vente. Dans ce cas, le tableau de bord, les métriques et les index créés dans la collection OpenSearch Service sont tous supprimés.

> **Note**
>
> Après avoir supprimé un tableau de bord contenant un journal automatique, attendez au moins six heures avant d'essayer de recréer ce même tableau de bord. Si vous n'attendez pas, le tableau de bord recréé ne fonctionnera pas correctement.

Pour supprimer un tableau de bord contenant un journal automatique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, choisissez Logs Insights, puis sélectionnez l' OpenSearchonglet Analyser avec.
3. Choisissez le tableau de bord dans la zone OpenSearch Tableaux de bord.

4. Sélectionnez Actions, Supprimer.
5. Confirmez votre décision en saisissant**`delete`**, puis en choisissant Supprimer.

Supprimer toutes les intégrations du tableau de bord des journaux automatiques avec Service OpenSearch

Vous pouvez supprimer l'intégralité de votre OpenSearch intégration. Dans ce cas, tous les tableaux de bord des journaux vendus et les données qui y étaient affichées sont supprimés.

> **Important**
>
> Pour éviter des coûts permanents, nous vous recommandons vivement de supprimer manuellement les ressources suivantes avant de supprimer l'intégration. La suppression de l'intégration ne supprime pas automatiquement ces ressources, et une fois l'intégration supprimée, vous ne pourrez plus accéder à ces ressources pour les supprimer. Pour trouver les noms des ressources à supprimer, reportez-vous à la procédure suivante.
>
> - La source de données
> - La collection
> - La politique d'accès aux données
> - La politique de chiffrement
> - La politique du réseau
> - La politique de cycle de vie

Pour supprimer l'intégralité de votre journal automatique, intégrer le tableau de bord à Service OpenSearch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Paramètres.
3. Sélectionnez l'onglet Logs (Journaux).
4. Dans la section OpenSearch Intégration, choisissez Supprimer l'intégration.

   L'écran suivant affiche les noms des ressources de OpenSearch service que vous devez supprimer avant de supprimer l'intégration.
5. Confirmez votre décision en saisissant**`delete`**, puis en choisissant Supprimer l'intégration.

Politiques IAM pour les utilisateurs

CloudWatch Logs a créé deux politiques IAM, CloudWatchOpenSearchDashboardsFullAccesset CloudWatchOpenSearchDashboardAccess. Le tableau suivant répertorie les actions autorisées par chacune de ces politiques.

Action	Politique IAM	Autorisations supplémentaires nécessaires
Création d'une intégration	CloudWatchOpenSearchDashboardsFullAccess	
Supprimer l'intégration	CloudWatchOpenSearchDashboardsFullAccess	
Créer un tableau de bord	CloudWatchOpenSearchDashboardsFullAccess	
Modifier le tableau de bord	CloudWatchOpenSearchDashboardsFullAccess	
Supprimer le tableau de bord	CloudWatchOpenSearchDashboardsFullAccess	
Actualiser le tableau de bord en utilisant Synchronize now	CloudWatchOpenSearchDashboardsFullAccess	
Afficher l'intégration dans les paramètres	CloudWatchOpenSearchDashboardAccess ou CloudWatchOpenSearchDashboardsFullAccess	
Afficher le tableau de bord	CloudWatchOpenSearchDashboardAccess ou CloudWatchOpenSearchDashboardsFullAccess	Spécifiez le rôle ou l'utilisateur lorsque vous créez l'intégration, ou modifiez la politique d'accès aux données pour la collection afin d'ajouter ces rôles ou utilisateurs. Pour plus d'informations,

Action	Politique IAM	Autorisations supplémentaires nécessaires
		consultez la section Contrôle d'accès aux données pour Amazon OpenSearch Service Serverless dans le Guide du développeur OpenSearch de services.
Afficher le tableau de bord dans la console OpenSearch de service	CloudWatchOpenSearchDashboardAccess ou CloudWatchOpenSearchDashboardsFullAccess	Spécifiez le rôle ou l'utilisateur lorsque vous créez l'intégration, ou modifiez la politique d'accès aux données pour la collection afin d'ajouter ces rôles ou utilisateurs. Pour plus d'informations, consultez la section Contrôle d'accès aux données pour Amazon OpenSearch Service Serverless dans le Guide du développeur OpenSearch de services.

Autorisations nécessaires à l'intégration

Si vous créez un rôle IAM à utiliser par l'intégration, au lieu d'autoriser CloudWatch Logs à créer le rôle, celui-ci doit inclure les autorisations et la politique de confiance suivantes. Pour plus d'informations sur la création d'un rôle IAM, voir Créer un rôle pour déléguer des autorisations à un AWS service.

JSON

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

```
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:StartQuery",
        "logs:GetLogGroupFields",
        "logs:GetQueryResults"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "CloudWatchLogsDescribeLogGroupsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    },
    {
        "Sid": "AmazonOpenSearchCollectionAccess",
        "Effect": "Allow",
        "Action": [
            "aoss:APIAccessAll"
        ],
        "Resource": "*",
        "Condition": {
            "StringLike": {
                "aoss:collection": "cloudwatch-logs-*"
            }
        }
    }
  ]
}
```

> **Note**
>
> Le rôle précédent donne accès à la lecture de tous les groupes de journaux du compte, afin de vous permettre de créer des tableaux de bord pour n'importe quel compte de journal, y compris les groupes de journaux intercomptes. Si vous souhaitez restreindre l'accès à des

groupes de journaux spécifiques et créer des tableaux de bord uniquement pour ces groupes de journaux, vous pouvez mettre à jour la première déclaration de cette politique comme suit :

```
{
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:StartQuery",
        "logs:GetLogGroupFields",
        "logs:GetQueryResults"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:123456789012:log-group:myLogGroup:*",
        "arn:aws:logs:us-east-1:123456789012:log-group:myLogGroup"
      ]
}
```

Création de métriques à partir d'événements du journal à l'aide de filtres

Vous pouvez rechercher et filtrer les données de journal entrant dans CloudWatch Logs en créant un ou plusieurs filtres métriques. Les filtres métriques définissent les termes et les modèles à rechercher dans les données du journal lorsqu'elles sont envoyées à CloudWatch Logs. CloudWatch Logs utilise ces filtres métriques pour transformer les données des journaux en CloudWatch indicateurs numériques que vous pouvez représenter graphiquement ou activer une alarme.

Lorsque vous créez une métrique à partir d'un filtre de journal, vous pouvez également choisir d'attribuer des dimensions et une unité à la métrique. Si vous spécifiez une unité, assurez-vous de spécifier l'unité correcte lorsque vous créez le filtre. La modification ultérieure de l'unité du filtre n'aura aucun effet.

Si un groupe de journaux abonné utilise la transformation des journaux, le modèle de filtre est appliqué aux versions transformées des événements du journal. Pour de plus amples informations, veuillez consulter Transformez les journaux lors de l'ingestion.

> **Note**
>
> Les filtres métriques ne sont pris en charge que pour les groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

Vous pouvez utiliser n'importe quel type de CloudWatch statistique, y compris les statistiques percentiles, lorsque vous consultez ces statistiques ou que vous configurez des alarmes.

> **Note**
>
> Les statistiques sur les centiles sont prises en charge pour une métrique uniquement si aucune des valeurs de la métrique n'est négative. Si vous configurez votre filtre de métrique afin qu'il puisse signaler des valeurs négatives, les statistiques sur les centiles ne seront pas disponibles pour cette métrique lorsqu'elle aura des nombres négatifs en tant que valeurs. Pour plus d'informations, consultez Percentiles.

Les filtres ne traitent pas les données de manière rétroactive. Les filtres publient uniquement les points de données de métriques pour les événements qui se produisent après la création des filtres. Les résultats filtrés renvoient les 50 premières lignes, qui ne seront pas affichées si l'horodatage sur les résultats filtrés est antérieur à l'heure de la création de la métrique.

Table des matières

- Concepts
- Syntaxe des modèles de filtres pour les filtres de métriques
- Création de filtres de métriques
- Liste des filtres de métriques
- Suppression d'un filtre de métrique

Concepts

Chaque filtre de métrique est composé des principaux éléments suivants :

valeur par défaut

La valeur indiquée au filtre de métrique au cours d'une période lorsque les journaux sont ingérés mais qu'aucun journal correspondant n'est trouvé. En définissant cette valeur sur 0, vous vous assurez que les données sont rapportées au cours de chacune de ces périodes, ce qui empêche des métriques irrégulières avec des périodes ne contenant pas de données correspondantes. Si aucun journal n'est ingéré pendant une période d'une minute, aucune valeur n'est rapportée.

Si vous attribuez des dimensions à une métrique créée par un filtre de métrique, vous ne pouvez pas attribuer une valeur par défaut pour cette métrique.

dimensions

Les dimensions sont les paires clé-valeur qui définissent plus précisément une métrique. Vous pouvez attribuer des dimensions à la métrique créée à partir d'un filtre de métrique. Comme les dimensions font partie de l'identifiant unique d'une métrique, chaque fois qu'une paire nom/valeur unique est extraite de vos journaux, vous créez une nouvelle variation de cette métrique.

filter pattern

Description symbolique de la façon dont CloudWatch Logs doit interpréter les données de chaque événement du journal. Par exemple, une entrée de journal peut contenir des horodatages, des

adresses IP, des chaînes et ainsi de suite. Le modèle permet de spécifier ce qu'il faut chercher dans le fichier journal.

metric name

Nom de la CloudWatch métrique dans laquelle les informations du journal surveillé doivent être publiées. Par exemple, vous pouvez publier selon une métrique appelée ErrorCount.

metric namespace

L'espace de noms de destination de la nouvelle CloudWatch métrique.

metric value

La valeur numérique à publier vers la métrique chaque fois qu'un journal correspondant est trouvé. Par exemple, si vous comptez les occurrences d'un terme particulier comme « Erreur », la valeur sera « 1 » pour chaque occurrence. Si vous calculez le nombre d'octets transférés, vous pouvez l'incrémenter par le nombre d'octets trouvés dans l'événement de journal.

Syntaxe des modèles de filtres pour les filtres de métriques

> **Note**
>
> En quoi les filtres métriques diffèrent-ils des requêtes CloudWatch Logs Insights
> Les filtres métriques diffèrent des requêtes CloudWatch Logs Insights en ce sens qu'une valeur numérique spécifiée est ajoutée à un filtre métrique chaque fois qu'un journal correspondant est trouvé. Pour de plus amples informations, veuillez consulter Configuration des valeurs de métriques pour un filtre de métriques.
> Pour plus d'informations sur la façon d'interroger vos groupes de CloudWatch journaux avec le langage de requête Amazon Logs Insights, consultezCloudWatch Syntaxe des requêtes en langage Logs Insights.
> Exemples de modèles de filtres génériques
> Pour plus d'informations sur la syntaxe des modèles de filtres génériques applicable aux filtres de métriques ainsi qu'aux filtres d'abonnements et aux filtres des événements du journal, consultez la section Syntaxe des modèles de filtres pour les filtres de métriques, les filtres d'abonnements et les filtres des événements du journal, qui inclut les exemples suivants :
>
> - Syntaxe des expressions régulières (regex) prise en charge
> - Correspondance des termes dans les événements du journal non structurés

- Établissement de correspondances dans les événements du journal JSON
- Correspondance des termes dans les événements du journal délimités par des espaces

Les filtres métriques vous permettent de rechercher et de filtrer les données de journal entrant dans CloudWatch Logs, d'extraire des observations métriques à partir des données de journal filtrées et de transformer les points de données en métriques CloudWatch Logs. Vous définissez les termes et les modèles à rechercher dans les données du journal lorsqu'elles sont envoyées à CloudWatch Logs. Les filtres de métriques sont associés aux groupes de journaux, et tous les filtres affectés à un groupe sont appliqués à ses flux de journaux.

Lorsqu'un filtre de métriques correspond à un terme, il incrémente le décompte de la métrique d'une valeur numérique spécifiée. Par exemple, vous pouvez créer un filtre de métriques qui compte le nombre d'apparitions du mot ERROR dans vos événements du journal.

Vous pouvez attribuer des unités de mesure et des dimensions aux métriques. Par exemple, si vous créez un filtre de métriques qui compte le nombre d'apparitions du mot ERROR (ERREUR) dans vos événements du journal, vous pouvez spécifier une dimension appelée `ErrorCode` pour afficher le nombre total d'événements du journal contenant le mot ERROR (ERREUR) et filtrer les données par codes d'erreur signalés.

> **Tip**
>
> Lorsque vous attribuez une unité de mesure à une métrique, assurez-vous de spécifier la bonne unité. Si vous modifiez l'unité plus tard, il se peut que votre modification ne prenne pas effet. Pour obtenir la liste complète des unités prises CloudWatch en charge, consultez MetricDatumle Amazon CloudWatch API Reference.

Rubriques

- Configuration des valeurs de métriques pour un filtre de métriques
- Publication de dimensions avec des métriques à partir de valeurs dans JSON ou d'événements du journal délimités par des espaces
- Utilisation de valeurs dans des événements du journal pour incrémenter la valeur d'une métrique

Configuration des valeurs de métriques pour un filtre de métriques

Lorsque vous créez un filtre de métriques, vous définissez votre modèle de filtre et spécifiez la valeur de vos métriques, ainsi que la valeur par défaut. Vous pouvez définir des valeurs de métriques pour des nombres, des identifiants nommés ou des identifiants numériques. Si vous ne spécifiez pas de valeur par défaut, les données CloudWatch ne seront pas communiquées lorsque votre filtre métrique ne trouve aucune correspondance. Nous vous recommandons de spécifier une valeur par défaut, même si la valeur est 0. La définition d'une valeur par défaut permet de CloudWatch rapporter les données avec plus de précision et CloudWatch d'éviter d'agréger des indicateurs irréguliers. CloudWatch agrège et rapporte les valeurs métriques toutes les minutes.

Lorsque votre filtre de métriques trouve une correspondance dans vos événements du journal, il incrémente le décompte de vos métriques en fonction de leurs valeurs. Si votre filtre de mesure ne trouve aucune correspondance, CloudWatch indique la valeur par défaut de la métrique. Par exemple, votre groupe de journaux publie deux registres toutes les minutes, avec une valeur de métrique égale à 1 et une valeur par défaut égale à 0. Si votre filtre de métriques trouve des correspondances dans les deux registres de journaux au cours de la première minute, la valeur de métrique pour cette minute est égale à 2. Si votre filtre de métriques ne trouve pas de correspondance dans les registres au cours de la seconde minute, la valeur par défaut pour cette minute est égale à 0. Si vous attribuez des dimensions à des métriques générées par des filtres de métriques, vous ne pouvez pas spécifier des valeurs par défaut pour ces métriques.

Vous pouvez également définir un filtre de métriques pour incrémenter une métrique dont la valeur est extraite d'un événement du journal, au lieu d'une valeur statique. Pour de plus amples informations, veuillez consulter Utilisation de valeurs dans des événements du journal pour incrémenter la valeur d'une métrique.

Publication de dimensions avec des métriques à partir de valeurs dans JSON ou d'événements du journal délimités par des espaces

Vous pouvez utiliser la CloudWatch console ou la AWS CLI pour créer des filtres métriques qui publient des dimensions avec des métriques générées par le JSON et des événements de journal délimités par des espaces. Les dimensions sont des paires de valeur nom/valeur et sont uniquement disponibles pour les modèles de filtres JSON et ceux délimités par des espaces. Vous pouvez créer des filtres de métriques JSON et délimités par des espaces avec trois dimensions maximum. Pour plus d'informations sur les dimensions et comment attribuer des dimensions à des métriques, consultez les sections suivantes :

- Dimensions indiquées dans le guide de CloudWatch l'utilisateur Amazon
- Exemple : extraire des champs d'un journal Apache et attribuer des dimensions dans le guide de l'utilisateur Amazon CloudWatch Logs

> **Important**
>
> Les dimensions contiennent des valeurs qui collectent des frais tout comme les métriques personnalisées. Pour éviter des charges inattendues, ne spécifiez pas de champs à cardinalité élevée tels que `IPAddress` ou `requestID`, comme dimensions.
> Si vous extrayez les métriques des événements du journal, vous êtes facturé pour les métriques personnalisées. Pour éviter que vous ne soyez accidentellement facturé des frais élevés, Amazon peut désactiver un filtre de métriques si celui-ci génère 1000 paires nom/valeur différentes pour des dimensions spécifiées dans un certain espace de temps.
> Vous pouvez créer des alarmes de facturation qui vous informent de vos frais estimés. Pour plus d'informations, consultez la section Création d'une alarme de facturation pour surveiller vos AWS frais estimés.

Publication de dimensions avec des métriques à partir des événements du journal JSON

Les exemples suivants contiennent des extraits de code qui décrivent comment spécifier des dimensions dans un filtre de métriques JSON.

Example: JSON log event

```
{
  "eventType": "UpdateTrail",
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {"name": "a",
         "id": 1
       },
       {"name": "b",
         "id": 2
       }
```

```
  ]

}
```

> **Note**
>
> Si vous testez les exemples de filtres de métriques avec l'exemple d'événement du journal JSON, vous devez saisir l'exemple de journal JSON sur une seule ligne.

Example: Metric filter

Le filtre de métriques incrémente la métrique chaque fois qu'un événement du journal JSON contient les propriétés `eventType` et `"sourceIPAddress"`.

```
{ $.eventType = "*" && $.sourceIPAddress != 123.123.* }
```

Lorsque vous créez un filtre de métriques JSON, vous pouvez spécifier n'importe laquelle des propriétés du filtre de métriques comme dimension. Par exemple, pour définir `eventType` comme dimension, utilisez les éléments suivants :

```
"eventType" : $.eventType
```

L'exemple de métrique contient une dimension nommée `"eventType"` et la valeur de la dimension dans l'exemple d'événement du journal est `"UpdateTrail"`.

Publication de dimensions avec des métriques à partir d'événements du journal délimités par des espaces

Les exemples suivants contiennent des extraits de code qui décrivent comment spécifier des dimensions dans un filtre de métriques délimité par des espaces.

Example: Space-delimited log event

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

Example: Metric filter

```
[ip, server, username, timestamp, request, status_code, bytes > 1000]
```

Le filtre de métriques incrémente la métrique lorsqu'un événement du journal délimité par des espaces inclut l'un des champs spécifiés dans le filtre. Par exemple, le filtre de métriques recherche les champs et les valeurs suivants dans l'exemple d'événement du journal délimité par des espaces.

```
{
   "$bytes": "1534",
   "$status_code": "404",

   "$request": "GET /index.html HTTP/1.0",
   "$timestamp": "10/Oct/2000:13:25:15 -0700",
   "$username": "frank",
   "$server": "Prod",
   "$ip": "127.0.0.1"
}
```

Lorsque vous créez un filtre de métriques délimité par des espaces, vous pouvez spécifier n'importe lequel des champs du filtre de métriques comme dimension. Par exemple, pour définir `server` comme dimension, utilisez les éléments suivants :

```
"server" : $server
```

L'exemple de filtre de métriques contient une dimension nommée `server` et la valeur de la dimension dans l'exemple d'événement du journal est `"Prod"`.

Example: Match terms with AND (&&) and OR (||)

Vous pouvez utiliser les opérateurs logiques AND (« && ») et OR (« || ») pour créer des filtres de métriques délimités par des espaces contenant des conditions. Le filtre de métriques suivant renvoie les événements du journal lorsque le premier mot des événements est ERROR ou une superchaîne de WARN.

```
[w1=ERROR || w1=%WARN%, w2]
```

Utilisation de valeurs dans des événements du journal pour incrémenter la valeur d'une métrique

Vous pouvez créer des filtres de métriques qui publient des valeurs numériques identifiées dans vos événements du journal. La procédure de cette section utilise l'exemple de filtre de métriques suivant pour montrer comment publier une valeur numérique d'un événement du journal JSON dans une métrique.

```
{ $.latency = * } metricValue: $.latency
```

Création d'un filtre de métriques qui publie une valeur dans un événement du journal

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), puis Log groups (Groupes de journaux).
3. Sélectionnez ou créez un groupe de journaux.

   Pour plus d'informations sur la création d'un groupe de journaux, consultez la section Créer un groupe de CloudWatch journaux dans Logs du guide de l'utilisateur Amazon CloudWatch Logs.
4. Choisissez Actions, puis Create metric filter (Créer un filtre de métrique).
5. Pour Filter Pattern (Modèle de filtre), saisissez **`{ $.latency = * }`**, puis choisissez Next (Suivant).
6. Pour Metric Name (Nom de la métrique), saisissez myMetric.
7. Pour Valeur de la métrique, saisissez **`$.latency`**.
8. (Facultatif) Pour Default Value (Valeur par défaut), saisissez 0, puis choisissez Next (Suivant).

Nous vous recommandons de spécifier une valeur par défaut, même si la valeur est 0. La définition d'une valeur par défaut permet de CloudWatch rapporter les données avec plus de précision et CloudWatch d'éviter d'agréger des indicateurs irréguliers. CloudWatch agrège et rapporte les valeurs métriques toutes les minutes.

9. Choisissez Créer un filtre de métriques.

L'exemple de filtre de métriques fait correspondre le terme `"latency"` dans l'exemple d'événement du journal JSON et publie une valeur numérique de 50 pour la métrique myMetric.

```
{
"latency": 50,
"requestType": "GET"
}
```

Création de filtres de métriques

La procédure et les exemples suivants montrent comment créer des filtres de métriques.

Exemples

- Créer un filtre de métrique pour un groupe de journaux
- Exemple : Comptage des événements du journal
- Exemple : Comptage des occurrences d'un terme
- Exemple : Comptage du nombre de codes HTTP 404
- Exemple : comptage de codes HTTP 4xx
- Exemple : Extraction des champs d'un journal Apache et attribution de dimensions

Créer un filtre de métrique pour un groupe de journaux

Pour créer un filtre de métrique pour un groupe de journaux, procédez comme suit. La métrique ne sera pas visible tant qu'il n'y aura pas de points de données pour elle.

Pour créer un filtre métrique à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), puis Log groups (Groupes de journaux).
3. Choisissez le nom du groupe de journaux.
4. Choisissez `Actions`, puis Create metric filter (Créer un filtre de métrique).
5. Pour Filter pattern, saisissez le modèle de filtre. Pour de plus amples informations, veuillez consulter Syntaxe des modèles de filtres pour les filtres de métriques, les filtres d'abonnements, les filtres d'événements du journal et Live Tail.
6. (Facultatif) Pour tester votre modèle de filtre, sous Test Pattern (Tester le modèle), saisissez un ou plusieurs événements du journal à utiliser à cet effet. Chaque événement du journal doit être mis en forme sur une seule ligne. Utilisez les sauts de ligne pour séparer les événements du journal dans la boîte Messages d'événements du journal.
7. Sélectionnez Next (Suivant), puis saisissez un nom pour le filtre de métrique.
8. Sous Détails de la métrique, pour l'espace de noms métrique, entrez le nom de l'espace de CloudWatch noms dans lequel la métrique sera publiée. Si l'espace réservé au nom n'existe pas encore, assurez-vous que l'option Create new (Créer un nouveau) est sélectionnée.
9. Pour Metric name (Nom de la métrique), saisissez un nom pour la nouvelle métrique.
10. Pour Metric value (valeur de la métrique), si votre filtre de métrique compte les occurrences des mots-clés dans le filtre, saisissez 1. Cela incrémente la métrique de 1 pour chaque événement de journal qui inclut l'un des mots-clés.

    Vous pouvez également saisir un jeton tel que **`$size`**. Cela incrémente la métrique de la valeur du nombre dans le champ `size` pour chaque événement de journal qui contient un champ `size`.
11. (Facultatif) Pour Unit (Unité), sélectionnez une unité à affecter à la métrique. Si vous ne spécifiez pas d'unité, elle est définie comme `None`.
12. (Facultatif) Saisissez les noms et les jetons pour trois dimensions maximum pour la métrique. Si vous attribuez des dimensions à des métriques créées par des filtres de métriques, vous ne pouvez pas affecter des valeurs par défaut pour ces métriques.

    > **Note**
    >
    > Les dimensions sont prises en charge uniquement dans JSON ou dans les filtres de métriques délimités par des espaces.
13. Choisissez Créer un filtre de métriques. Vous pouvez trouver le filtre de métrique que vous avez créé à partir du panneau de navigation. Choisissez Journaux, puis groupe de journaux.

Choisissez le nom du groupe de journaux pour lequel vous avez créé votre filtre de métrique, puis sélectionnez l'onglet Filtres de métriques.

Exemple : Comptage des événements du journal

Le type de surveillance des événements du journal le plus simple consiste à compter le nombre d'événements du journal qui se produisent. Vous pouvez souhaiter procéder ainsi pour garder un décompte de tous les événements, pour créer une surveillance de style « pulsation » ou tout simplement pour vous entraîner à créer des filtres de métriques.

Dans l'exemple de CLI suivant, un filtre métrique appelé MyAppAccessCount est appliqué au groupe de journaux MyApp /access.log pour créer la métrique EventCount dans l'espace de CloudWatch noms MyNamespace. Le filtre est configuré de manière à établir une correspondance avec n'importe quel contenu d'événement de journal et à augmenter la métrique de « 1 ».

Pour créer un filtre métrique à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Choisissez le nom d'un groupe de journaux.
4. Choisissez `Actions`, Créer un filtre de métriques.
5. Laissez les champs Modèle de filtre et Sélectionner les données de journal à tester vides.
6. Choisissez Suivant, puis, pour Nom du filtre, tapez **`EventCount`**.
7. Sous Metric Details, pour Metric Namespace, tapez **`MyNameSpace`**.
8. Dans Metric Name (Nom de la métrique), saisissez **`MyAppEventCount`**.
9. Confirmez que la Valeur de la métrique est égale à 1. Cela indique que le décompte est augmenté d'1 pour chaque événement de journal.
10. Dans Valeur par défaut saisissez 0, puis choisissez Suivant. Le fait de spécifier une valeur par défaut garantit que les données sont signalées même pendant les périodes où aucun événement de journal ne se produit, ce qui empêche les métriques irrégulières en cas d'absence de données.
11. Choisissez Créer un filtre de métriques.

Pour créer un filtre métrique à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
  --filter-pattern " " \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Vous pouvez tester cette nouvelle stratégie en publiant n'importe quelles données d'événements. Vous devriez voir les points de données publiés sur la métrique MyAppAccessEventCount.

Pour publier les données d'un événement à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Exemple : Comptage des occurrences d'un terme

Les événements du journal comprennent fréquemment des messages importants que vous pouvez souhaiter comptabiliser. Il peut s'agir de messages sur l'échec ou la réussite des opérations. Par exemple, une erreur peut se produire et être enregistrée dans un fichier journal en cas d'échec d'une opération spécifique. Il est possible de surveiller ces entrées pour comprendre l'évolution de vos erreurs.

Dans l'exemple ci-dessous, un filtre de métrique est créé pour surveiller le terme Error. La politique a été créée et ajoutée au groupe de journaux MyApp/message.log. CloudWatch Logs publie un point de données vers la métrique CloudWatch personnalisée ErrorCount dans l'espace de noms MyApp/message.log avec une valeur de « 1 » pour chaque événement contenant une erreur. Si aucun événement ne contient le mot Error, la valeur 0 est publiée. Lors de la représentation de ces données dans la console CloudWatch, veillez à utiliser la statistique de somme.

Après avoir créé un filtre de mesure, vous pouvez afficher la métrique dans la CloudWatch console. Lorsque vous sélectionnez la métrique à afficher, sélectionnez l'espace de noms de métrique

correspondant au nom du groupe de journaux. Pour de plus amples informations, consultez Affichage des métriques disponibles.

Pour créer un filtre métrique à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Choisissez le nom du groupe de journaux.
4. Choisissez Actions, Créer un filtre de métriques.
5. Pour Modèle de filtre, saisissez **`Error`**.

   > **Note**
   > Toutes les entrées figurant dans Filter Pattern sont sensibles à la casse.

6. (Facultatif) Pour tester votre modèle de filtre, sous Test Pattern (Modèle de test), saisissez un ou plusieurs événements du journal à utiliser pour tester le modèle. Chaque événement de journal doit se trouver sur une seule ligne, car des sauts de ligne sont utilisés pour séparer les événements du journal dans la boîte Messages d'événements du journal.
7. Choisissez Suivant, puis, sur la page Affecter une métrique, pour Nom du filtre, tapez **`MyAppErrorCount`**.
8. Sous Metric Details, pour Metric Namespace, tapez MyNameSpace.
9. Dans Metric Name (Nom de la métrique), saisissez ErrorCount.
10. Confirmez que la Valeur de la métrique est égale à 1. Cela indique que le décompte est augmenté d'1 pour chaque événement de journal contenant « Error ».
11. Pour Valeur par défaut tapez 0, puis choisissez Suivant.
12. Choisissez Créer un filtre de métriques.

Pour créer un filtre métrique à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
```

```
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Vous pouvez tester cette nouvelle stratégie en publiant les événements contenant le mot « Error » dans le message.

Pour publier des événements à l'aide du AWS CLI

À partir d'une invite de commande, exécutez la commande suivante. Notez que les modèles sont sensible à la casse.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Exemple : Comptage du nombre de codes HTTP 404

À l'aide CloudWatch des journaux, vous pouvez surveiller le nombre de fois que vos serveurs Apache renvoient une réponse HTTP 404, qui est le code de réponse pour une page introuvable. Vous pouvez surveiller ce phénomène afin d'appréhender le nombre de fois où les visiteurs de votre site ne trouvent pas la ressource qu'ils recherchent. Supposons que vos enregistrements de journaux sont structurés de manière à inclure les informations suivantes pour chaque événement de journal (visite du site) :

- Adresse IP du demandeur
- Identité RFC 1413
- Nom d'utilisateur
- Horodatage
- Méthode de la demande avec la ressource demandée et le protocole
- Code de réponse HTTP à la demande
- Nombre d'octets transférés dans la demande

Voici un exemple correspondant :

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

Vous pouvez spécifier une règle qui tente de relever les événements de cette structure pour les erreurs HTTP 404, comme illustré dans l'exemple suivant :

Pour créer un filtre métrique à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Choisissez `Actions`, Créer un filtre de métriques.
4. Pour Modèle de filtre, tapez **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**.
5. (Facultatif) Pour tester votre modèle de filtre, sous Test Pattern (Modèle de test), saisissez un ou plusieurs événements du journal à utiliser pour tester le modèle. Chaque événement de journal doit se trouver sur une seule ligne, car des sauts de ligne sont utilisés pour séparer les événements du journal dans la boîte Messages d'événements du journal.
6. Choisissez Suivant, puis pour Nom du filtre, tapez HTTP404Errors.
7. Sous Détails de la métrique, pour Espace de nom de la métrique, saisissez **`MyNameSpace`**.
8. Pour Nom de la métrique, saisissez **`ApacheNotFoundErrorCount`**.
9. Confirmez que la Valeur de la métrique est égale à 1. Cela indique que le décompte est augmenté d'1 pour chaque événement « 404 Error ».
10. Dans Valeur par défaut saisissez 0, puis choisissez Suivant.
11. Choisissez Créer un filtre de métriques.

Pour créer un filtre métrique à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
      metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

Dans cet exemple, les caractères littéraux, tels que les crochets gauche et droit, les doubles guillemets et la chaîne de caractères 404 ont été utilisés. Le modèle doit correspondre au message

tout entier de l'événement du journal pour que cet événement soit pris en compte à des fins de surveillance.

Vous pouvez vérifier la création du filtre de métrique à l'aide de la commande describe-metric-filters. Vous devriez obtenir un résultat du type suivant :

```
aws logs describe-metric-filters --log-group-name MyApp/access.log

{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

Désormais, vous pouvez publier quelques événements manuellement :

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

Peu après avoir enregistré ces exemples d'événements de journal, vous pouvez récupérer la métrique nommée dans la CloudWatch console sous le nom de ApacheNotFoundErrorCount.

Exemple : comptage de codes HTTP 4xx

Comme dans l'exemple précédent, il se peut que vous souhaitiez surveiller vos journaux d'accès aux services Web ainsi que les niveaux de code de réponse HTTP. Par exemple, vous pouvez souhaiter surveiller toutes les erreurs HTTP de niveau 400. Cependant, vous ne voudrez pas forcément spécifier un nouveau filtre de métrique pour chaque code de retour.

Vous trouverez ci-dessous comment créer une métrique qui inclut toutes les réponses de code HTTP de niveau 400 à partir d'un journal d'accès en utilisant le format de journal d'accès Apache de l'exemple Exemple : Comptage du nombre de codes HTTP 404.

Pour créer un filtre métrique à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Choisissez le nom du groupe de journaux du serveur Apache.
4. Choisissez `Actions`, Créer un filtre de métriques.
5. Pour Modèle de filtre, saisissez **`[ip, id, user, timestamp, request, status_code=4*, size]`**.
6. (Facultatif) Pour tester votre modèle de filtre, sous Test Pattern (Modèle de test), saisissez un ou plusieurs événements du journal à utiliser pour tester le modèle. Chaque événement de journal doit se trouver sur une seule ligne, car des sauts de ligne sont utilisés pour séparer les événements du journal dans la boîte Messages d'événements du journal.
7. Choisissez Next (Suivant), puis, pour Filter name (Nom du filtre), tapez **`HTTP4xxErrors`**.
8. Sous Détails de la métrique, pour Espace de nom de la métrique, saisissez **`MyNameSpace`**.
9. Dans Nom de la métrique, entrez HTTP4xxErrors.
10. Pour Valeur de la métrique, saisissez 1. Cela indique que le décompte est augmenté d'1 pour chaque événement de journal contenant « 4xx error ».
11. Pour Default value (Valeur par défaut) saisissez 0, puis choisissez Next (Suivant).
12. Choisissez Créer un filtre de métriques.

Pour créer un filtre métrique à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs put-metric-filter \
```

```
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
  metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Vous pouvez utiliser les données suivantes dans les appels PutEvents pour tester cette règle. Si vous n'a pas supprimé la règle de surveillance de l'exemple précédent, vous allez générer deux métriques différentes.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Exemple : Extraction des champs d'un journal Apache et attribution de dimensions

Parfois, au lieu de compter, il peut s'avérer utile d'utiliser des valeurs d'événements du journal individuels pour les valeurs de métriques. Cet exemple montre comment vous pouvez créer une règle d'extraction afin de créer une métrique qui mesure le nombre d'octets transférés par un serveur Apache.

Cet exemple montre également comment affecter des dimensions à la métrique que vous créez.

Pour créer un filtre métrique à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Choisissez le nom du groupe de journaux du serveur Apache.
4. Choisissez `Actions`, Créer un filtre de métriques.
5. Pour Modèle de filtre, saisissez **`[ip, id, user, timestamp, request, status_code, size]`**.
6. (Facultatif) Pour tester votre modèle de filtre, sous Test Pattern (Modèle de test), saisissez un ou plusieurs événements du journal à utiliser pour tester le modèle. Chaque événement de

journal doit se trouver sur une seule ligne, car des sauts de ligne sont utilisés pour séparer les événements du journal dans la boîte Messages d'événements du journal.

7. Choisissez Next (Suivant), puis, pour Filter name (Nom du filtre), tapez **`size`**.
8. Sous Metric details (Détails de la métrique), pour Metric namespace (Espace de nom de la métrique), saisissez **`MyNameSpace`**. Comme il s'agit d'un nouvel espace de nom, assurez-vous que l'option Create new (Créer un nouveau) est sélectionnée.
9. Pour Nom de la métrique, saisissez **`BytesTransferred`**
10. Pour Valeur de la métrique, saisissez **`$size`**.
11. Pour Unit (Unité), sélectionnez Bytes (Octets).
12. Pour Dimension Name, tapez **`IP`**.
13. Pour Dimension Value (Valeur de la dimension), tapez **`$ip`** et ensuite choisissez Next (Suivant).
14. Choisissez Créer un filtre de métriques.

Pour créer ce filtre métrique à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
 --filter-name BytesTransferred \
 --filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
 --metric-transformations  \
 metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size'
```

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations  \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size',unit=Bytes,dimensio
$ip}}'
```

> **Note**
>
> Dans cette commande, utilisez ce format pour spécifier plusieurs dimensions.
>
> ```
> aws logs put-metric-filter \
> ```

```
--log-group-name my-log-group-name \
--filter-name my-filter-name \
--filter-pattern 'my-filter-pattern' \
--metric-transformations  \
metricName=my-metric-name,metricNamespace=my-metric-namespace,metricValue=my-
token,unit=unit,dimensions='{dimension1=$dim,dimension2=$dim2,dim3=$dim3}'
```

Vous pouvez utiliser les données suivantes dans les put-log-event appels pour tester cette règle. Si vous n'avez pas supprimé la règle de surveillance de l'exemple précédent, vous allez générer deux métriques différentes.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Liste des filtres de métriques

Vous pouvez lister tous les filtres de métriques d'un groupe de journaux.

Pour répertorier les filtres métriques à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Dans le volet de contenu, dans la liste des groupes de journaux, et plus précisément dans la colonne Metric Filters, choisissez le nombre de filtres.

   La fenêtre Log Groups > Filters for répertorie tous les filtres de métriques associés au groupe de journaux.

Pour répertorier les filtres métriques à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

Voici un exemple de sortie :

```
{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

Suppression d'un filtre de métrique

Une stratégie est identifiée par son nom et le groupe de journaux auquel elle appartient.

Pour supprimer un filtre métrique à l'aide de la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Dans le volet de contenu, dans la colonne Metric Filter (Filtre de métrique), choisissez le nombre de filtres de métriques pour le groupe de journaux.
4. Sous l'écran Metric Filters (Filtres de métriques), cochez la case à droite du nom du filtre que vous voulez supprimer. Ensuite, choisissez Supprimer.
5. Lorsque vous êtes invité à confirmer l'opération, choisissez Supprimer.

Pour supprimer un filtre métrique à l'aide du AWS CLI

A partir d'une invite de commande, exécutez la commande suivante :

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
```

```
--filter-name MyFilterName
```

Traitement en temps réel des données du journal avec les abonnements

Vous pouvez utiliser des abonnements pour accéder à un flux en temps réel des événements du journal depuis CloudWatch Logs et le transmettre à d'autres services tels qu'un flux Amazon Kinesis, un flux Amazon Data Firehose, ou AWS Lambda pour un traitement, une analyse ou un chargement personnalisés sur d'autres systèmes. Lorsque les événements du journal sont envoyés au service récepteur, ils sont codés en base64 et compressés au format gzip.

Pour commencer à s'abonner aux événements du journal, créez la ressource de réception, telle qu'un flux Kinesis Data Streams, où les événements seront transférés. Un filtre d'abonnement définit le modèle de filtre à utiliser pour filtrer les événements du journal transmis à votre AWS ressource, ainsi que les informations indiquant à qui envoyer les événements de journal correspondants. Les événements du journal sont envoyés à la ressource réceptrice peu après avoir été ingérés, généralement en moins de trois minutes.

> **Note**
>
> Si un groupe de journaux abonné utilise la transformation des journaux, le modèle de filtre est comparé aux versions transformées des événements du journal. Pour de plus amples informations, veuillez consulter Transformez les journaux lors de l'ingestion.

Vous pouvez créer des abonnements au niveau du compte et au niveau du groupe de log. Chaque compte peut avoir un filtre d'abonnement au niveau du compte par région. Chaque groupe de journaux peut avoir jusqu'à deux filtres d'abonnement associés.

> **Note**
>
> Si le service de destination renvoie une erreur réessayable, telle qu'une exception de limitation ou une exception de service réessayable (HTTP 5xx par exemple), CloudWatch Logs continue de réessayer la livraison pendant 24 heures au maximum. CloudWatch Logs n'essaie pas de le renvoyer s'il s'agit d'une erreur non réessayable, telle que ou. AccessDeniedException ResourceNotFoundException Dans ces cas, le filtre d'abonnement est désactivé pendant 10 minutes au maximum, puis CloudWatch Logs tente à nouveau

d'envoyer les journaux à la destination. Pendant cette période de désactivation, les journaux sont ignorés.

CloudWatch Logs produit également CloudWatch des métriques concernant le transfert des événements du journal aux abonnements. Pour de plus amples informations, veuillez consulter Surveillance à l'aide de CloudWatch métriques.

Vous pouvez également utiliser un abonnement CloudWatch Logs pour diffuser les données des journaux en temps quasi réel vers un cluster Amazon OpenSearch Service. Pour plus d'informations, consultez la section Streaming CloudWatch Logs data to Amazon OpenSearch Service.

Les abonnements ne sont pris en charge que pour les groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

> **Note**
>
> Les filtres d'abonnement peuvent enregistrer les événements par lots afin d'optimiser la transmission et de réduire le nombre d'appels passés vers la destination. Le dosage n'est pas garanti mais est utilisé dans la mesure du possible.

Table des matières

- Concepts
- Filtres d'abonnement au niveau des groupes de journaux
- Filtres d'abonnement au niveau du compte
- Abonnements entre comptes et entre régions
- Prévention du député confus
- Prévention de la récursivité dans les journaux

Concepts

Chaque filtre d'abonnement est composé des principaux éléments suivants :

filter pattern

Une description symbolique de la façon dont les CloudWatch journaux doivent interpréter les données de chaque événement du journal, ainsi que des expressions de filtrage qui limitent ce qui est livré à la AWS ressource de destination. Pour plus d'informations sur la syntaxe des modèles de filtrage, consultez Syntaxe des modèles de filtres pour les filtres de métriques, les filtres d'abonnements, les filtres d'événements du journal et Live Tail.

destination arn

L'Amazon Resource Name (ARN) du flux Kinesis Data Streams, du flux Firehose ou de la fonction Lambda que vous souhaitez utiliser comme destination du flux d'abonnement.

ARN de rôle

Rôle IAM qui accorde à CloudWatch Logs les autorisations nécessaires pour placer des données dans la destination choisie. Ce rôle n'est pas nécessaire pour les destinations Lambda car CloudWatch Logs peut obtenir les autorisations nécessaires à partir des paramètres de contrôle d'accès de la fonction Lambda elle-même.

distribution

La méthode utilisée pour distribuer les données du journal à la destination, lorsque la destination est un flux dans Amazon Kinesis Data Streams. Par défaut, les données des journaux sont regroupées par flux de journaux. Pour assurer une meilleure répartition de la distribution, vous pouvez regrouper les données des journaux de manière aléatoire.

Pour les abonnements au niveau des groupes de journaux, l'élément clé suivant est également inclus :

log group name

Le groupe de journaux auquel associer le filtre d'abonnement. Tous les événements du journal téléchargés vers ce groupe de journaux sont soumis au filtre d'abonnement. Ceux qui correspondent au filtre sont envoyés au service de destination qui reçoit les événements du journal correspondants.

Pour les abonnements au niveau du compte, l'élément clé suivant est également inclus :

critères de sélection

Critères utilisés pour sélectionner les groupes de journaux auxquels le filtre d'abonnement au niveau du compte est appliqué. Si vous ne le spécifiez pas, le filtre d'abonnement au niveau du compte est appliqué à tous les groupes de journaux du compte. Ce champ est utilisé pour empêcher les boucles de log infinies. Pour plus d'informations sur le problème des boucles de log infinies, consultezPrévention de la récursivité dans les journaux.

Les critères de sélection ont une limite de taille de 25 Ko.

Filtres d'abonnement au niveau des groupes de journaux

Vous pouvez utiliser un filtre d'abonnement avec Amazon Kinesis Data Streams AWS Lambda, Amazon Data Firehose ou Amazon Service. OpenSearch Les journaux envoyés à un service via un filtre d'abonnement sont codés en base64 et compressés au format gzip. Cette section fournit des exemples que vous pouvez suivre pour créer un filtre d'abonnement aux CloudWatch journaux qui envoie les données des journaux à Firehose, Lambda et Kinesis Data Streams.

> **Note**
>
> Si vous souhaitez effectuer une recherche dans les données de votre journal, consultez la section Syntaxe des filtres et des modèles.

Exemples

- Exemple 1 : filtres d'abonnement avec Kinesis Data Streams
- Exemple 2 : filtres d'abonnement avec AWS Lambda
- Exemple 3 : filtres d'abonnement avec Amazon Data Firehose
- Exemple 4 : filtres d'abonnement avec Amazon OpenSearch Service

Exemple 1 : filtres d'abonnement avec Kinesis Data Streams

L'exemple suivant associe un filtre d'abonnement à un groupe de journaux contenant des AWS CloudTrail événements. Le filtre d'abonnement transmet toutes les activités enregistrées à l'aide des AWS informations d'identification « Root » à un flux appelé « RootAccess » dans Kinesis Data

Streams. Pour plus d'informations sur la façon d'envoyer AWS CloudTrail des événements aux CloudWatch journaux, consultez la section Envoyer CloudTrail des événements aux CloudWatch journaux dans le guide de AWS CloudTrail l'utilisateur.

> **Note**
>
> Avant de créer le flux , calculez le volume de données de journaux qui sera généré. Assurez-vous de créer un flux avec suffisamment de partitions pour gérer le volume. Si le flux n'a pas suffisamment de partitions, le flux de journaux sera limité. Pour plus d'informations sur les limites de volume de flux, consultez Quotas et limites (français non garanti).
> Les livrables limités sont réessayés pendant 24 heures au maximum. Au bout de 24 heures, les livrables ayant échoué sont supprimés.
> Pour réduire le risque de limitation, procédez comme suit :
>
> - Spécifiez `random` le `distribution` moment où vous créez le filtre d'abonnement avec PutSubscriptionFilterou put-subscription-filter. Par défaut, la distribution du filtre de flux se fait par flux de log, ce qui peut entraîner un ralentissement.
> - Surveillez votre stream à l'aide de CloudWatch métriques. Cela vous permet d'identifier toute limitation et d'ajuster votre configuration en conséquence. Par exemple, la `DeliveryThrottling` métrique peut être utilisée pour suivre le nombre d'événements de journal pour lesquels CloudWatch Logs a été limité lors du transfert des données vers la destination de l'abonnement. Pour de plus amples informations sur la surveillance, veuillez consulter Surveillance à l'aide de CloudWatch métriques.
> - Utilisez le mode de capacité à la demande pour votre flux dans Kinesis Data Streams. Le mode de capacité à la demande s'adapte instantanément à vos charges de travail à mesure qu'elles augmentent ou diminuent. Pour plus d'informations sur le mode de capacité à la demande, consultez Mode de capacité à la demande.
> - Limitez votre modèle de filtre CloudWatch d'abonnement pour qu'il corresponde à la capacité de votre flux dans Kinesis Data Streams. Si vous envoyez trop de données dans le flux, il se peut que vous deviez réduire la taille du filtre ou ajuster les critères de filtrage.

Pour créer un filtre d'abonnement pour Kinesis Data Streams

1. Créez un flux de destination à l'aide de la commande suivante :

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. Attendez que le flux devienne actif (cela peut prendre une minute ou deux). Vous pouvez utiliser la commande Kinesis Data Streams describe-stream suivante pour vérifier le. StreamDescription StreamStatuspropriété. Notez également la valeur StreamDescription.StreamArn, car vous en aurez besoin ultérieurement :

```
aws kinesis describe-stream --stream-name "RootAccess"
```

Voici un exemple de sortie :

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "49551135218688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Créez le rôle IAM qui autorisera CloudWatch Logs à insérer des données dans votre flux. Vous devrez tout d'abord créer une stratégie d'approbation dans un fichier (par exemple, `~/TrustPolicyForCWL-Kinesis.json`). Utilisez un éditeur de texte créer cette stratégie. N'utilisez pas la console IAM pour la créer.

   Cette politique comprend une clé de contexte de condition `aws:SourceArn` globale pour aider à prévenir le problème de sécurité du député confus. Pour plus d'informations, consultez Prévention du député confus.

```
{
  "Statement": {
```

```
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. Utilisez la commande create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Notez la valeur retournée de Role.Arn, car vous en aurez aussi besoin ultérieurement :

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

Voici un exemple de la sortie.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. Vous allez tout d'abord créer une stratégie d'autorisations dans un fichier (par exemple, `~/PermissionsForCWL-Kinesis.json`). Utilisez un éditeur de texte créer cette stratégie. N'utilisez pas la console IAM pour la créer.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
  ]
}
```

6. Associez la politique d'autorisations au rôle à l'aide de la put-role-policycommande suivante :

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Une fois que le flux est en état actif et que vous avez créé le rôle IAM, vous pouvez créer le filtre d'abonnement CloudWatch Logs. Le filtre d'abonnement lance immédiatement la transmission de données de journaux en temps réel à partir du groupe de journaux choisi vers votre flux :

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. Après avoir configuré le filtre d'abonnement, CloudWatch Logs transmet à votre stream tous les événements de journal entrants correspondant au modèle de filtre. Vous pouvez vérifier que cela se produit en saisissant un itérateur de partition Kinesis Data Streams et en utilisant la commande get-records de Kinesis Data Streams pour rechercher des enregistrements Kinesis Data Streams :

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id
 shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Notez que vous serez peut-être amené à renouveler cet appel plusieurs fois avant que Kinesis Data Streams commence à retourner des données.

Vous devriez obtenir une réponse constituée d'un tableau d'enregistrements. L'attribut Données dans un registre Kinesis Data Streams est codé en base64 et compressé au format gzip. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen des commandes Unix suivantes :

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Les données codées et décompressées en base64 sont formatées au format JSON avec la structure suivante :

```
{
    "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\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```

Les éléments clés de la structure de données ci-dessus sont les suivants :

owner

L'ID de AWS compte des données du journal d'origine.

logGroup

Le nom du groupe de journaux des données du journal source.

logStream

Le nom du flux de journaux des données du journal source.

subscriptionFilters

La liste des noms de filtres d'abonnements qui correspondaient aux données du journal source.

messageType

Les données de messages utiliseront le type « DATA_MESSAGE ». Parfois, CloudWatch Logs peut émettre des enregistrements Kinesis Data Streams de type « CONTROL_MESSAGE », principalement pour vérifier si la destination est accessible.

logEvents

Les données du journal réelles, représentées sous la forme d'un tableau d'enregistrements d'événements du journal. La propriété « id » est un identifiant unique pour chaque événement de journal.

Exemple 2 : filtres d'abonnement avec AWS Lambda

Dans cet exemple, vous allez créer un filtre d'abonnement CloudWatch aux journaux qui envoie les données des journaux à votre AWS Lambda fonction.

> **Note**
>
> Avant de créer la fonction Lambda, calculez le volume de données du journal qui sera généré. Veillez à créer une fonction qui peut gérer ce volume. Si la fonction n'a pas suffisamment de volume, le flux de journaux sera limité. Pour plus d'informations sur les limites Lambda, consultez Limites AWS Lambda.

Pour créer un filtre d'abonnements pour Lambda

1. Créez la AWS Lambda fonction.

   Assurez-vous d'avoir configuré le rôle d'exécution Lambda. Pour plus d'informations, consultez Étape 2.2 : Créer un rôle IAM (rôle d'exécution) dans le Guide du développeur AWS Lambda .

2. Ouvrez un éditeur de texte et créez un fichier nommé `helloWorld.js` avec le contenu suivant :

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
```

```
};
```

3. Zippez le fichier helloWorld.js et enregistrez-le sous le nom `helloWorld.zip`.
4. Utilisez la commande suivante, où le rôle correspond au rôle d'exécution Lambda que vous avez configuré à la première étape :

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. Accordez à CloudWatch Logs l'autorisation d'exécuter votre fonction. Utilisez la commande suivante en remplaçant l'espace réservé au compte par votre propre compte et l'espace réservé au groupe de journaux par le groupe de journaux à traiter :

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. Créez un filtre d'abonnements à l'aide de la commande suivante en remplaçant l'espace réservé au compte par votre propre compte et l'espace réservé au groupe de journaux par le groupe de journaux à traiter :

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. (Facultatif) Testez au moyen d'un exemple d'événement de journal. A l'invite de commande, exécutez la commande suivante, qui mettra un message de journal simple dans le flux abonné.

   Pour voir le résultat de votre fonction Lambda, accédez à la fonction Lambda où vous verrez le résultat dans/: aws/lambda/helloworld

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --
log-events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple
 Lambda Test\"}]"
```

Vous devriez voir une réponse comportant un tableau de Lambda. L'attribut Data du registre Lambda est codé en base64 et compressé au format gzip. La charge utile réelle reçue par Lambda est au format suivant `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen des commandes Unix suivantes :

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

Les données codées et décompressées en base64 sont formatées au format JSON avec la structure suivante :

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
```

```
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```

Les éléments clés de la structure de données ci-dessus sont les suivants :

owner

L'ID de AWS compte des données du journal d'origine.

logGroup

Le nom du groupe de journaux des données du journal source.

logStream

Le nom du flux de journaux des données du journal source.

subscriptionFilters

La liste des noms de filtres d'abonnements qui correspondaient aux données du journal source.

messageType

Les données de messages utiliseront le type « DATA_MESSAGE ». Parfois, CloudWatch Logs peut émettre des enregistrements Lambda de type « CONTROL_MESSAGE », principalement pour vérifier si la destination est accessible.

logEvents

Les données du journal réelles, représentées sous la forme d'un tableau d'enregistrements d'événements du journal. La propriété « id » est un identifiant unique pour chaque événement de journal.

Exemple 3 : filtres d'abonnement avec Amazon Data Firehose

Dans cet exemple, vous allez créer un abonnement CloudWatch Logs qui envoie tous les événements de log entrants correspondant aux filtres que vous avez définis à votre flux de diffusion Amazon Data Firehose. Les données envoyées par CloudWatch Logs à Amazon Data Firehose sont déjà compressées avec la compression gzip de niveau 6. Vous n'avez donc pas besoin d'utiliser

la compression dans votre flux de diffusion Firehose. Vous pouvez ensuite utiliser la fonction de décompression de Firehose pour décompresser automatiquement les journaux. Pour plus d'informations, consultez Envoyer CloudWatch des journaux à Firehose.

> **Note**
>
> Avant de créer le flux Firehose, calculez le volume de données de journal qui sera généré. Assurez-vous de créer un flux Firehose capable de gérer ce volume. Si le flux ne peut pas traiter le volume, le flux de journaux sera limité. Pour plus d'informations sur les limites de volume de flux Firehose, consultez Amazon Data Firehose Data Limits.

Pour créer un filtre d'abonnement pour Firehose

1. Créez un compartiment Amazon Simple Storage Service (Amazon S3). Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Toutefois, si vous souhaitez utiliser un compartiment existant, passez directement à l'étape 2.

   Exécutez la commande suivante en remplaçant l'espace réservé à la région par la région que vous voulez utiliser :

   ```
   aws s3api create-bucket --bucket amzn-s3-demo-bucket2 --create-bucket-configuration
    LocationConstraint=region
   ```

   Voici un exemple de sortie :

   ```
   {
       "Location": "/amzn-s3-demo-bucket2"
   }
   ```

2. Créez le rôle IAM qui autorise Amazon Data Firehose à placer des données dans votre compartiment Amazon S3.

   Pour plus d'informations, consultez la section Contrôler l'accès avec Amazon Data Firehose dans le manuel du développeur Amazon Data Firehose.

   D'abord, utilisez un éditeur de texte pour créer une politique d'approbation dans un fichier `~/TrustPolicyForFirehose.json` comme suit :

   ```
   {
   ```

```
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "firehose.amazonaws.com" },
    "Action": "sts:AssumeRole"
    }
}
```

3. Utilisez la commande create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Notez la valeur retournée de Role.Arn, car vous en aurez besoin ultérieurement :

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Créez une politique d'autorisation pour définir les actions que Firehose peut effectuer sur votre compte. D'abord, utilisez un éditeur de texte pour créer une stratégie d'autorisations dans un fichier `~/PermissionsForFirehose.json` :

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
```

```
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:ListBucketMultipartUploads",
            "s3:PutObject" ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket2",
            "arn:aws:s3:::amzn-s3-demo-bucket2/*" ]
      }
    ]
}
```

5. Associez la politique d'autorisations au rôle à l'aide de la put-role-policy commande suivante :

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Créez un flux de diffusion Firehose de destination comme suit, en remplaçant les valeurs d'espace réservé pour Rolearn et BucketArn par le rôle et le bucket que vous avez créés : ARNs

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket2"}'
```

   Notez que Firehose utilise automatiquement un préfixe au format YYYY/MM/DD/HH UTC pour les objets Amazon S3 livrés. Vous pouvez spécifier un préfixe supplémentaire à ajouter devant le préfixe de format temporel. Si le préfixe se termine par une barre oblique (/), il apparaît comme un dossier dans le compartiment Amazon S3.

7. Attendez que le flux devienne actif (cela peut prendre quelques minutes). Vous pouvez utiliser la describe-delivery-streamcommande Firehose pour vérifier le. DeliveryStreamDescription DeliveryStreamStatuspropriété. En outre, notez le DeliveryStreamDescription. DeliveryStreamValeur de l'ARN, dont vous aurez besoin ultérieurement :

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
```

```
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket2",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Créez le rôle IAM qui autorise CloudWatch Logs à insérer des données dans votre flux de diffusion Firehose. D'abord, utilisez un éditeur de texte pour créer une stratégie d'approbation dans un fichier `~/TrustPolicyForCWL.json` :

   Cette politique comprend une clé de contexte de condition `aws:SourceArn` globale pour aider à prévenir le problème de sécurité du député confus. Pour plus d'informations, consultez Prévention du député confus.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
```

```
  }
}
```

9. Utilisez la commande create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Notez la valeur retournée de Role.Arn, car vous en aurez besoin ultérieurement :

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. D'abord, utilisez un éditeur de texte pour créer une stratégie d'autorisations dans un fichier (par exemple, `~/PermissionsForCWL.json`) :

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
```

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

11. Associez la politique d'autorisations au rôle à l'aide de la put-role-policy commande :

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Une fois que le flux de diffusion Amazon Data Firehose est actif et que vous avez créé le rôle IAM, vous pouvez créer le filtre d'abonnement CloudWatch Logs. Le filtre d'abonnement lance immédiatement le flux de données de journal en temps réel entre le groupe de journaux choisi et votre flux de diffusion Amazon Data Firehose :

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. Une fois le filtre d'abonnement configuré, CloudWatch Logs transmet tous les événements de journal entrants correspondant au modèle de filtre à votre flux de diffusion Amazon Data Firehose. Vos données commenceront à apparaître dans votre Amazon S3 en fonction de l'intervalle de temps défini sur votre flux de diffusion Amazon Data Firehose. Après un délai suffisant, vous pouvez consulter votre compartiment Amazon S3 pour vérifier vos données.

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket2' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
```

```
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket2' --key 'firehose/2015/10/29/00/
my-delivery-stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Les données dans l'objet Amazon S3 sont comprimées au format gzip. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen de la commande Unix suivante :

```
zcat testfile.gz
```

Exemple 4 : filtres d'abonnement avec Amazon OpenSearch Service

Dans cet exemple, vous allez créer un abonnement CloudWatch Logs qui envoie à votre domaine de OpenSearch service les événements de journal entrants correspondant aux filtres que vous avez définis.

Pour créer un filtre d'abonnement pour le OpenSearch service

1. Créez un domaine OpenSearch de service. Pour plus d'informations, voir Création de domaines OpenSearch de service
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation, choisissez Groupes de journaux.
4. Sélectionnez le nom du groupe de journaux.
5. Choisissez Actions, Filtres d'abonnement, puis Créer un filtre d'abonnement Amazon OpenSearch Service.
6. Choisissez si vous souhaitez diffuser en continu vers un cluster de ce compte ou un autre compte.
   - Si vous avez choisi Ce compte, sélectionnez le domaine que vous avez créé à l'étape 1.
   - Si vous avez choisi Un autre compte, entrez l'ARN et le point de terminaison de ce domaine.
7. Si vous avez choisi un autre compte, fournissez l'ARN du domaine et le point de terminaison.
8. Pour le rôle d'exécution Lambda IAM, choisissez le rôle IAM que Lambda doit utiliser lors de l'exécution des appels au service. OpenSearch Le rôle IAM que vous choisissez doit répondre aux exigences suivantes :
   - Il doit avoir `lambda.amazonaws.com` dans la relation d'approbation.
   - Il doit inclure la stratégie suivante :

     JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
```

```
            "Resource": "arn:aws:es:us-east-1:111122223333:domain/target-
domain-name/*"
        }
    ]
}
```

- Si le domaine de OpenSearch service cible utilise un accès VPC, le rôle IAM doit également être associé à la AWSLambdaVPCAccessExecutionRolepolitique. Cette politique gérée par Amazon accorde à Lambda l'accès au VPC du client, ce qui permet à Lambda d'écrire sur le point de terminaison du VPC. OpenSearch

9. Choisissez un format de journal.
10. Pour le modèle de filtre d'abonnement, entrez les termes ou le modèle à rechercher dans les événements de votre journal. Cela garantit que vous n'envoyez que les données qui vous intéressent à votre cluster OpenSearch de services. Pour de plus amples informations, veuillez consulter Syntaxe des modèles de filtres pour les filtres de métriques.
11. (Facultatif) Pour Select log data to test (Sélectionner les données de journal à tester), sélectionnez un flux de journal, puis choisissez Test pattern (Modèle de test) afin de vérifier que votre filtre de recherche renvoie les résultats attendus.
12. Choisissez Start streaming (Démarrer la diffusion).

Filtres d'abonnement au niveau du compte

> **Important**
>
> Il existe un risque de créer une boucle récursive infinie avec des filtres d'abonnement, ce qui peut entraîner une forte augmentation de la facturation en cas d'ingestion si rien n'est fait pour y remédier. Pour atténuer ce risque, nous vous recommandons d'utiliser des critères de sélection dans les filtres d'abonnement au niveau de votre compte afin d'exclure les groupes de journaux qui ingèrent des données de journal provenant de ressources faisant partie du flux de travail de livraison des abonnements. Pour plus d'informations sur ce problème et pour déterminer les groupes de journaux à exclure, consultezPrévention de la récursivité dans les journaux.

Vous pouvez définir une politique d'abonnement au niveau du compte qui inclut un sous-ensemble de groupes de journaux dans le compte. La politique d'abonnement au compte peut fonctionner avec Amazon Kinesis Data Streams AWS Lambda ou Amazon Data Firehose. Les journaux

envoyés à un service via une politique d'abonnement au niveau du compte sont codés en base64 et compressés au format gzip. Cette section fournit des exemples que vous pouvez suivre pour créer un abonnement au niveau du compte pour Kinesis Data Streams, Lambda et Firehose.

> **Note**
>
> Pour consulter la liste de toutes les politiques de filtrage des abonnements de votre compte, utilisez la `describe-account-policies` commande dont la valeur est `SUBSCRIPTION_FILTER_POLICY` pour le `--policy-type` paramètre. Pour plus d'informations, consultez describe-account-policies¶.

Exemples

- Exemple 1 : filtres d'abonnement avec Kinesis Data Streams
- Exemple 2 : filtres d'abonnement avec AWS Lambda
- Exemple 3 : filtres d'abonnement avec Amazon Data Firehose

Exemple 1 : filtres d'abonnement avec Kinesis Data Streams

Avant de créer un flux de données Kinesis Data Streams à utiliser dans le cadre d'une politique d'abonnement au niveau du compte, calculez le volume de données de journal qui sera généré. Assurez-vous de créer un flux avec suffisamment de partitions pour gérer le volume. Si un flux ne contient pas suffisamment de partitions, il est limité. Pour plus d'informations sur les limites de volume de flux, consultez la section Quotas et limites dans la documentation de Kinesis Data Streams.

> **Warning**
>
> Les événements de journal de plusieurs groupes de journaux étant transférés vers la destination, il existe un risque de limitation. Les livrables limités sont réessayés pendant 24 heures au maximum. Au bout de 24 heures, les livrables ayant échoué sont supprimés. Pour réduire le risque de limitation, procédez comme suit :
>
> - Surveillez votre flux Kinesis Data Streams à l' CloudWatch aide de métriques. Cela vous permet d'identifier les ralentissements et d'ajuster votre configuration en conséquence. Par exemple, la `DeliveryThrottling` métrique suit le nombre d'événements de journal pour lesquels CloudWatch Logs a été limité lors du transfert des données vers la destination de

l'abonnement. Pour de plus amples informations, veuillez consulter Surveillance à l'aide de CloudWatch métriques.

- Utilisez le mode de capacité à la demande pour votre flux dans Kinesis Data Streams. Le mode de capacité à la demande s'adapte instantanément à vos charges de travail à mesure qu'elles augmentent ou diminuent. Pour plus d'informations, consultez la section Mode à la demande.
- Limitez le modèle de filtre de votre abonnement à CloudWatch Logs pour qu'il corresponde à la capacité de votre flux dans Kinesis Data Streams. Si vous envoyez trop de données dans le flux, il se peut que vous deviez réduire la taille du filtre ou ajuster les critères de filtrage.

L'exemple suivant utilise une politique d'abonnement au niveau du compte pour transférer tous les événements du journal vers un flux dans Kinesis Data Streams. Le modèle de filtre fait correspondre tous les événements du journal au texte `Test` et les transmet au flux dans Kinesis Data Streams.

Pour créer une politique d'abonnement au niveau du compte pour Kinesis Data Streams

1. Créez un flux de destination à l'aide de la commande suivante :

```
$ C:\>  aws kinesis create-stream –stream-name "TestStream" –shard-count 1
```

2. Patientez quelques minutes pour que le stream soit actif. Vous pouvez vérifier si le flux est actif en utilisant la commande describe-stream pour vérifier le. StreamDescription StreamStatuspropriété.

```
aws kinesis describe-stream --stream-name "TestStream"
```

Voici un exemple de sortie :

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "TestStream",
        "StreamARN": "arn:aws:kinesis:region:123456789012:stream/TestStream",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
```

```
                    "EndingHashKey": "EXAMPLE84634633746074317682114 55",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "EXAMPLE68881845667950383198145878459135270218157 2610"
                }
            }
        ]
    }
}
```

3. Créez le rôle IAM qui autorisera CloudWatch Logs à insérer des données dans votre flux. Vous devrez tout d'abord créer une stratégie d'approbation dans un fichier (par exemple, `~/TrustPolicyForCWL-Kinesis.json`). Utilisez un éditeur de texte créer cette stratégie.

   Cette politique comprend une clé de contexte de condition `aws:SourceArn` globale pour aider à prévenir le problème de sécurité du député confus. Pour plus d'informations, consultez Prévention du député confus.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. Utilisez la commande create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Notez la valeur retournée de Role.Arn, car vous en aurez aussi besoin ultérieurement :

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

   Voici un exemple de la sortie.

```
{
```

```
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "EXAMPLE450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. Vous allez tout d'abord créer une stratégie d'autorisations dans un fichier (par exemple, `~/PermissionsForCWL-Kinesis.json`). Utilisez un éditeur de texte créer cette stratégie. N'utilisez pas la console IAM pour le créer.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/TestStream"
    }
  ]
}
```

6. Associez la politique d'autorisations au rôle à l'aide de la put-role-policycommande suivante :

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Une fois que le flux est à l'état actif et que vous avez créé le rôle IAM, vous pouvez créer la politique de filtrage CloudWatch des abonnements aux journaux. La politique lance immédiatement le flux de données de journal en temps réel vers votre flux. Dans cet exemple, tous les événements du journal contenant la chaîne `ERROR` sont diffusés en continu, à l'exception de ceux des groupes de journaux nommés `LogGroupToExclude1` et`LogGroupToExclude2`.

```
aws logs put-account-policy \
    --policy-name "ExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisRole", "DestinationArn":"arn:aws:kinesis:region:123456789012:stream/
TestStream", "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

8. Après avoir configuré le filtre d'abonnement, CloudWatch Logs transmet à votre stream tous les événements de journal entrants correspondant au modèle de filtre et aux critères de sélection.

   Ce `selection-criteria` champ est facultatif, mais il est important pour exclure les groupes de journaux susceptibles de provoquer une récursivité infinie des journaux à partir d'un filtre d'abonnement. Pour plus d'informations sur ce problème et pour déterminer les groupes de journaux à exclure, consultez[Prévention de la récursivité dans les journaux]. NOT IN est actuellement le seul opérateur pris en charge pour`selection-criteria`.

   Vous pouvez vérifier le flux des événements du journal en utilisant un itérateur de partition Kinesis Data Streams et en utilisant la commande Kinesis Data Streams pour récupérer certains enregistrements Kinesis `get-records` Data Streams :

```
aws kinesis get-shard-iterator --stream-name TestStream --shard-id
 shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Vous devrez peut-être utiliser cette commande plusieurs fois avant que Kinesis Data Streams ne commence à renvoyer des données.

Vous devriez obtenir une réponse constituée d'un tableau d'enregistrements. L'attribut Données dans un registre Kinesis Data Streams est codé en base64 et compressé au format gzip. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen des commandes Unix suivantes :

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Les données codées et décompressées en base64 sont formatées au format JSON avec la structure suivante :

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicy"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
```

```
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

Les principaux éléments de la structure de données sont les suivants :

messageType

Les données de messages utiliseront le type « DATA_MESSAGE ». Parfois, CloudWatch Logs peut émettre des enregistrements Kinesis Data Streams de type « CONTROL_MESSAGE », principalement pour vérifier si la destination est accessible.

owner

L'ID de AWS compte des données du journal d'origine.

logGroup

Le nom du groupe de journaux des données du journal source.

logStream

Le nom du flux de journaux des données du journal source.

subscriptionFilters

La liste des noms de filtres d'abonnements qui correspondaient aux données du journal source.

logEvents

Les données du journal réelles, représentées sous la forme d'un tableau d'enregistrements d'événements du journal. La propriété « id » est un identifiant unique pour chaque événement de journal.

Niveau de la politique

Niveau auquel la politique a été appliquée. « ACCOUNT_LEVEL_POLICY » correspond à une politique de filtrage `policyLevel` d'abonnement au niveau du compte.

Exemple 2 : filtres d'abonnement avec AWS Lambda

Dans cet exemple, vous allez créer une politique de filtrage d'abonnement au niveau du compte CloudWatch Logs qui envoie les données des journaux à votre AWS Lambda fonction.

> **Warning**
>
> Avant de créer la fonction Lambda, calculez le volume de données du journal qui sera généré. Veillez à créer une fonction qui peut gérer ce volume. Si la fonction ne peut pas gérer le volume, le flux du journal sera limité. Étant donné que les événements de journal de tous les groupes de journaux ou d'un sous-ensemble des groupes de journaux du compte sont transférés vers la destination, il existe un risque de limitation. Pour plus d'informations sur les limites Lambda, consultez Limites AWS Lambda.

Pour créer une politique de filtrage des abonnements au niveau du compte pour Lambda

1. Créez la AWS Lambda fonction.

   Assurez-vous d'avoir configuré le rôle d'exécution Lambda. Pour plus d'informations, consultez Étape 2.2 : Créer un rôle IAM (rôle d'exécution) dans le Guide du développeur AWS Lambda .
2. Ouvrez un éditeur de texte et créez un fichier nommé `helloWorld.js` avec le contenu suivant :

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Zippez le fichier helloWorld.js et enregistrez-le sous le nom `helloWorld.zip`.
4. Utilisez la commande suivante, où le rôle correspond au rôle d'exécution Lambda que vous avez configuré à la première étape :

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs18.x
```

5. Accordez à CloudWatch Logs l'autorisation d'exécuter votre fonction. Utilisez la commande suivante pour remplacer le compte fictif par votre propre compte.

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789012:log-group:*" \
    --source-account "123456789012"
```

6. Créez une politique de filtrage des abonnements au niveau du compte à l'aide de la commande suivante, en remplaçant le compte réservé par votre propre compte. Dans cet exemple, tous les événements du journal contenant la chaîne `ERROR` sont diffusés en continu, à l'exception de ceux des groupes de journaux nommés `LogGroupToExclude1` et`LogGroupToExclude2`.

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyLambda" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:lambda:region:123456789012:function:helloWorld",
 "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Après avoir configuré le filtre d'abonnement, CloudWatch Logs transmet à votre stream tous les événements de journal entrants correspondant au modèle de filtre et aux critères de sélection.

Ce `selection-criteria` champ est facultatif, mais il est important pour exclure les groupes de journaux susceptibles de provoquer une récursivité infinie des journaux à partir d'un filtre d'abonnement. Pour plus d'informations sur ce problème et pour déterminer les groupes

de journaux à exclure, consultez Prévention de la récursivité dans les journaux. NOT IN est actuellement le seul opérateur pris en charge pour `selection-criteria`.

7. (Facultatif) Testez au moyen d'un exemple d'événement de journal. A l'invite de commande, exécutez la commande suivante, qui mettra un message de journal simple dans le flux abonné.

   Pour voir le résultat de votre fonction Lambda, accédez à la fonction Lambda où vous verrez le résultat dans/: aws/lambda/helloworld

```
aws logs put-log-events --log-group-name Example1 --log-stream-name logStream1 --
log-events "[{\"timestamp\":CURRENT TIMESTAMP MILLIS , \"message\": \"Simple Lambda
 Test\"}]"
```

   Vous devriez voir une réponse comportant un tableau de Lambda. L'attribut Data du registre Lambda est codé en base64 et compressé au format gzip. La charge utile réelle reçue par Lambda est au format suivant `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen des commandes Unix suivantes :

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

   Les données codées et décompressées en base64 sont formatées au format JSON avec la structure suivante :

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicyLambda"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
```

```
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

> **Note**
>
> Le filtre d'abonnement au niveau du compte ne sera pas appliqué au groupe de journaux de la fonction Lambda de destination. Cela permet d'éviter une récursivité infinie des logs susceptible d'entraîner une augmentation de la facturation d'ingestion. Pour plus d'informations sur ce problème, consultez Prévention de la récursivité dans les journaux.

Les principaux éléments de la structure de données sont les suivants :

messageType

Les données de messages utiliseront le type « DATA_MESSAGE ». Parfois, CloudWatch Logs peut émettre des enregistrements Kinesis Data Streams de type « CONTROL_MESSAGE », principalement pour vérifier si la destination est accessible.

owner

L'ID de AWS compte des données du journal d'origine.

logGroup

Le nom du groupe de journaux des données du journal source.

logStream

Le nom du flux de journaux des données du journal source.

subscriptionFilters

La liste des noms de filtres d'abonnements qui correspondaient aux données du journal source.

logEvents

Les données du journal réelles, représentées sous la forme d'un tableau d'enregistrements d'événements du journal. La propriété « id » est un identifiant unique pour chaque événement de journal.

Niveau de la politique

Niveau auquel la politique a été appliquée. « ACCOUNT_LEVEL_POLICY » correspond à une politique de filtrage `policyLevel` d'abonnement au niveau du compte.

Exemple 3 : filtres d'abonnement avec Amazon Data Firehose

Dans cet exemple, vous allez créer une politique de filtrage d'abonnement au niveau du compte CloudWatch Logs qui envoie les événements de journal entrants correspondant aux filtres que vous avez définis à votre flux de diffusion Amazon Data Firehose. Les données envoyées par CloudWatch Logs à Amazon Data Firehose sont déjà compressées avec la compression gzip de niveau 6. Vous n'avez donc pas besoin d'utiliser la compression dans votre flux de diffusion Firehose. Vous pouvez ensuite utiliser la fonction de décompression de Firehose pour décompresser automatiquement les journaux. Pour plus d'informations, consultez la section Écrire dans Kinesis Data CloudWatch Firehose à l'aide de journaux.

> **Warning**
>
> Avant de créer le flux Firehose, calculez le volume de données de journal qui sera généré. Assurez-vous de créer un flux Firehose capable de gérer ce volume. Si le flux ne peut pas traiter le volume, le flux de journaux sera limité. Pour plus d'informations sur les limites de volume de flux Firehose, consultez Amazon Data Firehose Data Limits.

Pour créer un filtre d'abonnement pour Firehose

1. Créez un compartiment Amazon Simple Storage Service (Amazon S3). Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Toutefois, si vous souhaitez utiliser un compartiment existant, passez directement à l'étape 2.

Exécutez la commande suivante en remplaçant l'espace réservé à la région par la région que vous voulez utiliser :

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket2 --create-bucket-configuration
 LocationConstraint=region
```

Voici un exemple de sortie :

```
{
    "Location": "/amzn-s3-demo-bucket2"
}
```

2. Créez le rôle IAM qui autorise Amazon Data Firehose à placer des données dans votre compartiment Amazon S3.

   Pour plus d'informations, consultez la section Contrôler l'accès avec Amazon Data Firehose dans le manuel du développeur Amazon Data Firehose.

   D'abord, utilisez un éditeur de texte pour créer une politique d'approbation dans un fichier `~/TrustPolicyForFirehose.json` comme suit :

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "firehose.amazonaws.com" },
    "Action": "sts:AssumeRole"
    }
}
```

3. Utilisez la commande create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Notez la valeur Role.Arn renvoyée, car vous en aurez besoin ultérieurement :

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
```

```
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "EXAMPLE50GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Créez une politique d'autorisation pour définir les actions que Firehose peut effectuer sur votre compte. D'abord, utilisez un éditeur de texte pour créer une stratégie d'autorisations dans un fichier `~/PermissionsForFirehose.json` :

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket2",
          "arn:aws:s3:::amzn-s3-demo-bucket2/*" ]
    }
  ]
}
```

5. Associez la politique d'autorisations au rôle à l'aide de la put-role-policy commande suivante :

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Créez un flux de diffusion Firehose de destination comme suit, en remplaçant les valeurs d'espace réservé pour Rolearn et BucketArn par le rôle et le bucket que vous avez créés : ARNs

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket2"}'
```

NFirehose utilise automatiquement un préfixe au format YYYY/MM/DD/HH UTC pour les objets Amazon S3 livrés. Vous pouvez spécifier un préfixe supplémentaire à ajouter devant le préfixe de format temporel. Si le préfixe se termine par une barre oblique (/), il apparaît comme un dossier dans le compartiment Amazon S3.

7. Patientez quelques minutes pour que le stream soit activé. Vous pouvez utiliser la describe-delivery-streamcommande Firehose pour vérifier le. DeliveryStreamDescription DeliveryStreamStatuspropriété. En outre, notez le DeliveryStreamDescription. DeliveryStreamValeur de l'ARN, dont vous aurez besoin ultérieurement :

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket2",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
```

```
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Créez le rôle IAM qui autorise CloudWatch Logs à insérer des données dans votre flux de diffusion Firehose. D'abord, utilisez un éditeur de texte pour créer une stratégie d'approbation dans un fichier `~/TrustPolicyForCWL.json` :

   Cette politique comprend une clé de contexte de condition `aws:SourceArn` globale pour aider à prévenir le problème de sécurité du député confus. Pour plus d'informations, consultez Prévention du député confus.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
  }
}
```

9. Utilisez la commande create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Notez la valeur Role.Arn renvoyée, car vous en aurez besoin ultérieurement :

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
```

```
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. D'abord, utilisez un éditeur de texte pour créer une stratégie d'autorisations dans un fichier (par exemple, `~/PermissionsForCWL.json`) :

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Associez la politique d'autorisations au rôle à l'aide de la put-role-policy commande :

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Une fois que le flux de diffusion Amazon Data Firehose est actif et que vous avez créé le rôle IAM, vous pouvez créer la politique de filtrage des abonnements au niveau du compte

CloudWatch Logs. La politique lance immédiatement le flux de données de journal en temps réel du groupe de journaux choisi vers votre flux de diffusion Amazon Data Firehose :

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyFirehose" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisFirehoseRole", "DestinationArn":"arn:aws:firehose:us-
east-1:123456789012:deliverystream/delivery-stream-name", "FilterPattern": "Test",
 "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

13. Après avoir configuré le filtre d'abonnement, CloudWatch Logs transmet les événements de journal entrants correspondant au modèle de filtre à votre flux de diffusion Amazon Data Firehose.

    Ce `selection-criteria` champ est facultatif, mais il est important pour exclure les groupes de journaux susceptibles de provoquer une récursivité infinie des journaux à partir d'un filtre d'abonnement. Pour plus d'informations sur ce problème et pour déterminer les groupes de journaux à exclure, consultezPrévention de la récursivité dans les journaux. NOT IN est actuellement le seul opérateur pris en charge pour`selection-criteria`.

    Vos données commenceront à apparaître dans votre Amazon S3 en fonction de l'intervalle de temps défini sur votre flux de diffusion Amazon Data Firehose. Après un délai suffisant, vous pouvez consulter votre compartiment Amazon S3 pour vérifier vos données.

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket2' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2023-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
```

```
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2023/10/29/00/my-delivery-stream-2023-10-29-00-35-40-
EXAMPLE-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "EXAMPLE6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket2' --key 'firehose/2023/10/29/00/
my-delivery-stream-2023-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2023 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Les données dans l'objet Amazon S3 sont comprimées au format gzip. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen de la commande Unix suivante :

```
zcat testfile.gz
```

Abonnements entre comptes et entre régions

Vous pouvez collaborer avec le propriétaire d'un autre AWS compte et recevoir ses événements de journal sur vos AWS ressources, comme un flux Amazon Kinesis ou Amazon Data Firehose (c'est ce que l'on appelle le partage de données entre comptes). Par exemple, les données de ce journal d'événements peuvent être lues à partir d'un flux Kinesis Data Streams ou Firehose centralisé pour

effectuer un traitement et une analyse personnalisés. Le traitement personnalisé est particulièrement utile lorsque vous collaborez et analysez les données sur plusieurs comptes.

Par exemple, le groupe de sécurité des informations d'une société pourrait souhaiter analyser des données relatives à la détection d'intrusions en temps réel ou aux comportements anormaux. Il peut donc procéder à un audit des comptes de tous les services de la société en recueillant leurs journaux de production séparés afin de les traiter de manière centrale. Un flux de diffusion en temps réel de données d'événements sur ces comptes peut être assemblé et acheminé vers les groupes de sécurité des informations, qui peuvent utiliser Kinesis Data Streams pour associer les données à leurs systèmes d'analyse de sécurité existants.

> **Note**
>
> Le groupe de journaux et la destination doivent se trouver dans la même AWS région. Cependant, la AWS ressource pointée par la destination peut être située dans une autre région. Dans les exemples des sections suivantes, toutes les ressources spécifiques à une région sont créées dans l'est des États-Unis (Virginie du Nord).

Rubriques

- Partage de données de journal entre comptes et entre régions à l'aide de Kinesis Data Streams
- Partage de données de journal entre comptes et entre régions à l'aide de Firehose
- Abonnements entre comptes et entre régions à l'aide de Kinesis Data Streams
- Abonnements entre comptes et entre régions à l'aide de Firehose

Partage de données de journal entre comptes et entre régions à l'aide de Kinesis Data Streams

Lorsque vous créez un abonnement entre comptes, vous pouvez spécifier un seul compte ou une organisation comme expéditeur. Si vous spécifiez une organisation, cette procédure permet à tous les comptes de l'organisation d'envoyer des journaux au compte récepteur.

Pour partager des données du journal entre comptes, vous devez spécifier un expéditeur et un récepteur :

- Expéditeur des données de journal : obtient les informations de destination auprès du destinataire et CloudWatch indique à Logs qu'il est prêt à envoyer ses événements de journal à la destination

spécifiée. Dans les procédures décrites dans le reste de cette section, l'expéditeur des données du journal est indiqué avec un numéro de AWS compte fictif 111111111111.

Si plusieurs comptes dans une organisation doivent envoyer des journaux au compte d'un destinataire, vous pouvez créer une politique qui accorde à tous les comptes de l'organisation l'autorisation d'envoyer des journaux au compte du destinataire. Vous devez toujours configurer des filtres d'abonnement distincts pour chaque compte d'expéditeur.

- Destinataire des données du journal : définit une destination qui encapsule un flux Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir les données du journal. Le destinataire partage ensuite les informations sur cette destination avec l'expéditeur. Dans les procédures décrites dans le reste de cette section, le destinataire des données du journal est indiqué avec un numéro de AWS compte fictif 999999999999.

Pour commencer à recevoir des événements de journal provenant d'utilisateurs multicomptes, le destinataire des données de journal crée d'abord une destination de CloudWatch journaux. Chaque destination comprend les éléments clés suivants :

Nom de destination

Le nom de la destination que vous souhaitez créer.

ARN cible

Le nom de ressource Amazon (ARN) de la AWS ressource que vous souhaitez utiliser comme destination du flux d'abonnement.

Role ARN (ARN de rôle)

Rôle AWS Identity and Access Management (IAM) qui accorde à CloudWatch Logs les autorisations nécessaires pour placer des données dans le flux choisi.

Stratégie d'accès

Un document de politique IAM (au format JSON, écrit à l'aide de la syntaxe des politiques IAM) régissant l'ensemble des utilisateurs qui sont autorisés à écrire à votre destination.

> **Note**
>
> Le groupe de journaux et la destination doivent se trouver dans la même AWS région. Par contre, la ressource AWS vers laquelle pointe la destination peut être située dans une autre

région. Dans les exemples des sections suivantes, toutes les ressources spécifiques à la région sont créées dans USA Est (Virginie du Nord).

Rubriques

- Configurer un nouvel abonnement entre comptes
- Mise à jour d'un abonnement existant entre comptes

Configurer un nouvel abonnement entre comptes

Suivez les étapes décrites dans ces sections pour configurer un nouvel abonnement au journal entre comptes.

Rubriques

- Étape 1 : créer une destination
- Étape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM
- Étape 3 : autorisations Add/validate IAM pour la destination multi-comptes
- Étape 4 : créer un filtre d'abonnement
- Validation du flux des événements des journaux
- Modification de l'appartenance à une destination au moment de l'exécution

Étape 1 : créer une destination

> **Important**
>
> Toutes les étapes de cette procédure doivent être réalisées dans le compte du destinataire des données du journal.

Dans cet exemple, le compte du destinataire des données du journal a un ID de compte 999999999999, tandis que l'identifiant du AWS compte de l'expéditeur AWS des données du journal est 111111111111.

Cet exemple crée une destination à l'aide d'un flux Kinesis Data Streams RecipientStream appelé, et d'un rôle qui CloudWatch permet à Logs d'y écrire des données.

Lorsque la destination est créée, CloudWatch Logs envoie un message test à la destination au nom du compte du destinataire. Lorsque le filtre d'abonnement est activé ultérieurement, CloudWatch Logs envoie les événements du journal à la destination au nom du compte source.

Pour créer une destination

1. Dans le compte du destinataire, créez un flux de diffusion de destination dans Kinesis Data Streams. À l'invite de commande, saisissez :

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Patientez jusqu'à ce que le flux devienne actif. Vous pouvez utiliser la commande aws kinesis describe-stream pour vérifier le. StreamDescription StreamStatuspropriété. Prenez également note de la valeur StreamDescription.StreamArn, car vous la transmettrez ultérieurement à CloudWatch Logs :

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768211EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702181EXAMPLE"
        }
      }
    ]
  }
}
```

Votre flux de données peut prendre une ou deux minutes avant d'apparaître avec le statut actif.

3. Créez le rôle IAM qui autorise CloudWatch Logs à insérer des données dans votre flux. Tout d'abord, vous devez créer une politique de confiance dans un fichier ~/ TrustPolicyFor CWL.json. Utilisez un éditeur de texte pour créer ce fichier de politique, n'utilisez pas la console IAM.

   Cette politique comprend une clé de contexte de condition globale `aws:SourceArn` qui spécifie le `sourceAccountId` pour aider à prévenir le problème de sécurité du député confus. Si vous ne connaissez pas encore l'ID du compte source lors du premier appel, nous vous recommandons de placer l'ARN de destination dans le champ ARN source. Dans les appels suivants, vous devez définir l'ARN source comme ARN source réel que vous avez recueilli lors du premier appel. Pour plus d'informations, consultez Prévention du député confus.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. Utilisez la commande aws iam create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Prenez note de la valeur Role.Arn renvoyée, car elle sera également transmise à CloudWatch Logs ultérieurement :

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
```

```
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. Utilisez d'abord un éditeur de texte pour créer une politique d'autorisations dans un fichier ~/ PermissionsFor CWL.json :

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Associez la politique d'autorisation au rôle à l'aide de la put-role-policy commande aws iam :

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
```

```
    --policy-document file://~/PermissionsForCWL.json
```

7. Une fois que le flux est actif et que vous avez créé le rôle IAM, vous pouvez créer la destination CloudWatch Logs.

   a. Cette étape n'associe aucune stratégie d'accès à votre destination. Il s'agit uniquement de la première de deux étapes pour créer une destination. Notez DestinationArnce qui est renvoyé dans la charge utile :

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Une fois l'étape 7a terminée, dans le compte des données du journal du destinataire, associez une stratégie d'accès à la destination. Cette politique doit spécifier l'PutSubscriptionFilteraction logs : et autoriser le compte expéditeur à accéder à la destination.

      La politique accorde l'autorisation au AWS compte qui envoie les journaux. Vous pouvez spécifier uniquement ce compte dans la politique. Si le compte de l'expéditeur est membre d'une organisation, la politique peut également spécifier l'ID de l'organisation. De cette façon, vous pouvez créer une seule politique pour autoriser plusieurs comptes d'une organisation à envoyer des journaux à ce compte de destination.

      Utilisez un éditeur de texte pour créer un fichier nommé `~/AccessPolicy.json` avec l'une des déclarations de politique suivantes.

      Ce premier exemple de politique autorise tous les comptes de l'organisation possédant un ID `o-1234567890` à envoyer les journaux au compte du destinataire.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

Cet exemple suivant permet uniquement au compte des données du journal de l'expéditeur (111111111111) d'envoyer des journaux au compte du destinataire des données du journal.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111111111111"
            },
            "Action": "logs:PutSubscriptionFilter",
```

```
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
        }
    ]
}
```

c. Associez la politique que vous avez créée à l'étape précédente à la destination.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Cette politique d'accès permet aux utilisateurs du AWS compte ayant l'ID 111111111111 d'appeler la destination avec l'ARN arn:aws:logs PutSubscriptionFilter::999999999999999999999:destination:testDestination. *region* Toute tentative d' PutSubscriptionFilter appel de la part d'un autre utilisateur vers cette destination sera rejetée.

Pour valider les privilèges d'un utilisateur par rapport à une stratégie d'accès, consultez Utilisation du programme de validation de politiques dans le Guide de l'utilisateur IAM.

Lorsque vous avez terminé, si vous utilisez AWS Organizations pour vos autorisations multi-comptes, suivez les étapes décrites dansÉtape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM. Si vous accordez des autorisations directement à l'autre compte au lieu d'utiliser des organisations, vous pouvez ignorer cette étape et passer à Étape 4 : créer un filtre d'abonnement.

Étape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM

Dans la section précédente, si vous avez créé la destination à l'aide d'une stratégie d'accès qui accorde des autorisations à l'organisation dans laquelle ce compte 111111111111 se trouve, au lieu d'accorder des autorisations directement au compte 111111111111, suivez les étapes de cette section. Sinon, vous pouvez passer à Étape 4 : créer un filtre d'abonnement.

Les étapes décrites dans cette section créent un rôle IAM, qui CloudWatch peut assumer et valider si le compte expéditeur est autorisé à créer un filtre d'abonnement par rapport à la destination du destinataire.

Suivez les étapes de cette section dans le compte de l'expéditeur. Le rôle doit exister dans le compte de l'expéditeur et vous devez spécifier l'ARN de ce rôle dans le filtre d'abonnement. Dans cet exemple, le compte de l'expéditeur est 111111111111.

Pour créer le rôle IAM nécessaire aux abonnements aux journaux entre comptes à l'aide de AWS Organizations

1. Créez la politique de confiance suivante dans un fichier `/TrustPolicyForCWLSubscriptionFilter.json`. Utilisez un éditeur de texte pour créer ce fichier de politique, n'utilisez pas la console IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Créez le rôle IAM qui utilise cette politique. Notez la valeur `Arn` qui est renvoyée par la commande ; vous en aurez besoin ultérieurement dans cette procédure. Dans cet exemple, nous utilisons `CWLtoSubscriptionFilterRole` pour connaître le nom du rôle que nous créons.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte.

   a. Utilisez d'abord un éditeur de texte pour créer la politique d'autorisations suivante dans un fichier nommé `~/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Saisissez la commande suivante pour associer la politique d'autorisations que vous venez de créer au rôle que vous avez créé à l'étape 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Lorsque vous aurez terminé, vous pouvez passer à Étape 4 : créer un filtre d'abonnement.

Étape 3 : autorisations Add/validate IAM pour la destination multi-comptes

Selon la logique d'évaluation des politiques AWS entre comptes, pour accéder à toute ressource intercomptes (telle qu'un flux Kinesis ou Firehose utilisé comme destination pour un filtre d'abonnement), vous devez disposer d'une politique basée sur l'identité dans le compte d'envoi qui fournit un accès explicite à la ressource de destination entre comptes. Pour plus d'informations sur la logique d'évaluation des politiques, consultez la section Logique d'évaluation des politiques entre comptes.

Vous pouvez attacher la politique basée sur l'identité au rôle IAM ou à l'utilisateur IAM que vous utilisez pour créer le filtre d'abonnement. Cette politique doit être présente dans le compte de l'expéditeur. Si vous utilisez le rôle d'administrateur pour créer le filtre d'abonnement, vous pouvez ignorer cette étape et passer à Étape 4 : créer un filtre d'abonnement.

Pour ajouter ou valider les autorisations IAM nécessaires pour les opérations entre comptes

1. Saisissez la commande suivante pour vérifier quel rôle IAM ou quel utilisateur IAM est utilisé pour exécuter les commandes de journalisation AWS .

```
aws sts get-caller-identity
```

La commande renvoie un résultat semblable à ce qui suit :

```
{
"UserId": "User ID",
"Account": "sending account id",
"Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
}
```

Prenez note de la valeur représentée par *RoleName* ou*UserName*.

2. Connectez-vous AWS Management Console au compte expéditeur et recherchez les politiques associées avec le rôle IAM ou l'utilisateur IAM renvoyé dans le résultat de la commande que vous avez saisie à l'étape 1.
3. Vérifiez que les politiques associées à ce rôle ou à cet utilisateur fournissent des autorisations explicites pour appeler `logs:PutSubscriptionFilter` au niveau de la ressource de destination entre comptes. L'exemple de politique suivant présente les autorisations recommandées.

   La politique suivante autorise la création d'un filtre d'abonnement sur n'importe quelle ressource de destination uniquement dans un seul AWS compte, un compte `123456789012` :

   JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
        {
            "Sid": "AllowSubscriptionFiltersOnAccountResources",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

   La politique suivante autorise la création d'un filtre d'abonnement uniquement sur une ressource de destination spécifique nommée `sampleDestination` dans un AWS compte unique, un compte `123456789012` :

   JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in
 one specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

Étape 4 : créer un filtre d'abonnement

Une fois que vous avez créé une destination, le compte du destinataire des données du journal peut partager l'ARN de destination (arn:aws:logs:us-east-1:999999999999:destination:testDestination) avec d'autres comptes AWS afin qu'ils puissent envoyer des événements du journal vers la même destination. Les utilisateurs des autres comptes d'expédition créent ensuite un filtre d'abonnement sur leurs groupes de journaux respectifs par rapport à cette destination. Ce filtre d'abonnement lance immédiatement la transmission de données du journal en temps réel à partir du groupe de journaux choisi vers la destination spécifiée.

> **Note**
>
> Si vous accordez des autorisations pour le filtre d'abonnement à l'ensemble d'une organisation, vous devez utiliser l'ARN du rôle IAM que vous avez créé dans Étape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM.

Dans l'exemple suivant, un filtre d'abonnement est créé dans un compte d'envoi. Le filtre est associé à un groupe de journaux contenant des AWS CloudTrail événements afin que chaque activité enregistrée à l'aide des AWS informations d'identification « root » soit transmise à la destination que vous avez créée précédemment. Cette destination encapsule un flux appelé « RecipientStream ».

Les autres étapes décrites dans les sections suivantes supposent que vous avez suivi les instructions de la section Envoyer des CloudTrail événements aux CloudWatch journaux dans le guide de

l'AWS CloudTrail utilisateur et que vous avez créé un groupe de journaux contenant vos CloudTrail événements. Ces étapes supposent que le nom de ce groupe de journaux est `CloudTrail/logs`.

Lorsque vous saisissez la commande suivante, assurez-vous d'être connecté en tant qu'utilisateur IAM ou que vous utilisez le rôle IAM pour lequel vous avez ajouté la politique, dans Étape 3 : autorisations Add/validate IAM pour la destination multi-comptes.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Le groupe de journaux et la destination doivent se trouver dans la même AWS région. Toutefois, la destination peut pointer vers une AWS ressource telle qu'un flux Kinesis Data Streams situé dans une autre région.

Validation du flux des événements des journaux

Après avoir créé le filtre d'abonnement, CloudWatch Logs transmet tous les événements de journal entrants correspondant au modèle de filtre au flux encapsulé dans le flux de destination appelé « RecipientStream». Le propriétaire de la destination peut vérifier que cela se produit en utilisant la get-shard-iterator commande aws kinesis pour récupérer une partition Kinesis Data Streams, et en utilisant la commande aws kinesis get-records pour récupérer certains enregistrements Kinesis Data Streams :

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
```

```
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Vous devrez peut-être réexécuter plusieurs fois la commande get-records avant que Kinesis Data Streams ne renvoie des données.

Vous devriez voir une réponse comportant un tableau d'enregistrements Kinesis Data Streams. L'attribut de données dans le registre Kinesis Data Streams est compressé au format gzip, puis codé en base64. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen de la commande Unix suivante :

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Les données codées et décompressées en base64 sont formatées au format JSON avec la structure suivante :

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
```

```
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Les éléments clés de cette structure de données sont les suivants :

owner

L'ID de AWS compte des données du journal d'origine.

logGroup

Le nom du groupe de journaux des données du journal source.

logStream

Le nom du flux de journaux des données du journal source.

subscriptionFilters

La liste des noms de filtres d'abonnements qui correspondaient aux données du journal source.

messageType

Les données de messages utilisent le type « DATA_MESSAGE ». Parfois, CloudWatch Logs peut émettre des enregistrements Kinesis Data Streams de type « CONTROL_MESSAGE », principalement pour vérifier si la destination est accessible.

logEvents

Les données du journal réelles, représentées sous la forme d'un tableau d'enregistrements d'événements du journal. La propriété ID est un identifiant unique pour chaque événement de journal.

Modification de l'appartenance à une destination au moment de l'exécution

Vous pouvez rencontrer des situations où vous devez ajouter ou supprimer l'adhésion de certains utilisateurs pour l'une de vos destinations. Vous pouvez utiliser la commande `put-destination-policy` sur votre destination avec une nouvelle stratégie d'accès. Dans l'exemple suivant, un compte

précédemment ajouté 111111111111 est écarté de l'envoi de données du journal supplémentaires, et le compte 222222222222 est activé.

1. Récupérez la politique actuellement associée à la destination TestDestination et notez : AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Mettez à jour la politique de façon à refléter l'arrêt du compte 111111111111 et l'activation du compte 222222222222. Insérez cette politique dans le fichier ~/ NewAccessPolicy .json :

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "222222222222"
            },
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
```

```
        }
    ]
}
```

3. Appelez PutDestinationPolicypour associer la politique définie dans le fichier NewAccessPolicy.json à la destination :

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Cela finira par désactiver les événements du journal à partir de l'ID de compte 111111111111. Les événements du journal de l'ID de compte 222222222222 commencent à passer à la destination dès que le propriétaire du compte 222222222222 crée un filtre d'abonnement.

Mise à jour d'un abonnement existant entre comptes

Si vous disposez actuellement d'un abonnement aux journaux entre comptes où le compte de destination n'accorde des autorisations qu'à des comptes d'expéditeur spécifiques et que vous souhaitez mettre à jour cet abonnement, afin que le compte de destination accorde l'accès à tous les comptes d'une organisation, suivez les étapes décrites dans cette section.

Rubriques

- Étape 1 : mettre à jour les filtres d'abonnement
- Étape 2 : mettre à jour la stratégie d'accès de la destination existante

Étape 1 : mettre à jour les filtres d'abonnement

> **Note**
>
> Cette étape est nécessaire uniquement pour les abonnements entre comptes pour les journaux créés par les services répertoriés dans Activer la journalisation à partir AWS des services. Si vous ne travaillez pas avec les journaux créés par l'un de ces groupes de journaux, vous pouvez passer à Étape 2 : mettre à jour la stratégie d'accès de la destination existante.

Dans certains cas, vous devez mettre à jour les filtres d'abonnement dans tous les comptes d'expéditeur qui envoient des journaux au compte de destination. La mise à jour ajoute un rôle IAM, qui CloudWatch peut supposer et valider que le compte expéditeur est autorisé à envoyer des journaux au compte destinataire.

Suivez les étapes de cette section pour chaque compte d'expéditeur que vous souhaitez mettre à jour, afin qu'il utilise l'ID d'organisation pour les autorisations d'abonnement entre comptes.

Dans les exemples présentés dans cette section, deux comptes, `111111111111` et `222222222222`, disposent de filtres d'abonnement pour envoyer des journaux au compte `999999999999`. Les valeurs de filtre d'abonnement existantes sont les suivantes :

```
## Existing Subscription Filter parameter values
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Si vous devez rechercher les valeurs actuelles des paramètres de filtre d'abonnement, saisissez la commande suivante.

```
aws logs describe-subscription-filters
    \ --log-group-name "my-log-group-name"
```

Pour mettre à jour un filtre d'abonnement afin de commencer à utiliser l'organisation IDs pour les autorisations de journalisation entre comptes

1. Créez la politique de confiance suivante dans un fichier `~/TrustPolicyForCWL.json`. Utilisez un éditeur de texte pour créer ce fichier de politique, n'utilisez pas la console IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Créez le rôle IAM qui utilise cette politique. Notez la valeur `Arn` de la valeur `Arn` qui est renvoyée par la commande ; vous en aurez besoin ultérieurement dans cette procédure. Dans

cet exemple, nous utilisons `CWLtoSubscriptionFilterRole` pour connaître le nom du rôle que nous créons.

```
aws iam create-role
    \ --role-name CWLtoSubscriptionFilterRole
    \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

3. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte.

   a. Utilisez d'abord un éditeur de texte pour créer la politique d'autorisations suivante dans un fichier nommé `/PermissionsForCWLSubscriptionFilter.json`.

   ```
   {
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "logs:PutLogEvents",
               "Resource": "arn:aws:logs:region:111111111111:log-
   group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
           }
       ]
   }
   ```

   b. Saisissez la commande suivante pour associer la politique d'autorisations que vous venez de créer au rôle que vous avez créé à l'étape 2.

   ```
   aws iam put-role-policy
       --role-name CWLtoSubscriptionFilterRole
       --policy-name Permissions-Policy-For-CWL-Subscription-filter
       --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
   ```

4. Saisissez la commande suivante pour mettre à jour le filtre d'abonnement.

```
aws logs put-subscription-filter
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn
 "arn:aws:logs:region:999999999999:destination:testDestination"
    \ --role-arn "arn:aws:iam::111111111111:role/CWLtoSubscriptionFilterRole"
```

Étape 2 : mettre à jour la stratégie d'accès de la destination existante

Une fois que vous avez mis à jour les filtres d'abonnement dans tous les comptes d'expéditeur, vous pouvez mettre à jour la stratégie d'accès de la destination dans le compte du destinataire.

Dans les exemples suivants, le compte du destinataire est `999999999999` et la destination est nommée `testDestination`.

La mise à jour permet à tous les comptes qui font partie de l'organisation ayant un ID `o-1234567890` d'envoyer des journaux au compte du destinataire. Seuls les comptes avec des filtres d'abonnement envoient des journaux au compte du destinataire.

Pour mettre à jour la stratégie d'accès de la destination dans le compte du destinataire, afin qu'il commence à utiliser un ID d'organisation pour les autorisations

1. Dans le compte du destinataire, utilisez un éditeur de texte pour créer un fichier `~/AccessPolicy.json` avec le contenu suivant.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

2. Saisissez la commande suivante pour attacher la politique que vous venez de créer à la destination existante. Pour mettre à jour une destination afin d'utiliser une politique d'accès avec un identifiant d'organisation au lieu d'une politique d'accès répertoriant un AWS compte spécifique IDs, incluez le `force` paramètre.

> **Warning**
>
> Si vous travaillez avec des journaux envoyés par un AWS service répertorié dansActiver la journalisation à partir AWS des services, avant de procéder à cette étape, vous devez d'abord avoir mis à jour les filtres d'abonnement dans tous les comptes d'expéditeurs, comme expliqué dansÉtape 1 : mettre à jour les filtres d'abonnement.

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Partage de données de journal entre comptes et entre régions à l'aide de Firehose

Pour partager des données du journal entre comptes, vous devez spécifier un expéditeur et un récepteur :

- Expéditeur des données de journal : obtient les informations de destination auprès du destinataire et CloudWatch indique à Logs qu'il est prêt à envoyer ses événements de journal à la destination spécifiée. Dans les procédures décrites dans le reste de cette section, l'expéditeur des données du journal est indiqué avec un numéro de AWS compte fictif 111111111111.
- Destinataire des données du journal : définit une destination qui encapsule un flux Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir les données du journal. Le destinataire partage ensuite les informations sur cette destination avec l'expéditeur. Dans les procédures décrites dans le reste de cette section, le destinataire des données du journal est indiqué avec un numéro de AWS compte fictif 222222222222.

L'exemple présenté dans cette section utilise un flux de diffusion Firehose avec un espace de stockage Amazon S3. Vous pouvez également configurer les flux de diffusion Firehose avec différents paramètres. Pour plus d'informations, consultez Création d'un flux de diffusion Firehose.

> **Note**
>
> Le groupe de journaux et la destination doivent se trouver dans la même AWS région. Par contre, la ressource AWS vers laquelle pointe la destination peut être située dans une autre région.

> **Note**
>
> Le filtre d'abonnement Firehose pour un même compte et un flux de diffusion interrégional sont pris en charge.

Rubriques

- Étape 1 : créer un flux de diffusion Firehose
- Étape 2 : créer une destination
- Étape 3 : autorisations Add/validate IAM pour la destination multi-comptes
- Étape 4 : créer un filtre d'abonnement
- Validation du flux des événements de journaux
- Modification de l'abonnement à la destination au moment de l'exécution

Étape 1 : créer un flux de diffusion Firehose

> **Important**
>
> Avant d'effectuer les étapes suivantes, vous devez utiliser une politique d'accès afin que Firehose puisse accéder à votre compartiment Amazon S3. Pour plus d'informations, consultez Controlling Access dans le manuel Amazon Data Firehose Developer Guide. Toutes les étapes de cette section (Étape 1) doivent être réalisées dans le compte du destinataire des données du journal.

USA Est (Virginie du Nord) est utilisé dans les exemples de commandes suivants. Remplacez la région par la celle appropriée pour votre déploiement.

Pour créer un flux de diffusion Firehose à utiliser comme destination

1. Créez un compartiment Amazon S3 :

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-1
```

2. Créez le rôle IAM qui autorise Firehose à placer des données dans le bucket.

   a. D'abord, utilisez un éditeur de texte pour créer une politique d'approbation dans un fichier `~/TrustPolicyForFirehose.json`.

   ```
   { "Statement": { "Effect": "Allow", "Principal": { "Service":
    "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
    { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
   ```

   b. Créez le rôle IAM, en spécifiant le fichier de politique d'approbation que vous venez de créer.

   ```
   aws iam create-role \
       --role-name FirehosetoS3Role \
       --assume-role-policy-document file://~/TrustPolicyForFirehose.json
   ```

   c. La sortie de cette commande ressemblera à ce qui suit. Notez le nom du rôle et l'ARN du rôle.

   ```
   {
       "Role": {
           "Path": "/",
           "RoleName": "FirehosetoS3Role",
           "RoleId": "AROAR3BXASEKW7K635M53",
           "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
           "CreateDate": "2021-02-02T07:53:10+00:00",
           "AssumeRolePolicyDocument": {
               "Statement": {
                   "Effect": "Allow",
                   "Principal": {
   ```

```
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Créez une politique d'autorisation pour définir les actions que Firehose peut effectuer sur votre compte.

   a. Utilisez d'abord un éditeur de texte pour créer la politique d'autorisations suivante dans un fichier nommé `~/PermissionsForFirehose.json`. Selon votre cas d'utilisation, il se peut que vous deviez ajouter des autorisations supplémentaires à ce fichier.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
    }]
}
```

   b. Saisissez la commande suivante pour associer la politique d'autorisation que vous venez de créer au rôle IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Entrez la commande suivante pour créer le flux de diffusion Firehose. Remplacez *my-role-arn* et *amzn-s3-demo-bucket2-arn* par les valeurs correctes pour votre déploiement.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket"}'
```

La sortie doit ressembler à ce qui suit :

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Étape 2 : créer une destination

> **Important**
>
> Toutes les étapes de cette procédure doivent être réalisées dans le compte du destinataire des données du journal.

Lorsque la destination est créée, CloudWatch Logs envoie un message test à la destination au nom du compte du destinataire. Lorsque le filtre d'abonnement est activé ultérieurement, CloudWatch Logs envoie les événements du journal à la destination au nom du compte source.

Pour créer une destination

1. Attendez que le stream Firehose que vous avez créé soit actifÉtape 1 : créer un flux de diffusion Firehose. Vous pouvez utiliser la commande suivante pour vérifier le StreamDescription. StreamStatuspropriété.

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
```

En outre, prenez note du DeliveryStreamDescription. DeliveryStreamValeur ARN, car vous devrez l'utiliser ultérieurement. Exemple de résultat de cette commande :

```
{
    "DeliveryStreamDescription": {
        "DeliveryStreamName": "my-delivery-stream",
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:222222222222:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamEncryptionConfiguration": {
            "Status": "DISABLED"
        },
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
```

```
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Votre flux de diffusion peut prendre une ou deux minutes avant d'apparaître avec le statut actif.

2. Lorsque le flux de diffusion est actif, créez le rôle IAM qui autorisera CloudWatch Logs à insérer des données dans votre flux Firehose. Tout d'abord, vous devez créer une politique de confiance dans un fichier ~/ TrustPolicyFor CWL.json. Utilisez un éditeur de texte créer cette stratégie. Pour plus d'informations sur les points de terminaison de CloudWatch Logs, consultez la section Points de terminaison et quotas Amazon CloudWatch Logs.

   Cette politique comprend une clé de contexte de condition globale `aws:SourceArn` qui spécifie le `sourceAccountId` pour aider à prévenir le problème de sécurité du député confus. Si vous ne connaissez pas encore l'ID du compte source lors du premier appel, nous vous recommandons de placer l'ARN de destination dans le champ ARN source. Dans les appels suivants, vous devez définir l'ARN source comme ARN source réel que vous avez recueilli lors du premier appel. Pour plus d'informations, consultez Prévention du député confus.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.region.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
     }
}
```

3. Utilisez la commande aws iam create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation que vous venez de créer.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

Voici un exemple de sortie. Prenez note de la valeur `Role.Arn` retournée, car vous devrez l'utiliser dans une étape ultérieure.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2021-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. Utilisez d'abord un éditeur de texte pour créer une politique d'autorisations dans un fichier ~/ PermissionsFor CWL.json :

```
{
```

```
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Associez la politique d'autorisations au rôle en entrant la commande suivante :

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Une fois que le flux de diffusion Firehose est actif et que vous avez créé le rôle IAM, vous pouvez créer la CloudWatch destination Logs.

   a. Cette étape n'associera pas de stratégie d'accès à votre destination. Il s'agit uniquement de la première de deux étapes pour créer une destination. Notez l'ARN de la nouvelle destination qui est renvoyé dans la charge utile, car vous l'utiliserez comme `destination.arn` dans une étape ultérieure.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

   b. Une fois l'étape précédente terminée, dans le compte des données du journal du destinataire (222222222222), associez une stratégie d'accès à la destination.

Cette politique permet au compte de l'expéditeur des données du journal (111111111111) d'accéder à la destination dans le compte du destinataire des données du journal (222222222222). Vous pouvez utiliser un éditeur de texte pour mettre cette politique dans le fichier ~/ AccessPolicy .json :

JSON

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. Cela crée une stratégie définissant qui a accès en écriture à la destination. Cette politique doit spécifier les logs : PutSubscriptionFilter action pour accéder à la destination. Les utilisateurs multicomptes utiliseront l'PutSubscriptionFilteraction pour envoyer les événements du journal à la destination :

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Étape 3 : autorisations Add/validate IAM pour la destination multi-comptes

Selon la logique d'évaluation des politiques AWS entre comptes, pour accéder à toute ressource intercomptes (telle qu'un flux Kinesis ou Firehose utilisé comme destination pour un filtre d'abonnement), vous devez disposer d'une politique basée sur l'identité dans le compte d'envoi qui fournit un accès explicite à la ressource de destination entre comptes. Pour plus d'informations sur

la logique d'évaluation des politiques, consultez la section Logique d'évaluation des politiques entre comptes.

Vous pouvez attacher la politique basée sur l'identité au rôle IAM ou à l'utilisateur IAM que vous utilisez pour créer le filtre d'abonnement. Cette politique doit être présente dans le compte de l'expéditeur. Si vous utilisez le rôle d'administrateur pour créer le filtre d'abonnement, vous pouvez ignorer cette étape et passer à Étape 4 : créer un filtre d'abonnement.

Pour ajouter ou valider les autorisations IAM nécessaires pour les opérations entre comptes

1. Saisissez la commande suivante pour vérifier quel rôle IAM ou quel utilisateur IAM est utilisé pour exécuter les commandes de journalisation AWS .

   ```
   aws sts get-caller-identity
   ```

   La commande renvoie un résultat semblable à ce qui suit :

   ```
   {
   "UserId": "User ID",
   "Account": "sending account id",
   "Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
   }
   ```

   Prenez note de la valeur représentée par `RoleName` ou`UserName`.

2. Connectez-vous AWS Management Console au compte expéditeur et recherchez les politiques associées avec le rôle IAM ou l'utilisateur IAM renvoyé dans le résultat de la commande que vous avez saisie à l'étape 1.

3. Vérifiez que les politiques associées à ce rôle ou à cet utilisateur fournissent des autorisations explicites pour appeler `logs:PutSubscriptionFilter` au niveau de la ressource de destination entre comptes. L'exemple de politique suivant présente les autorisations recommandées.

   La politique suivante autorise la création d'un filtre d'abonnement sur n'importe quelle ressource de destination uniquement dans un seul AWS compte, un compte `123456789012` :

   JSON

   ```
   {
   ```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid":
 "AllowSubscriptionFiltersOnAnyResourceInOneSpecificAccount",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

La politique suivante autorise la création d'un filtre d'abonnement uniquement sur une ressource de destination spécifique nommée `sampleDestination` dans un AWS compte unique, un compte `123456789012` :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Sid": "AllowSubscriptionFiltersOnSpecificResource",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

Étape 4 : créer un filtre d'abonnement

Passez au compte d'envoi, qui est 111111111111 dans cet exemple. Vous allez maintenant créer le filtre d'abonnement dans le compte d'envoi. Dans cet exemple, le filtre est associé à un groupe de journaux contenant des AWS CloudTrail événements afin que chaque activité enregistrée avec les AWS informations d'identification « root » soit transmise à la destination que vous avez créée précédemment. Pour plus d'informations sur la façon d'envoyer AWS CloudTrail des événements aux CloudWatch journaux, consultez la section Envoyer CloudTrail des événements aux CloudWatch journaux dans le guide de AWS CloudTrail l'utilisateur.

Lorsque vous saisissez la commande suivante, assurez-vous d'être connecté en tant qu'utilisateur IAM ou que vous utilisez le rôle IAM pour lequel vous avez ajouté la politique, dans Étape 3 : autorisations Add/validate IAM pour la destination multi-comptes.

```
aws logs put-subscription-filter \
    --log-group-name "aws-cloudtrail-logs-111111111111-300a971e" \
    --filter-name "firehose_test" \
    --filter-pattern "{$.userIdentity.type = AssumedRole}" \
    --destination-arn "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
```

Le groupe de journaux et la destination doivent se trouver dans la même AWS région. Cependant, la destination peut pointer vers une AWS ressource telle qu'un flux Firehose situé dans une autre région.

Validation du flux des événements de journaux

Une fois que vous avez créé le filtre d'abonnement, CloudWatch Logs transmet tous les événements de journal entrants correspondant au modèle de filtre au flux de diffusion Firehose. Les données commencent à apparaître dans votre compartiment Amazon S3 en fonction de l'intervalle de temps défini dans le flux de diffusion Firehose. Après un délai suffisant, vous pouvez consulter le compartiment Amazon S3 pour vérifier vos données. Pour vérifier le compartiment, saisissez la commande suivante :

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket'
```

La sortie de cette commande sera similaire à ce qui suit :

```
{
```

```
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2021-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

Vous pouvez ensuite récupérer un objet spécifique du compartiment en entrant la commande suivante. Remplacez la valeur de `key` par la valeur que vous avez trouvée dans la commande précédente.

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket' --key '2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Les données dans l'objet Amazon S3 sont comprimées au format gzip. Vous pouvez examiner les données brutes à partir de la ligne de commande en utilisant l'une des commandes suivantes :

Linux :

```
zcat testfile.gz
```

macOS :

```
zcat <testfile.gz
```

Modification de l'abonnement à la destination au moment de l'exécution

Vous pouvez rencontrer des situations où vous devez ajouter ou supprimer des expéditeurs de journaux d'une destination qui vous appartient. Vous pouvez utiliser l'PutDestinationPolicyaction sur votre destination avec une nouvelle politique d'accès. Dans l'exemple suivant, un compte

précédemment ajouté 111111111111 est écarté de l'envoi de données du journal supplémentaires, et le compte 333333333333 est activé.

1. Récupérez la politique actuellement associée à la destination TestDestination et notez : AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testFirehoseDestination"

{
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Mettez à jour la politique de façon à refléter l'arrêt du compte 111111111111 et l'activation du compte 333333333333. Insérez cette politique dans le fichier ~/ NewAccessPolicy .json :

   JSON

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
```

```
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Utilisez la commande suivante pour associer la politique définie dans le fichier NewAccessPolicy.json à la destination :

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

Cela désactive éventuellement les événements du journal à partir de l'ID de compte 111111111111. Les événements du journal de l'ID de compte 333333333333 commencent à passer à la destination dès que le propriétaire du compte 333333333333 crée un filtre d'abonnement.

Abonnements entre comptes et entre régions à l'aide de Kinesis Data Streams

Lorsque vous créez un abonnement entre comptes, vous pouvez spécifier un seul compte ou une organisation comme expéditeur. Si vous spécifiez une organisation, cette procédure permet à tous les comptes de l'organisation d'envoyer des journaux au compte récepteur.

Pour partager des données du journal entre comptes, vous devez spécifier un expéditeur et un récepteur :

- Expéditeur des données de journal : obtient les informations de destination auprès du destinataire et CloudWatch indique à Logs qu'il est prêt à envoyer ses événements de journal à la destination spécifiée. Dans les procédures décrites dans le reste de cette section, l'expéditeur des données du journal est indiqué avec un numéro de AWS compte fictif 111111111111.

  Si plusieurs comptes dans une organisation doivent envoyer des journaux au compte d'un destinataire, vous pouvez créer une politique qui accorde à tous les comptes de l'organisation l'autorisation d'envoyer des journaux au compte du destinataire. Vous devez toujours configurer des filtres d'abonnement distincts pour chaque compte d'expéditeur.

- Destinataire des données du journal : définit une destination qui encapsule un flux Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir les données du journal. Le destinataire partage ensuite les informations sur cette destination avec l'expéditeur. Dans les procédures décrites dans le reste de cette section, le destinataire des données du journal est indiqué avec un numéro de AWS compte fictif 999999999999.

Pour commencer à recevoir des événements de journal provenant d'utilisateurs multicomptes, le destinataire des données de journal crée d'abord une destination de CloudWatch journaux. Chaque destination comprend les éléments clés suivants :

Nom de destination

Le nom de la destination que vous souhaitez créer.

ARN cible

Le nom de ressource Amazon (ARN) de la AWS ressource que vous souhaitez utiliser comme destination du flux d'abonnement.

Role ARN (ARN de rôle)

Rôle AWS Identity and Access Management (IAM) qui accorde à CloudWatch Logs les autorisations nécessaires pour placer des données dans le flux choisi.

Stratégie d'accès

Un document de politique IAM (au format JSON, écrit à l'aide de la syntaxe des politiques IAM) régissant l'ensemble des utilisateurs qui sont autorisés à écrire à votre destination.

> **Note**
>
> Le groupe de journaux et la destination doivent se trouver dans la même AWS région. Par contre, la ressource AWS vers laquelle pointe la destination peut être située dans une autre région. Dans les exemples des sections suivantes, toutes les ressources spécifiques à la région sont créées dans USA Est (Virginie du Nord).

Rubriques

- Configurer un nouvel abonnement entre comptes
- Mise à jour d'un abonnement existant entre comptes

Configurer un nouvel abonnement entre comptes

Suivez les étapes décrites dans ces sections pour configurer un nouvel abonnement au journal entre comptes.

Rubriques

- Étape 1 : créer une destination
- Étape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM
- Étape 3 : Création d'une politique de filtrage des abonnements au niveau du compte
- Validation du flux des événements des journaux
- Modification de l'appartenance à une destination au moment de l'exécution

Étape 1 : créer une destination

> **Important**
>
> Toutes les étapes de cette procédure doivent être réalisées dans le compte du destinataire des données du journal.

Dans cet exemple, le compte du destinataire des données du journal a un ID de compte 999999999999, tandis que l'identifiant du AWS compte de l'expéditeur AWS des données du journal est 111111111111.

Cet exemple crée une destination à l'aide d'un flux Kinesis Data Streams RecipientStream appelé, et d'un rôle qui CloudWatch permet à Logs d'y écrire des données.

Lorsque la destination est créée, CloudWatch Logs envoie un message test à la destination au nom du compte du destinataire. Lorsque le filtre d'abonnement est activé ultérieurement, CloudWatch Logs envoie les événements du journal à la destination au nom du compte source.

Pour créer une destination

1. Dans le compte du destinataire, créez un flux de diffusion de destination dans Kinesis Data Streams. À l'invite de commande, saisissez :

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Patientez jusqu'à ce que le flux devienne actif. Vous pouvez utiliser la commande aws kinesis describe-stream pour vérifier le. StreamDescription StreamStatuspropriété. Prenez également note de la valeur StreamDescription.StreamArn, car vous la transmettrez ultérieurement à CloudWatch Logs :

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768211EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702181EXAMPLE"
        }
      }
    ]
  }
}
```

Votre flux de données peut prendre une ou deux minutes avant d'apparaître avec le statut actif.

3. Créez le rôle IAM qui autorise CloudWatch Logs à insérer des données dans votre flux. Tout d'abord, vous devez créer une politique de confiance dans un fichier ~/ TrustPolicyFor CWL.json. Utilisez un éditeur de texte pour créer ce fichier de politique, n'utilisez pas la console IAM.

   Cette politique comprend une clé de contexte de condition globale `aws:SourceArn` qui spécifie le `sourceAccountId` pour aider à prévenir le problème de sécurité du député confus. Si vous ne connaissez pas encore l'ID du compte source lors du premier appel, nous vous recommandons de placer l'ARN de destination dans le champ ARN source. Dans les appels suivants, vous devez définir l'ARN source comme ARN source réel que vous avez recueilli lors du premier appel. Pour plus d'informations, consultez Prévention du député confus.

```
{
```

```
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. Utilisez la commande aws iam create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation. Prenez note de la valeur Role.Arn renvoyée, car elle sera également transmise à CloudWatch Logs ultérieurement :

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
```

```
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. Utilisez d'abord un éditeur de texte pour créer une politique d'autorisations dans un fichier ~/ PermissionsFor CWL.json :

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Associez la politique d'autorisation au rôle à l'aide de la put-role-policy commande aws iam :

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Une fois que le flux est actif et que vous avez créé le rôle IAM, vous pouvez créer la destination CloudWatch Logs.

   a. Cette étape n'associe aucune stratégie d'accès à votre destination. Il s'agit uniquement de la première de deux étapes pour créer une destination. Notez DestinationArnce qui est renvoyé dans la charge utile :

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
```

```
{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

b. Une fois l'étape 7a terminée, dans le compte des données du journal du destinataire, associez une stratégie d'accès à la destination. Cette politique doit spécifier l'PutSubscriptionFilteraction logs : et autoriser le compte expéditeur à accéder à la destination.

   La politique accorde l'autorisation au AWS compte qui envoie les journaux. Vous pouvez spécifier uniquement ce compte dans la politique. Si le compte de l'expéditeur est membre d'une organisation, la politique peut également spécifier l'ID de l'organisation. De cette façon, vous pouvez créer une seule politique pour autoriser plusieurs comptes d'une organisation à envoyer des journaux à ce compte de destination.

   Utilisez un éditeur de texte pour créer un fichier nommé `~/AccessPolicy.json` avec l'une des déclarations de politique suivantes.

   Ce premier exemple de politique autorise tous les comptes de l'organisation possédant un ID `o-1234567890` à envoyer les journaux au compte du destinataire.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

Cet exemple suivant permet uniquement au compte des données du journal de l'expéditeur (111111111111) d'envoyer des journaux au compte du destinataire des données du journal.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111111111111"
            },
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
        }
    ]
}
```

c. Associez la politique que vous avez créée à l'étape précédente à la destination.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Cette politique d'accès permet aux utilisateurs du AWS compte ayant l'ID 111111111111 d'appeler la destination avec l'ARN arn:aws:logs PutSubscriptionFilter::999999999999999999999:destination:testDestination. *region* Toute tentative d' PutSubscriptionFilter appel de la part d'un autre utilisateur vers cette destination sera rejetée.

Pour valider les privilèges d'un utilisateur par rapport à une stratégie d'accès, consultez Utilisation du programme de validation de politiques dans le Guide de l'utilisateur IAM.

Lorsque vous avez terminé, si vous utilisez AWS Organizations pour vos autorisations multi-comptes, suivez les étapes décrites dansÉtape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM. Si vous accordez des autorisations directement à l'autre compte au lieu d'utiliser des organisations, vous pouvez ignorer cette étape et passer à Étape 3 : Création d'une politique de filtrage des abonnements au niveau du compte.

Étape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM

Dans la section précédente, si vous avez créé la destination à l'aide d'une stratégie d'accès qui accorde des autorisations à l'organisation dans laquelle ce compte `111111111111` se trouve, au lieu d'accorder des autorisations directement au compte `111111111111`, suivez les étapes de cette section. Sinon, vous pouvez passer à Étape 3 : Création d'une politique de filtrage des abonnements au niveau du compte.

Les étapes décrites dans cette section créent un rôle IAM, qui CloudWatch peut assumer et valider si le compte expéditeur est autorisé à créer un filtre d'abonnement par rapport à la destination du destinataire.

Suivez les étapes de cette section dans le compte de l'expéditeur. Le rôle doit exister dans le compte de l'expéditeur et vous devez spécifier l'ARN de ce rôle dans le filtre d'abonnement. Dans cet exemple, le compte de l'expéditeur est `111111111111`.

Pour créer le rôle IAM nécessaire aux abonnements aux journaux entre comptes à l'aide de AWS Organizations

1. Créez la politique de confiance suivante dans un fichier `/TrustPolicyForCWLSubscriptionFilter.json`. Utilisez un éditeur de texte pour créer ce fichier de politique, n'utilisez pas la console IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Créez le rôle IAM qui utilise cette politique. Notez la valeur `Arn` qui est renvoyée par la commande ; vous en aurez besoin ultérieurement dans cette procédure. Dans cet exemple, nous utilisons `CWLtoSubscriptionFilterRole` pour connaître le nom du rôle que nous créons.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte.

   a. Utilisez d'abord un éditeur de texte pour créer la politique d'autorisations suivante dans un fichier nommé `~/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

   b. Saisissez la commande suivante pour associer la politique d'autorisations que vous venez de créer au rôle que vous avez créé à l'étape 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
```

```
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Lorsque vous aurez terminé, vous pouvez passer à Étape 3 : Création d'une politique de filtrage des abonnements au niveau du compte.

Étape 3 : Création d'une politique de filtrage des abonnements au niveau du compte

Une fois que vous avez créé une destination, le compte du destinataire des données du journal peut partager l'ARN de destination (arn:aws:logs:us-east-1:999999999999:destination:testDestination) avec d'autres comptes AWS afin qu'ils puissent envoyer des événements du journal vers la même destination. Les utilisateurs des autres comptes d'expédition créent ensuite un filtre d'abonnement sur leurs groupes de journaux respectifs par rapport à cette destination. Ce filtre d'abonnement lance immédiatement la transmission de données du journal en temps réel à partir du groupe de journaux choisi vers la destination spécifiée.

> **Note**
>
> Si vous accordez des autorisations pour le filtre d'abonnement à l'ensemble d'une organisation, vous devez utiliser l'ARN du rôle IAM que vous avez créé dans Étape 2 : (uniquement si vous utilisez une organisation) créer un rôle IAM.

Dans l'exemple suivant, une politique de filtrage d'abonnement au niveau du compte est créée dans un compte expéditeur. Le filtre est associé au compte expéditeur `111111111111` afin que chaque événement du journal correspondant au filtre et aux critères de sélection soit envoyé à la destination que vous avez créée précédemment. Cette destination encapsule un flux appelé « RecipientStream ».

Ce `selection-criteria` champ est facultatif, mais il est important pour exclure les groupes de journaux susceptibles de provoquer une récursivité infinie des journaux à partir d'un filtre d'abonnement. Pour plus d'informations sur ce problème et pour déterminer les groupes de journaux à exclure, consultezPrévention de la récursivité dans les journaux. NOT IN est actuellement le seul opérateur pris en charge pour`selection-criteria`.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
```

```
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Les groupes de journaux du compte expéditeur et la destination doivent se trouver dans la même AWS région. Toutefois, la destination peut pointer vers une AWS ressource telle qu'un flux Kinesis Data Streams situé dans une autre région.

Validation du flux des événements des journaux

Une fois que vous avez créé la politique de filtrage des abonnements au niveau du compte, CloudWatch Logs transmet tous les événements de journal entrants qui correspondent au modèle de filtre et aux critères de sélection au flux encapsulé dans le flux de destination appelé « ». RecipientStream Le propriétaire de la destination peut vérifier que cela se produit en utilisant la get-shard-iterator commande aws kinesis pour récupérer une partition Kinesis Data Streams, et en utilisant la commande aws kinesis get-records pour récupérer certains enregistrements Kinesis Data Streams :

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Vous devrez peut-être réexécuter la `get-records` commande plusieurs fois avant que Kinesis Data Streams ne commence à renvoyer des données.

Vous devriez voir une réponse comportant un tableau d'enregistrements Kinesis Data Streams. L'attribut de données dans le registre Kinesis Data Streams est compressé au format gzip, puis codé en base64. Vous pouvez examiner les données brutes à partir de la ligne de commande au moyen de la commande Unix suivante :

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Les données codées et décompressées en base64 sont formatées au format JSON avec la structure suivante :

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
```

```
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Les principaux éléments de la structure de données sont les suivants :

messageType

Les données de messages utiliseront le type « DATA_MESSAGE ». Parfois, CloudWatch Logs peut émettre des enregistrements Kinesis Data Streams de type « CONTROL_MESSAGE », principalement pour vérifier si la destination est accessible.

owner

L'ID de AWS compte des données du journal d'origine.

logGroup

Le nom du groupe de journaux des données du journal source.

logStream

Le nom du flux de journaux des données du journal source.

subscriptionFilters

La liste des noms de filtres d'abonnements qui correspondaient aux données du journal source.

logEvents

Les données du journal réelles, représentées sous la forme d'un tableau d'enregistrements d'événements du journal. La propriété « id » est un identifiant unique pour chaque événement de journal.

Niveau de la politique

Niveau auquel la politique a été appliquée. « ACCOUNT_LEVEL_POLICY » correspond à une politique de filtrage `policyLevel` d'abonnement au niveau du compte.

Modification de l'appartenance à une destination au moment de l'exécution

Vous pouvez rencontrer des situations où vous devez ajouter ou supprimer l'adhésion de certains utilisateurs pour l'une de vos destinations. Vous pouvez utiliser la commande `put-destination-`

`policy` sur votre destination avec une nouvelle stratégie d'accès. Dans l'exemple suivant, un compte précédemment ajouté 111111111111 est écarté de l'envoi de données du journal supplémentaires, et le compte 222222222222 est activé.

1. Récupérez la politique actuellement associée à la destination TestDestination et notez : AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Mettez à jour la politique de façon à refléter l'arrêt du compte 111111111111 et l'activation du compte 222222222222. Insérez cette politique dans le fichier ~/ NewAccessPolicy .json :

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "222222222222"
            },
            "Action": [
                "logs:PutSubscriptionFilter",
```

```
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
        }
    ]
}
```

3. Appelez PutDestinationPolicypour associer la politique définie dans le fichier NewAccessPolicy.json à la destination :

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Cela finira par désactiver les événements du journal à partir de l'ID de compte 111111111111. Les événements du journal de l'ID de compte 222222222222 commencent à passer à la destination dès que le propriétaire du compte 222222222222 crée un filtre d'abonnement.

Mise à jour d'un abonnement existant entre comptes

Si vous disposez actuellement d'un abonnement aux journaux entre comptes où le compte de destination n'accorde des autorisations qu'à des comptes d'expéditeur spécifiques et que vous souhaitez mettre à jour cet abonnement, afin que le compte de destination accorde l'accès à tous les comptes d'une organisation, suivez les étapes décrites dans cette section.

Rubriques

- Étape 1 : mettre à jour les filtres d'abonnement
- Étape 2 : mettre à jour la stratégie d'accès de la destination existante

Étape 1 : mettre à jour les filtres d'abonnement

> **Note**
>
> Cette étape est nécessaire uniquement pour les abonnements entre comptes pour les journaux créés par les services répertoriés dans Activer la journalisation à partir AWS des services. Si vous ne travaillez pas avec les journaux créés par l'un de ces groupes de

journaux, vous pouvez passer à Étape 2 : mettre à jour la stratégie d'accès de la destination existante.

Dans certains cas, vous devez mettre à jour les filtres d'abonnement dans tous les comptes d'expéditeur qui envoient des journaux au compte de destination. La mise à jour ajoute un rôle IAM, qui CloudWatch peut supposer et valider que le compte expéditeur est autorisé à envoyer des journaux au compte destinataire.

Suivez les étapes de cette section pour chaque compte d'expéditeur que vous souhaitez mettre à jour, afin qu'il utilise l'ID d'organisation pour les autorisations d'abonnement entre comptes.

Dans les exemples présentés dans cette section, deux comptes, `111111111111` et `222222222222`, disposent de filtres d'abonnement pour envoyer des journaux au compte `999999999999`. Les valeurs de filtre d'abonnement existantes sont les suivantes :

```
## Existing Subscription Filter parameter values
{
    "DestinationArn": "arn:aws:logs:region:999999999999:destination:testDestination",
    "FilterPattern": "{$.userIdentity.type = Root}",
    "Distribution": "Random"
}
```

Si vous devez rechercher les valeurs actuelles des paramètres de filtre d'abonnement, saisissez la commande suivante.

```
aws logs describe-account-policies \
--policy-type "SUBSCRIPTION_FILTER_POLICY" \
--policy-name "CrossAccountStreamsExamplePolicy"
```

Pour mettre à jour un filtre d'abonnement afin de commencer à utiliser l'organisation IDs pour les autorisations de journalisation entre comptes

1. Créez la politique de confiance suivante dans un fichier `~/TrustPolicyForCWL.json`. Utilisez un éditeur de texte pour créer ce fichier de politique, n'utilisez pas la console IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
   ```

```
    "Action": "sts:AssumeRole"
  }
}
```

2. Créez le rôle IAM qui utilise cette politique. Notez la valeur `Arn` de la valeur `Arn` qui est renvoyée par la commande ; vous en aurez besoin ultérieurement dans cette procédure. Dans cet exemple, nous utilisons `CWLtoSubscriptionFilterRole` pour connaître le nom du rôle que nous créons.

```
aws iam create-role
    \ --role-name CWLtoSubscriptionFilterRole
    \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

3. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte.

   a. Utilisez d'abord un éditeur de texte pour créer la politique d'autorisations suivante dans un fichier nommé `/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

   b. Saisissez la commande suivante pour associer la politique d'autorisations que vous venez de créer au rôle que vous avez créé à l'étape 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. Entrez la commande suivante pour mettre à jour la politique de filtrage des abonnements.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
```

```
   --policy-type "SUBSCRIPTION_FILTER_POLICY" \
   --policy-document
'{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
"FilterPattern": "{$.userIdentity.type = Root}", "Distribution": "Random"}' \
   --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
"LogGroupToExclude2"]' \
   --scope "ALL"
```

Étape 2 : mettre à jour la stratégie d'accès de la destination existante

Une fois que vous avez mis à jour les filtres d'abonnement dans tous les comptes d'expéditeur, vous pouvez mettre à jour la stratégie d'accès de la destination dans le compte du destinataire.

Dans les exemples suivants, le compte du destinataire est `999999999999` et la destination est nommée `testDestination`.

La mise à jour permet à tous les comptes qui font partie de l'organisation ayant un ID `o-1234567890` d'envoyer des journaux au compte du destinataire. Seuls les comptes avec des filtres d'abonnement envoient des journaux au compte du destinataire.

Pour mettre à jour la stratégie d'accès de la destination dans le compte du destinataire, afin qu'il commence à utiliser un ID d'organisation pour les autorisations

1. Dans le compte du destinataire, utilisez un éditeur de texte pour créer un fichier `~/AccessPolicy.json` avec le contenu suivant.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

2. Saisissez la commande suivante pour attacher la politique que vous venez de créer à la destination existante. Pour mettre à jour une destination afin d'utiliser une politique d'accès avec un identifiant d'organisation au lieu d'une politique d'accès répertoriant un AWS compte spécifique IDs, incluez le `force` paramètre.

> **Warning**
>
> Si vous travaillez avec des journaux envoyés par un AWS service répertorié dansActiver la journalisation à partir AWS des services, avant de procéder à cette étape, vous devez d'abord avoir mis à jour les filtres d'abonnement dans tous les comptes d'expéditeurs, comme expliqué dansÉtape 1 : mettre à jour les filtres d'abonnement.

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Abonnements entre comptes et entre régions à l'aide de Firehose

Pour partager des données du journal entre comptes, vous devez spécifier un expéditeur et un récepteur :

- Expéditeur des données de journal : obtient les informations de destination auprès du destinataire et CloudWatch indique à Logs qu'il est prêt à envoyer ses événements de journal à la destination spécifiée. Dans les procédures décrites dans le reste de cette section, l'expéditeur des données du journal est indiqué avec un numéro de AWS compte fictif 111111111111.

- Destinataire des données du journal : définit une destination qui encapsule un flux Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir les données du journal. Le destinataire partage ensuite les informations sur cette destination avec l'expéditeur. Dans les procédures décrites dans le reste de cette section, le destinataire des données du journal est indiqué avec un numéro de AWS compte fictif 222222222222.

L'exemple présenté dans cette section utilise un flux de diffusion Firehose avec un espace de stockage Amazon S3. Vous pouvez également configurer les flux de diffusion Firehose avec différents paramètres. Pour plus d'informations, consultez Création d'un flux de diffusion Firehose.

> **Note**
>
> Le groupe de journaux et la destination doivent se trouver dans la même AWS région. Par contre, la ressource AWS vers laquelle pointe la destination peut être située dans une autre région.

> **Note**
>
> Le filtre d'abonnement Firehose pour un même compte et un flux de diffusion interrégional sont pris en charge.

Rubriques

- Étape 1 : créer un flux de diffusion Firehose
- Étape 2 : créer une destination
- Étape 3 : Création d'une politique de filtrage des abonnements au niveau du compte
- Validation du flux des événements de journaux
- Modification de l'abonnement à la destination au moment de l'exécution

Étape 1 : créer un flux de diffusion Firehose

> **Important**
>
> Avant d'effectuer les étapes suivantes, vous devez utiliser une politique d'accès afin que Firehose puisse accéder à votre compartiment Amazon S3. Pour plus d'informations, consultez Controlling Access dans le manuel Amazon Data Firehose Developer Guide.
> Toutes les étapes de cette section (Étape 1) doivent être réalisées dans le compte du destinataire des données du journal.
> USA Est (Virginie du Nord) est utilisé dans les exemples de commandes suivants. Remplacez la région par la celle appropriée pour votre déploiement.

Pour créer un flux de diffusion Firehose à utiliser comme destination

1. Créez un compartiment Amazon S3 :

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-1
```

2. Créez le rôle IAM qui autorise Firehose à placer des données dans le bucket.

   a. D'abord, utilisez un éditeur de texte pour créer une politique d'approbation dans un fichier `~/TrustPolicyForFirehose.json`.

   ```
   { "Statement": { "Effect": "Allow", "Principal": { "Service":
    "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
    { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
   ```

   b. Créez le rôle IAM, en spécifiant le fichier de politique d'approbation que vous venez de créer.

   ```
   aws iam create-role \
       --role-name FirehosetoS3Role \
       --assume-role-policy-document file://~/TrustPolicyForFirehose.json
   ```

   c. La sortie de cette commande ressemblera à ce qui suit. Notez le nom du rôle et l'ARN du rôle.

   ```
   {
   ```

```
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Créez une politique d'autorisation pour définir les actions que Firehose peut effectuer sur votre compte.

   a. Utilisez d'abord un éditeur de texte pour créer la politique d'autorisations suivante dans un fichier nommé `~/PermissionsForFirehose.json`. Selon votre cas d'utilisation, il se peut que vous deviez ajouter des autorisations supplémentaires à ce fichier.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
    }]
```

```
}
```

b. Saisissez la commande suivante pour associer la politique d'autorisation que vous venez de créer au rôle IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Entrez la commande suivante pour créer le flux de diffusion Firehose. Remplacez *my-role-arn* et *amzn-s3-demo-bucket2-arn* par les valeurs correctes pour votre déploiement.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket"}'
```

La sortie doit ressembler à ce qui suit :

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Étape 2 : créer une destination

> **Important**
>
> Toutes les étapes de cette procédure doivent être réalisées dans le compte du destinataire des données du journal.

Lorsque la destination est créée, CloudWatch Logs envoie un message test à la destination au nom du compte du destinataire. Lorsque le filtre d'abonnement est activé ultérieurement, CloudWatch Logs envoie les événements du journal à la destination au nom du compte source.

Pour créer une destination

1. Attendez que le stream Firehose que vous avez créé soit actifÉtape 1 : créer un flux de diffusion Firehose. Vous pouvez utiliser la commande suivante pour vérifier le StreamDescription. StreamStatuspropriété.

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
```

En outre, prenez note du DeliveryStreamDescription. DeliveryStreamValeur ARN, car vous devrez l'utiliser ultérieurement. Exemple de résultat de cette commande :

```
{
    "DeliveryStreamDescription": {
        "DeliveryStreamName": "my-delivery-stream",
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:222222222222:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamEncryptionConfiguration": {
            "Status": "DISABLED"
        },
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
```

```
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Votre flux de diffusion peut prendre une ou deux minutes avant d'apparaître avec le statut actif.

2. Lorsque le flux de diffusion est actif, créez le rôle IAM qui autorisera CloudWatch Logs à insérer des données dans votre flux Firehose. Tout d'abord, vous devez créer une politique de confiance dans un fichier ~/ TrustPolicyFor CWL.json. Utilisez un éditeur de texte créer cette stratégie. Pour plus d'informations sur les points de terminaison de CloudWatch Logs, consultez la section Points de terminaison et quotas Amazon CloudWatch Logs.

   Cette politique comprend une clé de contexte de condition globale `aws:SourceArn` qui spécifie le `sourceAccountId` pour aider à prévenir le problème de sécurité du député confus. Si vous ne connaissez pas encore l'ID du compte source lors du premier appel, nous vous recommandons de placer l'ARN de destination dans le champ ARN source. Dans les appels suivants, vous devez définir l'ARN source comme ARN source réel que vous avez recueilli lors du premier appel. Pour plus d'informations, consultez Prévention du député confus.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
```

```
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
    }
}
```

3. Utilisez la commande aws iam create-role pour créer le rôle IAM, en spécifiant le fichier de politique d'approbation que vous venez de créer.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

Voici un exemple de sortie. Prenez note de la valeur `Role.Arn` retournée, car vous devrez l'utiliser dans une étape ultérieure.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2023-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
```

```
                    }
                }
            }
        }
    }
}
```

4. Créez une politique d'autorisation pour définir les actions que CloudWatch Logs peut effectuer sur votre compte. Utilisez d'abord un éditeur de texte pour créer une politique d'autorisations dans un fichier ~/ PermissionsFor CWL.json :

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Associez la politique d'autorisations au rôle en entrant la commande suivante :

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Une fois que le flux de diffusion Firehose est actif et que vous avez créé le rôle IAM, vous pouvez créer la CloudWatch destination Logs.

   a. Cette étape n'associera pas de stratégie d'accès à votre destination. Il s'agit uniquement de la première de deux étapes pour créer une destination. Notez l'ARN de la nouvelle destination qui est renvoyé dans la charge utile, car vous l'utiliserez comme `destination.arn` dans une étape ultérieure.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
```

```
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

b. Une fois l'étape précédente terminée, dans le compte des données du journal du destinataire (222222222222), associez une stratégie d'accès à la destination. Cette politique permet au compte de l'expéditeur des données du journal (111111111111) d'accéder à la destination dans le compte du destinataire des données du journal (222222222222). Vous pouvez utiliser un éditeur de texte pour mettre cette politique dans le `~/AccessPolicy.json` fichier :

   JSON

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. Cela crée une stratégie définissant qui a accès en écriture à la destination. Cette politique doit spécifier les `logs:PutAccountPolicy` actions `logs:PutSubscriptionFilter` et pour accéder à la destination. Les utilisateurs multicomptes utiliseront les `PutAccountPolicy` actions `PutSubscriptionFilter` et pour envoyer les événements du journal à la destination.

```
aws logs put-destination-policy \
```

```
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Étape 3 : Création d'une politique de filtrage des abonnements au niveau du compte

Passez au compte d'envoi, qui est 111111111111 dans cet exemple. Vous allez maintenant créer la politique de filtrage des abonnements au niveau du compte dans le compte expéditeur. Dans cet exemple, le filtre fait en sorte que chaque événement de journal contenant la chaîne ERROR de tous les groupes de journaux sauf deux soit livré à la destination que vous avez créée précédemment.

```
aws logs put-account-policy \
    --policy-name "CrossAccountFirehoseExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"DestinationArn":"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination", "FilterPattern":
 "{$.userIdentity.type = AssumedRole}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Les groupes de log du compte expéditeur et la destination doivent se trouver dans la même AWS région. Cependant, la destination peut pointer vers une AWS ressource telle qu'un flux Firehose situé dans une autre région.

Validation du flux des événements de journaux

Après avoir créé le filtre d'abonnement, CloudWatch Logs transmet tous les événements de journal entrants qui correspondent au modèle de filtre et aux critères de sélection au flux de diffusion Firehose. Les données commencent à apparaître dans votre compartiment Amazon S3 en fonction de l'intervalle de temps défini dans le flux de diffusion Firehose. Après un délai suffisant, vous pouvez consulter le compartiment Amazon S3 pour vérifier vos données. Pour vérifier le compartiment, saisissez la commande suivante :

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket'
```

La sortie de cette commande sera similaire à ce qui suit :

```
{
```

```
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2023-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

Vous pouvez ensuite récupérer un objet spécifique du compartiment en entrant la commande suivante. Remplacez la valeur de `key` par la valeur que vous avez trouvée dans la commande précédente.

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket' --key '2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Les données dans l'objet Amazon S3 sont comprimées au format gzip. Vous pouvez examiner les données brutes à partir de la ligne de commande en utilisant l'une des commandes suivantes :

Linux :

```
zcat testfile.gz
```

macOS :

```
zcat <testfile.gz
```

Modification de l'abonnement à la destination au moment de l'exécution

Vous pouvez rencontrer des situations où vous devez ajouter ou supprimer des expéditeurs de journaux d'une destination qui vous appartient. Vous pouvez utiliser les `PutAccountPolicy` actions PutDestinationPolicyet sur votre destination avec la nouvelle politique d'accès. Dans l'exemple

suivant, un compte précédemment ajouté 111111111111 est écarté de l'envoi de données du journal supplémentaires, et le compte 333333333333 est activé.

1. Récupérez la politique actuellement associée à la destination TestDestination et notez : AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testFirehoseDestination"
```

Les données renvoyées peuvent ressembler à ceci.

```
{
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Mettez à jour la politique de façon à refléter l'arrêt du compte 111111111111 et l'activation du compte 333333333333. Insérez cette politique dans le fichier ~/ NewAccessPolicy .json :

JSON

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

```
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Utilisez la commande suivante pour associer la politique définie dans le fichier NewAccessPolicy.json à la destination :

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

Cela désactive éventuellement les événements du journal à partir de l'ID de compte 111111111111. Les événements du journal de l'ID de compte 333333333333 commencent à passer à la destination dès que le propriétaire du compte 333333333333 crée un filtre d'abonnement.

Prévention du député confus

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services auprès des principaux fournisseurs de services qui ont obtenu l'accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés `aws:SourceOrgID`contextuelles `aws:SourceArnaws:SourceAccount`,, et de condition `aws:SourceOrgPaths`globale dans les politiques de ressources afin de limiter les autorisations qui accordent un autre service à la ressource. Utilisez `aws:SourceArn` pour associer une seule ressource à l'accès interservice. Utilisez `aws:SourceAccount` pour permettre à n'importe quelle ressource de ce compte d'être

associée à l'utilisation interservice. Utilisez `aws:SourceOrgID` pour permettre à n'importe quelle ressource de n'importe quel compte au sein d'une organisation d'être associée à l'utilisation interservice. Utilisez `aws:SourceOrgPaths` pour associer n'importe quelle ressource des comptes d'un chemin d'accès AWS Organizations à l'utilisation interservice. Pour plus d'informations sur l'utilisation et la compréhension des chemins, voir Comprendre le chemin de l' AWS Organizations entité.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (`*`) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:servicename:*:123456789012:*`.

Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser à la fois `aws:SourceAccount` et `aws:SourceArn` pour limiter les autorisations.

Pour se protéger contre le problème de l'adjoint confus à grande échelle, utilisez la clé de contexte de condition globale `aws:SourceOrgID` ou `aws:SourceOrgPaths` avec l'ID de l'organisation ou le chemin de l'organisation de la ressource dans vos politiques basées sur les ressources. Lorsque vous ajoutez, supprimez et déplacez des comptes dans votre organisation, les politiques qui contiennent la clé `aws:SourceOrgID` ou `aws:SourceOrgPaths` incluront automatiquement les bons comptes et vous n'avez pas besoin de mettre manuellement à jour les polices.

Les politiques documentées pour autoriser l'accès aux CloudWatch journaux afin d'y écrire des données Étape 1 : créer une destination dans Kinesis Data Streams Étape 2 : créer une destination et Firehose montrent comment vous pouvez utiliser la clé aws SourceArn : global condition context pour éviter le problème de confusion des adjoints.

Prévention de la récursivité dans les journaux

Les filtres d'abonnement risquent de provoquer une récursivité infinie des journaux, ce qui peut entraîner une forte augmentation de la facturation d'ingestion, à la fois dans les CloudWatch journaux et dans votre destination, s'ils ne sont pas évités. Cela peut se produire lorsqu'un filtre d'abonnement est associé à un groupe de journaux qui reçoit les événements du journal à la suite du flux de livraison de votre abonnement. Les journaux ingérés dans le groupe de journaux seront livrés à la destination, ce qui obligera le groupe de journaux à ingérer d'autres journaux qui seront ensuite redirigés vers la destination, créant ainsi une boucle de récursivité.

Par exemple, imaginez un filtre d'abonnement dont la destination est Firehose, qui transmet les événements du journal à Amazon S3. En outre, il existe également une fonction Lambda qui traite les nouveaux événements transmis à Amazon S3 et produit elle-même certains journaux. Si le filtre d'abonnement est appliqué au groupe de journaux de la fonction Lambda, les événements de journal produits par la fonction seront transmis à Firehose et Amazon S3 à destination, qui invoqueront ensuite à nouveau la fonction, ce qui entraînera la production et le transfert d'autres journaux vers Firehose et Amazon S3, provoquant une nouvelle invocation de la fonction, etc. Cela se produira en boucle infinie, entraînant une augmentation inattendue de la facturation lors de l'ingestion des journaux, de Firehose et d'Amazon S3.

Si la fonction Lambda est attachée à un VPC dont les journaux de flux sont activés pour les journaux, le groupe de CloudWatch journaux du VPC peut également provoquer une récursivité des journaux.

Nous vous recommandons de ne pas appliquer de filtres d'abonnement aux groupes de journaux qui font partie du flux de livraison de vos abonnements. Pour les filtres d'abonnement au niveau du compte, utilisez le `selectionCriteria` paramètre de l'`PutAccountPolicy`API pour exclure ces groupes de journaux de la politique.

Lorsque vous excluez des groupes de journaux, considérez les AWS services suivants qui produisent des journaux et peuvent faire partie des flux de travail de livraison de vos abonnements :

- Amazon EC2 avec Fargate
- Lambda
- AWS Step Functions
- Journaux de flux Amazon VPC activés pour les journaux CloudWatch

> **Note**
>
> Les événements de journal produits par le groupe de journaux d'une destination Lambda ne seront pas renvoyés à la fonction Lambda pour une politique de filtrage des abonnements au niveau du compte. Dans ce cas, il n'`selectionCriteria`est pas nécessaire d'exclure l'utilisation du groupe de journaux par la fonction Lambda de destination pour les politiques d'abonnement au compte.

Syntaxe des modèles de filtres pour les filtres de métriques, les filtres d'abonnements, les filtres d'événements du journal et Live Tail

> **Note**
>
> Pour plus d'informations sur la façon d'interroger vos groupes de CloudWatch journaux avec le langage de requête Amazon Logs Insights, consultezCloudWatch Syntaxe des requêtes en langage Logs Insights.

Avec CloudWatch Logs, vous pouvez utiliser des filtres métriques pour transformer les données des journaux en indicateurs exploitables, des filtres d'abonnement pour acheminer les événements du journal vers d'autres AWS services, filtrer les événements du journal pour rechercher des événements du journal et Live Tail pour visualiser de manière interactive vos journaux en temps réel au fur et à mesure qu'ils sont ingérés.

Les modèles de filtre constituent la syntaxe utilisée par les filtres métriques, les filtres d'abonnement, les événements de journal et Live Tail pour faire correspondre les termes des événements de journal. Les termes peuvent être des mots, des phrases exactes ou des valeurs numériques. Les expressions régulières (regex) peuvent être utilisées pour créer des modèles de filtres autonomes ou peuvent être intégrées aux modèles de filtres JSON et aux modèles de filtres délimités par des espaces.

Créez des modèles de filtre à partir des termes que vous souhaitez faire correspondre. Les modèles de filtre renvoient uniquement les événements du journal contenant les termes que vous définissez. Vous pouvez tester les modèles de filtre dans la CloudWatch console.

Rubriques

- Syntaxe des expressions régulières (regex) prise en charge
- Utilisation de modèles de filtres pour faire correspondre les termes à une expression régulière (regex)
- Utilisation de modèles de filtres pour faire correspondre les termes dans les événements du journal non structurés
- Utilisation de modèles de filtres pour faire correspondre les termes dans les événements du journal JSON

- Utilisation des modèles de filtres pour faire correspondre des termes dans des événements du journal délimités par des espaces

Syntaxe des expressions régulières (regex) prise en charge

Syntaxe regex prise en charge

Lorsque vous utilisez des expressions régulières pour rechercher et filtrer les données de journaux, vous devez entourer vos expressions de `%`.

Les modèles de filtres contenant des expressions régulières ne peuvent inclure que les éléments suivants :

- Des caractères alphanumériques : un caractère alphanumérique est un caractère qui est soit une lettre (de A à Z ou de a à z) soit un chiffre (de 0 à 9).
- Caractères symboliques pris en charge : « `:` », « `_` », « », # « », = « », « `@` », « `/` », « `;` », « `,` », « » et « `-` ». Par exemple, `%something!%` serait rejeté, car « `!` » n'est pas pris en charge.
- Des opérateurs pris en charge : « `^` », « `$` », « `?` », « `[` », « `]` », « `{` », « `}` », « `|` », « `\` », « `*` », « `+` » et « `.` ».

Les opérateurs `(` et `)` ne sont pas pris en charge. Vous ne pouvez pas utiliser de parenthèses pour définir un sous-modèle.

Les caractères multi-octets ne sont pas pris en charge.

> **Note**
>
> Quotas
> Il existe un maximum de 5 modèles de filtres contenant des expressions régulières pour chaque groupe de journaux lors de la création de filtres de métriques ou de filtres d'abonnements.
> Il existe une limite de 2 expressions régulières pour chaque modèle de filtres lors de la création d'un modèle de filtres délimités ou JSON pour les filtres de métriques et les filtres d'abonnements ou lors du filtrage des événements des journaux ou Live Tail.

Utilisation des opérateurs pris en charge

- `^` : ancre la correspondance au début d'une chaîne. Par exemple, `%^[hc]at%` correspond à « hat » et « cat », mais uniquement au début d'une chaîne.
- `$` : ancre la correspondance à la fin d'une chaîne. Par exemple, `%[hc]at$%` correspond à « hat » et « cat », mais uniquement à la fin d'une chaîne.
- `?`: Correspond à zéro ou à une occurrence du terme précédent. Par exemple, `%colou?r%` peut correspondre à la fois « color » et « colour ».
- `[]` : définit une classe de caractères. Correspond à la liste de caractères ou à la plage de caractères figurant entre crochets. Par exemple, `%[abc]%` correspond à « a », « b » ou « c » ; `%[a-z]%` correspond à n'importe quelle lettre minuscule comprise entre « a » et « z » ; et `%[abcx-z]%` correspond à « a », « b », « c », « x », « y » ou « z ».
- `{m, n}` : correspond au terme précédent au moins m fois et pas plus de n fois. Par exemple, `%a{3,5}%` correspond uniquement à « aaa », « aaaa » et « aaaaa ».

> **Note**
>
> m ou n peuvent être omis si vous choisissez de ne pas définir de minimum ou de maximum.

- `|` : valeur booléenne « Or », qui correspond au terme situé de part et d'autre de la barre verticale. Par exemple :
  - `%gra|ey%`peut correspondre à « gris » ou « gris »
  - `%^starting|^initializing|^shutting down%`peut correspondre à un match « commençant par... «, ou « initialisation... «, ou « arrêt », mais ne correspondra pas à « sauter l'initialisation... »
  - `%abcc|ab[^c]$`peut correspondre à « abcc... » et « aba... » mais ne correspond pas à « aac... »
- `\` : caractère d'échappement, qui vous permet d'utiliser la signification littérale d'un opérateur au lieu de sa signification particulière. Par exemple, `%\[.\]%` correspond à n'importe quel caractère entouré de « [» et «] » puisque les crochets ne sont pas inclus, tels que « [a] », « [b] », « [7] », « [@] », « []] » et « [ ] ».

> **Note**
>
> `%10\.10\.0\.1%` est la bonne façon de créer une expression régulière correspondant à l'adresse IP 10.10.0.1.

- `*`: Correspond à zéro ou plusieurs instances du terme précédent. Par exemple, `%ab*c%` peut correspondre à « ac », « abc » et « abbbc » ; `%ab[0-9]*%` peut correspondre à « ab », « ab0 » et « ab129 ».
- `+`: correspond à une ou plusieurs instances du terme précédent. Par exemple, `%ab+c%` peut correspondre à « abc », « abbc » et « abbbc », mais pas à « ac ».
- `.` : correspond à un seul caractère. Par exemple, `%.at%` correspond à une chaîne de trois caractères se terminant par « at », y compris « hat », « cat », « bat », « 4at », « #at » et « at » (en commençant par un espace).

> **Note**
>
> Lorsque vous créez une expression régulière pour faire correspondre les adresses IP, il est important d'échapper à l'opérateur `.`. Par exemple, `%10.10.0.1%` peut correspondre à « 10010,051 », ce qui n'est peut-être pas l'objectif réel de l'expression.

- `\d`, `\D` : correspond à un caractère numérique ou non numérique. Par exemple, `%\d%` est équivalent à `%[0-9]%` et `%\D%` est équivalent à `%[^0-9]%`.

> **Note**
>
> L'opérateur majuscule indique l'inverse de son équivalent en minuscules.

- `\s`, `\S` : correspond à un caractère d'espace ou à un caractère autre qu'un espace.

> **Note**
>
> L'opérateur majuscule indique l'inverse de son équivalent en minuscules. Les espaces incluent les caractères de tabulation (`\t`), d'espace ( ) et de nouvelle ligne (`\n`).

- `\w`, `\W` : correspond à un caractère alphanumérique ou non alphanumérique. Par exemple, `%\w%` est équivalent à `%[a-zA-Z_0-9]%` et `%\W%` est équivalent à `%[^a-zA-Z_0-9]%`.

> **Note**
>
> L'opérateur majuscule indique l'inverse de son équivalent en minuscules.

- `\xhh` : correspond à la mise en correspondance ASCII d'un caractère hexadécimal à deux chiffres. `\x` est la séquence d'échappement qui indique que les caractères suivants représentent la valeur

hexadécimale ASCII. hh spécifie les deux chiffres hexadécimaux (0-9 et A-F) qui pointent vers un caractère de la table ASCII.

> **Note**
>
> Vous pouvez utiliser `\xhh` pour faire correspondre les caractères de symbole qui ne sont pas pris en charge par le modèle de filtres. Par exemple, `%\x3A%` correspond à `:` ; `%\x28%` correspond à `(`.

Utilisation de modèles de filtres pour faire correspondre les termes à une expression régulière (regex)

Faire correspondre les termes à l'aide de regex

Vous pouvez faire correspondre les termes des événements de votre journal à l'aide d'un modèle regex entouré de `%` (signes de pourcentage avant et après le modèle regex). L'extrait de code suivant montre un exemple de modèle de filtre qui renvoie tous les événements du journal contenant le mot-clé AUTHORIZED.

Pour obtenir la liste des expressions régulières prises en charge, consultez la section Expressions régulières prises en charge.

```
%AUTHORIZED%
```

Ce modèle de filtre renvoie les messages d'événements du journal, tels que les suivants :

- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[SUCCESS 200] AUTHORIZED REQUEST`

Utilisation de modèles de filtres pour faire correspondre les termes dans les événements du journal non structurés

Faire correspondre les termes dans les événements du journal non structurés

Les exemples suivants contiennent des extraits de code qui montrent comment utiliser des modèles de filtres pour faire correspondre des termes dans les événements du journal non structurés.

> **Note**
>
> Les modèles de filtre sont sensibles à la casse. Placez les phrases et les termes exacts qui incluent des caractères non alphanumériques entre des guillemets doubles ("").

Example: Match a single term

L'extrait de code suivant montre un exemple de modèle de filtre à terme unique qui renvoie tous les événements du journal où les messages contiennent le mot ERROR.

```
ERROR
```

Ce modèle de filtre fait correspondre des messages d'événements du journal, tels que les suivants :

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match multiple terms

L'extrait de code suivant montre un exemple de modèle de filtre à termes multiples qui renvoie tous les événements du journal où les messages contiennent les mots ERROR et ARGUMENTS.

```
ERROR ARGUMENTS
```

Le filtre renvoie les messages d'événements du journal, comme l'exemple suivant :

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Ce modèle de filtre ne renvoie pas les messages d'événements du journal suivants, car ils ne contiennent pas les deux termes spécifiés dans le modèle de filtre.

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Example: Match optional terms

Vous pouvez utiliser la correspondance de modèles pour créer des modèles de filtres qui renvoient des événements du journal contenant des termes facultatifs. Insérez un point d'interrogation (« ? ») avant les termes que vous souhaitez faire correspondre. L'extrait de code suivant montre un exemple de modèle de filtre qui renvoie tous les événements du journal dont les messages contiennent le mot ERREUR ou le mot ARGUMENTS.

```
?ERROR ?ARGUMENTS
```

Ce modèle de filtre fait correspondre des messages d'événements du journal, tels que les suivants :

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

> **Note**
>
> Vous ne pouvez pas combiner le point d'interrogation (« ? ») avec d'autres modèles de filtre, tels que les termes d'inclusion et d'exclusion. Si vous combinez « ? » avec d'autres modèles de filtre, tous les termes en forme de point d'interrogation seront ignorés.
> Par exemple, le modèle de filtre suivant correspond à tous les événements contenant le motREQUEST, à l'exception du point d'interrogation (« ? ») les termes du filtre sont ignorés et n'ont aucun effet.
>
> ```
> ?ERROR ?ARGUMENTS REQUEST
> ```
>
> Correspondance des événements du journal
>
> - `[INFO] REQUEST FAILED`
> - `[WARN] UNAUTHORIZED REQUEST`
> - `[ERROR] 400 BAD REQUEST`

Example: Match exact phrases

L'extrait de code suivant montre un exemple de modèle de filtre qui renvoie tous les événements du journal lorsque les messages contiennent la phrase exacte INTERNAL SERVER ERROR.

```
"INTERNAL SERVER ERROR"
```

Ce modèle de filtre renvoie le message d'événement du journal suivant :

- `[ERROR 500] INTERNAL SERVER ERROR`

Example: Include and exclude terms

Vous pouvez créer des modèles de filtre qui renvoient des événements du journal lorsque les messages incluent certains termes et en excluent d'autres. Insérez un symbole moins (« - ») avant les termes que vous souhaitez exclure. L'extrait de code suivant montre un exemple de

modèle de filtre qui renvoie tous les événements du journal lorsque les messages incluent le terme ERROR et excluent le terme ARGUMENTS.

```
ERROR -ARGUMENTS
```

Ce modèle de filtre renvoie les messages d'événements du journal, tels que les suivants :

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Ce modèle de filtre ne renvoie pas les messages d'événements du journal suivants, car ils contiennent le mot ARGUMENTS.

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match everything

Vous pouvez faire correspondre tout dans vos événements du journal à l'aide des guillemets doubles. L'extrait de code suivant montre un exemple de modèle de filtre qui renvoie tous les événements du journal.

```
" "
```

Utilisation de modèles de filtres pour faire correspondre les termes dans les événements du journal JSON

Écriture de modèles de filtres pour les événements du journal JSON

La procédure suivante décrit comment écrire la syntaxe des modèles de filtres qui font correspondre les termes JSON contenant des chaînes et des valeurs numériques.

Writing filter patterns that match strings

Vous pouvez créer des modèles de filtres pour faire correspondre des chaînes dans les événements du journal JSON. L'extrait de code suivant montre un exemple de syntaxe pour les modèles de filtres basés sur des chaînes.

```
{ PropertySelector EqualityOperator String }
```

Placez les modèles de filtres entre accolades (« {} »). Les modèles de filtres basés sur des chaînes doivent contenir les parties suivantes :

- Sélecteur de propriétés

  Définissez les sélecteurs de propriétés à l'aide d'un symbole dollar suivi d'un point (« $. »). Les sélecteurs de propriétés constituent des chaînes alphanumériques qui prennent en charge le trait d'union « - » et le caractère de soulignement « _ ». Les chaînes ne prennent pas en charge la notation scientifique. Les sélecteurs de propriétés pointent vers des nœuds de valeur dans les événements du journal JSON. Les nœuds de valeur peuvent être des chaînes ou des nombres. Insérez les matrices après les sélecteurs de propriétés. Les éléments des tableaux suivent un système de numérotation basé sur le zéro, ce qui signifie que le premier élément du tableau est l'élément 0, le deuxième élément est l'élément 1, et ainsi de suite. Placez les éléments entre crochets (« [] »). Si un sélecteur de propriétés pointe vers une matrice ou vers un objet, le modèle de filtre ne pourra pas correspondre au format du journal. Si la propriété JSON contient un point (`"."`), la notation entre crochets peut être utilisée pour sélectionner cette propriété.

  > **Note**
  >
  > Sélecteur de caractères génériques
  > Vous pouvez utiliser le caractère générique JSON pour sélectionner n'importe quel élément du tableau ou n'importe quel champ d'objet JSON.
  > Quotas
  > Vous ne pouvez utiliser qu'un seul sélecteur de caractères génériques dans un sélecteur de propriétés.

- Opérateur d'égalité

Définissez les opérateurs d'égalité à l'aide de l'un des symboles suivants : égal (« = ») ou inégal (« != »). Les opérateurs d'égalité renvoient une valeur booléenne (vrai ou faux).

- Chaîne

  Vous pouvez placer des chaînes entre guillemets doubles ("")). Les chaînes contenant des types différents des caractères alphanumériques et le symbole de soulignement doivent être placées entre guillemets doubles. Utilisez l'astérisque (« * ») comme joker pour faire correspondre du texte.

> **Note**
>
> Vous pouvez utiliser n'importe quelle expression régulière conditionnelle lors de la création des modèles de filtres pour faire correspondre les termes des événements du journal JSON. Pour obtenir la liste des expressions régulières prises en charge, consultez la section Expressions régulières prises en charge.

L'extrait de code suivant contient un exemple de modèle de filtre qui montre comment formater un modèle de filtre pour faire corresponde un terme JSON à une chaîne.

```
{ $.eventType = "UpdateTrail" }
```

Writing filter patterns that match numeric values

Vous pouvez créer des modèles de filtres pour faire correspondre des valeurs numériques dans les événements du journal JSON. L'extrait de code suivant montre un exemple de syntaxe pour les modèles de filtres qui font correspondre des valeurs numériques.

```
{ PropertySelector NumericOperator Number }
```

Placez les modèles de filtres entre accolades (« {} »). Les modèles de filtres qui font correspondre des valeurs numériques doivent comporter les parties suivantes :

- Sélecteur de propriétés

Définissez les sélecteurs de propriétés à l'aide d'un symbole dollar suivi d'un point (« $. »). Les sélecteurs de propriétés constituent des chaînes alphanumériques qui prennent en charge le trait d'union « - » et le caractère de soulignement « _ ». Les chaînes ne prennent pas en charge la notation scientifique. Les sélecteurs de propriétés pointent vers des nœuds de valeur dans les événements du journal JSON. Les nœuds de valeur peuvent être des chaînes ou des nombres. Insérez les matrices après les sélecteurs de propriétés. Les éléments des tableaux suivent un système de numérotation basé sur le zéro, ce qui signifie que le premier élément du tableau est l'élément 0, le deuxième élément est l'élément 1, et ainsi de suite. Placez les éléments entre crochets (« [] »). Si un sélecteur de propriétés pointe vers une matrice ou vers un objet, le modèle de filtre ne pourra pas correspondre au format du journal. Si la propriété JSON contient un point (`"."`), la notation entre crochets peut être utilisée pour sélectionner cette propriété.

> **Note**
>
> Sélecteur de caractères génériques
>
> Vous pouvez utiliser le caractère générique JSON pour sélectionner n'importe quel élément du tableau ou n'importe quel champ d'objet JSON.
>
> Quotas
>
> Vous ne pouvez utiliser qu'un seul sélecteur de caractères génériques dans un sélecteur de propriétés.

- Opérateur numérique

  Définissez les opérateurs numériques à l'aide de l'un des symboles suivants : supérieur à (« > »), inférieur à (« < »), égal (« = »), inégal (« != »), supérieur ou égal à (« >= ») ou inférieur ou égal à (« <= »).

- Nombre

  Vous pouvez utiliser des entiers contenant les symboles plus (« + ») ou moins (« - »), et suivre une notation scientifique. Utilisez l'astérisque (« * ») comme joker pour faire correspondre des nombres.

L'extrait de code suivant contient des exemples qui montrent comment formater des modèles de filtres pour faire correspondre des termes JSON à des valeurs numériques.

```
// Filter pattern with greater than symbol
```

```
{ $.bandwidth > 75 }
// Filter pattern with less than symbol
{ $.latency < 50 }
// Filter pattern with greater than or equal to symbol
{ $.refreshRate >= 60 }
// Filter pattern with less than or equal to symbol
{ $.responseTime <= 5 }
// Filter pattern with equal sign
{ $.errorCode = 400}
// Filter pattern with not equal sign
{ $.errorCode != 500 }
// Filter pattern with scientific notation and plus symbol
{ $.number[0] = 1e-3 }
// Filter pattern with scientific notation and minus symbol
{ $.number[0] != 1e+3 }
```

Faire correspondre les termes des événements du journal JSON à l'aide d'expressions simples

Les exemples suivants contiennent des extraits de code qui montrent comment les modèles de filtres peuvent faire correspondre des termes dans un événement du journal JSON.

> **Note**
>
> Si vous testez un exemple de modèles de filtre avec l'exemple d'événement du journal JSON, vous devez saisir l'exemple de journal JSON sur une seule ligne.

Événement du journal JSON

```
{
      "eventType": "UpdateTrail",
      "sourceIPAddress": "111.111.111.111",
      "arrayKey": [
            "value",
            "another value"
      ],
      "objectList": [
           {
             "name": "a",
             "id": 1
```

```
            },
            {
              "name": "b",
              "id": 2
            }
        ],
        "SomeObject": null,
        "cluster.name": "c"
}
```

Example: Filter pattern that matches string values

Ce modèle de filtre fait correspondre la chaîne `"UpdateTrail"` dans la propriété `"eventType"`.

```
{ $.eventType = "UpdateTrail" }
```

Example: Filter pattern that matches string values (IP address)

Ce modèle de filtre contient un joker et fait correspondre la propriété `"sourceIPAddress"`, car elle ne contient pas de numéro avec le préfixe `"123.123."`.

```
{ $.sourceIPAddress != 123.123.* }
```

Example: Filter pattern that matches a specific array element with a string value

Ce modèle de filtre fait correspondre l'élément `"value"` dans la matrice `"arrayKey"`.

```
{ $.arrayKey[0] = "value" }
```

Example: Filter pattern that matches a string using regex

Ce modèle de filtre fait correspondre la chaîne `"Trail"` dans la propriété `"eventType"`.

```
{ $.eventType = %Trail% }
```

Example: Filter pattern that uses a wildcard to match values of any element in the array using regex

Le modèle de filtre contient une expression régulière qui fait correspondre l'élément `"value"` dans la matrice `"arrayKey"`.

```
{ $.arrayKey[*] = %val.{2}% }
```

Example: Filter pattern that uses a wildcard to match values of any element with a specific prefix and subnet using regex (IP address)

Ce modèle de filtre contient une expression régulière qui fait correspondre l'élément `"111.111.111.111"` dans la propriété `"sourceIPAddress"`.

```
{ $.* = %111\.111\.111\.1[0-9]{1,2}% }
```

> **Note**
>
> Quotas
> Vous ne pouvez utiliser qu'un seul sélecteur de caractères génériques dans un sélecteur de propriétés.

Example: Filter pattern that matches a JSON property with a period (.) in the key

```
{ $.['cluster.name'] = "c" }
```

Example: Filter pattern that matches JSON logs using IS

Vous pouvez créer des modèles de filtres qui font correspondre des champs dans les journaux JSON avec la variable `IS`. La variable `IS` peut faire correspondre des champs contenant les valeurs `NULL`, `TRUE` ou `FALSE`. Le modèle de filtre suivant renvoie les journaux JSON où la valeur de `SomeObject` est `NULL`.

```
{ $.SomeObject IS NULL }
```

Example: Filter pattern that matches JSON logs using NOT EXISTS

Vous pouvez créer des modèles de filtre avec la `NOT EXISTS` variable pour renvoyer des journaux JSON qui ne contiennent pas de champs spécifiques dans les données des journaux. Le modèle de filtre suivant utilise `NOT EXISTS` pour renvoyer des journaux JSON qui ne contiennent pas le champ `SomeOtherObject`.

```
{ $.SomeOtherObject NOT EXISTS }
```

> **Note**
>
> Les variables `IS NOT` et `EXISTS` ne sont actuellement pas prises en charge.

Faire correspondre les termes dans des objets JSON à l'aide d'expressions composées

Vous pouvez utiliser les opérateurs logiques AND (« && ») et OR (« || ») dans les modèles de filtres pour créer des expressions composées faisant correspondre des événements du journal où deux ou plusieurs conditions sont vraies. Les expressions composées prennent en charge l'utilisation de parenthèses (« () ») et l'ordre d'opérations standard suivant : () > && > ||. Les exemples suivants contiennent des extraits de code qui montrent comment utiliser des modèles de filtres avec des expressions composées pour faire correspondre des termes dans un objet JSON.

Objet JSON

```
{
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
```

```
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
         "email": "Jane.Doe@example.com"
        }
    ],
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Example: Expression that matches using AND (&&)

Ce modèles de filtre contient une expression composée qui fait correspondre `"id"` dans `"user"` avec une valeur numérique de 1 et `"email"` dans le premier élément de la matrice `"users"` avec la chaîne `"John.Doe@example.com"`.

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

Example: Expression that matches using OR (||)

Ce modèle de filtre contient une expression composée qui fait correspondre `"email"` dans `"user"` avec la chaîne `"John.Stiles@example.com"`.

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch" &&
 $.actions[2] = "nonmatch" }
```

Example: Expression that doesn't match using AND (&&)

Ce modèle de filtre contient une expression composée qui ne trouve pas de correspondance, car l'expression ne correspond pas à la troisième action de `"actions"`.

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch") &&
 $.actions[2] = "nonmatch" }
```

> **Note**
>
> Quotas
>
> Vous ne pouvez utiliser qu'un seul sélecteur de caractères génériques dans un sélecteur de propriétés, et jusqu'à trois sélecteurs de caractères génériques dans un modèle de filtre avec des expressions composées.

Example: Expression that doesn't match using OR (||)

Ce modèle de filtre contient une expression composée qui ne trouve pas de correspondance, car l'expression ne correspond pas à la première propriété de `"users"` ou à la troisième action de `"actions"`.

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

Utilisation des modèles de filtres pour faire correspondre des termes dans des événements du journal délimités par des espaces

Écriture de modèles de filtres pour les événements du journal délimités par des espaces

Vous pouvez créer des modèles de filtres pour faire correspondre les termes des événements du journal délimités par des espaces. La procédure suivante fournit un exemple d'événement du

journal délimité par des espaces et décrit comment écrire la syntaxe des modèles de filtres qui font correspondre les termes dans l'événement du journal délimité par des espaces.

> **Note**
>
> Vous pouvez utiliser n'importe quelle expression régulière conditionnelle lorsque vous créez des modèles de filtre pour faire correspondre les termes des événements du journal délimités par des espaces. Pour obtenir la liste des expressions régulières prises en charge, consultez la section Expressions régulières prises en charge.

Example: Space-delimited log event

L'extrait de code suivant montre un événement du journal délimité par des espaces contenant sept champs : `ip`, `user`, `username`, `timestamp`, `request`, `status_code` et `bytes`.

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

> **Note**
>
> Les caractères entre crochets (« [] ») et les guillemets doubles (« ») sont considérés comme des champs uniques.

Writing filter patterns that match terms in a space-delimited log event

Pour créer un modèle de filtre qui fait correspondre les termes dans un événement du journal délimité par des espaces, placez le modèle de filtre entre crochets (« [] ») et spécifiez des champs dont les noms sont séparés par des virgules (« , »). Le modèle de filtre suivant analyse sept champs.

```
[ip=%127\.0\.0\.[1-9]%, user, username, timestamp, request =*.html*, status_code =
 4*, bytes]
```

Vous pouvez utiliser des opérateurs numériques (>, <, =, !=, >= ou <=) et l'astérisque (*) comme caractère générique ou expression régulière pour définir les conditions de votre modèle de filtre. Dans l'exemple de modèle de filtre, `ip` utilise une expression régulière qui correspond à la plage d'adresses IP 127.0.0.1 à 127.0.0.9, `request` contient un caractère générique qui indique qu'elle doit extraire une valeur avec `.html` et `status_code` contient un caractère générique qui indique qu'elle doit extraire une valeur commençant par `4`.

Si vous ne connaissez pas le nombre de champs que vous analysez dans un événement du journal délimité par des espaces, vous pouvez utiliser des points de suspension (...) pour référencer n'importe quel champ non nommé. Les points de suspension peuvent référencer autant de champs que nécessaire. L'exemple suivant montre un modèle de filtre avec des points de suspension représentant les quatre premiers champs non nommés montrés dans l'exemple de modèle de filtre précédent.

```
[..., request =*.html*, status_code = 4*, bytes]
```

Vous pouvez également utiliser les opérateurs logiques AND (&&) et OR (||) pour créer des expressions composées. Le modèle de filtre suivant contient une expression composée qui indique que la valeur de `status_code` doit être `404` ou `410`.

```
[ip, user, username, timestamp, request =*.html*, status_code = 404 || status_code =
 410, bytes]
```

Faire correspondre les termes dans les événements du journal délimités par des espaces à l'aide de la mise en correspondance de modèles

Vous pouvez utiliser la correspondance de modèles pour créer des modèles de filtres délimités par des espaces qui font correspondre des termes dans un ordre spécifique. Spécifiez l'ordre de vos termes à l'aide d'indicateurs. Utilisez w1 pour représenter votre premier terme, puis w2, et ainsi de suite, pour représenter l'ordre de vos termes ultérieurs. Insérez des virgules (« , ») entre vos

termes. Les exemples suivants contiennent des extraits de code qui montrent comment utiliser la correspondance de modèles avec des modèles de filtres délimités par des espaces.

> **Note**
>
> Vous pouvez utiliser n'importe quelle expression régulière conditionnelle lorsque vous créez des modèles de filtre pour faire correspondre les termes des événements du journal délimités par des espaces. Pour obtenir la liste des expressions régulières prises en charge, consultez la section Expressions régulières prises en charge.

Événement du journal délimité par des espaces

```
INFO 09/25/2014 12:00:00 GET /service/resource/67 1200
INFO 09/25/2014 12:00:01 POST /service/resource/67/part/111 1310
WARNING 09/25/2014 12:00:02 Invalid user request
ERROR 09/25/2014 12:00:02 Failed to process request
```

Example: Match terms in order

Le modèle de filtre délimité par des espaces suivant renvoie les événements du journal où le premier mot des événements du journal est ERROR.

```
[w1=ERROR, w2]
```

> **Note**
>
> Lorsque vous créez des modèles de filtres délimités par des espaces qui utilisent la correspondance de modèles, vous devez inclure un indicateur vide après avoir spécifié l'ordre de vos termes. Par exemple, si vous créez un modèle de filtre qui renvoie les événements du journal où le premier mot est ERROR, incluez un indicateur vide w2 après le terme w1.

Example: Match terms with AND (&&) and OR (||)

Vous pouvez utiliser les opérateurs logiques AND (« && ») et OR (« || ») pour créer des modèles de filtres délimités par des espaces contenant des conditions. Le modèle de filtre suivant renvoie les événements du journal lorsque le premier mot des événements est ERROR ou WARNING.

```
[w1=ERROR || w1=WARNING, w2]
```

Example: Exclude terms from matches

Vous pouvez créer des modèles de filtres délimités par des espaces qui renvoient des événements du journal qui excluent un ou plusieurs termes. Placez un symbole moins (« - ») avant les termes que vous souhaitez exclure. L'extrait de code suivant montre un exemple de modèle de filtre qui renvoie les événements du journal lorsque les premiers mots ne sont pas ERROR et WARNING.

```
[w1!=ERROR && w1!=WARNING, w2]
```

Example: Match the top level item in a resource URI

L'extrait de code suivant montre un exemple de modèle de filtre qui correspond à l'élément de premier niveau d'un URI de ressource à l'aide d'une expression régulière.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+$%, response_time]
```

Example: Match the child level item in a resource URI

L'extrait de code suivant montre un exemple de modèle de filtre qui correspond à l'élément de niveau enfant d'un URI de ressource à l'aide d'une expression régulière.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+/part/[0-9]+$%,
 response_time]
```

Activer la journalisation à partir AWS des services

Alors que de nombreux services publient des CloudWatch journaux uniquement dans Logs, certains AWS services peuvent publier des journaux directement sur Amazon Simple Storage Service ou Amazon Data Firehose. Si votre principale exigence en matière de journaux est le stockage ou le traitement dans l'un de ces services, vous pouvez facilement demander au service qui produit les journaux de les envoyer directement à Amazon S3 ou Firehose sans configuration supplémentaire.

Même lorsque les journaux sont publiés directement sur Amazon S3 ou Firehose, des frais s'appliquent. Pour plus d'informations, consultez la section Vended Logs dans l'onglet Logs d'Amazon CloudWatch Pricing.

Certains AWS services utilisent une infrastructure commune pour envoyer leurs journaux. Pour activer la journalisation à partir de ces services, vous devez être connecté en tant qu'utilisateur disposant de certaines autorisations. En outre, vous devez accorder des autorisations AWS pour permettre l'envoi des journaux.

Pour les services qui exigent ces autorisations, il existe deux versions des autorisations nécessaires. Les services qui exigent ces autorisations supplémentaires sont indiqués comme étant Pris en charge [Autorisations V1] et Pris en charge [Autorisations V2] dans le tableau. Pour plus d'informations sur ces autorisations requises, consultez les sections situées après le tableau.

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
Journaux d'accès Amazon API Gateway	Journaux vended	Pris en charge [Autorisations V1]		
AWS AppSync journaux	Journaux personnalisés	Pris en charge		

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
Journaux Amazon Aurora MySQL	Journaux personnalisés	Pris en charge		
Amazon Bedrock Journalisation des bases de connaissances	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
Amazon Bedrock AgentCore	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
Journaux de métriques de qualité multimédia Amazon Chime et journaux de messages SIP	Journaux vended	Pris en charge [Autorisations V1]		
CloudFront: journaux d'accès	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
AWS CloudHSM journaux d'audit	Journaux personnalisés	Pris en charge		

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
CloudWatch De toute évidence, journaux des événements d'évaluation	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	
CloudWatch Journaux du moniteur Internet	Journaux vended		Pris en charge [Autorisations V1]	
CloudTrail journaux	Journaux personnalisés	Pris en charge		
AWS CodeBuild journaux	Journaux personnalisés	Pris en charge		
Amazon CodeWhisperer journaux d'événements	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
Amazon Cognito journaux	Journaux vended	Pris en charge [Autorisations V1]		
Journaux Amazon Connect	Journaux personnalisés	Pris en charge		

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
AWS DataSync journaux	Journaux personnalisés	Pris en charge		
Journaux Amazon ElastiCache (Redis OSS)	Journaux vended	Pris en charge [Autorisations V1]		Pris en charge [Autorisations V1]
AWS Elastic Beanstalk Journaux	Journaux personnalisés	Pris en charge		
Journaux Amazon Elastic Container Service	Journaux personnalisés	Pris en charge		
Journaux de plan de contrôle d'Amazon Elastic Kubernetes Service	Journaux vended	Pris en charge		
AWS Elemental MediaPackage journaux d'accès	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
AWS Elemental MediaTailor journaux	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
Résolution des entités AWS Journaux	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
Amazon EventBridge Enregistrement des canalisations	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Amazon EventBridge bus événementiels	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
AWS Fargate Journaux	Journaux personnalisés	Pris en charge		
Journaux d'expériences AWS Fault Injection Service	Journaux vended		Pris en charge [Autorisations V1]	
Amazon FinSpace	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
AWS Global Accelerator journaux de flux	Journaux vended		Pris en charge [Autorisations V1]	
AWS Glue journaux de tâches	Journaux personnalisés	Pris en charge		
Journaux d'erreurs d'IAM Identity Center	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
Journaux de conversation Amazon Interactive Video Service	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
AWS IoT Journaux	Journaux personnalisés	Pris en charge		
AWS IoT FleetWise journaux	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
AWS Lambda Journaux	Journaux vended	Pris en charge	Pris en charge	Pris en charge

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
Journaux Amazon Macie	Journaux personnalisés	Pris en charge		
Journaux Amazon SES	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
AWS Mainframe Modernization	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Journaux Amazon Managed Service for Prometheus	Journaux vended	Pris en charge [Autorisations V1]		
Journaux de l'agent Amazon MSK	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Journaux Amazon MSK Connect	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
Journaux Amazon MQ	Journaux personnalisés	Pris en charge		
AWS Journaux de Network Firewall	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Journaux d'accès de Network Load Balancer	Journaux vended		Pris en charge [Autorisations V1]	
OpenSearch journaux	Journaux personnalisés	Pris en charge		
Journaux d'ingestion d'Amazon OpenSearch Service	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
AWS OpsWorks Journaux	Journaux personnalisés	Pris en charge		
Journaux PCS AWS	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
Journaux ServicePostgre SQL de la base de données relationnelle Amazon	Journaux personnalisés	Pris en charge		
Journaux de conversation Amazon Q Business	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
AWS RoboMaker journaux	Journaux personnalisés	Pris en charge		
Journaux des requêtes DNS publiques Amazon Route 53	Journaux vended	Pris en charge		
Journaux de requête Amazon Route 53 Resolver	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	
Événements Amazon SageMaker AI	Journaux vended	Pris en charge [Autorisations V1]		
Événements destinés aux employés d'Amazon SageMaker AI	Journaux vended	Pris en charge [Autorisations V1]		

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
AWS Journaux VPN de site à site	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Journaux d'Amazon Simple Email Service	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]
Journaux Amazon Simple Notification Service	Journaux personnalisés	Pris en charge		
Journaux des politiques de protection des données d'Amazon Simple Notification Service	Journaux personnalisés	Pris en charge		
EC2 Fichiers de flux de données des instances Spot	Journaux vended		Pris en charge [Autorisations V1]	
AWS Step Functions Journaux du flux de travail Express et du flux de travail standard	Journaux vended	Pris en charge [Autorisations V1]		

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
Journaux d'audit et journaux d'état de Storage Gateway	Journaux vended	Pris en charge [Autorisations V1]		
AWS Transfer Family journaux	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Accès vérifié par AWS journaux	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Journaux de flux d'Amazon Virtual Private Cloud	Journaux vended	Pris en charge	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Journaux d'accès Amazon VPC Lattice	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]
Serveur de routage Amazon VPC	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]

Source des journaux	Type de journal	the section called "Logs envoyés à CloudWatch Logs"	the section called "Journaux envoyés à Amazon S3"	the section called "Logs envoyés à Firehose"
AWS WAF Journaux	Journaux vended	Pris en charge [Autorisations V1]	Pris en charge [Autorisations V1]	Pris en charge
Amazon WorkMail journaux d'audit	Journaux vended	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]	Pris en charge [Autorisations V2]

Journalisation nécessitant des autorisations supplémentaires [V1]

Certains AWS services utilisent une infrastructure commune pour envoyer leurs CloudWatch journaux à Logs, Amazon S3 ou Firehose. Pour permettre aux AWS services répertoriés dans le tableau suivant d'envoyer leurs journaux vers ces destinations, vous devez être connecté en tant qu'utilisateur disposant de certaines autorisations.

En outre, des autorisations doivent être accordées AWS pour permettre l'envoi des journaux. AWS peut créer automatiquement ces autorisations lors de la configuration des journaux, ou vous pouvez les créer vous-même avant de configurer la journalisation. Pour la livraison entre comptes, vous devez créer vous-même les politiques d'autorisation manuellement.

Si vous choisissez de configurer AWS automatiquement les autorisations et les politiques de ressources nécessaires lorsque vous ou un membre de votre organisation configurez l'envoi des journaux pour la première fois, l'utilisateur qui configure l'envoi des journaux doit disposer de certaines autorisations, comme expliqué plus loin dans cette section. Vous pouvez également créer les politiques de ressources vous-même, de sorte que les utilisateurs qui configurent l'envoi des journaux n'ont pas besoin d'autant d'autorisations.

Les rubriques suivantes fournissent plus de détails pour chacune de ces destinations.

Rubriques

- Logs envoyés à CloudWatch Logs
- Journaux envoyés à Amazon S3
- Logs envoyés à Firehose

Logs envoyés à CloudWatch Logs

> **Important**
>
> Lorsque vous configurez les types de journaux dans la liste suivante à envoyer à CloudWatch Logs, vous AWS créez ou modifiez les politiques de ressources associées au groupe de journaux recevant les journaux, si nécessaire. Continuez à lire cette section pour voir les détails.

Cette section s'applique lorsque les types de journaux répertoriés dans le tableau de la section précédente sont envoyés à CloudWatch Logs :

Autorisations des utilisateurs

Pour pouvoir configurer l'envoi de l'un de ces types de CloudWatch journaux à Logs pour la première fois, vous devez être connecté à un compte avec les autorisations suivantes.

- `logs:CreateLogDelivery`
- `logs:PutResourcePolicy`
- `logs:DescribeResourcePolicies`
- `logs:DescribeLogGroups`

> **Note**
>
> Lorsque vous spécifiez l'autorisation `logs:DescribeLogGroupslogs:DescribeResourcePolicies`, ou l'`logs:PutResourcePolicy`autorisation, veillez à définir l'ARN de sa `Resource` ligne de manière à utiliser un `*` caractère générique, au lieu de ne spécifier qu'un seul nom de groupe de journaux. Par exemple, `"Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*"`

Si l'un de ces types de journaux est déjà envoyé à un groupe de CloudWatch journaux dans Logs, vous n'avez besoin que de l'`logs:CreateLogDelivery`autorisation pour configurer l'envoi d'un autre de ces types de journaux à ce même groupe de journaux.

Politique de ressources du groupe de journaux

Le groupe de journaux dans lequel les journaux sont envoyés doit avoir une politique de ressources qui inclut certaines autorisations. Si le groupe de journaux n'a actuellement aucune politique de ressources et que l'utilisateur qui configure la journalisation dispose des `logs:DescribeLogGroups` autorisations `logs:PutResourcePolicylogs:DescribeResourcePolicies`, et pour le groupe de journaux, crée AWS automatiquement la politique suivante pour celui-ci lorsque vous commencez à envoyer les CloudWatch journaux à Logs.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:my-log-group:log-
stream:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
```

```
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Si le groupe de journaux dispose d'une politique de ressources mais que cette politique ne contient pas l'instruction indiquée dans la politique précédente, et que l'utilisateur qui configure la journalisation dispose des autorisations `logs:PutResourcePolicy`, `logs:DescribeResourcePolicies` et `logs:DescribeLogGroups` pour le groupe de journaux, cette instruction est ajoutée à la politique de ressources du groupe de journaux.

Journaux envoyés à Amazon S3

Lorsque vous configurez les journaux à envoyer à Amazon S3, vous AWS créez ou modifiez les politiques de ressources associées au compartiment S3 qui reçoit les journaux, si nécessaire.

Les journaux publiés directement sur Amazon S3 le sont dans un compartiment existant que vous spécifiez. Un ou plusieurs fichiers journaux sont créés toutes les cinq minutes dans le compartiment spécifié.

Lorsque vous livrez des journaux pour la première fois à un compartiment Amazon S3, le service qui livre les journaux enregistre le propriétaire du compartiment pour s'assurer que les journaux sont livrés uniquement à un compartiment appartenant à ce compte. Par conséquent, pour changer le propriétaire du compartiment Amazon S3, vous devez recréer ou mettre à jour l'abonnement au journal dans le service d'origine.

> **Note**
>
> CloudFront utilise un modèle d'autorisation différent de celui des autres services qui envoient des journaux vendus à S3. Pour plus d'informations, consultez Autorisations requises pour configurer la journalisation standard et pour accéder à vos fichiers journaux.

En outre, si vous utilisez le même compartiment S3 pour les journaux CloudFront d'accès et une autre source de journaux, l'activation de l'ACL sur le compartiment permet CloudFront également d'autoriser toutes les autres sources de journaux qui utilisent ce compartiment.

Important

Si vous envoyez des journaux vers un compartiment Amazon S3 et que la politique du compartiment contient un `NotPrincipal` élément `NotAction` ou, l'ajout automatique des autorisations de livraison des journaux au compartiment et la création d'un abonnement aux journaux échoueront. Pour créer un abonnement aux journaux avec succès, vous devez ajouter manuellement les autorisations de livraison des journaux à la politique de compartiment, puis créer l'abonnement aux journaux. Pour plus d'informations, consultez les instructions de cette section.

Si le compartiment est chiffré côté serveur à l'aide d'une AWS KMS clé gérée par le client, vous devez également ajouter la politique de clé pour votre clé gérée par le client. Pour de plus amples informations, veuillez consulter Amazon S3.

Si le SSE-KMS et une clé de compartiment sont activés dans le compartiment de destination, la politique de clé KMS gérée par le client associée ne fonctionne plus comme prévu pour toutes les demandes. Pour plus d'informations, consultez Réduire le coût du SSE-KMS avec les clés de compartiment Amazon S3.

Si vous utilisez des journaux vendus et un chiffrement S3 avec une AWS KMS clé gérée par le client, vous devez utiliser un ARN de AWS KMS clé entièrement qualifié au lieu d'un identifiant de clé lorsque vous configurez le compartiment. Pour plus d'informations, consultez put-bucket-encryption.

Autorisations des utilisateurs

Pour pouvoir configurer l'envoi de l'un de ces types de journaux à Amazon S3 pour la première fois, vous devez être connecté à un compte disposant des autorisations suivantes.

- `logs:CreateLogDelivery`
- `S3:GetBucketPolicy`
- `S3:PutBucketPolicy`

Si l'un de ces types de journaux est déjà envoyé vers un compartiment Amazon S3, il vous suffit de disposer de l'autorisation `logs:CreateLogDelivery` pour configurer l'envoi d'un autre de ces types de journaux vers le même compartiment.

Politique de ressources du compartiment S3

Le compartiment S3 où les journaux sont envoyés doit avoir une politique de ressources qui inclut certaines autorisations. Si le compartiment ne comporte aucune politique de ressources et que l'utilisateur qui configure la journalisation dispose des autorisations `S3:GetBucketPolicy` et `S3:PutBucketPolicy` pour le compartiment, AWS crée automatiquement la politique suivante lorsque vous commencez à envoyer les journaux à Amazon S3.

JSON

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Dans la politique précédente, pour `aws:SourceAccount`, spécifiez la liste des ID de comptes pour lesquels les journaux sont transmis à ce compartiment. Pour`aws:SourceArn`, spécifiez la liste ARNs des ressources qui génèrent les journaux, dans le formulaire`arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Si le compartiment dispose d'une politique de ressources, mais que celle-ci ne contient pas la déclaration indiquée dans la politique précédente, et que l'utilisateur qui configure la journalisation dispose des autorisations `S3:GetBucketPolicy` et `S3:PutBucketPolicy` pour le compartiment, cette déclaration est ajoutée à la politique de ressources du compartiment.

> **Note**
>
> Dans certains cas, des `AccessDenied` erreurs peuvent s'afficher AWS CloudTrail si l'`s3:ListBucket`autorisation n'a pas été accordée`delivery.logs.amazonaws.com`. Pour éviter ces erreurs dans vos CloudTrail journaux, vous devez accorder l'`s3:ListBucket`autorisation `delivery.logs.amazonaws.com` et inclure les `Condition` paramètres indiqués dans l'`s3:GetBucketAcl`autorisation définie dans la politique de compartiment précédente. Pour simplifier les choses, au lieu de créer un nouveau

`Statement`, vous pouvez directement mettre à jour le `AWSLogDeliveryAclCheck` pour qu'il devienne `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Chiffrement côté serveur des compartiments Amazon S3

Vous pouvez protéger les données de votre compartiment Amazon S3 en activant le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) ou le chiffrement côté serveur avec une clé stockée dans (SSE-KMS). AWS KMS AWS Key Management Service Pour plus d'informations, consultez Protection des données à l'aide du chiffrement côté serveur.

Si vous choisissez l'option SSE-S3, aucune configuration supplémentaire n'est requise. Amazon S3 gère la clé de chiffrement.

> **Warning**
>
> Si vous choisissez SSE-KMS, vous devez utiliser une clé gérée par le client, car l'utilisation d'une clé AWS gérée n'est pas prise en charge dans ce scénario. Si vous configurez le chiffrement à l'aide d'une clé AWS gérée, les journaux seront fournis dans un format illisible.

Lorsque vous utilisez une AWS KMS clé gérée par le client, vous pouvez spécifier le nom de ressource Amazon (ARN) de la clé gérée par le client lorsque vous activez le chiffrement des compartiments. Vous devez ajouter les informations suivantes à la politique de votre clé gérée par le client (pas à la politique du compartiment S3), afin que le compte de livraison des journaux puisse écrire des données dans votre compartiment S3.

Si vous choisissez l'option SSE-KMS, vous devez utiliser une clé gérée par le client, car les clés gérées AWS ne sont pas prises en charge pour ce scénario. Lorsque vous utilisez une AWS KMS clé gérée par le client, vous pouvez spécifier le nom de ressource Amazon (ARN) de la clé gérée par le client lorsque vous activez le chiffrement des compartiments. Vous devez ajouter les informations suivantes à la politique de votre clé gérée par le client (pas à la politique du compartiment S3), afin que le compte de livraison des journaux puisse écrire des données dans votre compartiment S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
```

```
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
        }
}
```

Pour `aws:SourceAccount`, spécifiez la liste des ID de comptes pour lesquels les journaux sont transmis à ce compartiment. Pour`aws:SourceArn`, spécifiez la liste ARNs des ressources qui génèrent les journaux, dans le formulaire`arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Logs envoyés à Firehose

Cette section s'applique lorsque les types de journaux répertoriés dans le tableau de la section précédente sont envoyés à Firehose :

Autorisations des utilisateurs

Pour pouvoir configurer l'envoi de l'un de ces types de journaux à Firehose pour la première fois, vous devez être connecté à un compte avec les autorisations suivantes.

- `logs:CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam:CreateServiceLinkedRole`

Si l'un de ces types de journaux est déjà envoyé à Firehose, vous devez uniquement disposer des autorisations et pour configurer l'envoi d'un autre de ces types de journaux à Firehose. `logs:CreateLogDelivery` `firehose:TagDeliveryStream`

Rôles IAM utilisés pour les autorisations

Comme Firehose n'utilise pas de politiques de ressources, il AWS utilise les rôles IAM lors de la configuration de ces journaux à envoyer à Firehose. AWS crée un rôle lié à un service nommé. AWSServiceRoleForLogDelivery Ce rôle lié à un service comprend les autorisations suivantes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Ce rôle lié à un service accorde l'autorisation à tous les flux de diffusion Firehose dont la `LogDeliveryEnabled` balise est définie sur. `true` AWS attribue cette balise au flux de diffusion de destination lorsque vous configurez la journalisation.

Ce rôle lié à un service possède également une politique d'approbation qui permet au principal du service `delivery.logs.amazonaws.com` d'assumer le rôle lié au service nécessaire. Cette politique d'approbation est la suivante :

JSON

```
{
```

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

Journalisation nécessitant des autorisations supplémentaires [V2]

Certains AWS services utilisent une nouvelle méthode pour envoyer leurs journaux. Il s'agit d'une méthode flexible qui vous permet de configurer la livraison des journaux depuis ces services vers une ou plusieurs des destinations suivantes : CloudWatch Logs, Amazon S3 ou Firehose and X-Ray pour la livraison de traces.

La livraison d'un journal de travail comprend trois éléments :

- `ADeliverySource`, qui est un objet logique qui représente la ou les ressources qui envoient réellement les journaux.
- `ADeliveryDestination`, qui est un objet logique qui représente la destination de livraison réelle.
- `ADelivery`, qui connecte une source de livraison à une destination de livraison

Pour configurer la livraison des journaux entre un AWS service pris en charge et une destination, vous devez effectuer les opérations suivantes :

- Créez une source de diffusion avec PutDeliverySource.
- Créez une destination de livraison avec PutDeliveryDestination.
- Si vous distribuez des journaux entre comptes, vous devez les utiliser PutDeliveryDestinationPolicydans le compte de destination pour attribuer une IAM politique à la destination. Cette politique autorise la création d'une livraison depuis la source de livraison dans le compte A vers la destination de livraison dans le compte B. Pour la livraison entre comptes, vous devez créer manuellement les politiques d'autorisation vous-même.

- Créez une livraison en associant exactement une source de livraison et une destination de livraison, en utilisant CreateDelivery.

Les sections suivantes fournissent les détails des autorisations dont vous avez besoin lorsque vous êtes connecté pour configurer la livraison des journaux à chaque type de destination, à l'aide du processus V2. Ces autorisations peuvent être accordées à un rôle IAM avec lequel vous êtes connecté.

> **Important**
>
> Il est de votre responsabilité de supprimer les ressources de livraison de journaux après avoir supprimé la ressource génératrice de journaux. Pour ce faire, procédez comme suit.
>
> 1. `Delivery`Supprimez-le à l'aide de l'DeleteDeliveryopération.
> 2. `DeliverySource`Supprimez-le à l'aide de l'DeleteDeliverySourceopération.
> 3. Si le `DeliveryDestination` fichier associé à `DeliverySource` celui que vous venez de supprimer n'est utilisé que pour ce `DeliverySource` paramètre spécifique, vous pouvez le supprimer en utilisant l'DeleteDeliveryDestinationsopération.

Table des matières

- Logs envoyés à CloudWatch Logs
- Journaux envoyés à Amazon S3
  - Chiffrement côté serveur des compartiments Amazon S3
- Logs envoyés à Firehose
- Traces envoyées à X-Ray

Logs envoyés à CloudWatch Logs

Autorisations des utilisateurs

Pour activer l'envoi de CloudWatch journaux à Logs, vous devez être connecté avec les autorisations suivantes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
            "arn:aws:logs:us-east-1:444455556666:delivery-source:*",
            "arn:aws:logs:us-east-1:777788889999:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyCWL",
            "Effect": "Allow",
```

```
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:123456789012:*"
            ]
        }
    ]
}
```

Politique de ressources du groupe de journaux

Le groupe de journaux dans lequel les journaux sont envoyés doit avoir une politique de ressources qui inclut certaines autorisations. Si le groupe de journaux n'a actuellement aucune politique de ressources et que l'utilisateur qui configure la journalisation dispose des `logs:DescribeLogGroups` autorisations `logs:PutResourcePolicylogs:DescribeResourcePolicies`, et pour le groupe de journaux, crée AWS automatiquement la politique suivante pour celui-ci lorsque vous commencez à envoyer les CloudWatch journaux à Logs.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
```

```
                "arn:aws:logs:us-east-1:111122223333:log-group:my-log-group:log-
stream:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Journaux envoyés à Amazon S3

Autorisations des utilisateurs

Pour activer l'envoi de journaux à Amazon S3, vous devez être connecté avec les autorisations suivantes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
```

```
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

Le compartiment S3 où les journaux sont envoyés doit avoir une politique de ressources qui inclut certaines autorisations. Si le compartiment ne comporte aucune politique de ressources et que l'utilisateur qui configure la journalisation dispose des autorisations `S3:GetBucketPolicy` et `S3:PutBucketPolicy` pour le compartiment, AWS crée automatiquement la politique suivante lorsque vous commencez à envoyer les journaux à Amazon S3.

JSON

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:delivery-source:*"
                    ]
                }
            }
        }
    ]
}
```

Dans la politique précédente, pour `aws:SourceAccount`, spécifiez la liste des ID de comptes pour lesquels les journaux sont transmis à ce compartiment. Pour`aws:SourceArn`, spécifiez la liste ARNs des ressources qui génèrent les journaux, dans le formulaire`arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Si le compartiment dispose d'une politique de ressources, mais que celle-ci ne contient pas la déclaration indiquée dans la politique précédente, et que l'utilisateur qui configure la journalisation dispose des autorisations `S3:GetBucketPolicy` et `S3:PutBucketPolicy` pour le compartiment, cette déclaration est ajoutée à la politique de ressources du compartiment.

> **Note**
>
> Dans certains cas, des `AccessDenied` erreurs peuvent s'afficher AWS CloudTrail si l'`s3:ListBucket`autorisation n'a pas été accordée`delivery.logs.amazonaws.com`. Pour éviter ces erreurs dans vos CloudTrail journaux, vous devez accorder l'`s3:ListBucket`autorisation `delivery.logs.amazonaws.com` et inclure les `Condition` paramètres indiqués dans l'`s3:GetBucketAcl`autorisation définie dans la politique de compartiment précédente. Pour simplifier les choses, au lieu de créer un nouveau `Statement`, vous pouvez directement mettre à jour le `AWSLogDeliveryAclCheck` pour qu'il devienne `"Action": ["s3:GetBucketAcl", "s3:ListBucket"]`

Chiffrement côté serveur des compartiments Amazon S3

Vous pouvez protéger les données de votre compartiment Amazon S3 en activant le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) ou le chiffrement côté serveur avec une clé stockée dans (SSE-KMS). AWS KMS AWS Key Management Service Pour plus d'informations, consultez Protection des données à l'aide du chiffrement côté serveur.

Si vous choisissez l'option SSE-S3, aucune configuration supplémentaire n'est requise. Amazon S3 gère la clé de chiffrement.

> **Warning**
>
> Si vous choisissez SSE-KMS, vous devez utiliser une clé gérée par le client, car l'utilisation d'une clé AWS gérée n'est pas prise en charge dans ce scénario. Si vous configurez le chiffrement à l'aide d'une clé AWS gérée, les journaux seront fournis dans un format illisible.

Lorsque vous utilisez une AWS KMS clé gérée par le client, vous pouvez spécifier le nom de ressource Amazon (ARN) de la clé gérée par le client lorsque vous activez le chiffrement des compartiments. Vous devez ajouter les informations suivantes à la politique de votre clé gérée par le client (pas à la politique du compartiment S3), afin que le compte de livraison des journaux puisse écrire des données dans votre compartiment S3.

Si vous choisissez l'option SSE-KMS, vous devez utiliser une clé gérée par le client, car les clés gérées AWS ne sont pas prises en charge pour ce scénario. Lorsque vous utilisez une AWS KMS clé gérée par le client, vous pouvez spécifier le nom de ressource Amazon (ARN) de la clé gérée par

le client lorsque vous activez le chiffrement des compartiments. Vous devez ajouter les informations suivantes à la politique de votre clé gérée par le client (pas à la politique du compartiment S3), afin que le compte de livraison des journaux puisse écrire des données dans votre compartiment S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source:*"]
        }
        }
}
```

Pour `aws:SourceAccount`, spécifiez la liste des ID de comptes pour lesquels les journaux sont transmis à ce compartiment. Pour`aws:SourceArn`, spécifiez la liste ARNs des ressources qui génèrent les journaux, dans le formulaire`arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Logs envoyés à Firehose

Autorisations des utilisateurs

Pour activer l'envoi de logs à Firehose, vous devez être connecté avec les autorisations suivantes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyFH",
            "Effect": "Allow",
```

```
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/*"
            ]
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        }
    ]
}
```

Rôles IAM utilisés pour les autorisations de ressources

Comme Firehose n'utilise pas de politiques de ressources, il AWS utilise les rôles IAM lors de la configuration de ces journaux à envoyer à Firehose. AWS crée un rôle lié à un service nommé. AWSServiceRoleForLogDelivery Ce rôle lié à un service comprend les autorisations suivantes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
```

```
            },
            "Effect": "Allow"
        }
    ]
}
```

Ce rôle lié à un service accorde l'autorisation à tous les flux de diffusion Firehose dont la `LogDeliveryEnabled` balise est définie sur. `true` AWS attribue cette balise au flux de diffusion de destination lorsque vous configurez la journalisation.

Ce rôle lié à un service possède également une politique d'approbation qui permet au principal du service `delivery.logs.amazonaws.com` d'assumer le rôle lié au service nécessaire. Cette politique d'approbation est la suivante :

JSON

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

Traces envoyées à X-Ray

Autorisations des utilisateurs

Pour activer l'envoi de traces à AWS X-Ray, vous devez être connecté avec les autorisations suivantes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
                "arn:aws:logs:US East (N. Virginia):111122223333:delivery:*",
                "arn:aws:logs:US East (N. Virginia):111122223333:delivery-
source:*",
                "arn:aws:logs:US East (N. Virginia):111122223333:delivery-
destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "AllowUpdatesToResourcePolicyXRay",
            "Effect": "Allow",
            "Action": [
                "xray:PutResourcePolicy",
                "xray:ListResourcePolicies"
            ],
            "Resource": "*"
        }
    ]
}
```

Politique en matière de ressources X-Ray

Le compte de destination vers lequel les traces sont envoyées doit disposer d'une politique de ressources incluant certaines autorisations. Lorsque X-Ray n'a actuellement pas de politique de ressources et que l'utilisateur qui configure le suivi possède `xray:PutResourcePolicy` des `xray:ListResourcePolicies` autorisations dans le compte, AWS créera automatiquement la politique suivante lorsque vous commencerez à envoyer des traces à X-Ray.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite20150319",
            "Effect": "Allow",
            "Action": [
                "xray:PutTraceSegments"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:US East (N. Virginia):111122223333:delivery-
source:my-delivery-source"
```

```
                    ]
                }
            }
        }
    ]
}
```

Activer la recherche de transactions

Pour activer l'envoi de traces à X-Ray, vous devez activer la recherche de transactions.

Autorisations spécifiques au service

Outre les autorisations spécifiques à la destination répertoriées dans les sections précédentes, certains services nécessitent une autorisation explicite autorisant les clients à envoyer des journaux à partir de leurs ressources, comme couche de sécurité supplémentaire. Il autorise l'`AllowVendedLogDeliveryForResource`action pour les ressources qui vendent des journaux au sein de ce service. Pour ces services, appliquez la politique suivante et remplacez *`service`* et *`resource-type`* par les valeurs appropriées. Pour les valeurs spécifiques aux services pour ces champs, consultez la page de documentation de ces services pour les journaux vendus.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceLevelAccessForLogDelivery",
            "Effect": "Allow",
            "Action": [
                "service:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:CloudWatch Logs:us-east-1:123456789012:resource-
type/*"
        }
    ]
}
```

Autorisations spécifiques à la console

Outre les autorisations répertoriées dans les sections précédentes, si vous configurez la livraison des journaux à l'aide de la console au lieu de la APIs, vous avez également besoin des autorisations supplémentaires suivantes :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryActionsConsoleCWL",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleS3",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleFH",
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "firehose:DescribeDeliveryStream"
            ],
            "Resource": [
```

```
                "*"
            ]
        }
    ]
}
```

Exemple de livraison entre comptes

Dans cet exemple, deux comptes sont concernés. Le compte contenant la ressource génératrice de journaux est le compte A, ID :*123456789012*, et le compte contenant la ressource consommant le journal est le compte B, ID :. *111122223333*

Le compte A souhaite fournir des logs provenant de la base de Amazon Bedrock connaissances de son compte avec l'ARN arn:aws:bedrock : :knowledge base/. *us-east-1 123456789012 kb-12345678*

Pour cet exemple, le compte A a besoin des autorisations suivantes :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowVendedLogDeliveryForKnowledgeBase",
            "Effect": "Allow",
            "Action": [
                "bedrock:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:bedrock:us-east-1:123456789012:knowledge-
base/XXXXXXXXXX"
        },
        {
            "Sid": "CreateLogDeliveryPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliverySource",
                "logs:CreateDelivery"
            ],
            "Resource": [
```

```
            "arn:aws:logs:us-east-1:123456789012:delivery-source:*",
    "arn:aws:logs:us-east-1:123456789012:delivery:*",
    "arn:aws:logs:us-east-1:444455556666:delivery-destination:*"
            ]
        }
    ]
}
```

Création d'une source de livraison

Pour commencer, le compte A crée une source de diffusion à partir de sa base de connaissances de base :

```
aws logs put-delivery-source --name my-delivery-source --log-type APPLICATION_LOGS --
resource-arn arn:aws:bedrock:region:AAAAAAAAAAAA:knowledge-base/XXXXXXXXXX
```

Ensuite, le compte B doit créer la destination de livraison en utilisant l'un des flux ci-dessous :

- Configuration de la livraison vers un compartiment Amazon S3
- Configurer la diffusion vers un flux Firehose

Configuration de la livraison vers un compartiment Amazon S3

Le compte B souhaite recevoir les logs dans son compartiment S3 avec l'ARN arn:aws:s3 : ::amzn-s3-demo-bucket. Dans cet exemple, le compte B aura besoin des autorisations suivantes :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PutLogDestinationPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliveryDestination",
                "logs:PutDeliveryDestinationPolicy"
            ],
```

```
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:*"
        }
    ]
}
```

Le bucket aura besoin des autorisations suivantes dans sa politique de bucket :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogsDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/AWSLogs/123456789012/
*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                    "123456789012"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:delivery-source:my-
delivery-source"
                    ]
                }
            }
        }
    ]
}
```

Si le compartiment est chiffré avec SSE-KMS, assurez-vous que la politique de AWS KMS clés dispose des autorisations appropriées. Par exemple, si la clé KMS est`arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`, utilisez ce qui suit :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogsGenerateDataKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:BBBBBBBBBBBB:key/X",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "AAAAAAAAAAAA"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:AAAAAAAAAAAA:delivery-source:my-
delivery-source"
                    ]
                }
            }
        }
    ]
}
```

Le compte B peut ensuite créer une destination de livraison avec le compartiment S3 comme ressource de destination :

```
aws logs put-delivery-destination --name my-s3-delivery-destination --delivery-
destination-configuration "destinationResourceArn=arn:aws:s3:::amzn-s3-demo-bucket"
```

Ensuite, le compte B crée une politique de destination de livraison pour sa nouvelle destination de livraison, qui autorisera le compte A à créer un journal de livraison. La politique qui sera ajoutée à la destination de livraison nouvellement créée est la suivante :

JSON

```
    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
                "AWS": "123456789012"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:amzn-s3-demo-bucket"
        }
    ]
}
```

Cette politique sera enregistrée sur l'ordinateur du compte B car `destination-policy-s3.json` pour associer cette ressource, le compte B exécutera la commande suivante :

```
aws logs put-delivery-destination-policy --delivery-destination-name my-s3-delivery-
destination --delivery-destination-policy file://destination-policy-s3.json
```

Enfin, le compte A crée la livraison, qui lie la source de livraison dans le compte A à la destination de livraison dans le compte B.

```
aws logs create-delivery --delivery-source-name my-delivery-source --delivery-
destination-arn arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-s3-delivery-
destination
```

Configurer la diffusion vers un flux Firehose

Dans cet exemple, le compte B souhaite recevoir des logs dans son stream Firehose. Le flux Firehose possède l'ARN suivant et est configuré pour utiliser le type de flux de DirectPut diffusion :

`arn:aws:firehose:us-east-1:111122223333:deliverystream/log-delivery-stream`

Pour cet exemple, le compte B a besoin des autorisations suivantes :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFirehoseCreateSLR",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        },
        {
            "Sid": "AllowFirehoseTagging",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": "arn:aws:firehose:us-
east-1:111122223333:deliverystream/X"
        },
        {
            "Sid": "AllowFirehoseDeliveryDestination",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliveryDestination",
```

```
                "logs:PutDeliveryDestinationPolicy"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:*"
        }
    ]
}
```

Le stream Firehose doit avoir la balise `LogDeliveryEnabled` définie sur. `true`

Le compte B créera ensuite une destination de livraison avec le stream Firehose comme ressource de destination :

```
aws logs put-delivery-destination --name my-fh-delivery-destination --delivery-
destination-configuration
 "destinationResourceArn=arn:aws:firehose:region:BBBBBBBBBBBB:deliverystream/X"
```

Ensuite, le compte B crée une politique de destination de livraison pour sa nouvelle destination de livraison, qui autorisera le compte A à créer un journal de livraison. La politique à ajouter à la destination de livraison nouvellement créée est la suivante :

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
            "AWS": "123456789012"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:amzn-s3-demo-bucket"
        }
    ]
}
```

Cette politique sera enregistrée sur l'ordinateur du compte B car `destination-policy-fh.json` pour associer cette ressource, le compte B exécute la commande suivante :

```
aws logs put-delivery-destination-policy --delivery-destination-name my-fh-delivery-
destination --delivery-destination-policy file://destination-policy-fh.json
```

Enfin, le compte A crée la livraison, qui lie la source de livraison dans le compte A à la destination de livraison dans le compte B.

```
aws logs create-delivery --delivery-source-name my-delivery-source --delivery-
destination-arn arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-fh-delivery-
destination
```

Prévention du cas de figure de l'adjoint désorienté entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services auprès des principaux fournisseurs de services qui ont obtenu l'accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés `aws:SourceAccount`contextuelles `aws:SourceArn``aws:SourceOrgID`,, et de condition `aws:SourceOrgPaths`globale dans les politiques de ressources afin de limiter les autorisations que CloudWatch Logs accorde à un autre service à la ressource. Utilisez `aws:SourceArn` pour associer une seule ressource à l'accès interservice. Utilisez `aws:SourceAccount` pour permettre à n'importe quelle ressource de ce compte d'être associée à l'utilisation interservice. Utilisez `aws:SourceOrgID` pour permettre à n'importe quelle ressource de n'importe quel compte au sein d'une organisation d'être associée à l'utilisation interservice. `aws:SourceOrgPaths`À utiliser pour associer toute ressource provenant de comptes situés dans un AWS Organizations chemin à l'utilisation interservices. Pour plus d'informations sur l'utilisation et la compréhension des chemins, voir Comprendre le chemin de l' AWS Organizations entité.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous

ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (`*`) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:servicename:*:123456789012:*`.

Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser à la fois `aws:SourceAccount` et `aws:SourceArn` pour limiter les autorisations.

Pour se protéger contre le problème de l'adjoint confus à grande échelle, utilisez la clé de contexte de condition globale `aws:SourceOrgID` ou `aws:SourceOrgPaths` avec l'ID de l'organisation ou le chemin de l'organisation de la ressource dans vos politiques basées sur les ressources. Lorsque vous ajoutez, supprimez et déplacez des comptes dans votre organisation, les politiques qui contiennent la clé `aws:SourceOrgID` ou `aws:SourceOrgPaths` incluront automatiquement les bons comptes et vous n'avez pas besoin de mettre manuellement à jour les polices.

Les politiques des sections précédentes de cette page indiquent comment utiliser les clés de contexte de condition globale `aws:SourceArn` et `aws:SourceAccount` pour éviter le problème de député confus.

Exporter les données du journal vers Amazon S3

Ce chapitre fournit des informations qui vous permettent d'exporter les données de journal de vos groupes de journaux vers un compartiment Amazon S3 pour un traitement et une analyse personnalisés, ou pour les charger sur d'autres systèmes. Vous pouvez exporter vers un compartiment du même compte ou d'un autre compte.

Vous pouvez effectuer les actions suivantes :

- Exportez les données du journal vers des compartiments S3 chiffrés par SSE-KMS dans () AWS Key Management Service AWS KMS
- Exporter les données du journal vers les compartiments S3 dont le verrouillage d'objet S3 est activé avec une période de rétention

> **Note**
>
> L'exportation vers Amazon S3 n'est prise en charge que pour les groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

Nous vous recommandons de ne pas exporter régulièrement vers Amazon S3 afin d'archiver en permanence vos journaux. Dans ce cas d'utilisation, nous vous recommandons plutôt d'utiliser des abonnements. Pour plus d'informations sur les abonnements, consultezTraitement en temps réel des données du journal avec les abonnements.

Pour lancer le processus d'exportation, vous devez créer un compartiment S3 pour stocker les données de journal exportées. Vous pouvez stocker les fichiers exportés dans votre compartiment S3 et définir des règles du cycle de vie Amazon S3 pour archiver ou supprimer automatiquement les fichiers exportés.

Vous pouvez effectuer l'exportation vers des compartiments S3 qui sont chiffrés avec AES-256 ou avec SSE-KMS. L'exportation vers les compartiments chiffrés avec DSSE-KMS n'est pas prise en charge.

Vous pouvez exporter les journaux de plusieurs groupes de journaux ou de plusieurs plages de temps dans le même compartiment S3. Pour séparer les données de journal pour chaque tâche

d'exportation, vous pouvez spécifier un préfixe qui sera utilisé comme préfixe de clé Amazon S3 pour tous les objets exportés.

> **Note**
>
> Le tri temporel sur des fragments de données de journal dans un fichier exporté n'est pas garanti. Vous pouvez trier les données du fichier journal exporté à l'aide des utilitaires Linux. Par exemple, la commande utilitaire suivante permet de trier les événements dans tous les fichiers `.gz` dans un seul dossier.
>
> ```
> find . -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
> ```
>
> La commande utilitaire suivante permet de trier les fichiers .gz à partir de plusieurs sous-dossiers.
>
> ```
> find ./*/ -type f -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
> ```
>
> En outre, vous pouvez utiliser une autre commande `stdout` pour diriger la sortie triée vers un autre fichier afin de la sauvegarder.

Un délai de 12 h peut s'avérer nécessaire pour que les données des journaux deviennent disponibles pour l'exportation. Les tâches d'exportation expirent au bout de 24 heures. Si vos tâches d'exportation arrivent à expiration, réduisez l'intervalle de temps lors de la création de la tâche d'exportation.

Pour procéder à une analyse quasiment en temps réel des données des journaux, consultez plutôt Analyse des données des CloudWatch journaux avec Logs Insights ou Traitement en temps réel des données du journal avec les abonnements.

Table des matières

- Concepts
- Exporter les données du journal vers Amazon S3 à l'aide de la console
- Exportez les données du journal vers Amazon S3 à l'aide du AWS CLI
- Décrire les tâches d'exportation
- Annuler une tâche d'exportation

Concepts

Avant de commencer, familiarisez-vous avec les concepts d'exportation suivants :

log group name

Nom du groupe de journaux associé à une tâche d'exportation. Les données de journaux dans ce groupe de journaux seront exportées dans le compartiment S3 spécifié.

de (horodatage)

Horodatage requis (en nombre de millisecondes) depuis le 1er janvier 1970 00:00:00 UTC. Tous les événements du groupe de journaux qui ont été ingérés à cette date ou après cette date seront exportés.

à (horodatage)

Horodatage requis (en nombre de millisecondes) depuis le 1er janvier 1970 00:00:00 UTC. Tous les événements de journal du groupe de journaux consignés avant cette heure seront exportés.

compartiment de destination

Nom du compartiment S3 associé à une tâche d'exportation. Ce compartiment est utilisé pour exporter les données de journaux du groupe de journaux spécifié.

préfixe de destination

Attribut facultatif utilisé comme préfixe de clé Amazon S3 pour tous les objets exportés. Cela permet de créer une organisation de type dossier dans votre compartiment.

Exporter les données du journal vers Amazon S3 à l'aide de la console

Dans les exemples suivants, vous utilisez la CloudWatch console Amazon pour exporter toutes les données d'un groupe de CloudWatch journaux Amazon Logs nommé `my-log-group` vers un compartiment Amazon S3 nommé`my-exported-logs`.

L'exportation de données de journal vers des compartiments S3 qui sont chiffrés par SSE-KMS est prise en charge. L'exportation vers les compartiments chiffrés avec DSSE-KMS n'est pas prise en charge.

Les détails de la configuration de l'exportation varient selon que le compartiment Amazon S3 vers lequel vous souhaitez exporter se trouve dans le même compte que vos journaux à exporter ou dans un compte différent.

Rubriques

- Exportation vers le même compte
- Exportation intercomptes

Exportation vers le même compte

Si le compartiment Amazon S3 se trouve dans le même compte que les journaux à exporter, suivez les instructions de cette section.

Rubriques

- Étape 1 : Créer un compartiment Amazon S3
- Étape 2 : définir les autorisations d'accès
- Étape 3 : définir les autorisations sur un compartiment S3
- (Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS
- Étape 5 : créer une tâche d'exportation

Étape 1 : Créer un compartiment Amazon S3

Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Cependant, si vous souhaitez utiliser un compartiment existant, vous pouvez passer à l'étape 2.

> **Note**
>
> Le compartiment S3 doit résider dans la même région que les données du journal à exporter. CloudWatch Logs ne prend pas en charge l'exportation de données vers des compartiments S3 d'une autre région.

Pour créer un compartiment S3

1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.

2. Si nécessaire, changez la région. Dans la barre de navigation, choisissez la région dans laquelle se trouvent vos CloudWatch journaux.
3. Choisissez Créer un compartiment.
4. Dans Bucket Name (Nom de compartiment), attribuez un nom au compartiment.
5. Pour Région, sélectionnez la région dans laquelle se trouvent vos données de CloudWatch journalisation.
6. Choisissez Créer.

Étape 2 : définir les autorisations d'accès

Pour créer la tâche d'exportation à l'étape 5, vous devez être connecté avec le rôle IAM `AmazonS3ReadOnlyAccess` et disposer des autorisations suivantes :

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Étape 3 : définir les autorisations sur un compartiment S3

Par défaut, tous les objets et les compartiments S3 sont privés. Seul le propriétaire de la ressource, le Compte AWS ayant créé le compartiment, peut accéder au compartiment et aux objets qu'il contient. Le propriétaire de la ressource peut toutefois accorder des autorisations d'accès à d'autres ressources et à d'autres utilisateurs en créant une stratégie d'accès.

Lorsque vous définissez la stratégie, nous vous recommandons d'inclure une chaîne générée de façon aléatoire comme préfixe pour le compartiment, afin que seuls les flux de journaux prévus soient exportés vers le compartiment.

> **Important**
>
> Pour sécuriser davantage les exportations vers les compartiments S3, nous vous demandons désormais de spécifier la liste des comptes sources autorisés à exporter les données des journaux vers votre compartiment S3.
> Dans l'exemple suivant, la liste des comptes figurant IDs dans la `aws:SourceAccount` clé correspond aux comptes à partir desquels un utilisateur peut exporter des données de journal vers votre compartiment S3. La clé `aws:SourceArn` correspond à la ressource pour laquelle l'action est entreprise. Vous pouvez limiter cela à un groupe de journaux spécifique ou utiliser un caractère générique, comme indiqué dans cet exemple.
> Nous vous recommandons d'inclure également l'ID du compte sur lequel le compartiment S3 est créé, afin de permettre l'exportation au sein du même compte.

Pour définir des autorisations sur un compartiment Amazon S3

1. Dans la console Amazon S3, choisissez le compartiment que vous avez créé à l'étape 1.
2. Choisissez Permissions, Bucket policy.
3. Dans Bucket Policy Editor (Éditeur de stratégie de compartiment), ajoutez la politique ci-dessous. Remplacez `my-exported-logs` par le nom de votre compartiment S3. Assurez-vous de spécifier le point de terminaison correct de la région, tel que `us-west-1`, pour Principal.

   JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
      {
          "Sid": "AllowCloudWatchLogsGetBucketAcl",
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "123456789012",
                    "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:123456789012:log-group:*",
                        "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
            }
          }
      },
      {
          "Sid": "AllowCloudWatchLogsPutObject",
          "Action": "s3:PutObject",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "123456789012",
                    "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:123456789012:log-group:*",
                        "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
            }
          }
      }
```

```
    ]
}
```

4. Choisissez Save pour définir la stratégie que vous venez d'ajouter en tant que stratégie d'accès à votre compartiment. Cette politique permet à CloudWatch Logs d'exporter les données des journaux vers votre compartiment S3. Le propriétaire du compartiment dispose des autorisations d'accès complet à tous les objets exportés.

> **Warning**
>
> Si une ou plusieurs politiques sont déjà associées au bucket existant, ajoutez les instructions pour l'accès aux CloudWatch journaux à cette ou ces politiques. Nous vous recommandons d'évaluer le jeu d'autorisations obtenu pour vérifier son adéquation pour les utilisateurs appelés à accéder au compartiment.

(Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS

Cette étape n'est nécessaire que si vous exportez vers un compartiment S3 qui utilise le chiffrement côté serveur avec. AWS KMS keys Ce chiffrement est connu sous le nom de SSE-KMS.

Exportation vers un compartiment chiffré avec SSE-KMS

1. Ouvrez la AWS KMS console à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
3. Dans le panneau de navigation de gauche, choisissez Customer manages keys (Clés gérées par le client).

   Choisissez Create key (Créer une clé).
4. Pour Type de clé, choisissez Symétrique.
5. Pour Key usage (Utilisation de la clé), choisissez Encrypt and decrypt (Chiffrer et déchiffrer), puis choisissez Next (Suivant).
6. Sous Add labels (Ajouter des étiquettes) saisissez un alias pour la clé et ajoutez éventuellement une description ou des balises. Ensuite, sélectionnez Suivant.
7. Sous Key administrators (Administrateurs de clés), sélectionnez qui peut administrer cette clé, puis choisissez Next (Suivant).

8. Sous Define key usage permissions (Définir les autorisations d'utilisation des clés), n'apportez aucune modification et choisissez Next (Suivant).
9. Passez en revue vos paramètres, puis choisissez Finish (Terminer).
10. De retour sur la page Customer managed keys (Clés gérées par le client), choisissez le nom de la clé que vous venez de créer.
11. Choisissez l'onglet Key policy (Politique de clé) et choisissez Switch to policy view (Passer à la vue de la politique).
12. Dans la section Key policy (Politique de clé), choisissez Edit (Modifier).
13. Ajoutez la déclaration suivante à la liste des déclarations de politique de clé. Lorsque vous le faites, remplacez-le *Region* par la région de vos journaux et remplacez-le *account-ARN* par l'ARN du compte propriétaire de la clé KMS.

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
```

```
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

14. Sélectionnez Enregistrer les modifications.
15. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
16. Trouvez le compartiment que vous avez créé dans Étape 1 : créer un compartiment S3 et choisissez son nom.
17. Choisissez l'onglet Propriétés. Ensuite, sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).
18. Sous Server-side encryption (Chiffrement côté serveur), choisissez Enable (Activer).
19. Sous Encryption type (Type de chiffrement), choisissez AWS Key Management Service key (SSE-KMS) (Clé KMS (SSE-KMS)).
20. Choisissez Choisir parmi vos AWS KMS clés et recherchez la clé que vous avez créée.
21. Pour Bucket name (Nom du compartiment), choisissez Enable (Activer).
22. Sélectionnez Enregistrer les modifications.

Étape 5 : créer une tâche d'exportation

Dans cette étape, vous créez la tâche d'exportation pour exporter des journaux d'un groupe de journaux.

Pour exporter des données vers Amazon S3 à l'aide de la CloudWatch console

1. Connectez-vous avec les autorisations suffisantes, comme indiqué dans Étape 2 : définir les autorisations d'accès.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation, choisissez Groupes de journaux.
4. Dans l'écran Groupes de journaux choisissez le nom du groupe de journaux.
5. Choisissez Actions, Export data to Amazon S3 (Exporter les données vers Amazon S3).

6. Dans l'écran Export data to Amazon S3 (Exporter les données vers Amazon S3), sous Define data export (Définir les données à exporter), définissez la plage de temps pour les données à exporter grâce aux champs From (De) et To (À).
7. Si votre groupe de journaux a plusieurs flux de journal, vous pouvez indiquer un préfixe de flux de journal pour limiter les données du groupe de journaux à un flux spécifique. Choisissez Advanced (Avancé), puis indiquez le préfixe de flux de journal dans Stream prefix (Préfixe de flux).
8. Sous Choose S3 bucket (Choisir le compartiment S3), choisissez le compte associé au compartiment S3.
9. Pour S3 bucket name (Nom du compartiment S3), choisissez un compartiment S3.
10. Pour Préfixe du compartiment S3, indiquez la chaîne générée de façon aléatoire que vous avez spécifiée dans la stratégie de compartiment.
11. Choisissez Export (Exporter) pour exporter les données de journal vers Amazon S3.
12. Pour afficher l'état des données de journal que vous avez exportées vers Amazon S3, choisissez Actions, puis View all exports to Amazon S3 (Afficher tous les exports vers Amazon S3).

Exportation intercomptes

Si le compartiment Amazon S3 se trouve dans un compte différent de celui des journaux à exporter, suivez les instructions de cette section.

Rubriques

- Étape 1 : Créer un compartiment Amazon S3
- Étape 2 : définir les autorisations d'accès
- Étape 3 : définir les autorisations sur un compartiment S3
- (Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS
- Étape 5 : créer une tâche d'exportation

Étape 1 : Créer un compartiment Amazon S3

Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Cependant, si vous souhaitez utiliser un compartiment existant, vous pouvez passer à l'étape 2.

> **Note**
>
> Le compartiment S3 doit résider dans la même région que les données du journal à exporter. CloudWatch Logs ne prend pas en charge l'exportation de données vers des compartiments S3 d'une autre région.

Pour créer un compartiment S3

1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Si nécessaire, changez la région. Dans la barre de navigation, choisissez la région dans laquelle se trouvent vos CloudWatch journaux.
3. Choisissez Créer un compartiment.
4. Dans Bucket Name (Nom de compartiment), attribuez un nom au compartiment.
5. Pour Région, sélectionnez la région dans laquelle se trouvent vos données de CloudWatch journalisation.
6. Choisissez Créer.

Étape 2 : définir les autorisations d'accès

Tout d'abord, vous devez créer une nouvelle politique IAM pour permettre à CloudWatch Logs d'avoir l'`s3:PutObject`autorisation d'accéder au compartiment Amazon S3 de destination dans le compte de destination.

La politique que vous créez dépend de l'utilisation ou non du AWS KMS chiffrement par le compartiment de destination.

Pour créer une politique IAM afin d'exporter les journaux vers un compartiment Amazon S3

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Politiques.
3. Choisissez Create Policy (Créer une politique).
4. Dans la section Éditeur de politique, sélectionnez JSON.
5. Si le compartiment de destination n'utilise pas le AWS KMS chiffrement, collez la politique suivante dans l'éditeur.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Si le compartiment de destination utilise le AWS KMS chiffrement, collez la politique suivante dans l'éditeur.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}
```

6. Choisissez Suivant.

7. Entrez un nom de stratégie. Vous utiliserez ce nom pour associer la politique à votre rôle IAM.
8. Sélectionnez Créer une politique pour enregistrer la nouvelle politique.

Pour créer la tâche d'exportation à l'étape 5, vous devez vous connecter avec le rôle IAM `AmazonS3ReadOnlyAccess`. Vous devez également vous connecter avec la politique IAM que vous venez de créer et les autorisations suivantes :

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Étape 3 : définir les autorisations sur un compartiment S3

Par défaut, tous les objets et les compartiments S3 sont privés. Seul le propriétaire de la ressource, le Compte AWS ayant créé le compartiment, peut accéder au compartiment et aux objets qu'il

contient. Le propriétaire de la ressource peut toutefois accorder des autorisations d'accès à d'autres ressources et à d'autres utilisateurs en créant une stratégie d'accès.

Lorsque vous définissez la stratégie, nous vous recommandons d'inclure une chaîne générée de façon aléatoire comme préfixe pour le compartiment, afin que seuls les flux de journaux prévus soient exportés vers le compartiment.

> **Important**
>
> Pour sécuriser davantage les exportations vers les compartiments S3, nous vous demandons désormais de spécifier la liste des comptes sources autorisés à exporter les données des journaux vers votre compartiment S3.
> Dans l'exemple suivant, la liste des comptes figurant IDs dans la `aws:SourceAccount` clé correspond aux comptes à partir desquels un utilisateur peut exporter des données de journal vers votre compartiment S3. La clé `aws:SourceArn` correspond à la ressource pour laquelle l'action est entreprise. Vous pouvez limiter cela à un groupe de journaux spécifique ou utiliser un caractère générique, comme indiqué dans cet exemple.
> Nous vous recommandons d'inclure également l'ID du compte sur lequel le compartiment S3 est créé, afin de permettre l'exportation au sein du même compte.

Pour définir des autorisations sur un compartiment Amazon S3

1. Dans la console Amazon S3, choisissez le compartiment que vous avez créé à l'étape 1.
2. Choisissez Permissions, Bucket policy.
3. Dans Bucket Policy Editor (Éditeur de stratégie de compartiment), ajoutez la politique ci-dessous. Remplacez `my-exported-logs` par le nom de votre compartiment S3. Assurez-vous de spécifier le point de terminaison correct de la région, tel que `us-west-1`, pour Principal.

   JSON

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

```
          "Resource": "arn:aws:s3:::<replaceable>&example-s3-bucket;</
replaceable>",
          "Principal": { "Service": "logs.<replaceable>&region_api_default;</
replaceable>.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "<replaceable>&account_id_example;</replaceable>",
                    "<replaceable>&ExampleAWSAccountNo1;</replaceable>"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:<replaceable>&region_api_default;</
replaceable>:<replaceable>&account_id_example;</replaceable>:log-group:*",
                        "arn:aws:logs:<replaceable>&region_api_default;</
replaceable>:<replaceable>&ExampleAWSAccountNo1;</replaceable>:log-group:*"
                    ]
            }
          }
      },
      {
          "Action": "s3:PutObject",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::<replaceable>&example-s3-bucket;</
replaceable>/*",
          "Principal": { "Service": "logs.<replaceable>&region_api_default;</
replaceable>.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "<replaceable>&account_id_example;</replaceable>",
                    "<replaceable>&ExampleAWSAccountNo1;</replaceable>"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:<replaceable>&region_api_default;</
replaceable>:<replaceable>&account_id_example;</replaceable>:log-group:*",
                        "arn:aws:logs:<replaceable>&region_api_default;</
replaceable>:<replaceable>&ExampleAWSAccountNo1;</replaceable>:log-group:*"
                    ]
            }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
              "AWS": "arn:aws:iam::<replaceable>&ExampleAWSAccountNo1;</
replaceable>:role/<replaceable>role_name</replaceable>"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<replaceable>&example-s3-bucket;</
replaceable>/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
         }
    ]
}
```

4. Choisissez Save pour définir la stratégie que vous venez d'ajouter en tant que stratégie d'accès à votre compartiment. Cette politique permet à CloudWatch Logs d'exporter les données des journaux vers votre compartiment S3. Le propriétaire du compartiment dispose des autorisations d'accès complet à tous les objets exportés.

> **Warning**
>
> Si une ou plusieurs politiques sont déjà associées au bucket existant, ajoutez les instructions pour l'accès aux CloudWatch journaux à cette ou ces politiques. Nous vous recommandons d'évaluer le jeu d'autorisations obtenu pour vérifier son adéquation pour les utilisateurs appelés à accéder au compartiment.

(Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS

Cette étape n'est nécessaire que si vous exportez vers un compartiment S3 qui utilise le chiffrement côté serveur avec. AWS KMS keys Ce chiffrement est connu sous le nom de SSE-KMS.

Exportation vers un compartiment chiffré avec SSE-KMS

1. Ouvrez la AWS KMS console à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
3. Dans le panneau de navigation de gauche, choisissez Customer manages keys (Clés gérées par le client).

   Choisissez Create key (Créer une clé).
4. Pour Type de clé, choisissez Symétrique.
5. Pour Key usage (Utilisation de la clé), choisissez Encrypt and decrypt (Chiffrer et déchiffrer), puis choisissez Next (Suivant).
6. Sous Add labels (Ajouter des étiquettes) saisissez un alias pour la clé et ajoutez éventuellement une description ou des balises. Ensuite, sélectionnez Suivant.
7. Sous Key administrators (Administrateurs de clés), sélectionnez qui peut administrer cette clé, puis choisissez Next (Suivant).
8. Sous Define key usage permissions (Définir les autorisations d'utilisation des clés), n'apportez aucune modification et choisissez Next (Suivant).
9. Passez en revue vos paramètres, puis choisissez Finish (Terminer).
10. De retour sur la page Customer managed keys (Clés gérées par le client), choisissez le nom de la clé que vous venez de créer.
11. Choisissez l'onglet Key policy (Politique de clé) et choisissez Switch to policy view (Passer à la vue de la politique).
12. Dans la section Key policy (Politique de clé), choisissez Edit (Modifier).
13. Ajoutez la déclaration suivante à la liste des déclarations de politique de clé. Lorsque vous le faites, remplacez-le *Region* par la région de vos journaux et remplacez-le *account-ARN* par l'ARN du compte propriétaire de la clé KMS.

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
```

```
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id"
        }
    ]
}
```

14. Sélectionnez Enregistrer les modifications.
15. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.

16. Trouvez le compartiment que vous avez créé dans Étape 1 : créer un compartiment S3 et choisissez son nom.
17. Choisissez l'onglet Propriétés. Ensuite, sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).
18. Sous Server-side encryption (Chiffrement côté serveur), choisissez Enable (Activer).
19. Sous Encryption type (Type de chiffrement), choisissez AWS Key Management Service key (SSE-KMS) (Clé KMS (SSE-KMS)).
20. Choisissez Choisir parmi vos AWS KMS clés et recherchez la clé que vous avez créée.
21. Pour Bucket name (Nom du compartiment), choisissez Enable (Activer).
22. Sélectionnez Enregistrer les modifications.

Étape 5 : créer une tâche d'exportation

Dans cette étape, vous créez la tâche d'exportation pour exporter des journaux d'un groupe de journaux.

Pour exporter des données vers Amazon S3 à l'aide de la CloudWatch console

1. Connectez-vous avec les autorisations suffisantes, comme indiqué dans Étape 2 : définir les autorisations d'accès.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation, choisissez Groupes de journaux.
4. Dans l'écran Groupes de journaux choisissez le nom du groupe de journaux.
5. Choisissez Actions, Export data to Amazon S3 (Exporter les données vers Amazon S3).
6. Dans l'écran Export data to Amazon S3 (Exporter les données vers Amazon S3), sous Define data export (Définir les données à exporter), définissez la plage de temps pour les données à exporter grâce aux champs From (De) et To (À).
7. Si votre groupe de journaux a plusieurs flux de journal, vous pouvez indiquer un préfixe de flux de journal pour limiter les données du groupe de journaux à un flux spécifique. Choisissez Advanced (Avancé), puis indiquez le préfixe de flux de journal dans Stream prefix (Préfixe de flux).
8. Sous Choose S3 bucket (Choisir le compartiment S3), choisissez le compte associé au compartiment S3.
9. Pour S3 bucket name (Nom du compartiment S3), choisissez un compartiment S3.

10. Pour Préfixe du compartiment S3, indiquez la chaîne générée de façon aléatoire que vous avez spécifiée dans la stratégie de compartiment.
11. Choisissez Export (Exporter) pour exporter les données de journal vers Amazon S3.
12. Pour afficher l'état des données de journal que vous avez exportées vers Amazon S3, choisissez Actions, puis View all exports to Amazon S3 (Afficher tous les exports vers Amazon S3).

Exportez les données du journal vers Amazon S3 à l'aide du AWS CLI

Dans l'exemple suivant, vous utilisez une tâche d'exportation pour exporter toutes les données d'un groupe de CloudWatch journaux Logs nommé `my-log-group` vers un compartiment Amazon S3 nommé`my-exported-logs`. Cet exemple suppose que vous avez déjà créé un groupe de journaux appelé `my-log-group`.

L'exportation des données du journal vers des compartiments S3 chiffrés par AWS KMS est prise en charge. L'exportation vers les compartiments chiffrés avec DSSE-KMS n'est pas prise en charge.

Les détails de la configuration de l'exportation varient selon que le compartiment Amazon S3 vers lequel vous souhaitez exporter se trouve dans le même compte que vos journaux à exporter ou dans un compte différent.

Rubriques

- Exportation vers le même compte
- Exportation intercomptes

Exportation vers le même compte

Si le compartiment Amazon S3 se trouve dans le même compte que les journaux à exporter, suivez les instructions de cette section.

Rubriques

- Étape 1 : créer un compartiment S3
- Étape 2 : définir les autorisations d'accès
- Étape 3 : définir les autorisations sur un compartiment S3
- (Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS

- Étape 5 : créer une tâche d'exportation

Étape 1 : créer un compartiment S3

Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Cependant, si vous souhaitez utiliser un compartiment existant, vous pouvez passer à l'étape 2.

> **Note**
>
> Le compartiment S3 doit résider dans la même région que les données du journal à exporter. CloudWatch Logs ne prend pas en charge l'exportation de données vers des compartiments S3 d'une autre région.

Pour créer un compartiment S3 à l'aide du AWS CLI

À partir d'une invite de commande, exécutez la commande create-bucket suivante, où `LocationConstraint` correspond à la région dans laquelle vous exportez les données de journal.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Voici un exemple de sortie.

```
{
    "Location": "/my-exported-logs"
}
```

Étape 2 : définir les autorisations d'accès

Pour créer la tâche d'exportation à l'étape 5, vous devez être connecté avec le rôle IAM `AmazonS3ReadOnlyAccess` et disposer des autorisations suivantes :

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Étape 3 : définir les autorisations sur un compartiment S3

Par défaut, tous les objets et les compartiments S3 sont privés. Seul le propriétaire de la ressource, le compte ayant créé le compartiment, peut accéder au compartiment et aux objets qu'il contient. Le propriétaire de la ressource peut toutefois accorder des autorisations d'accès à d'autres ressources et à d'autres utilisateurs en créant une stratégie d'accès.

> **Important**
>
> Pour sécuriser davantage les exportations vers les compartiments S3, nous vous demandons désormais de spécifier la liste des comptes sources autorisés à exporter les données des journaux vers votre compartiment S3.
> Dans l'exemple suivant, la liste des comptes figurant IDs dans la `aws:SourceAccount` clé correspond aux comptes à partir desquels un utilisateur peut exporter des données de journal vers votre compartiment S3. La clé `aws:SourceArn` correspond à la ressource pour laquelle l'action est entreprise. Vous pouvez limiter cela à un groupe de journaux spécifique ou utiliser un caractère générique, comme indiqué dans cet exemple.
> Nous vous recommandons d'inclure également l'ID du compte sur lequel le compartiment S3 est créé, afin de permettre l'exportation au sein du même compte.

Définition des autorisations sur un compartiment S3

1. Créez un fichier nommé `policy.json` et ajoutez la stratégie d'accès suivante, en changeant `my-exported-logs` par le nom de votre compartiment S3 et `Principal` par le point de terminaison de la région où vous exportez les données du journal, tel que `us-west-1`. Utilisez un éditeur de texte créer ce fichier de stratégie. N'utilisez pas la console IAM.

   JSON

```
{
"Version": "2012-10-17",
"Statement": [
  {
      "Sid": "AllowGetBucketAcl",
      "Action": "s3:GetBucketAcl",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
      "Condition": {
        "StringEquals": {
            "aws:SourceAccount": [
                "123456789012",
                "111122223333"
            ]
        },
        "ArnLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:log-group:*",
                    "arn:aws:logs:us-east-1:111122223333:log-group:*"
                ]
        }
      }
  },
  {
      "Sid": "AllowPutObject",
      "Action": "s3:PutObject",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
      "Condition": {
        "StringEquals": {
```

```
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "123456789012",
                    "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:123456789012:log-group:*",
                        "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
            }
        }
    }
  ]
}
```

2. Définissez la politique que vous venez d'ajouter comme politique d'accès à votre compartiment à l'aide de la put-bucket-policycommande. Cette politique permet à CloudWatch Logs d'exporter les données des journaux vers votre compartiment S3. Le propriétaire du compartiment disposera des autorisations d'accès complet à tous les objets exportés.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> Si une ou plusieurs politiques sont déjà associées au bucket existant, ajoutez les instructions pour l'accès aux CloudWatch journaux à cette ou ces politiques. Nous vous recommandons d'évaluer le jeu d'autorisations obtenu pour vérifier son adéquation pour les utilisateurs appelés à accéder au compartiment.

(Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS

Cette étape n'est nécessaire que si vous exportez vers un compartiment S3 qui utilise le chiffrement côté serveur avec. AWS KMS keys Ce chiffrement est connu sous le nom de SSE-KMS.

Exportation vers un compartiment chiffré avec SSE-KMS

1. Utilisez un éditeur de texte pour créer un fichier nommé `key_policy.json` et ajoutez la stratégie d'accès suivante. Lorsque vous ajoutez la politique, apportez les modifications suivantes :

   - Remplacez *`Region`* par la région de vos journaux.
   - *`account-ARN`*Remplacez-le par l'ARN du compte propriétaire de la clé KMS.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

2. Entrez la commande suivante :

```
aws kms create-key --policy file://key_policy.json
```

Voici un exemple de sortie de la commande :

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. À l'aide d'un éditeur de texte, créez un fichier nommé `bucketencryption.json` avec le contenu suivant.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
```

```
  ]
}
```

4. Entrez la commande suivante, en la *bucket-name* remplaçant par le nom du compartiment vers lequel vous exportez les journaux.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

Si la commande ne renvoie pas d'erreur, le processus est réussi.

Étape 5 : créer une tâche d'exportation

Utilisez la commande suivante pour créer la tâche d'exportation. Une fois que vous l'avez créée, la tâche d'exportation peut prendre de quelques secondes à quelques heures, en fonction de la taille des données à exporter.

Pour exporter des données vers Amazon S3 à l'aide du AWS CLI

1. Connectez-vous avec les autorisations suffisantes, comme indiqué dans Étape 2 : définir les autorisations d'accès.
2. À l'invite de commandes, utilisez la create-export-taskcommande suivante pour créer la tâche d'exportation.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-
group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --
to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-
task-output"
```

Voici un exemple de sortie.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Exportation intercomptes

Si le compartiment Amazon S3 se trouve dans un compte différent de celui des journaux à exporter, suivez les instructions de cette section.

Rubriques

- Étape 1 : créer un compartiment S3
- Étape 2 : définir les autorisations d'accès
- Étape 3 : définir les autorisations sur un compartiment S3
- (Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS
- Étape 5 : créer une tâche d'exportation

Étape 1 : créer un compartiment S3

Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Cependant, si vous souhaitez utiliser un compartiment existant, vous pouvez passer à l'étape 2.

> **Note**
>
> Le compartiment S3 doit résider dans la même région que les données du journal à exporter. CloudWatch Logs ne prend pas en charge l'exportation de données vers des compartiments S3 d'une autre région.

Pour créer un compartiment S3 à l'aide du AWS CLI

À partir d'une invite de commande, exécutez la commande create-bucket suivante, où `LocationConstraint` correspond à la région dans laquelle vous exportez les données de journal.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Voici un exemple de sortie.

```
{
    "Location": "/my-exported-logs"
}
```

Étape 2 : définir les autorisations d'accès

Tout d'abord, vous devez créer une nouvelle politique IAM pour permettre à CloudWatch Logs d'avoir l'`s3:PutObject`autorisation d'accéder au compartiment Amazon S3 de destination.

Pour créer la tâche d'exportation à l'étape 5, vous devez vous connecter avec le rôle IAM `AmazonS3ReadOnlyAccess` et d'autres autorisations spécifiques. Vous pouvez créer une politique contenant certaines de ces autres autorisations nécessaires.

La politique que vous créez dépend de l'utilisation ou non du AWS KMS chiffrement par le compartiment de destination. S'il n'utilise pas AWS KMS le chiffrement, créez une politique avec le contenu suivant.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Si le compartiment de destination utilise AWS KMS le chiffrement, créez une politique avec le contenu suivant.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::>amzn-s3-demo-bucket/*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Pour créer la tâche d'exportation à l'étape 5, vous devez vous connecter avec le rôle IAM `AmazonS3ReadOnlyAccess`, la politique IAM que vous venez de créer et les autorisations suivantes :

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Étape 3 : définir les autorisations sur un compartiment S3

Par défaut, tous les objets et les compartiments S3 sont privés. Seul le propriétaire de la ressource, le compte ayant créé le compartiment, peut accéder au compartiment et aux objets qu'il contient. Le propriétaire de la ressource peut toutefois accorder des autorisations d'accès à d'autres ressources et à d'autres utilisateurs en créant une stratégie d'accès.

> **Important**
>
> Pour sécuriser davantage les exportations vers les compartiments S3, nous vous demandons désormais de spécifier la liste des comptes sources autorisés à exporter les données des journaux vers votre compartiment S3.
> Dans l'exemple suivant, la liste des comptes figurant IDs dans la `aws:SourceAccount` clé correspond aux comptes à partir desquels un utilisateur peut exporter des données de journal vers votre compartiment S3. La clé `aws:SourceArn` correspond à la ressource pour laquelle l'action est entreprise. Vous pouvez limiter cela à un groupe de journaux spécifique ou utiliser un caractère générique, comme indiqué dans cet exemple.
> Nous vous recommandons d'inclure également l'ID du compte sur lequel le compartiment S3 est créé, afin de permettre l'exportation au sein du même compte.

Définition des autorisations sur un compartiment S3

1. Créez un fichier nommé `policy.json` et ajoutez la stratégie d'accès suivante, en changeant `my-exported-logs` par le nom de votre compartiment S3 et `Principal` par le point de terminaison de la région où vous exportez les données du journal, tel que `us-west-1`. Utilisez un éditeur de texte créer ce fichier de stratégie. N'utilisez pas la console IAM.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
```

```
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                  "123456789012",
                  "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                      "arn:aws:logs:us-east-1:123456789012:log-group:*",
                      "arn:aws:logs:us-east-1:111122223333:log-group:*"
                       ]
              }
            }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                  "123456789012",
                  "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                      "arn:aws:logs:us-east-1:123456789012:log-group:*",
                      "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::>amzn-s3-demo-bucket/*",
            "Condition": {
```

```
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

2. Définissez la politique que vous venez d'ajouter comme politique d'accès à votre compartiment à l'aide de la put-bucket-policycommande. Cette politique permet à CloudWatch Logs d'exporter les données des journaux vers votre compartiment S3. Le propriétaire du compartiment disposera des autorisations d'accès complet à tous les objets exportés.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> Si une ou plusieurs politiques sont déjà associées au bucket existant, ajoutez les instructions pour l'accès aux CloudWatch journaux à cette ou ces politiques. Nous vous recommandons d'évaluer le jeu d'autorisations obtenu pour vérifier son adéquation pour les utilisateurs appelés à accéder au compartiment.

(Facultatif) Étape 4 : exportation vers un compartiment chiffré avec SSE-KMS

Cette étape n'est nécessaire que si vous exportez vers un compartiment S3 qui utilise le chiffrement côté serveur avec. AWS KMS keys Ce chiffrement est connu sous le nom de SSE-KMS.

Exportation vers un compartiment chiffré avec SSE-KMS

1. Utilisez un éditeur de texte pour créer un fichier nommé `key_policy.json` et ajoutez la stratégie d'accès suivante. Lorsque vous ajoutez la politique, apportez les modifications suivantes :

   - Remplacez *`Region`* par la région de vos journaux.
   - *`account-ARN`*Remplacez-le par l'ARN du compte propriétaire de la clé KMS.

JSON

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Sid": "AllowCWLServicePrincipalUsage",
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
        },
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "*"
    },
    {
        "Sid": "EnableIAMUserPermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
        },
        "Action": [
            "kms:GetKeyPolicy*",
            "kms:PutKeyPolicy*",
            "kms:DescribeKey*",
            "kms:CreateAlias*",
            "kms:ScheduleKeyDeletion*",
            "kms:Decrypt"
        ],
        "Resource": "*"
    },
    {
        "Sid": "EnableIAMRolePermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/role_name"
        },
        "Action": [
```

```
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

2. Entrez la commande suivante :

```
aws kms create-key --policy file://key_policy.json
```

Voici un exemple de sortie de la commande :

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. À l'aide d'un éditeur de texte, créez un fichier nommé `bucketencryption.json` avec le contenu suivant.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
```

```
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Entrez la commande suivante, en la *bucket-name* remplaçant par le nom du compartiment vers lequel vous exportez les journaux.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

Si la commande ne renvoie pas d'erreur, le processus est réussi.

Étape 5 : créer une tâche d'exportation

Utilisez la commande suivante pour créer la tâche d'exportation. Une fois que vous l'avez créée, la tâche d'exportation peut prendre de quelques secondes à quelques heures, en fonction de la taille des données à exporter.

Pour exporter des données vers Amazon S3 à l'aide du AWS CLI

1. Connectez-vous avec les autorisations suffisantes, comme indiqué dans Étape 2 : définir les autorisations d'accès.
2. À l'invite de commandes, utilisez la create-export-taskcommande suivante pour créer la tâche d'exportation.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-
group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --
to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-
task-output"
```

Voici un exemple de sortie.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Décrire les tâches d'exportation

Une fois que vous avez créé une tâche d'exportation, vous pouvez en obtenir le statut actuel.

Pour décrire les tâches d'exportation à l'aide du AWS CLI

À l'invite de commande, utilisez la describe-export-tasckscommande suivante.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id
 "cda45419-90ea-4db5-9833-aade86253e66"
```

Voici un exemple de sortie.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "tTo": 1441494000000
   }]
}
```

Vous pouvez utiliser la commande `describe-export-tasks` de trois façons :

- Sans filtres : répertorie toutes vos tâches d'exportation, dans l'ordre inverse de leur création.
- Filtrer en fonction de l'identifiant de la tâche : liste la tâche d'exportation, si elle existe, avec l'ID spécifié.
- Filtrer en fonction du statut de la tâche : liste les tâches d'exportation avec le statut spécifié.

Par exemple, utilisez la commande suivante pour filtrer sur le statut `FAILED`.

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

Voici un exemple de sortie.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

Annuler une tâche d'exportation

Vous pouvez annuler une tâche d'exportation dont l'état est `PENDING` ou `RUNNING`.

Pour annuler une tâche d'exportation à l'aide du AWS CLI

À l'invite de commande, utilisez la cancel-export-taskcommande suivante :

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-
aade86253e66"
```

Vous pouvez utiliser la describe-export-tasckscommande pour vérifier que la tâche a bien été annulée.

Streaming : CloudWatch enregistre les données vers Amazon OpenSearch Service

Vous pouvez configurer un groupe de CloudWatch journaux dans Amazon Logs afin de pouvoir diffuser des données vers votre cluster Amazon OpenSearch Service en temps quasi réel. Pour de plus amples informations, veuillez consulter Traitement en temps réel des données du journal avec les abonnements.

> **Note**
>
> Le streaming vers le OpenSearch service n'est pris en charge que pour les groupes de journaux de la classe de journaux standard. Pour plus d'informations sur les classes de log, consultezClasses de log.

En fonction de la quantité de données de journal diffusées, envisagez de définir une limite de simultanéité au niveau des fonctions. Pour plus d'informations, consultez Capacité de mise à l'échelle d'une fonction Lambda.

> **Note**
>
> Étant donné que le transfert de grandes quantités de données de CloudWatch journaux vers le OpenSearch service peut entraîner des frais d'utilisation élevés, nous vous recommandons de créer un budget dans la AWS Billing and Cost Management console. Pour plus d'informations, consultez la section Gestion de vos coûts avec AWS les budgets.

Cette section décrit les conditions préalables que vous devez remplir avant de souscrire un groupe de journaux au OpenSearch Service. Il décrit également comment abonner un groupe de journaux à OpenSearch Service.

Prérequis

Avant de commencer, créez un domaine OpenSearch de service. Le domaine peut avoir un accès publique ou un accès VPC, mais vous ne pouvez pas modifier le type d'accès une fois le domaine créé. Vous souhaiterez peut-être revoir les paramètres de votre domaine de OpenSearch service

ultérieurement et modifier la configuration de votre cluster en fonction de la quantité de données que votre cluster traitera. Pour obtenir des instructions sur la création d'un domaine, consultez la section Création OpenSearch de domaines de service.

Pour plus d'informations sur le OpenSearch service, consultez le manuel Amazon OpenSearch Service Developer Guide.

Abonnement d'un groupe de logs au OpenSearch Service

Vous pouvez utiliser la CloudWatch console pour abonner un groupe de journaux au OpenSearch Service.

Pour abonner un groupe de logs au OpenSearch Service

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Sélectionnez le nom du groupe de journaux.
4. Choisissez Actions, Filtres d'abonnement, puis Créer un filtre d'abonnement Amazon OpenSearch Service.
5. Choisissez si vous souhaitez diffuser en continu vers un cluster de ce compte ou un autre compte.
   - Si vous avez choisi ce compte, sélectionnez le domaine que vous avez créé à l'étape précédente.
   - Si vous avez choisi un autre compte, fournissez l'ARN du domaine et le point de terminaison.
6. Pour le rôle d'exécution Lambda IAM, choisissez le rôle IAM que Lambda doit utiliser lors de l'exécution d'appels vers. OpenSearch

   Le rôle IAM que vous choisissez doit répondre à ces exigences :

   - Il doit avoir `lambda.amazonaws.com` dans la relation d'approbation.
   - Il doit inclure la stratégie suivante :

     JSON

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

```
        {
            "Sid": "AllowOpenSearchStreamingAccess",
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:us-
east-1:123456789012:domain/cloudwatch-logs/*"
        }
    ]
}
```

- Si le domaine OpenSearch de service cible utilise l'accès VPC, la AWSLambdaVPCAccessExecutionRolepolitique doit être attachée au rôle. Cette politique gérée par Amazon accorde à Lambda l'accès au VPC du client, ce qui permet à Lambda d'écrire sur le point de terminaison du VPC. OpenSearch

7. Pour Log format (Format de journal), choisissez un format de journal.
8. Pour Modèle de filtre d'abonnement, tapez les termes ou le modèle à rechercher dans vos événements du journal. Cela garantit que vous n'envoyez que les données qui vous intéressent à votre OpenSearch cluster. Pour de plus amples informations, veuillez consulter Création de métriques à partir d'événements du journal à l'aide de filtres.
9. (Facultatif) Pour Select log data to test (Sélectionner les données de journal à tester), sélectionnez un flux de journal, puis choisissez Test pattern (Modèle de test) afin de vérifier que votre filtre de recherche renvoie les résultats attendus.
10. Choisissez Start streaming (Démarrer la diffusion).

Exemples de code pour CloudWatch les journaux utilisant AWS SDKs

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

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- Exemples de base pour l'utilisation de CloudWatch Logs AWS SDKs
  - Actions pour les CloudWatch journaux utilisant AWS SDKs
    - Utilisation AssociateKmsKey avec un AWS SDK
    - Utilisation CancelExportTask avec un AWS SDK
    - Utilisation CreateExportTask avec un AWS SDK
    - Utilisation CreateLogGroup avec un AWS SDK ou une CLI
    - Utilisation CreateLogStream avec un AWS SDK ou une CLI
    - Utilisation DeleteLogGroup avec un AWS SDK ou une CLI
    - Utilisation DeleteSubscriptionFilter avec un AWS SDK
    - Utilisation DescribeExportTasks avec un AWS SDK
    - Utilisation DescribeLogGroups avec un AWS SDK ou une CLI
    - Utilisation DescribeLogStreams avec un AWS SDK ou une CLI
    - Utilisation DescribeSubscriptionFilters avec un AWS SDK
    - Utilisation GetLogEvents avec un AWS SDK ou une CLI

    - Utilisation GetQueryResults avec un AWS SDK
    - Utilisation PutSubscriptionFilter avec un AWS SDK
    - Utilisation StartLiveTail avec un AWS SDK
    - Utilisation StartQuery avec un AWS SDK
- Scénarios d'utilisation CloudWatch des journaux AWS SDKs
  - Utiliser CloudWatch les journaux pour exécuter une requête volumineuse
  - Utilisent des événements planifiés pour invoquer une fonction Lambda

Exemples de base pour l'utilisation de CloudWatch Logs AWS SDKs

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

Exemples

- Actions pour les CloudWatch journaux utilisant AWS SDKs
  - Utilisation AssociateKmsKey avec un AWS SDK
  - Utilisation CancelExportTask avec un AWS SDK
  - Utilisation CreateExportTask avec un AWS SDK
  - Utilisation CreateLogGroup avec un AWS SDK ou une CLI
  - Utilisation CreateLogStream avec un AWS SDK ou une CLI
  - Utilisation DeleteLogGroup avec un AWS SDK ou une CLI
  - Utilisation DeleteSubscriptionFilter avec un AWS SDK
  - Utilisation DescribeExportTasks avec un AWS SDK
  - Utilisation DescribeLogGroups avec un AWS SDK ou une CLI
  - Utilisation DescribeLogStreams avec un AWS SDK ou une CLI
  - Utilisation DescribeSubscriptionFilters avec un AWS SDK
  - Utilisation GetLogEvents avec un AWS SDK ou une CLI
  - Utilisation GetQueryResults avec un AWS SDK
  - Utilisation PutSubscriptionFilter avec un AWS SDK
  - Utilisation StartLiveTail avec un AWS SDK

- Utilisation StartQuery avec un AWS SDK

Actions pour les CloudWatch journaux utilisant AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions de CloudWatch journalisation 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 appellent l'API CloudWatch Logs et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Vous pouvez voir les actions dans leur contexte dans Scénarios d'utilisation CloudWatch des journaux AWS SDKs .

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

Exemples

- Utilisation AssociateKmsKey avec un AWS SDK
- Utilisation CancelExportTask avec un AWS SDK
- Utilisation CreateExportTask avec un AWS SDK
- Utilisation CreateLogGroup avec un AWS SDK ou une CLI
- Utilisation CreateLogStream avec un AWS SDK ou une CLI
- Utilisation DeleteLogGroup avec un AWS SDK ou une CLI
- Utilisation DeleteSubscriptionFilter avec un AWS SDK
- Utilisation DescribeExportTasks avec un AWS SDK
- Utilisation DescribeLogGroups avec un AWS SDK ou une CLI
- Utilisation DescribeLogStreams avec un AWS SDK ou une CLI
- Utilisation DescribeSubscriptionFilters avec un AWS SDK
- Utilisation GetLogEvents avec un AWS SDK ou une CLI
- Utilisation GetQueryResults avec un AWS SDK
- Utilisation PutSubscriptionFilter avec un AWS SDK
- Utilisation StartLiveTail avec un AWS SDK
- Utilisation StartQuery avec un AWS SDK

Utilisation **`AssociateKmsKey`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`AssociateKmsKey`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to associate an AWS Key Management Service (AWS KMS) key with
    /// an Amazon CloudWatch Logs log group.
    /// </summary>
    public class AssociateKmsKey
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string kmsKeyId = "arn:aws:kms:us-west-2:<account-
number>:key/7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
            string groupName = "cloudwatchlogs-example-loggroup";

            var request = new AssociateKmsKeyRequest
            {
                KmsKeyId = kmsKeyId,
                LogGroupName = groupName,
            };
```

```
            var response = await client.AssociateKmsKeyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully associated KMS key ID:
{kmsKeyId} with log group: {groupName}.");
            }
            else
            {
                Console.WriteLine("Could not make the association between:
{kmsKeyId} and {groupName}.");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous AssociateKmsKeyà la section Référence des AWS SDK pour .NET API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CancelExportTask`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`CancelExportTask`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
```

```
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to cancel an Amazon CloudWatch Logs export task.
/// </summary>
public class CancelExportTask
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string taskId = "exampleTaskId";

        var request = new CancelExportTaskRequest
        {
            TaskId = taskId,
        };

        var response = await client.CancelExportTaskAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"{taskId} successfully canceled.");
        }
        else
        {
            Console.WriteLine($"{taskId} could not be canceled.");
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CancelExportTaskà la section Référence des AWS SDK pour .NET API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CreateExportTask`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`CreateExportTask`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Export Task to export the contents of the Amazon
    /// CloudWatch Logs to the specified Amazon Simple Storage Service (Amazon
S3)
    /// bucket.
    /// </summary>
    public class CreateExportTask
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string taskName = "export-task-example";
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string destination = "amzn-s3-demo-bucket";
            var fromTime = 1437584472382;
```

```
            var toTime = 1437584472833;

            var request = new CreateExportTaskRequest
            {
                From = fromTime,
                To = toTime,
                TaskName = taskName,
                LogGroupName = logGroupName,
                Destination = destination,
            };

            var response = await client.CreateExportTaskAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"The task, {taskName} with ID: " +
                                  $"{response.TaskId} has been created
successfully.");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateExportTaskà la section Référence des AWS SDK pour .NET API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CreateLogGroup`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `CreateLogGroup`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to create an Amazon CloudWatch Logs log group.
/// </summary>
public class CreateLogGroup
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();

        string logGroupName = "cloudwatchlogs-example-loggroup";

        var request = new CreateLogGroupRequest
        {
            LogGroupName = logGroupName,
        };

        var response = await client.CreateLogGroupAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"Successfully create log group with ID:
{logGroupName}.");
        }
```

```
            else
            {
                Console.WriteLine("Could not create log group.");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateLogGroupà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

La commande suivante crée un groupe de journaux nommé `my-logs` :

```
aws logs create-log-group --log-group-name my-logs
```

- Pour plus de détails sur l'API, voir CreateLogGroupla section Référence des AWS CLI commandes.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import { CreateLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new CreateLogGroupCommand({
    // The name of the log group.
```

```
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Pour plus de détails sur l'API, reportez-vous CreateLogGroupà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CreateLogStream`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `CreateLogStream`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
```

```
    /// Shows how to create an Amazon CloudWatch Logs stream for a CloudWatch
    /// log group.
    /// </summary>
    public class CreateLogStream
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string logStreamName = "cloudwatchlogs-example-logstream";

            var request = new CreateLogStreamRequest
            {
                LogGroupName = logGroupName,
                LogStreamName = logStreamName,
            };

            var response = await client.CreateLogStreamAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{logStreamName} successfully created for
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create stream.");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateLogStreamà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

La commande suivante crée un flux de journal nommé `20150601` dans le groupe de journaux `my-logs` :

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Pour plus de détails sur l'API, voir CreateLogStreamla section Référence des AWS CLI commandes.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteLogGroup`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DeleteLogGroup`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Uses the Amazon CloudWatch Logs Service to delete an existing
/// CloudWatch Logs log group.
/// </summary>
public class DeleteLogGroup
```

```
    {
        public static async Task Main()
        {
            var client = new AmazonCloudWatchLogsClient();
            string logGroupName = "cloudwatchlogs-example-loggroup";

            var request = new DeleteLogGroupRequest
            {
                LogGroupName = logGroupName,
            };

            var response = await client.DeleteLogGroupAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully deleted CloudWatch log group,
{logGroupName}.");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteLogGroupà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

La commande suivante supprime un groupe de journaux nommé `my-logs` :

```
aws logs delete-log-group --log-group-name my-logs
```

- Pour plus de détails sur l'API, voir DeleteLogGroupla section Référence des AWS CLI commandes.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import { DeleteLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Pour plus de détails sur l'API, reportez-vous DeleteLogGroupà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteSubscriptionFilter`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `DeleteSubscriptionFilter`.

C++

Kit de développement logiciel (SDK) for C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Joignez les fichiers requis.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DeleteSubscriptionFilterRequest.h>
#include <iostream>
```

Supprimez un filtre d'abonnement.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DeleteSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetLogGroupName(log_group);

        auto outcome = cwl.DeleteSubscriptionFilter(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Failed to delete CloudWatch log subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        } else {
            std::cout << "Successfully deleted CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous DeleteSubscriptionFilterà la section Référence des AWS SDK pour C++ API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DeleteSubscriptionFilterRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <logGroup>

                Where:
                  filter - The name of the subscription filter (for example,
 MyFilter).
                  logGroup - The name of the log group. (for example, testgroup).
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String filter = args[0];
        String logGroup = args[1];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .build();

        deleteSubFilter(logs, filter, logGroup);
        logs.close();
    }

    public static void deleteSubFilter(CloudWatchLogsClient logs, String filter,
 String logGroup) {
        try {
            DeleteSubscriptionFilterRequest request =
 DeleteSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .logGroupName(logGroup)
                    .build();

            logs.deleteSubscriptionFilter(request);
            System.out.printf("Successfully deleted CloudWatch logs subscription
 filter %s", filter);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteSubscriptionFilterà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import { DeleteSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteSubscriptionFilterCommand({
    // The name of the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Pour plus de détails sur l'API, reportez-vous DeleteSubscriptionFilterà la section Référence des AWS SDK pour JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
```

```
  filterName: "FILTER",
  logGroupName: "LOG_GROUP",
};

cwl.deleteSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK pour JavaScript.
- Pour plus de détails sur l'API, reportez-vous DeleteSubscriptionFilterà la section Référence des AWS SDK pour JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun deleteSubFilter(
    filter: String?,
    logGroup: String?,
) {
    val request =
        DeleteSubscriptionFilterRequest {
            filterName = filter
            logGroupName = logGroup
        }

    CloudWatchLogsClient.fromEnvironment { region = "us-west-2" }.use { logs ->
        logs.deleteSubscriptionFilter(request)
```

```
        println("Successfully deleted CloudWatch logs subscription filter named
 $filter")
    }
}
```

- Pour plus de détails sur l'API, consultez DeleteSubscriptionFilterla section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeExportTasks`** avec un AWS SDK

L'exemple de code suivant montre comment utiliser`DescribeExportTasks`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to retrieve a list of information about Amazon CloudWatch
/// Logs export tasks.
/// </summary>
public class DescribeExportTasks
{
    public static async Task Main()
    {
```

```
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            var request = new DescribeExportTasksRequest
            {
                Limit = 5,
            };

            var response = new DescribeExportTasksResponse();

            do
            {
                response = await client.DescribeExportTasksAsync(request);
                response.ExportTasks.ForEach(t =>
                {
                    Console.WriteLine($"{t.TaskName} with ID: {t.TaskId} has
status: {t.Status}");
                });
            }
            while (response.NextToken is not null);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeExportTasksà la section Référence des AWS SDK pour .NET API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeLogGroups`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeLogGroups`.

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Retrieves information about existing Amazon CloudWatch Logs log groups
/// and displays the information on the console.
/// </summary>
public class DescribeLogGroups
{
    public static async Task Main()
    {
        // Creates a CloudWatch Logs client using the default
        // user. If you need to work with resources in another
        // AWS Region than the one defined for the default user,
        // pass the AWS Region as a parameter to the client constructor.
        var client = new AmazonCloudWatchLogsClient();

        bool done = false;
        string newToken = null;

        var request = new DescribeLogGroupsRequest
        {
            Limit = 5,
        };

        DescribeLogGroupsResponse response;

        do
        {
            if (newToken is not null)
```

```
                {
                    request.NextToken = newToken;
                }

                response = await client.DescribeLogGroupsAsync(request);

                response.LogGroups.ForEach(lg =>
                {
                    Console.WriteLine($"{lg.LogGroupName} is associated with the
key: {lg.KmsKeyId}.");
                    Console.WriteLine($"Created on:
{lg.CreationTime.Date.Date}");
                    Console.WriteLine($"Date for this group will be stored for:
{lg.RetentionInDays} days.\n");
                });

                if (response.NextToken is null)
                {
                    done = true;
                }
                else
                {
                    newToken = response.NextToken;
                }
            }
            while (!done);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeLogGroupsà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

La commande suivante décrit un groupe de journaux nommé `my-logs` :

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

Sortie :

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeLogGroupsla section Référence des AWS CLI commandes.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import {
  paginateDescribeLogGroups,
  CloudWatchLogsClient,
} from "@aws-sdk/client-cloudwatch-logs";

const client = new CloudWatchLogsClient({});

export const main = async () => {
  const paginatedLogGroups = paginateDescribeLogGroups({ client }, {});
  const logGroups = [];

  for await (const page of paginatedLogGroups) {
    if (page.logGroups?.every((lg) => !!lg)) {
```

```
      logGroups.push(...page.logGroups);
    }
  }

  console.log(logGroups);
  return logGroups;
};
```

- Pour plus de détails sur l'API, reportez-vous DescribeLogGroupsà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeLogStreams`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeLogStreams`.

CLI

AWS CLI

La commande suivante affiche tous les flux de journaux `2015` en commençant par le préfixe du groupe `my-logs` de journaux :

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-
prefix 2015
```

Sortie :

```
{
    "logStreams": [
        {
            "creationTime": 1433189871774,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150531",
            "logStreamName": "20150531",
            "storedBytes": 0
        },
        {
```

```
            "creationTime": 1433189873898,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150601",
            "logStreamName": "20150601",
            "storedBytes": 0
        }
    ]
}
```

- Pour plus de détails sur l'API, voir DescribeLogStreamsla section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CloudWatchLogQuery {
    public static void main(final String[] args) {
        final String usage = """
                Usage:
                  <logGroupName>

                Where:
                  logGroupName - The name of the log group (for example, /aws/
lambda/ChatAIHandler).
                """;
```

```
        if (args.length != 1) {
            System.out.print(usage);
            System.exit(1);
        }

        String logGroupName = "/aws/lambda/ChatAIHandler" ; //args[0];
        CloudWatchLogsClient logsClient = CloudWatchLogsClient.builder()
                .region(Region.US_EAST_1)
                .build();

        describeMostRecentLogStream(logsClient, logGroupName);
    }

    /**
     * Describes and prints metadata about the most recent log stream in the
specified log group.
     *
     * @param logsClient   the CloudWatchLogsClient used to interact with AWS
CloudWatch Logs
     * @param logGroupName the name of the log group
     */
    public static void describeMostRecentLogStream(CloudWatchLogsClient
logsClient, String logGroupName) {
        DescribeLogStreamsRequest streamsRequest =
DescribeLogStreamsRequest.builder()
                .logGroupName(logGroupName)
                .orderBy(OrderBy.LAST_EVENT_TIME)
                .descending(true)
                .limit(1)
                .build();

        try {
            DescribeLogStreamsResponse streamsResponse =
logsClient.describeLogStreams(streamsRequest);
            List<LogStream> logStreams = streamsResponse.logStreams();

            if (logStreams.isEmpty()) {
                System.out.println("No log streams found for log group: " +
logGroupName);
                return;
            }

            LogStream stream = logStreams.get(0);
```

```
            System.out.println("Most Recent Log Stream:");
            System.out.println("  Name: " + stream.logStreamName());
            System.out.println("  ARN: " + stream.arn());
            System.out.println("  Creation Time: " + stream.creationTime());
            System.out.println("  First Event Time: " +
 stream.firstEventTimestamp());
            System.out.println("  Last Event Time: " +
 stream.lastEventTimestamp());
            System.out.println("  Stored Bytes: " + stream.storedBytes());
            System.out.println("  Upload Sequence Token: " +
 stream.uploadSequenceToken());

        } catch (CloudWatchLogsException e) {
            System.err.println("Failed to describe log stream: " +
 e.awsErrorDetails().errorMessage());
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeLogStreamsà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeSubscriptionFilters`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `DescribeSubscriptionFilters`.

C++

Kit de développement logiciel (SDK) for C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Joignez les fichiers requis.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DescribeSubscriptionFiltersRequest.h>
#include <aws/logs/model/DescribeSubscriptionFiltersResult.h>
#include <iostream>
#include <iomanip>
```

Répertoriez les filtres d'abonnement.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DescribeSubscriptionFiltersRequest request;
        request.SetLogGroupName(log_group);
        request.SetLimit(1);

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = cwl.DescribeSubscriptionFilters(
                    request);
            if (!outcome.IsSuccess()) {
                std::cout << "Failed to describe CloudWatch subscription filters
 "
                    << "for log group " << log_group << ": " <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(32) << "Name" <<
                    std::setw(64) << "FilterPattern" << std::setw(64) <<
                    "DestinationArn" << std::endl;
                header = true;
            }

            const auto &filters = outcome.GetResult().GetSubscriptionFilters();
            for (const auto &filter : filters) {
                std::cout << std::left << std::setw(32) <<
                    filter.GetFilterName() << std::setw(64) <<
                    filter.GetFilterPattern() << std::setw(64) <<
```

```
                    filter.GetDestinationArn() << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Pour plus de détails sur l'API, reportez-vous DescribeSubscriptionFiltersà la section Référence des AWS SDK pour C++ API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersResponse
import software.amazon.awssdk.services.cloudwatchlogs.model.SubscriptionFilter;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeSubscriptionFilters {
```

```
    public static void main(String[] args) {

        final String usage = """

                Usage:
                  <logGroup>

                Where:
                  logGroup - A log group name (for example, myloggroup).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String logGroup = args[0];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        describeFilters(logs, logGroup);
        logs.close();
    }

    public static void describeFilters(CloudWatchLogsClient logs, String
 logGroup) {
        try {
            boolean done = false;
            String newToken = null;

            while (!done) {
                DescribeSubscriptionFiltersResponse response;
                if (newToken == null) {
                    DescribeSubscriptionFiltersRequest request =
 DescribeSubscriptionFiltersRequest.builder()
                            .logGroupName(logGroup)
                            .limit(1).build();

                    response = logs.describeSubscriptionFilters(request);
                } else {
                    DescribeSubscriptionFiltersRequest request =
 DescribeSubscriptionFiltersRequest.builder()
                            .nextToken(newToken)
```

```
                            .logGroupName(logGroup)
                            .limit(1).build();
                    response = logs.describeSubscriptionFilters(request);
                }

                for (SubscriptionFilter filter : response.subscriptionFilters())
 {
                    System.out.printf("Retrieved filter with name %s, " +
 "pattern %s " + "and destination arn %s",
                            filter.filterName(),
                            filter.filterPattern(),
                            filter.destinationArn());
                }

                if (response.nextToken() == null) {
                    done = true;
                } else {
                    newToken = response.nextToken();
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.printf("Done");
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeSubscriptionFiltersà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import { DescribeSubscriptionFiltersCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  // This will return a list of all subscription filters in your account
  // matching the log group name.
  const command = new DescribeSubscriptionFiltersCommand({
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
    limit: 1,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Pour plus de détails sur l'API, reportez-vous DescribeSubscriptionFiltersà la section Référence des AWS SDK pour JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
```

```
  logGroupName: "GROUP_NAME",
  limit: 5,
};

cwl.describeSubscriptionFilters(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.subscriptionFilters);
  }
});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK pour JavaScript.
- Pour plus de détails sur l'API, reportez-vous DescribeSubscriptionFiltersà la section Référence des AWS SDK pour JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun describeFilters(logGroup: String) {
    val request =
        DescribeSubscriptionFiltersRequest {
            logGroupName = logGroup
            limit = 1
        }

    CloudWatchLogsClient.fromEnvironment { region = "us-west-2" }.use { cwlClient
 ->
        val response = cwlClient.describeSubscriptionFilters(request)
        response.subscriptionFilters?.forEach { filter ->
            println("Retrieved filter with name  ${filter.filterName} pattern
 ${filter.filterPattern} and destination ${filter.destinationArn}")
```

```
        }
    }
}
```

- Pour plus de détails sur l'API, consultez DescribeSubscriptionFilterslа section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetLogEvents`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `GetLogEvents`.

CLI

AWS CLI

La commande suivante extrait les événements du journal à partir d'un flux de journaux nommé `20150601` dans le groupe `my-logs` de journaux :

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

Sortie :

```
{
    "nextForwardToken":
 "f/31961209122447488583055879464742346735121166569214640130",
    "events": [
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
            "ingestionTime": 1433190516679,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
```

```
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184358,
            "message": "Example Event 2"
        }
    ],
    "nextBackwardToken":
 "b/31961209122358285602261756944988674324553373268216709120"
}
```

- Pour plus de détails sur l'API, voir GetLogEventsla section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import software.amazon.awssdk.services.cloudwatchlogs.model.GetLogEventsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class GetLogEvents {

    public static void main(String[] args) {

        final String usage = """
```

```
                Usage:
                  <logGroupName> <logStreamName>

                Where:
                  logGroupName - The name of the log group (for example,
myloggroup).
                  logStreamName - The name of the log stream (for example,
mystream).

                """;

        if (args.length != 2) {
            System.out.print(usage);
            System.exit(1);

        }

        String logGroupName = args[0];
        String logStreamName = args[1];

        Region region = Region.US_WEST_2;
        CloudWatchLogsClient cloudWatchLogsClient =
CloudWatchLogsClient.builder()
                .region(region)
                .build();

        getCWLogEvents(cloudWatchLogsClient, logGroupName, logStreamName);
        cloudWatchLogsClient.close();
    }

    public static void getCWLogEvents(CloudWatchLogsClient cloudWatchLogsClient,
String logGroupName,
            String logStreamName) {
        try {
            GetLogEventsRequest getLogEventsRequest =
GetLogEventsRequest.builder()
                    .logGroupName(logGroupName)
                    .logStreamName(logStreamName)
                    .startFromHead(true)
                    .build();

            int logLimit =
cloudWatchLogsClient.getLogEvents(getLogEventsRequest).events().size();
```

```
            for (int c = 0; c < logLimit; c++) {

 System.out.println(cloudWatchLogsClient.getLogEvents(getLogEventsRequest).events().get(c
            }

            System.out.println("Successfully got CloudWatch log events!");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetLogEventsà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetQueryResults`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `GetQueryResults`.

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 :

- Exécuter une requête volumineuse

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
```

- Pour plus de détails sur l'API, reportez-vous GetQueryResultsà la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
```

```
                return results.get("results", [])
```

- Pour plus de détails sur l'API, consultez GetQueryResultsle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutSubscriptionFilter`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `PutSubscriptionFilter`.

C++

Kit de développement logiciel (SDK) for C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Joignez les fichiers requis.

```
#include <aws/core/Aws.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/PutSubscriptionFilterRequest.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Créer le filtre d'abonnement

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::PutSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetFilterPattern(filter_pattern);
        request.SetLogGroupName(log_group);
```

```
        request.SetDestinationArn(dest_arn);
        auto outcome = cwl.PutSubscriptionFilter(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch logs subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutSubscriptionFilterà la section Référence des AWS SDK pour C++ API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.PutSubscriptionFilterRequest;

/**
 * Before running this code example, you need to grant permission to CloudWatch
 * Logs the right to execute your Lambda function.
 * To perform this task, you can use this CLI command:
 *
 * aws lambda add-permission --function-name "lamda1" --statement-id "lamda1"
```

```
 * --principal "logs.us-west-2.amazonaws.com" --action "lambda:InvokeFunction"
 * --source-arn "arn:aws:logs:us-west-2:111111111111:log-group:testgroup:*"
 * --source-account "111111111111"
 *
 * Make sure you replace the function name with your function name and replace
 * '111111111111' with your account details.
 * For more information, see "Subscription Filters with AWS Lambda" in the
 * Amazon CloudWatch Logs Guide.
 *
 *
 * Also, before running this Java V2 code example,set up your development
 * environment,including your credentials.
 *
 * For more information,see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 */

public class PutSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <pattern> <logGroup> <functionArn>\s

                Where:
                  filter - A filter name (for example, myfilter).
                  pattern - A filter pattern (for example, ERROR).
                  logGroup - A log group name (testgroup).
                  functionArn - An AWS Lambda function ARN (for example,
 arn:aws:lambda:us-west-2:111111111111:function:lambda1) .
                """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String filter = args[0];
        String pattern = args[1];
        String logGroup = args[2];
        String functionArn = args[3];
```

```
        Region region = Region.US_WEST_2;
        CloudWatchLogsClient cwl = CloudWatchLogsClient.builder()
                .region(region)
                .build();

        putSubFilters(cwl, filter, pattern, logGroup, functionArn);
        cwl.close();
    }

    public static void putSubFilters(CloudWatchLogsClient cwl,
            String filter,
            String pattern,
            String logGroup,
            String functionArn) {

        try {
            PutSubscriptionFilterRequest request =
 PutSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .filterPattern(pattern)
                    .logGroupName(logGroup)
                    .destinationArn(functionArn)
                    .build();

            cwl.putSubscriptionFilter(request);
            System.out.printf(
                    "Successfully created CloudWatch logs subscription filter
 %s",
                    filter);

        } catch (CloudWatchLogsException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutSubscriptionFilterà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import { PutSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new PutSubscriptionFilterCommand({
    // An ARN of a same-account Kinesis stream, Kinesis Firehose
    // delivery stream, or Lambda function.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
SubscriptionFilters.html
    destinationArn: process.env.CLOUDWATCH_LOGS_DESTINATION_ARN,

    // A name for the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,

    // A filter pattern for subscribing to a filtered stream of log events.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
FilterAndPatternSyntax.html
    filterPattern: process.env.CLOUDWATCH_LOGS_FILTER_PATTERN,

    // The name of the log group. Messages in this group matching the filter
 pattern
    // will be sent to the destination ARN.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};
```

```
export default run();
```

- Pour plus de détails sur l'API, reportez-vous PutSubscriptionFilterà la section Référence des AWS SDK pour JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  destinationArn: "LAMBDA_FUNCTION_ARN",
  filterName: "FILTER_NAME",
  filterPattern: "ERROR",
  logGroupName: "LOG_GROUP",
};

cwl.putSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK pour JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutSubscriptionFilterà la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez Utilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`StartLiveTail`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `StartLiveTail`.

.NET

SDK pour .NET

Joignez les fichiers requis.

```
using Amazon;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

Démarrez la session Live Tail.

```
            var client = new AmazonCloudWatchLogsClient();
            var request = new StartLiveTailRequest
            {
                LogGroupIdentifiers = logGroupIdentifiers,
                LogStreamNames = logStreamNames,
                LogEventFilterPattern = filterPattern,
            };

            var response = await client.StartLiveTailAsync(request);

            // Catch if request fails
            if (response.HttpStatusCode != System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("Failed to start live tail session");
                return;
            }
```

Vous pouvez gérer les événements de la session Live Tail de deux manières :

```
            /* Method 1
```

```
            * 1). Asynchronously loop through the event stream
            * 2). Set a timer to dispose the stream and stop the Live Tail
session at the end.
            */
            var eventStream = response.ResponseStream;
            var task = Task.Run(() =>
            {
                foreach (var item in eventStream)
                {
                    if (item is LiveTailSessionUpdate liveTailSessionUpdate)
                    {
                        foreach (var sessionResult in
liveTailSessionUpdate.SessionResults)
                        {
                            Console.WriteLine("Message : {0}",
sessionResult.Message);
                        }
                    }
                    if (item is LiveTailSessionStart)
                    {
                        Console.WriteLine("Live Tail session started");
                    }
                    // On-stream exceptions are processed here
                    if (item is CloudWatchLogsEventStreamException)
                    {
                        Console.WriteLine($"ERROR: {item}");
                    }
                }
            });
            // Close the stream to stop the session after a timeout
            if (!task.Wait(TimeSpan.FromSeconds(10))){
                eventStream.Dispose();
                Console.WriteLine("End of line");
            }
```

```
            /* Method 2
            * 1). Add event handlers to each event variable
            * 2). Start processing the stream and wait for a timeout using
AutoResetEvent
            */
            AutoResetEvent endEvent = new AutoResetEvent(false);
            var eventStream = response.ResponseStream;
```

```
            using (eventStream) // automatically disposes the stream to stop the
 session after execution finishes
            {
                eventStream.SessionStartReceived += (sender, e) =>
                {
                    Console.WriteLine("LiveTail session started");
                };
                eventStream.SessionUpdateReceived += (sender, e) =>
                {
                    foreach (LiveTailSessionLogEvent logEvent in
 e.EventStreamEvent.SessionResults){
                        Console.WriteLine("Message: {0}", logEvent.Message);
                    }
                };
                // On-stream exceptions are captured here
                eventStream.ExceptionReceived += (sender, e) =>
                {
                    Console.WriteLine($"ERROR:
 {e.EventStreamException.Message}");
                };

                eventStream.StartProcessing();
                // Stream events for this amount of time.
                endEvent.WaitOne(TimeSpan.FromSeconds(10));
                Console.WriteLine("End of line");
            }
```

- Pour plus de détails sur l'API, reportez-vous StartLiveTailà la section Référence des AWS SDK pour .NET API.

Go

Kit SDK for Go V2

Joignez les fichiers requis.

```
import (
 "context"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/config"
```

```
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)
```

Gérez les événements de la session Live Tail.

```
func handleEventStreamAsync(stream *cloudwatchlogs.StartLiveTailEventStream) {
 eventsChan := stream.Events()
 for {
  event := <-eventsChan
  switch e := event.(type) {
  case *types.StartLiveTailResponseStreamMemberSessionStart:
   log.Println("Received SessionStart event")
  case *types.StartLiveTailResponseStreamMemberSessionUpdate:
   for _, logEvent := range e.Value.SessionResults {
    log.Println(*logEvent.Message)
   }
  default:
   // Handle on-stream exceptions
   if err := stream.Err(); err != nil {
    log.Fatalf("Error occured during streaming: %v", err)
   } else if event == nil {
    log.Println("Stream is Closed")
    return
   } else {
    log.Fatalf("Unknown event type: %T", e)
   }
  }
 }
}
```

Démarrez la session Live Tail.

```
 cfg, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  panic("configuration error, " + err.Error())
 }
 client := cloudwatchlogs.NewFromConfig(cfg)

 request := &cloudwatchlogs.StartLiveTailInput{
  LogGroupIdentifiers:   logGroupIdentifiers,
```

```
 LogStreamNames:        logStreamNames,
 LogEventFilterPattern: logEventFilterPattern,
}

response, err := client.StartLiveTail(context.TODO(), request)
// Handle pre-stream Exceptions
if err != nil {
 log.Fatalf("Failed to start streaming: %v", err)
}

// Start a Goroutine to handle events over stream
stream := response.GetStream()
go handleEventStreamAsync(stream)
```

Arrêtez la session Live Tail au bout d'un certain temps.

```
// Close the stream (which ends the session) after a timeout
time.Sleep(10 * time.Second)
stream.Close()
log.Println("Event stream closed")
```

- Pour plus de détails sur l'API, reportez-vous StartLiveTailà la section Référence des AWS SDK pour Go API.

Java

SDK pour Java 2.x

Joignez les fichiers requis.

```
import io.reactivex.FlowableSubscriber;
import io.reactivex.annotations.NonNull;
import org.reactivestreams.Subscription;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsAsyncClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionStart;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionUpdate;
```

```
import software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseHandler;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseStream;

import java.util.Date;
import java.util.List;
import java.util.concurrent.atomic.AtomicReference;
```

Gérez les événements de la session Live Tail.

```
    private static StartLiveTailResponseHandler
 getStartLiveTailResponseStreamHandler(
            AtomicReference<Subscription> subscriptionAtomicReference) {
        return StartLiveTailResponseHandler.builder()
            .onResponse(r -> System.out.println("Received initial response"))
            .onError(throwable -> {
                CloudWatchLogsException e = (CloudWatchLogsException)
 throwable.getCause();
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            })
            .subscriber(() -> new FlowableSubscriber<>() {
                @Override
                public void onSubscribe(@NonNull Subscription s) {
                    subscriptionAtomicReference.set(s);
                    s.request(Long.MAX_VALUE);
                }

                @Override
                public void onNext(StartLiveTailResponseStream event) {
                    if (event instanceof LiveTailSessionStart) {
                        LiveTailSessionStart sessionStart =
 (LiveTailSessionStart) event;
                        System.out.println(sessionStart);
                    } else if (event instanceof LiveTailSessionUpdate) {
                        LiveTailSessionUpdate sessionUpdate =
 (LiveTailSessionUpdate) event;
```

```
                        List<LiveTailSessionLogEvent> logEvents =
sessionUpdate.sessionResults();
                        logEvents.forEach(e -> {
                            long timestamp = e.timestamp();
                            Date date = new Date(timestamp);
                            System.out.println("[" + date + "] " + e.message());
                        });
                    } else {
                        throw CloudWatchLogsException.builder().message("Unknown
event type").build();
                    }
                }

                @Override
                public void onError(Throwable throwable) {
                    System.out.println(throwable.getMessage());
                    System.exit(1);
                }

                @Override
                public void onComplete() {
                    System.out.println("Completed Streaming Session");
                }
            })
            .build();
    }
```

Démarrez la session Live Tail.

```
        CloudWatchLogsAsyncClient cloudWatchLogsAsyncClient =
                CloudWatchLogsAsyncClient.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();

        StartLiveTailRequest request =
                StartLiveTailRequest.builder()
                    .logGroupIdentifiers(logGroupIdentifiers)
                    .logStreamNames(logStreamNames)
                    .logEventFilterPattern(logEventFilterPattern)
                    .build();

        /* Create a reference to store the subscription */
```

```
        final AtomicReference<Subscription> subscriptionAtomicReference = new
 AtomicReference<>(null);

        cloudWatchLogsAsyncClient.startLiveTail(request,
 getStartLiveTailResponseStreamHandler(subscriptionAtomicReference));
```

Arrêtez la session Live Tail au bout d'un certain temps.

```
        /* Set a timeout for the session and cancel the subscription. This will:
         * 1). Close the stream
         * 2). Stop the Live Tail session
         */
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        if (subscriptionAtomicReference.get() != null) {
            subscriptionAtomicReference.get().cancel();
            System.out.println("Subscription to stream closed");
        }
```

- Pour plus de détails sur l'API, reportez-vous StartLiveTailà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Joignez les fichiers requis.

```
import { CloudWatchLogsClient, StartLiveTailCommand } from "@aws-sdk/client-
cloudwatch-logs";
```

Gérez les événements de la session Live Tail.

```
async function handleResponseAsync(response) {
    try {
```

```
      for await (const event of response.responseStream) {
        if (event.sessionStart !== undefined) {
          console.log(event.sessionStart);
        } else if (event.sessionUpdate !== undefined) {
          for (const logEvent of event.sessionUpdate.sessionResults) {
            const timestamp = logEvent.timestamp;
            const date = new Date(timestamp);
            console.log("[" + date + "] " + logEvent.message);
          }
        } else {
            console.error("Unknown event type");
        }
      }
    } catch (err) {
        // On-stream exceptions are captured here
        console.error(err)
    }
}
```

Démarrez la session Live Tail.

```
    const client = new CloudWatchLogsClient();

    const command = new StartLiveTailCommand({
        logGroupIdentifiers: logGroupIdentifiers,
        logStreamNames: logStreamNames,
        logEventFilterPattern: filterPattern
    });
    try{
        const response = await client.send(command);
        handleResponseAsync(response);
    } catch (err){
        // Pre-stream exceptions are captured here
        console.log(err);
    }
```

Arrêtez la session Live Tail au bout d'un certain temps.

```
    /* Set a timeout to close the client. This will stop the Live Tail session.
 */
    setTimeout(function() {
```

```
        console.log("Client timeout");
        client.destroy();
      }, 10000);
```

- Pour plus de détails sur l'API, reportez-vous StartLiveTailà la section Référence des AWS SDK pour JavaScript API.

Kotlin

SDK pour Kotlin

Joignez les fichiers requis.

```
import aws.sdk.kotlin.services.cloudwatchlogs.CloudWatchLogsClient
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailRequest
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailResponseStream
import kotlinx.coroutines.flow.takeWhile
```

Démarrez la session Live Tail.

```
    val client = CloudWatchLogsClient.fromEnvironment()

    val request = StartLiveTailRequest {
        logGroupIdentifiers = logGroupIdentifiersVal
        logStreamNames = logStreamNamesVal
        logEventFilterPattern = logEventFilterPatternVal
    }

    val startTime = System.currentTimeMillis()

    try {
        client.startLiveTail(request) { response ->
            val stream = response.responseStream
            if (stream != null) {
                /* Set a timeout to unsubcribe from the flow. This will:
                * 1). Close the stream
                * 2). Stop the Live Tail session
                */
                stream.takeWhile { System.currentTimeMillis() - startTime <
 10000 }.collect { value ->
```

```
                    if (value is StartLiveTailResponseStream.SessionStart) {
                        println(value.asSessionStart())
                    } else if (value is
StartLiveTailResponseStream.SessionUpdate) {
                        for (e in value.asSessionUpdate().sessionResults!!) {
                            println(e)
                        }
                    } else {
                        throw IllegalArgumentException("Unknown event type")
                    }
                }
            } else {
                throw IllegalArgumentException("No response stream")
            }
        }
    } catch (e: Exception) {
        println("Exception occurred during StartLiveTail: $e")
        System.exit(1)
    }
```

- Pour plus de détails sur l'API, consultez StartLiveTailla section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

Joignez les fichiers requis.

```
import boto3
import time
from datetime import datetime
```

Démarrez la session Live Tail.

```
    # Initialize the client
    client = boto3.client('logs')

    start_time = time.time()
```

```
    try:
        response = client.start_live_tail(
            logGroupIdentifiers=log_group_identifiers,
            logStreamNames=log_streams,
            logEventFilterPattern=filter_pattern
        )
        event_stream = response['responseStream']
        # Handle the events streamed back in the response
        for event in event_stream:
            # Set a timeout to close the stream.
            # This will end the Live Tail session.
            if (time.time() - start_time >= 10):
                event_stream.close()
                break
            # Handle when session is started
            if 'sessionStart' in event:
                session_start_event = event['sessionStart']
                print(session_start_event)
            # Handle when log event is given in a session update
            elif 'sessionUpdate' in event:
                log_events = event['sessionUpdate']['sessionResults']
                for log_event in log_events:
                    print('[{date}]
{log}'.format(date=datetime.fromtimestamp(log_event['timestamp']/1000),log=log_event['me
            else:
                # On-stream exceptions are captured here
                raise RuntimeError(str(event))
    except Exception as e:
        print(e)
```

- Pour plus de détails sur l'API, consultez StartLiveTaille AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`StartQuery`** avec un AWS SDK

Les exemples de code suivants illustrent comment utiliser `StartQuery`.

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 :

- Exécuter une requête volumineuse

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
 * Wrapper for the StartQueryCommand. Uses a static query string
 * for consistency.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 * @returns {Promise<{ queryId: string }>}
 */
async _startQuery([startDate, endDate], maxLogs = 10000) {
  try {
    return await this.client.send(
      new StartQueryCommand({
        logGroupNames: this.logGroupNames,
        queryString: "fields @timestamp, @message | sort @timestamp asc",
        startTime: startDate.valueOf(),
        endTime: endDate.valueOf(),
        limit: maxLogs,
      }),
    );
  } catch (err) {
    /** @type {string} */
    const message = err.message;
    if (message.startsWith("Query's end date and time")) {
      // This error indicates that the query's start or end date occur
      // before the log group was created.
      throw new DateOutOfBoundsError(message);
```

```
    }

    throw err;
  }
}
```

- Pour plus de détails sur l'API, reportez-vous StartQueryà la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
```

```
                response = client.start_query(
                    logGroupName=self.log_group,
                    startTime=start_time,
                    endTime=end_time,
                    queryString=self.query_string,
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
```

```
            )
            response = client.start_query(
                logGroupName=self.log_group,
                startTime=start_time,
                endTime=end_time,
                queryString=self.query_string,
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")
```

- Pour plus de détails sur l'API, consultez StartQuerylle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Scénarios d'utilisation CloudWatch des journaux AWS SDKs

Les exemples de code suivants vous montrent comment implémenter des scénarios courants dans CloudWatch Logs with AWS SDKs. Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions dans CloudWatch Logs ou en les combinant avec d'autres Services AWS. Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la configuration et l'exécution du code.

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

Exemples

- Utiliser CloudWatch les journaux pour exécuter une requête volumineuse
- Utilisent des événements planifiés pour invoquer une fonction Lambda

Utiliser CloudWatch les journaux pour exécuter une requête volumineuse

Les exemples de code suivants montrent comment utiliser CloudWatch les journaux pour interroger plus de 10 000 enregistrements.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

C'est le point d'entrée.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { CloudWatchLogsClient } from "@aws-sdk/client-cloudwatch-logs";
import { CloudWatchQuery } from "./cloud-watch-query.js";

console.log("Starting a recursive query...");

if (!process.env.QUERY_START_DATE || !process.env.QUERY_END_DATE) {
  throw new Error(
    "QUERY_START_DATE and QUERY_END_DATE environment variables are required.",
  );
}

const cloudWatchQuery = new CloudWatchQuery(new CloudWatchLogsClient({}), {
  logGroupNames: ["/workflows/cloudwatch-logs/large-query"],
  dateRange: [
    new Date(Number.parseInt(process.env.QUERY_START_DATE)),
    new Date(Number.parseInt(process.env.QUERY_END_DATE)),
  ],
});

await cloudWatchQuery.run();

console.log(
```

```
  `Queries finished in ${cloudWatchQuery.secondsElapsed} seconds.\nTotal logs
 found: ${cloudWatchQuery.results.length}`,
);
```

Il s'agit d'une classe qui divise les requêtes en plusieurs étapes si nécessaire.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  StartQueryCommand,
  GetQueryResultsCommand,
} from "@aws-sdk/client-cloudwatch-logs";
import { splitDateRange } from "@aws-doc-sdk-examples/lib/utils/util-date.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

class DateOutOfBoundsError extends Error {}

export class CloudWatchQuery {
  /**
   * Run a query for all CloudWatch Logs within a certain date range.
   * CloudWatch logs return a max of 10,000 results. This class
   * performs a binary search across all of the logs in the provided
   * date range if a query returns the maximum number of results.
   *
   * @param {import('@aws-sdk/client-cloudwatch-logs').CloudWatchLogsClient}
 client
   * @param {{ logGroupNames: string[], dateRange: [Date, Date], queryConfig:
 { limit: number } }} config
   */
  constructor(client, { logGroupNames, dateRange, queryConfig }) {
    this.client = client;
    /**
     * All log groups are queried.
     */
    this.logGroupNames = logGroupNames;

    /**
     * The inclusive date range that is queried.
     */
    this.dateRange = dateRange;

    /**
```

```
     * CloudWatch Logs never returns more than 10,000 logs.
     */
    this.limit = queryConfig?.limit ?? 10000;

    /**
     * @type {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]}
     */
    this.results = [];
  }

  /**
   * Run the query.
   */
  async run() {
    this.secondsElapsed = 0;
    const start = new Date();
    this.results = await this._largeQuery(this.dateRange);
    const end = new Date();
    this.secondsElapsed = (end - start) / 1000;
    return this.results;
  }

  /**
   * Recursively query for logs.
   * @param {[Date, Date]} dateRange
   * @returns {Promise<import("@aws-sdk/client-cloudwatch-logs").ResultField[]
[]>}
   */
  async _largeQuery(dateRange) {
    const logs = await this._query(dateRange, this.limit);

    console.log(
      `Query date range: ${dateRange
        .map((d) => d.toISOString())
        .join(" to ")}. Found ${logs.length} logs.`,
    );

    if (logs.length < this.limit) {
      return logs;
    }

    const lastLogDate = this._getLastLogDate(logs);
    const offsetLastLogDate = new Date(lastLogDate);
    offsetLastLogDate.setMilliseconds(lastLogDate.getMilliseconds() + 1);
```

```
    const subDateRange = [offsetLastLogDate, dateRange[1]];
    const [r1, r2] = splitDateRange(subDateRange);
    const results = await Promise.all([
      this._largeQuery(r1),
      this._largeQuery(r2),
    ]);
    return [logs, ...results].flat();
  }

  /**
   * Find the most recent log in a list of logs.
   * @param {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]} logs
   */
  _getLastLogDate(logs) {
    const timestamps = logs
      .map(
        (log) =>
          log.find((fieldMeta) => fieldMeta.field === "@timestamp")?.value,
      )
      .filter((t) => !!t)
      .map((t) => `${t}Z`)
      .sort();

    if (!timestamps.length) {
      throw new Error("No timestamp found in logs.");
    }

    return new Date(timestamps[timestamps.length - 1]);
  }

  /**
   * Simple wrapper for the GetQueryResultsCommand.
   * @param {string} queryId
   */
  _getQueryResults(queryId) {
    return this.client.send(new GetQueryResultsCommand({ queryId }));
  }

  /**
   * Starts a query and waits for it to complete.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   */
  async _query(dateRange, maxLogs) {
```

```
    try {
      const { queryId } = await this._startQuery(dateRange, maxLogs);
      const { results } = await this._waitUntilQueryDone(queryId);
      return results ?? [];
    } catch (err) {
      /**
       * This error is thrown when StartQuery returns an error indicating
       * that the query's start or end date occur before the log group was
       * created.
       */
      if (err instanceof DateOutOfBoundsError) {
        return [];
      }
      throw err;
    }
  }

  /**
   * Wrapper for the StartQueryCommand. Uses a static query string
   * for consistency.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   * @returns {Promise<{ queryId: string }>}
   */
  async _startQuery([startDate, endDate], maxLogs = 10000) {
    try {
      return await this.client.send(
        new StartQueryCommand({
          logGroupNames: this.logGroupNames,
          queryString: "fields @timestamp, @message | sort @timestamp asc",
          startTime: startDate.valueOf(),
          endTime: endDate.valueOf(),
          limit: maxLogs,
        }),
      );
    } catch (err) {
      /** @type {string} */
      const message = err.message;
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }
```

```
      throw err;
    }
  }

  /**
   * Call GetQueryResultsCommand until the query is done.
   * @param {string} queryId
   */
  _waitUntilQueryDone(queryId) {
    const getResults = async () => {
      const results = await this._getQueryResults(queryId);
      const queryDone = [
        "Complete",
        "Failed",
        "Cancelled",
        "Timeout",
        "Unknown",
      ].includes(results.status);

      return { queryDone, results };
    };

    return retry(
      { intervalInMs: 1000, maxRetries: 60, quiet: true },
      async () => {
        const { queryDone, results } = await getResults();
        if (!queryDone) {
          throw new Error("Query not done.");
        }

        return results;
      },
    );
  }
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK pour JavaScript .
  - GetQueryResults
  - StartQuery

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Ce fichier invoque un module d'exemple pour gérer les CloudWatch requêtes dépassant 10 000 résultats.

```
import logging
import os
import sys

import boto3
from botocore.config import Config

from cloudwatch_query import CloudWatchQuery
from date_utilities import DateUtilities

# Configure logging at the module level.
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(filename)s:%(lineno)d - %(message)s",
)

DEFAULT_QUERY_LOG_GROUP = "/workflows/cloudwatch-logs/large-query"


class CloudWatchLogsQueryRunner:
    def __init__(self):
        """
        Initializes the CloudWatchLogsQueryRunner class by setting up date
 utilities
        and creating a CloudWatch Logs client with retry configuration.
        """
        self.date_utilities = DateUtilities()
        self.cloudwatch_logs_client = self.create_cloudwatch_logs_client()
```

```
    def create_cloudwatch_logs_client(self):
        """
        Creates and returns a CloudWatch Logs client with a specified retry
configuration.

        :return: A CloudWatch Logs client instance.
        :rtype: boto3.client
        """
        try:
            return boto3.client("logs", config=Config(retries={"max_attempts":
10}))
        except Exception as e:
            logging.error(f"Failed to create CloudWatch Logs client: {e}")
            sys.exit(1)

    def fetch_environment_variables(self):
        """
        Fetches and validates required environment variables for query start and
end dates.
        Fetches the environment variable for log group, returning the default
value if it
        does not exist.

        :return: Tuple of query start date and end date as integers and the log
group.
        :rtype: tuple
        :raises SystemExit: If required environment variables are missing or
invalid.
        """
        try:
            query_start_date = int(os.environ["QUERY_START_DATE"])
            query_end_date = int(os.environ["QUERY_END_DATE"])
        except KeyError:
            logging.error(
                "Both QUERY_START_DATE and QUERY_END_DATE environment variables
are required."
            )
            sys.exit(1)
        except ValueError as e:
            logging.error(f"Error parsing date environment variables: {e}")
            sys.exit(1)

        try:
            log_group = os.environ["QUERY_LOG_GROUP"]
```

```
        except KeyError:
            logging.warning("No QUERY_LOG_GROUP environment variable, using
 default value")
            log_group = DEFAULT_QUERY_LOG_GROUP

        return query_start_date, query_end_date, log_group

    def convert_dates_to_iso8601(self, start_date, end_date):
        """
        Converts UNIX timestamp dates to ISO 8601 format using DateUtilities.

        :param start_date: The start date in UNIX timestamp.
        :type start_date: int
        :param end_date: The end date in UNIX timestamp.
        :type end_date: int
        :return: Start and end dates in ISO 8601 format.
        :rtype: tuple
        """
        start_date_iso8601 =
 self.date_utilities.convert_unix_timestamp_to_iso8601(
            start_date
        )
        end_date_iso8601 = self.date_utilities.convert_unix_timestamp_to_iso8601(
            end_date
        )
        return start_date_iso8601, end_date_iso8601

    def execute_query(
        self,
        start_date_iso8601,
        end_date_iso8601,
        log_group="/workflows/cloudwatch-logs/large-query",
        query="fields @timestamp, @message | sort @timestamp asc"
    ):
        """
        Creates a CloudWatchQuery instance and executes the query with provided
 date range.

        :param start_date_iso8601: The start date in ISO 8601 format.
        :type start_date_iso8601: str
        :param end_date_iso8601: The end date in ISO 8601 format.
        :type end_date_iso8601: str
        :param log_group: Log group to search: "/workflows/cloudwatch-logs/large-
query"
```

```
        :type log_group: str
        :param query: Query string to pass to the CloudWatchQuery instance
        :type query: str
        """
        cloudwatch_query = CloudWatchQuery(
            log_group=log_group,
            query_string=query
        )
        cloudwatch_query.query_logs((start_date_iso8601, end_date_iso8601))
        logging.info("Query executed successfully.")
        logging.info(
            f"Queries completed in {cloudwatch_query.query_duration} seconds.
 Total logs found: {len(cloudwatch_query.query_results)}"
        )


def main():
    """
    Main function to start a recursive CloudWatch logs query.
    Fetches required environment variables, converts dates, and executes the
 query.
    """
    logging.info("Starting a recursive CloudWatch logs query...")
    runner = CloudWatchLogsQueryRunner()
    query_start_date, query_end_date, log_group =
 runner.fetch_environment_variables()
    start_date_iso8601 = DateUtilities.convert_unix_timestamp_to_iso8601(
        query_start_date
    )
    end_date_iso8601 =
 DateUtilities.convert_unix_timestamp_to_iso8601(query_end_date)
    runner.execute_query(start_date_iso8601, end_date_iso8601,
 log_group=log_group)


if __name__ == "__main__":
    main()
```

Ce module traite les CloudWatch requêtes de plus de 10 000 résultats.

```
import logging
import time
```

```
from datetime import datetime
import threading
import boto3

from date_utilities import DateUtilities

DEFAULT_QUERY = "fields @timestamp, @message | sort @timestamp asc"
DEFAULT_LOG_GROUP = "/workflows/cloudwatch-logs/large-query"

class DateOutOfBoundsError(Exception):
    """Exception raised when the date range for a query is out of bounds."""

    pass


class CloudWatchQuery:
    """
    A class to query AWS CloudWatch logs within a specified date range.

    :vartype date_range: tuple
    :ivar limit: Maximum number of log entries to return.
    :vartype limit: int
    :log_group str: Name of the log group to query
    :query_string str: query
    """

    def __init__(self, log_group: str = DEFAULT_LOG_GROUP, query_string:
 str=DEFAULT_QUERY) -> None:
        self.lock = threading.Lock()
        self.log_group = log_group
        self.query_string = query_string
        self.query_results = []
        self.query_duration = None
        self.datetime_format = "%Y-%m-%d %H:%M:%S.%f"
        self.date_utilities = DateUtilities()
        self.limit = 10000

    def query_logs(self, date_range):
        """
        Executes a CloudWatch logs query for a specified date range and
 calculates the execution time of the query.

        :return: A batch of logs retrieved from the CloudWatch logs query.
        :rtype: list
```

```
        """
        start_time = datetime.now()

        start_date, end_date = self.date_utilities.normalize_date_range_format(
            date_range, from_format="unix_timestamp", to_format="datetime"
        )

        logging.info(
            f"Original query:"
            f"\n       START:     {start_date}"
            f"\n       END:       {end_date}"
            f"\n       LOG GROUP: {self.log_group}"
        )
        self.recursive_query((start_date, end_date))
        end_time = datetime.now()
        self.query_duration = (end_time - start_time).total_seconds()

    def recursive_query(self, date_range):
        """
        Processes logs within a given date range, fetching batches of logs
recursively if necessary.

        :param date_range: The date range to fetch logs for, specified as a tuple
(start_timestamp, end_timestamp).
        :type date_range: tuple
        :return: None if the recursive fetching is continued or stops when the
final batch of logs is processed.
                 Although it doesn't explicitly return the query results, this
method accumulates all fetched logs
                 in the `self.query_results` attribute.
        :rtype: None
        """
        batch_of_logs = self.perform_query(date_range)
        # Add the batch to the accumulated logs
        with self.lock:
            self.query_results.extend(batch_of_logs)
        if len(batch_of_logs) == self.limit:
            logging.info(f"Fetched {self.limit}, checking for more...")
            most_recent_log = self.find_most_recent_log(batch_of_logs)
            most_recent_log_timestamp = next(
                item["value"]
                for item in most_recent_log
                if item["field"] == "@timestamp"
            )
```

```
            new_range = (most_recent_log_timestamp, date_range[1])
            midpoint = self.date_utilities.find_middle_time(new_range)

            first_half_thread = threading.Thread(
                target=self.recursive_query,
                args=((most_recent_log_timestamp, midpoint),),
            )
            second_half_thread = threading.Thread(
                target=self.recursive_query, args=((midpoint, date_range[1]),)
            )

            first_half_thread.start()
            second_half_thread.start()

            first_half_thread.join()
            second_half_thread.join()

    def find_most_recent_log(self, logs):
        """
        Search a list of log items and return most recent log entry.
        :param logs: A list of logs to analyze.
        :return: log
        :type :return List containing log item details
        """
        most_recent_log = None
        most_recent_date = "1970-01-01 00:00:00.000"

        for log in logs:
            for item in log:
                if item["field"] == "@timestamp":
                    logging.debug(f"Compared: {item['value']} to
{most_recent_date}")
                    if (
                        self.date_utilities.compare_dates(
                            item["value"], most_recent_date
                        )
                        == item["value"]
                    ):
                        logging.debug(f"New most recent: {item['value']}")
                        most_recent_date = item["value"]
                        most_recent_log = log
        logging.info(f"Most recent log date of batch: {most_recent_date}")
        return most_recent_log
```

```
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_group,
                    startTime=start_time,
                    endTime=end_time,
                    queryString=self.query_string,
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []
```

```
    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_group,
                startTime=start_time,
                endTime=end_time,
                queryString=self.query_string,
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")


    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
```

```
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
  - GetQueryResults
  - StartQuery

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisent des événements planifiés pour invoquer une fonction Lambda

Les exemples de code suivants montrent comment créer une AWS Lambda fonction invoquée par un événement EventBridge planifié par Amazon.

Java

SDK pour Java 2.x

Montre comment créer un événement EventBridge planifié Amazon qui invoque une AWS Lambda fonction. Configurez EventBridge pour utiliser une expression cron afin de planifier le moment où la fonction Lambda est invoquée. Dans cet exemple, vous créez une fonction Lambda à l'aide de l'API d'exécution Lambda. Cet exemple fait appel à différents AWS services pour réaliser un cas d'utilisation spécifique. Cet exemple montre comment créer une application qui envoie un message texte mobile à vos employés pour les féliciter à leur date d'anniversaire.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- CloudWatch Journaux
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

JavaScript

SDK pour JavaScript (v3)

Montre comment créer un événement EventBridge planifié Amazon qui invoque une AWS Lambda fonction. Configurez EventBridge pour utiliser une expression cron afin de planifier le moment où la fonction Lambda est invoquée. Dans cet exemple, vous créez une fonction Lambda à l'aide de l'API d'exécution JavaScript Lambda. Cet exemple fait appel à différents AWS services pour réaliser un cas d'utilisation spécifique. Cet exemple montre comment créer une application qui envoie un message texte mobile à vos employés pour les féliciter à leur date d'anniversaire.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Cet exemple est également disponible dans le AWS SDK pour JavaScript guide du développeur v3.

Les services utilisés dans cet exemple

- CloudWatch Journaux
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

Python

SDK pour Python (Boto3)

Cet exemple montre comment enregistrer une AWS Lambda fonction en tant que cible d'un EventBridge événement Amazon planifié. Le gestionnaire Lambda écrit un message convivial et les données complètes de l'événement dans Amazon CloudWatch Logs pour une récupération ultérieure.

- Déploie une fonction Lambda.
- Crée un événement EventBridge planifié et fait de la fonction Lambda la cible.
- Accorde l'autorisation de laisser EventBridge invoquer la fonction Lambda.
- Imprime les dernières données des CloudWatch journaux pour afficher le résultat des appels planifiés.
- Nettoie toutes les ressources créées lors de la démonstration.

Il est préférable de visionner cet exemple sur GitHub. Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- CloudWatch Journaux
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch des journaux avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Sécurité dans Amazon CloudWatch Logs

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

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à WorkSpaces, voir AWS Services concernés par programme de conformitéAWS .
- 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 entreprise et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon CloudWatch Logs. Il vous explique comment configurer Amazon CloudWatch Logs pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources CloudWatch Logs.

Table des matières

- Protection des données dans Amazon CloudWatch Logs
- Gestion des identités et des accès pour Amazon CloudWatch Logs
- Validation de conformité pour Amazon CloudWatch Logs
- Résilience dans Amazon CloudWatch Logs
- Sécurité de l'infrastructure dans Amazon CloudWatch Logs
- Utilisation des CloudWatch journaux avec les points de terminaison VPC de l'interface

Protection des données dans Amazon CloudWatch Logs

> **Note**
>
> Outre les informations suivantes sur la protection générale des données dans CloudWatch Logs AWS, vous pouvez également protéger les données sensibles lors des événements du journal en les masquant. Pour de plus amples informations, veuillez consulter Aider à protéger les données sensibles des journaux grâce au masquage.

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans Amazon CloudWatch Logs. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.

- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec des CloudWatch journaux ou d'autres outils Services AWS à l'aide de la console AWS CLI, de l'API ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement au repos

CloudWatch Les journaux protègent les données au repos grâce au chiffrement. Tous les groupes de journaux sont chiffrés. Par défaut, le service CloudWatch Logs gère le chiffrement côté serveur et utilise le chiffrement côté serveur avec le mode Galois/Counter Standard Galois/Counter Mode (AES-GCM) 256 bits pour chiffrer les données des journaux au repos.

Si vous souhaitez gérer les clés utilisées pour chiffrer et déchiffrer vos journaux, utilisez des clés. AWS KMS Pour de plus amples informations, veuillez consulter Chiffrez les données du journal dans CloudWatch Logs à l'aide de AWS Key Management Service.

Chiffrement en transit

CloudWatch Les journaux utilisent end-to-end le chiffrement des données en transit. Le service CloudWatch Logs gère les clés de chiffrement côté serveur.

Gestion des identités et des accès pour Amazon CloudWatch Logs

L'accès à Amazon CloudWatch Logs nécessite des informations d'identification qui AWS peuvent être utilisées pour authentifier vos demandes. Ces informations d'identification doivent être autorisées à accéder aux AWS ressources, par exemple pour récupérer CloudWatch les données des journaux concernant vos ressources cloud. Les sections suivantes fournissent des informations détaillées sur la manière dont vous pouvez utiliser AWS Identity and Access Management (IAM) et CloudWatch les journaux pour sécuriser vos ressources en contrôlant les personnes autorisées à y accéder :

- Authentification
- Contrôle d'accès

Authentification

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Contrôle d'accès

Vous pouvez disposer d'informations d'identification valides pour authentifier vos demandes, mais vous ne pouvez pas créer de ressources de CloudWatch journaux ou y accéder si vous n'êtes pas autorisé à le faire. Par exemple, vous devez disposer d'autorisations pour créer des flux de journaux, des groupes de journaux et ainsi de suite.

Les sections suivantes décrivent comment gérer les autorisations pour les CloudWatch journaux. Nous vous recommandons de lire d'abord la présentation.

- Vue d'ensemble de la gestion des autorisations d'accès à vos ressources CloudWatch Logs
- Utilisation de politiques basées sur l'identité (politiques IAM) pour les journaux CloudWatch
- CloudWatch Référence des autorisations de journalisation

Vue d'ensemble de la gestion des autorisations d'accès à vos ressources CloudWatch Logs

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Suivez les instructions de la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

Rubriques

- CloudWatch Enregistre les ressources et les opérations
- Présentation de la propriété des ressources
- Gestion de l'accès aux ressources
- Spécification des éléments d'une politique : actions, effets et mandataires
- Spécification de conditions dans une politique

CloudWatch Enregistre les ressources et les opérations

Dans CloudWatch Logs, les principales ressources sont les groupes de journaux, les flux de journaux et les destinations. CloudWatch Les journaux ne prennent pas en charge les sous-ressources (autres ressources à utiliser avec la ressource principale).

Ces ressources et sous-ressources sont associées à des noms de ressources Amazon uniques (ARNs), comme indiqué dans le tableau suivant.

Type de ressource	Format ARN
Groupe de journaux	Les deux éléments suivants sont utilisés. Le second, avec le `:*` à la fin, est ce qui est renvoyé par la commande `describe-log-groups` CLI et l'DescribeLogGroupsAPI. arn:aws:logs : ::log-group : *region account-id log_group_name* arn:aws:logs : ::log-group : : * *region account-id log_group_name* Utilisez la première version, sans la suite `:*`, dans les situations suivantes : • Dans le champ de `logGroupIdentifier` saisie dans de nombreux cas CloudWatch Logs APIs. • Sur le `resourceArn` terrain lors du balisage APIs • Dans IAM les politiques, lorsque vous spécifiez des autorisations pour TagResourceUntagResource, et ListTagsForResource. Utilisez la deuxième version, avec la fin `:*`, pour faire référence à l'ARN lorsque vous spécifiez des autorisations dans les politiques IAM pour toutes les autres actions d'API.
Flux de journaux	arn:aws:logs : ::log-group ::log-stream : *region account-id log_group_name log-stream-name*

Type de ressource	Format ARN
Destination	arn:aws:logs : ::destination : *region account-id destination_name*

Pour plus d'informations ARNs, consultez le Guide de ARNsl'utilisateur d'IAM. Pour plus d'informations sur CloudWatch les journaux ARNs, consultez Amazon Resource Names (ARNs) dans Référence générale d'Amazon Web Services. Pour un exemple de politique couvrant les CloudWatch journaux, consultezUtilisation de politiques basées sur l'identité (politiques IAM) pour les journaux CloudWatch .

CloudWatch Logs fournit un ensemble d'opérations permettant d'utiliser les ressources CloudWatch Logs. Pour obtenir la liste des opérations disponibles, consultez CloudWatch Référence des autorisations de journalisation.

Présentation de la propriété des ressources

Le AWS compte possède les ressources créées dans le compte, quelle que soit la personne qui les a créées. Plus précisément, le propriétaire de la ressource est le AWS compte de l'entité principale (c'est-à-dire le compte root, un utilisateur ou un rôle IAM) qui authentifie la demande de création de ressource. Les exemples suivants illustrent comment cela fonctionne :

- Si vous utilisez les informations d'identification du compte root de votre AWS compte pour créer un groupe de journaux, votre AWS compte est le propriétaire de la ressource CloudWatch Logs.
- Si vous créez un utilisateur dans votre AWS compte et que vous accordez l'autorisation de créer CloudWatch des ressources Logs à cet utilisateur, celui-ci peut créer des ressources CloudWatch Logs. Cependant, votre AWS compte, auquel appartient l'utilisateur, possède les ressources CloudWatch Logs.
- Si vous créez un rôle IAM dans votre AWS compte avec les autorisations nécessaires pour créer des ressources de CloudWatch journaux, toute personne pouvant assumer ce rôle peut créer des ressources de CloudWatch journaux. Votre AWS compte, auquel appartient le rôle, possède les ressources CloudWatch Logs.

Gestion de l'accès aux ressources

Une politique d'autorisation décrit qui a accès à quoi. La section suivante explique les options disponibles pour créer des politiques d'autorisations.

> **Note**
>
> Cette section décrit l'utilisation d'IAM dans le contexte des CloudWatch journaux. Elle ne fournit pas d'informations détaillées sur le service IAM. Pour une documentation complète sur IAM, consultez la rubrique Qu'est-ce que IAM ? dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la syntaxe et les descriptions des politiques IAM, consultez la Référence des politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques associées à une identité IAM sont appelées politiques basées sur l'identité (politiques IAM) et les politiques associées à une ressource sont appelées politiques basées sur les ressources. CloudWatch Logs prend en charge les politiques basées sur l'identité et les politiques basées sur les ressources pour les destinations, qui sont utilisées pour activer les abonnements entre comptes. Pour de plus amples informations, veuillez consulter Abonnements entre comptes et entre régions.

Rubriques

- Autorisations de groupe de journaux et informations sur Contributor Insights
- Politiques basées sur les ressources

Autorisations de groupe de journaux et informations sur Contributor Insights

Contributor Insights est une fonctionnalité CloudWatch qui vous permet d'analyser les données des groupes de journaux et de créer des séries chronologiques qui affichent les données des contributeurs. Vous pouvez voir les mesures concernant les premiers contributeurs, le nombre total de contributeurs uniques et leur utilisation. Pour plus d'informations, consultez Utilisation de Contributor Insights pour analyser des données de cardinalité élevée.

Lorsque vous accordez à un utilisateur les `cloudwatch:GetInsightRuleReport` autorisations `cloudwatch:PutInsightRule` et, celui-ci peut créer une règle qui évalue n'importe quel groupe de CloudWatch journaux dans Logs, puis en voir les résultats. Les résultats peuvent contenir des données de contributeur pour ces groupes de journaux. Veillez à n'accorder ces autorisations qu'aux utilisateurs qui doivent pouvoir visualiser ces données.

Politiques basées sur les ressources

CloudWatch Logs prend en charge les politiques basées sur les ressources pour les destinations, que vous pouvez utiliser pour activer les abonnements entre comptes. Pour de plus amples informations, veuillez consulter Étape 1 : créer une destination. Les destinations peuvent être créées à l'aide de l'PutDestinationAPI, et vous pouvez ajouter une politique de ressources à la destination à l'aide de l'PutDestinationPolicyAPI. L'exemple suivant permet à un autre compte AWS avec l'ID de compte 111122223333 d'abonner ses groupes de journaux à la destination `arn:aws:logs:us-east-1:123456789012:destination:testDestination`.

JSON

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:123456789012:destination:testDestination"
    }
  ]
}
```

Spécification des éléments d'une politique : actions, effets et mandataires

Pour chaque ressource CloudWatch Logs, le service définit un ensemble d'opérations d'API. Pour accorder des autorisations pour ces opérations d'API, CloudWatch Logs définit un ensemble d'actions que vous pouvez spécifier dans une politique. Certaines opérations d'API peuvent exiger des autorisations pour plusieurs actions afin de réaliser l'opération d'API. Pour plus d'informations sur les ressources et les opérations de l'API, consultez CloudWatch Enregistre les ressources et les opérations et CloudWatch Référence des autorisations de journalisation.

Voici les éléments de base d'une politique :

- Ressource : vous utilisez un nom Amazon Resource Name (ARN) pour identifier la ressource à laquelle s'applique la politique. Pour plus d'informations, consultez CloudWatch Enregistre les ressources et les opérations.
- Action : vous utilisez des mots clés d'action pour identifier les opérations de ressource que vous voulez accorder ou refuser. Par exemple, l'autorisation `logs.DescribeLogGroups` permet à l'utilisateur d'effectuer l'opération `DescribeLogGroups`.
- Effet - Vous spécifiez l'effet produit, autorisation ou refus, lorsque l'utilisateur demande l'action spécifique. Si vous n'accordez pas explicitement l'accès pour (autoriser) une ressource, l'accès est implicitement refusé. Vous pouvez aussi explicitement refuser l'accès à une ressource, ce que vous pouvez faire afin de vous assurer qu'un utilisateur n'y a pas accès, même si une politique différente accorde l'accès.
- Principal – dans les politiques basées sur une identité (politiques IAM), l'utilisateur auquel la politique est attachée est le principal implicite. Pour les politiques basées sur les ressources, vous spécifiez l'utilisateur, le compte, le service ou toute autre entité pour lequel vous souhaitez recevoir des autorisations (s'applique uniquement aux politiques basées sur les ressources). CloudWatch Logs prend en charge les politiques basées sur les ressources pour les destinations.

Pour en savoir plus sur la syntaxe des stratégies IAM et pour obtenir des descriptions, consultez Référence de stratégie IAM AWS dans le Guide de l'utilisateur IAM.

Pour un tableau présentant toutes les actions de l'API CloudWatch Logs et les ressources auxquelles elles s'appliquent, consultezCloudWatch Référence des autorisations de journalisation.

Spécification de conditions dans une politique

Lorsque vous accordez des autorisations, vous pouvez utiliser le langage d'access policy pour spécifier les conditions définissant quand une politique doit prendre effet. Par exemple, il est possible d'appliquer une politique après seulement une date spécifique. Pour plus d'informations sur la spécification de conditions dans un langage de politique, consultez Condition dans le Guide de l'utilisateur IAM.

Pour exprimer des conditions, vous utilisez des clés de condition prédéfinies. Pour obtenir la liste des clés de contexte prises en charge par chaque AWS service et une liste des AWS clés de politique générales, consultez les sections Actions, ressources et clés de condition pour les AWS services et clés contextuelles de condition AWS globale.

> **Note**
>
> Vous pouvez utiliser des balises pour contrôler l'accès aux ressources CloudWatch des journaux, notamment aux groupes de journaux et aux destinations. L'accès aux flux de journaux est contrôlé au niveau du groupe de journaux, en raison de la relation hiérarchique entre les groupes de journaux et les flux de journaux. Pour plus d'informations sur l'utilisation d'identifications pour contrôler l'accès, consultez Contrôle de l'accès aux ressources Amazon Web Services à l'aide de balises.

Utilisation de politiques basées sur l'identité (politiques IAM) pour les journaux CloudWatch

Cette rubrique fournit des exemples de politiques basées sur une identité dans lesquelles un administrateur de compte peut attacher des politiques d'autorisation aux identités IAM (c'est-à-dire aux utilisateurs, groupes et rôles).

> **Important**
>
> Nous vous recommandons de consulter d'abord les rubriques d'introduction qui expliquent les concepts de base et les options disponibles pour gérer l'accès à vos ressources CloudWatch Logs. Pour de plus amples informations, veuillez consulter Vue d'ensemble de la gestion des autorisations d'accès à vos ressources CloudWatch Logs.

Cette rubrique aborde les points suivants :

- Autorisations requises pour utiliser la console CloudWatch
- AWS politiques gérées (prédéfinies) pour les CloudWatch journaux
- Exemples de politiques gérées par le client

Un exemple de stratégie d'autorisation est exposé ci-dessous :

JSON

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Cette stratégie comporte une instruction qui accorde des autorisations pour créer des groupes et des flux de journaux, pour télécharger des événements de journaux et pour répertorier des détails sur les flux de journaux.

Le caractère générique (*) à la fin de la valeur `Resource` signifie que l'instruction permet l'autorisation des actions `logs:CreateLogGroup`, `logs:CreateLogStream`, `logs:PutLogEvents` et `logs:DescribeLogStreams` sur tout groupe de journaux. Pour limiter cette autorisation à un groupe de journaux spécifique, remplacez le caractère générique (*) de l'ARN de la ressource par l'ARN du groupe de journaux spécifique. Pour plus d'informations sur les éléments d'une déclaration de politique IAM, consultez Références des éléments de politique IAM dans le Guide de l'utilisateur IAM. Pour obtenir la liste de toutes les actions de CloudWatch journalisation, consultezCloudWatch Référence des autorisations de journalisation.

Autorisations requises pour utiliser la console CloudWatch

Pour qu'un utilisateur puisse utiliser les CloudWatch journaux dans la CloudWatch console, il doit disposer d'un ensemble minimal d'autorisations lui permettant de décrire les autres AWS ressources de son AWS compte. Pour utiliser les CloudWatch journaux dans la CloudWatch console, vous devez disposer des autorisations des services suivants :

- CloudWatch
- CloudWatch Journaux
- OpenSearch Service

- IAM
- Kinesis
- Lambda
- Amazon S3

Si vous créez une politique IAM plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les utilisateurs dotés de cette politique IAM. Pour garantir que ces utilisateurs peuvent toujours utiliser la CloudWatch console, attachez également la politique `CloudWatchReadOnlyAccess` gérée à l'utilisateur, comme décrit dansAWS politiques gérées (prédéfinies) pour les CloudWatch journaux.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API Logs AWS CLI ou l'API CloudWatch Logs.

L'ensemble complet des autorisations requises pour utiliser la CloudWatch console pour un utilisateur qui n'utilise pas la console pour gérer les abonnements aux journaux est le suivant :

- surveillance des nuages : GetMetricData
- surveillance des nuages : ListMetrics
- journaux : CancelExportTask
- journaux : CreateExportTask
- journaux : CreateLogGroup
- journaux : CreateLogStream
- journaux : DeleteLogGroup
- journaux : DeleteLogStream
- journaux : DeleteMetricFilter
- journaux : DeleteQueryDefinition
- journaux : DeleteRetentionPolicy
- journaux : DeleteSubscriptionFilter
- journaux : DescribeExportTasks
- journaux : DescribeLogGroups
- journaux : DescribeLogStreams
- journaux : DescribeMetricFilters

- journaux : DescribeQueryDefinitions
- journaux : DescribeQueries
- journaux : DescribeSubscriptionFilters
- journaux : FilterLogEvents
- journaux : GetLogEvents
- journaux : GetLogGroupFields
- journaux : GetLogRecord
- journaux : GetQueryResults
- journaux : PutMetricFilter
- journaux : PutQueryDefinition
- journaux : PutRetentionPolicy
- journaux : StartQuery
- journaux : StopQuery
- journaux : PutSubscriptionFilter
- journaux : TestMetricFilter

Pour un utilisateur qui se servira également de la console pour gérer les abonnements aux journaux, les autorisations suivantes sont requises :

- Oui : DescribeElasticsearchDomain
- Oui : ListDomainNames
- iam : AttachRolePolicy
- iam : CreateRole
- iam : GetPolicy
- iam : GetPolicyVersion
- iam : GetRole
- iam : ListAttachedRolePolicies
- iam : ListRoles
- kinésie : DescribeStreams
- kinésie : ListStreams
- lambda : AddPermission

- lambda : CreateFunction
- lambda : GetFunctionConfiguration
- lambda : ListAliases
- lambda : ListFunctions
- lambda : ListVersionsByFunction
- lambda : RemovePermission
- s3 : ListBuckets

AWS politiques gérées (prédéfinies) pour les CloudWatch journaux

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par. AWS Les politiques gérées octroient les autorisations requises dans les cas d'utilisation courants et vous évitent d'avoir à réfléchir aux autorisations qui sont requises. Pour plus d'informations, consultez Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs et aux rôles de votre compte, sont spécifiques aux CloudWatch journaux :

- CloudWatchLogsFullAccess— Accorde un accès complet aux CloudWatch journaux.
- CloudWatchLogsReadOnlyAccess— Accorde un accès en lecture seule aux journaux. CloudWatch

CloudWatchLogsFullAccess

La CloudWatchLogsFullAccesspolitique accorde un accès complet aux CloudWatch journaux. La politique inclut les `cloudwatch:GenerateQueryResultsSummary` autorisations `cloudwatch:GenerateQuery` et, de sorte que les utilisateurs dotés de cette politique peuvent générer une chaîne de requête CloudWatch Logs Insights à partir d'une invite en langage naturel. Pour consulter le contenu complet de la politique, consultez CloudWatchLogsFullAccessle Guide de référence des politiques AWS gérées.

CloudWatchLogsReadOnlyAccess

La CloudWatchLogsReadOnlyAccesspolitique accorde un accès en lecture seule aux journaux. CloudWatch Elle inclut les `cloudwatch:GenerateQueryResultsSummary` autorisations `cloudwatch:GenerateQuery` et, de sorte que les utilisateurs soumis à cette politique peuvent générer une chaîne de requête CloudWatch Logs Insights à partir d'une invite en langage naturel.

Pour consulter le contenu complet de la politique, consultez CloudWatchLogsReadOnlyAccessle Guide de référence des politiques AWS gérées.

CloudWatchOpenSearchDashboardsFullAccess

La CloudWatchOpenSearchDashboardsFullAccesspolitique accorde l'accès pour créer, gérer et supprimer des intégrations avec OpenSearch Service, ainsi que pour créer, supprimer et gérer des tableaux de bord de journaux vendus dans ces intégrations. Pour de plus amples informations, veuillez consulter Analyser avec Amazon OpenSearch Service.

Pour consulter le contenu complet de la politique, consultez CloudWatchOpenSearchDashboardsFullAccessle Guide de référence des politiques AWS gérées.

CloudWatchOpenSearchDashboardAccess

La CloudWatchOpenSearchDashboardAccesspolitique autorise l'accès aux tableaux de bord des journaux vendus créés à l'aide Amazon OpenSearch Service d'analyses. Pour de plus amples informations, veuillez consulter Analyser avec Amazon OpenSearch Service.

> **Important**
>
> Outre l'octroi de cette politique, pour permettre à un rôle ou à un utilisateur de consulter les tableaux de bord des journaux vendus, vous devez également les spécifier lors de la création de l'intégration avec OpenSearch Service. Pour de plus amples informations, veuillez consulter Étape 1 : Création de l'intégration avec le OpenSearch service.

Pour consulter le contenu complet de la politique, consultez CloudWatchOpenSearchDashboardAccessle Guide de référence des politiques AWS gérées.

CloudWatchLogsCrossAccountSharingConfiguration

La CloudWatchLogsCrossAccountSharingConfigurationpolitique accorde l'accès à la création, à la gestion et à l'affichage des liens d'Observability Access Manager pour partager les ressources CloudWatch des journaux entre les comptes. Pour plus d'informations, consultez la section Observabilité inter-comptes de CloudWatch (français non garanti).

Pour consulter le contenu complet de la politique, consultez CloudWatchLogsCrossAccountSharingConfigurationle Guide de référence des politiques AWS gérées.

CloudWatch Enregistre les mises à jour des politiques AWS gérées

Consultez les détails des mises à jour apportées aux politiques AWS gérées pour CloudWatch les journaux depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique du document CloudWatch Logs.

Modification	Description	Date
CloudWatchLogsFullAccess - Mettre à jour vers une politique existante.	CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsFullAccess. Des autorisations pour `cloudwatch:GenerateQueryResultsSummary` ont été ajoutées pour permettre la génération d'un résumé en langage naturel des résultats de la requête.	20 mai 2025
CloudWatchLogsReadOnlyAccess - Mettre à jour vers une politique existante.	CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsReadOnlyAccess. Des autorisations pour `cloudwatch:GenerateQueryResultsSummary` ont été ajoutées pour permettre la génération d'un résumé en langage naturel des résultats de la requête.	20 mai 2025
		1er décembre 2024

Modification	Description	Date
CloudWatchLogsFullAccess - Mettre à jour vers une politique existante.	CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsFullAccess. Des autorisations pour Amazon OpenSearch Service et IAM ont été ajoutées, afin de permettre l'intégration CloudWatch des journaux au OpenSearch service pour certaines fonctionnalités.	
CloudWatchOpenSearchDashboardsFullAccess— Nouvelle politique IAM.	CloudWatch Logs a ajouté une nouvelle politique IAM, CloudWatchOpenSearchDashboardsFullAccess.- Cette politique accorde l'accès pour créer, gérer et supprimer des intégrations avec OpenSearch Service, ainsi que pour créer, gérer et supprimer des tableaux de bord de journaux vendus dans ces intégrations. Pour de plus amples informations, veuillez consulter Analyser avec Amazon OpenSearch Service.	1er décembre 2024

Modification	Description	Date
CloudWatchOpenSearchDashboardAccess— Nouvelle politique IAM.	CloudWatch Logs a ajouté une nouvelle politique IAM, CloudWatchOpenSearchDashboardAccess.- Cette politique permet d'accéder aux tableaux de bord des journaux vendus alimentés par. Amazon OpenSearch Service Pour de plus amples informations, veuillez consulter Analyser avec Amazon OpenSearch Service.	1er décembre 2024
CloudWatchLogsFullAccess - Mettre à jour vers une politique existante.	CloudWatch Logs a ajouté une autorisation à CloudWatchLogsFullAccess. L'`cloudwatch:GenerateQuery` autorisation a été ajoutée afin que les utilisateurs soumis à cette politique puissent générer une chaîne de requête CloudWatch Logs Insights à partir d'une invite en langage naturel.	27 novembre 2023

Modification	Description	Date
CloudWatchLogsReadOnlyAccess - Mettre à jour vers une politique existante.	CloudWatch a ajouté une autorisation à CloudWatchLogsReadOnlyAccess. L'`cloudwatch:GenerateQuery` autorisation a été ajoutée afin que les utilisateurs soumis à cette politique puissent générer une chaîne de requête CloudWatch Logs Insights à partir d'une invite en langage naturel.	27 novembre 2023
CloudWatchLogsReadOnlyAccess – Mise à jour d'une politique existante	CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsReadOnlyAccess. Les `logs:StopLiveTail` autorisations `logs:StartLiveTail` et ont été ajoutées afin que les utilisateurs soumis à cette politique puissent utiliser la console pour démarrer et arrêter CloudWatch les sessions Logs Live Tail. Pour plus d'informations, veuillez consulter Utilisation de Live Tail pour visualiser les journaux en temps quasi réel.	6 juin 2023

Modification	Description	Date
CloudWatchLogsCrossAccountSharingConfiguration : nouvelle politique	CloudWatch Logs a ajouté une nouvelle politique pour vous permettre de gérer les liens d'observabilité CloudWatch entre comptes qui partagent des groupes de CloudWatch journaux Logs. Pour plus d'informations, consultez la CloudWatch section Observabilité entre comptes	27 novembre 2022
CloudWatchLogsReadOnlyAccess – Mise à jour d'une politique existante	CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsReadOnlyAccess. Les `oam:ListAttachedLinks` autorisations `oam:ListSinks` et ont été ajoutées afin que les utilisateurs soumis à cette politique puissent utiliser la console pour consulter les données partagées à partir de comptes sources dans le cadre d'une CloudWatch observabilité entre comptes.	27 novembre 2022

Exemples de politiques gérées par le client

Vous pouvez créer vos propres politiques IAM personnalisées pour autoriser les actions et les ressources des CloudWatch journaux. Vous pouvez attacher ces stratégies personnalisées aux utilisateurs ou groupes qui nécessitent ces autorisations.

Dans cette section, vous trouverez des exemples de politiques utilisateur qui accordent des autorisations pour diverses actions de CloudWatch journalisation. Ces politiques fonctionnent lorsque vous utilisez l'API CloudWatch Logs AWS SDKs, ou le AWS CLI.

Exemples

- Exemple 1 : Autoriser l'accès complet aux CloudWatch journaux
- Exemple 2 : Autoriser l'accès en lecture seule aux journaux CloudWatch
- Exemple 3 : Autoriser l'accès à un groupe de journaux

Exemple 1 : Autoriser l'accès complet aux CloudWatch journaux

La politique suivante permet à un utilisateur d'accéder à toutes les actions CloudWatch des journaux.

JSON

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

Exemple 2 : Autoriser l'accès en lecture seule aux journaux CloudWatch

AWS fournit une CloudWatchLogsReadOnlyAccesspolitique qui permet l'accès en lecture seule aux CloudWatch données des journaux. Cette politique inclut les autorisations suivantes.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Exemple 3 : Autoriser l'accès à un groupe de journaux

La stratégie suivante permet à un utilisateur de lire et d'écrire des événements de journaux dans un groupe de journaux spécifié.

> **Important**
>
> Le `:*` à la fin du nom du groupe de journaux dans la ligne `Resource` est obligatoire pour indiquer que la stratégie s'applique à tous les flux de journaux de ce groupe de journaux. Si vous omettez `:*`, la politique ne sera pas appliquée.

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-
group:SampleLogGroupName:*"
      }
   ]
}
```

Utilisation de l'étiquetage, et des politiques IAM pour le contrôle au niveau du groupe de journaux

Vous pouvez accorder aux utilisateurs l'accès à certains groupes de journaux tout en leur empêchant d'accéder à d'autres groupes de journaux. Pour ce faire, étiquetez vos groupes de journaux et utilisez les politiques IAM qui font référence à ces identifications. Pour appliquer des balises à un groupe de journaux, vous devez disposer de l'autorisation `logs:TagResource` ou `logs:TagLogGroup`. Cela s'applique à la fois si vous attribuez des balises au groupe de journaux lorsque vous le créez ou si vous les attribuez ultérieurement.

Pour plus d'informations sur le balisage des groupes de journaux, consultez Étiqueter les groupes de journaux dans Amazon CloudWatch Logs.

Lorsque vous étiquetez les groupes de journaux, vous pouvez ensuite accorder une politique IAM à un utilisateur pour autoriser l'accès uniquement aux groupes de journaux associés à une identification particulière. Par exemple, la déclaration de stratégie suivante accorde l'accès uniquement aux groupes de journaux avec la valeur de `Green` pour la clé de balise `Team`.

JSON

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

```
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

Les opérations StopQueryet StopLiveTailAPI n'interagissent pas avec les AWS ressources au sens traditionnel du terme. Elles ne renvoient aucune donnée, ne mettent aucune donnée et ne modifient aucune ressource de quelque façon que ce soit. Au lieu de cela, ils ne fonctionnent que sur une session de live tail donnée ou une requête CloudWatch Logs Insights donnée, qui ne sont pas classées dans la catégorie des ressources. Par conséquent, lorsque vous spécifiez le champ `Resource` dans les politiques IAM pour ces opérations, vous devez définir la valeur du champ `Resource` en tant que `*`, comme dans l'exemple de commande suivant.

JSON

```
{
    "Version": "2012-10-17",
    "Statement":
        [ {
            "Effect": "Allow",
            "Action": [
                "logs:StopQuery",
                "logs:StopLiveTail"
            ],
            "Resource": "*"
            }
        ]
}
```

Pour plus d'informations sur l'utilisation des instructions de politique IAM, consultez Contrôle de l'accès à l'aide des politiques dans le Guide de l'utilisateur IAM.

CloudWatch Référence des autorisations de journalisation

Lorsque vous configurez des politiques d'autorisation d'écriture et de Contrôle d'accès que vous pouvez attacher à une entité IAM (politiques basées sur une identité), vous pouvez utiliser le tableau ci-dessous comme référence. Le tableau répertorie chaque opération de l'API CloudWatch Logs et les actions correspondantes pour lesquelles vous pouvez accorder des autorisations pour effectuer l'action. Vous spécifiez les actions dans le champ `Action` de la politique. Pour le `Resource` champ, vous pouvez spécifier l'ARN d'un groupe de journaux ou d'un flux de journaux, ou spécifier `*` pour représenter toutes les ressources de CloudWatch journaux.

Vous pouvez utiliser des AWS clés de condition larges dans vos politiques de CloudWatch journalisation pour exprimer des conditions. Pour obtenir la liste complète des clés « AWS wide », reportez-vous à la section Clés contextuelles AWS globales et IAM Condition du guide de l'utilisateur IAM.

> **Note**
>
> Pour spécifier une action, utilisez le préfixe `logs:` suivi du nom de l'opération d'API. Par exemple : `logs:CreateLogGrouplogs:CreateLogStream`, ou `logs:*` (pour toutes les actions CloudWatch Logs).

CloudWatch Enregistre les opérations de l'API et les autorisations requises pour les actions

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
CancelExportTask	`logs:CancelExportTask` Exigé pour annuler une tâche d'exportation en attente ou en cours d'exécution.
CreateExportTask	`logs:CreateExportTask` Exigé pour exporter des données d'un groupe de journaux vers un compartiment Amazon S3.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
CreateLogGroup	`logs:CreateLogGroup` Exigé pour créer un nouveau groupe de journaux.
CreateLogStream	`logs:CreateLogStream` Exigé pour créer un nouveau flux de journaux dans un groupe de journaux.
DeleteDestination	`logs:DeleteDestination` Exigé pour supprimer une destination de journal et désactive tous les filtres d'abonnem ent connexes.
DeleteLogGroup	`logs:DeleteLogGroup` Exigé pour supprimer un groupe de journaux et tous les événements du journal archivés associés.
DeleteLogStream	`logs:DeleteLogStream` Exigé pour supprimer un flux de journaux et tous les événements du journal archivés associés.
DeleteMetricFilter	`logs:DeleteMetricFilter` Exigé pour supprimer un filtre de métrique associé à un groupe de journaux.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Nécessaire pour supprimer une définition de requête enregistrée dans CloudWatch Logs Insights.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Nécessaire pour supprimer une politique de ressources CloudWatch Logs.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Exigé pour supprimer la politique de rétention d'un groupe de journaux.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Exigé pour supprimer le filtre d'abonnement associé à un groupe de journaux.
DescribeDestinations	`logs:DescribeDestinations` Exigé pour afficher toutes les destinations associées au compte.
DescribeExportTasks	`logs:DescribeExportTasks` Exigé pour afficher toutes les tâches d'exporta tion associées au compte.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
DescribeLogGroups	`logs:DescribeLogGroups` Exigé pour afficher tous les groupes de journaux associés au compte.
DescribeLogStreams	`logs:DescribeLogStreams` Exigé pour afficher tous les flux de journaux associés à un groupe de journaux.
DescribeMetricFilters	`logs:DescribeMetricFilters` Exigé pour afficher toutes les métriques associées à un groupe de journaux.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Nécessaire pour voir la liste des définitions de requêtes enregistrées dans CloudWatch Logs Insights.
DescribeQueries	`logs:DescribeQueries` Obligatoire pour voir la liste des requêtes CloudWatch Logs Insights planifiées, en cours d'exécution ou récemment exécutées.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Obligatoire pour consulter la liste des politique s relatives CloudWatch aux ressources des journaux.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Exigé pour afficher tous les filtres d'abonnem ent associés à un groupe de journaux.
FilterLogEvents	`logs:FilterLogEvents` Exigé pour trier les événements du journal par modèle de filtres de groupes de journaux.
GetLogEvents	`logs:GetLogEvents` Exigé pour récupérer les événements du journal à partir d'un flux de journaux.
GetLogGroupFields	`logs:GetLogGroupFields` Obligatoire pour récupérer la liste des champs qui sont inclus dans les événements du journal d'un groupe de journaux.
GetLogRecord	`logs:GetLogRecord` Obligatoire pour récupérer des informations à partir d'un seul événement du journal.
GetLogObject	`logs:GetLogRecord` Nécessaire pour récupérer le contenu d'une grande partie des événements du journal qui ont été ingérés via l' PutOpenTelemetryLogs API.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
GetQueryResults	`logs:GetQueryResults` Nécessaire pour récupérer les résultats des requêtes CloudWatch Logs Insights.
ListEntitiesForLogGroup (autorisation CloudWatch réservée à la console)	`logs:ListEntitiesForLogGroup` Obligatoire pour trouver les entités associées à un groupe de journaux. Nécessaire pour explorer les journaux associés dans la CloudWatch console.
ListLogGroupsForEntity (autorisation CloudWatch réservée à la console)	`logs:ListLogGroupsForEntity` Nécessaire pour trouver les groupes de journaux associés à une entité. Nécessaire pour explorer les journaux associés dans la CloudWatch console.
ListTagsLogGroup	`logs:ListTagsLogGroup` Exigé pour afficher toutes les étiquettes associées à un groupe de journaux.
ListLogGroups	`logs:DescribeLogGroups` Exigé pour afficher tous les groupes de journaux associés au compte.
PutDestination	`logs:PutDestination` Exigé pour créer ou mettre à jour un flux de journaux de destination (comme un flux Kinesis).

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
PutDestinationPolicy	`logs:PutDestinationPolicy` Exigé pour créer ou mettre à jour une politique d'accès associée à une destination de journal existante.
PutLogEvents	`logs:PutLogEvents` Exigé pour charger un lot d'événements du journal dans un flux de journaux.
PutMetricFilter	`logs:PutMetricFilter` Exigé pour créer ou mettre à jour un filtre de métrique et l'associer à un groupe de journaux.
PutQueryDefinition	`logs:PutQueryDefinition` Nécessaire pour enregistrer une requête dans CloudWatch Logs Insights.
PutResourcePolicy	`logs:PutResourcePolicy` Nécessaire pour créer une politique de ressources CloudWatch Logs.
PutRetentionPolicy	`logs:PutRetentionPolicy` Exigé pour définir le nombre de jours de conservation des événements du journal (rétention) dans un groupe de journaux.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Exigé pour créer ou mettre à jour un filtre d'abonnement et l'associer à un groupe de journaux.
StartQuery	`logs:StartQuery` Nécessaire pour démarrer CloudWatch les requêtes Logs Insights.
StopQuery	`logs:StopQuery` Nécessaire pour arrêter une requête CloudWatch Logs Insights en cours.
TagLogGroup	`logs:TagLogGroup` Exigé pour ajouter ou mettre à jour des étiquettes de groupe de journaux.
TestMetricFilter	`logs:TestMetricFilter` Exigé pour tester un modèle de filtre par rapport à un échantillonnage de messages d'événements du journal.

Utilisation de rôles liés à un service pour les journaux CloudWatch

Amazon CloudWatch Logs utilise des AWS Identity and Access Management rôles liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié aux journaux. CloudWatch Les rôles liés au service sont prédéfinis par CloudWatch Logs et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Un rôle lié à un service rend la configuration CloudWatch des journaux plus efficace, car vous n'êtes pas obligé d'ajouter manuellement les autorisations nécessaires. CloudWatch Logs définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul CloudWatch Logs peut assumer ces rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisations. Cette politique d'autorisations ne peut pas être attachée à une autre entité IAM.

Pour obtenir des informations sur les autres services qui prennent en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez les services qui comportent un Oui dans la colonne Rôle lié à un service. Choisissez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Autorisations de rôle liées au service pour les journaux CloudWatch

CloudWatch Logs utilise le rôle lié au service nommé. AWSServiceRoleForLogDelivery CloudWatch Logs utilise ce rôle lié à un service pour écrire des journaux directement dans Firehose. Pour de plus amples informations, veuillez consulter Activer la journalisation à partir AWS des services.

Le rôle lié à un service AWSServiceRoleForLogDelivery approuve les services suivants pour endosser le rôle :

- `logs.amazonaws.com`

La politique d'autorisation des rôles permet à CloudWatch Logs d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `firehose:PutRecord` et `firehose:PutRecordBatch` sur tous les streams Firehose dotés d'une balise dont la `LogDeliveryEnabled` clé a une valeur de. `True` Cette balise est automatiquement attachée à un stream Firehose lorsque vous créez un abonnement pour transmettre les journaux à Firehose.

Vous devez configurer les autorisations pour permettre à une entité IAM de créer, modifier ou supprimer un rôle lié à un service. Cette entité peut être un utilisateur, un groupe ou un rôle. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM

Création d'un rôle lié à un service pour Logs CloudWatch

Vous n'êtes pas obligé de créer manuellement un rôle lié à un service. Lorsque vous configurez des journaux pour qu'ils soient envoyés directement à un flux Firehose via l' AWS Management Console AWS API AWS CLI, CloudWatch Logs crée le rôle lié au service pour vous.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous configurez à nouveau les journaux pour qu'ils soient envoyés directement à un stream Firehose, CloudWatch Logs crée à nouveau le rôle lié au service pour vous.

Modification d'un rôle lié à un service pour Logs CloudWatch

CloudWatch Les journaux ne vous permettent pas de modifier le rôle, ni de le modifier AWSServiceRoleForLogDelivery, ni de l'utiliser pour tout autre rôle lié à un service, une fois que vous l'avez créé. Vous ne pouvez pas modifier le nom du rôle car diverses entités pourraient y faire référence. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Supprimer un rôle lié à un service pour Logs CloudWatch

Si vous n'avez plus besoin d'utiliser une fonctionnalité ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

> **Note**
>
> Si le service CloudWatch Logs utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression risque d'échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer les ressources CloudWatch Logs utilisées par le rôle lié à AWSServiceRoleForLogDeliveryun service

- Arrêtez d'envoyer des logs directement aux streams Firehose.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM AWS CLI, le ou l' AWS API pour supprimer le rôle lié au AWSServiceRoleForLogDeliveryservice. Pour plus d'informations, consultez Suppression d'un rôle lié à un service

Régions prises en charge pour les CloudWatch rôles liés au service Logs

CloudWatch Logs prend en charge l'utilisation de rôles liés au service dans toutes les AWS régions où le service est disponible. Pour plus d'informations, consultez CloudWatch Logs, régions et points de terminaison.

CloudWatch Enregistre les mises à jour des rôles liés aux AWS services

Consultez les détails des mises à jour apportées au rôle lié au AWS service pour CloudWatch Logs depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique du document CloudWatch Logs.

Modification	Description	Date
AWSServiceRoleForLogDelivery politique de rôle liée au service — Mise à jour d'une politique existante	CloudWatch Les journaux ont modifié les autorisations de la politique IAM associée au rôle lié au AWSServiceRoleForLogDeliveryservice. La modification suivante a été apportée : • La clé de condition `firehose:ResourceTag/LogDeliveryEnabled": "true"` a été modifiée en `aws:ResourceTag/LogDeliveryEnabled": "true"` .	15 juillet 2021

Modification	Description	Date
CloudWatch Les journaux ont commencé à suivre les modifications	CloudWatch Logs a commencé à suivre les modifications apportées AWS à ses politiques gérées.	10 juin 2021

Validation de conformité pour Amazon CloudWatch Logs

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

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 .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Conformité et gouvernance de la sécurité : ces guides de mise en œuvre de solutions traitent des considérations architecturales et fournissent les étapes à suivre afin de déployer des fonctionnalités de sécurité et de conformité.
- Référence des services éligibles HIPAA : liste les services éligibles HIPAA. Tous ne Services AWS sont pas éligibles à la loi HIPAA.
- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).

- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config 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— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dans Amazon CloudWatch Logs

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Sécurité de l'infrastructure dans Amazon CloudWatch Logs

En tant que service géré, Amazon CloudWatch Logs est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure,

consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder aux CloudWatch journaux via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Utilisation des CloudWatch journaux avec les points de terminaison VPC de l'interface

Si vous utilisez Amazon Virtual Private Cloud (Amazon VPC) pour héberger vos AWS ressources, vous pouvez établir une connexion privée entre votre VPC et Logs. CloudWatch Vous pouvez utiliser cette connexion pour envoyer des CloudWatch journaux à Logs sans les envoyer via Internet. CloudWatch Logs prend en charge les points de terminaison IPv4 VPC dans toutes les régions et prend en charge les IPv6 points de terminaison dans toutes les régions, à l'exception de l'Asie-Pacifique (Malaisie), de l'Asie-Pacifique (Thaïlande) et du Mexique (centre).

Amazon VPC est un AWS service que vous pouvez utiliser pour lancer AWS des ressources dans un réseau virtuel que vous définissez. Avec un VPC, vous contrôlez des paramètres réseau, tels que la plage d'adresses IP, les sous-réseaux, les tables de routage et les passerelles réseau. Pour connecter votre VPC aux CloudWatch journaux, vous devez définir un point de terminaison VPC d'interface pour les journaux. CloudWatch Ce type de point de terminaison vous permet de connecter votre VPC à des services AWS . Le point de terminaison fournit une connectivité fiable et évolutive aux CloudWatch journaux sans nécessiter de passerelle Internet, d'instance de traduction d'adresses réseau (NAT) ou de connexion VPN. Pour de plus amples informations, consultez Qu'est-ce qu'Amazon VPC ? dans le Guide de l'utilisateur Amazon VPC.

Les points de terminaison VPC d'interface sont alimentés par AWS PrivateLink une AWS technologie qui permet une communication privée entre les AWS services à l'aide d'une interface Elastic Network avec des adresses IP privées. Pour plus d'informations, voir Nouveau — AWS PrivateLink pour les AWS services.

Les étapes suivantes s'adressent aux utilisateurs d'Amazon VPC. Pour plus d'informations, consultez Démarrez dans le Amazon VPC Guide de l'utilisateur.

Disponibilité

CloudWatch Logs prend actuellement en charge les points de terminaison VPC dans toutes les AWS régions, y compris les régions. AWS GovCloud (US)

Création d'un point de terminaison VPC pour les journaux CloudWatch

Pour commencer à utiliser CloudWatch les journaux avec votre VPC, créez un point de terminaison VPC d'interface pour les journaux. CloudWatch Le service à choisir est com.amazonaws.***Region***.journaux. Il n'est pas nécessaire de modifier les paramètres des CloudWatch journaux. Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Amazon VPC Guide de l'utilisateur.

Test de la connexion entre votre VPC et Logs CloudWatch

Une fois que vous avez créé le point de terminaison, vous pouvez tester la connexion.

Pour tester la connexion entre votre VPC et votre CloudWatch point de terminaison Logs

1. Connectez-vous à une EC2 instance Amazon qui réside dans votre VPC. Pour plus d'informations sur la connexion, consultez Connect to your Linux instance ou Connecting to your Windows Instance dans la EC2 documentation Amazon.
2. À partir de l'instance, utilisez le AWS CLI pour créer une entrée de journal dans l'un de vos groupes de journaux existants.

   Commencez par créer un fichier JSON avec un événement de journal. L'horodatage doit être spécifié en nombre de millisecondes après le 1er janvier 1970 00:00:00 UTC.

```
[
  {
    "timestamp": 1533854071310,
```

```
    "message": "VPC Connection Test"
  }
]
```

Utilisez ensuite la commande `put-log-events` pour créer une entrée de journal :

```
aws logs put-log-events --log-group-name LogGroupName --log-stream-
name LogStreamName --log-events file://JSONFileName
```

Si la réponse à la commande inclut `nextSequenceToken`, cela signifie que la commande a réussi et que votre point de terminaison d'un VPC fonctionne.

Contrôle de l'accès à votre point de CloudWatch terminaison Logs VPC

Une stratégie de point de terminaison d'un VPC est une stratégie de ressource IAM que vous attachez à un point de terminaison lorsque vous le créez ou le modifiez. Si vous ne définissez pas de politique lorsque vous créez un point de terminaison, nous définissons une politique par défaut pour vous, qui autorise un accès total au service. Une politique de point de terminaison n'annule pas et ne remplace pas les politiques IAM ou les politiques spécifiques aux services. Il s'agit d'une politique distincte qui contrôle l'accès depuis le point de terminaison jusqu'au service spécifié.

Les politiques de point de terminaison doivent être écrites au format JSON.

Pour en savoir plus, consultez Contrôle de l'accès aux services avec des points de terminaison d'un VPC dans le guide de l'utilisateur Amazon VPC.

Voici un exemple de politique de point de terminaison pour CloudWatch Logs. Cette politique permet aux utilisateurs qui se connectent aux CloudWatch journaux via le VPC de créer des flux de journaux et d'envoyer des journaux aux CloudWatch journaux, et les empêche d'effectuer d'autres actions liées CloudWatch aux journaux.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
```

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

Pour modifier la politique de point de terminaison VPC pour les journaux CloudWatch

1. Ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le volet de navigation, choisissez Points de terminaison.
3. Si vous n'avez pas encore créé le point de terminaison pour CloudWatch Logs, choisissez Create Endpoint. Sélectionnez ensuite com.amazonaws. ***Region***.logs et choisissez Create endpoint.
4. Sélectionnez com.amazonaws. ***Region***.logs endpoint, et choisissez l'onglet Policy dans la partie inférieure de l'écran.
5. Choisissez Modifier la politique, puis apportez les modifications souhaitées à la politique.

Prise en charge des clés de contexte de VPC

CloudWatch Les journaux prennent en charge les clés de `aws:SourceVpce` contexte `aws:SourceVpc` et qui peuvent limiter l'accès à des points de terminaison VPC spécifiques VPCs ou spécifiques. Ces clés fonctionnent uniquement lorsque l'utilisateur utilise des points de terminaison d'un VPC. Pour plus d'informations, consultez Clés disponibles pour certains services dans le Guide de l'utilisateur IAM.

Logging CloudWatch Logs, API et opérations de console dans AWS CloudTrail

Amazon CloudWatch Logs est intégré à AWS CloudTrailun service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS. CloudTrail capture les appels d'API pour les CloudWatch journaux sous forme d'événements. Les appels capturés incluent des appels provenant de la console CloudWatch Logs et des appels de code vers les opérations de l'API CloudWatch Logs. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à CloudWatch Logs, l'adresse IP à partir de laquelle la demande a été faite, la date à laquelle elle a été faite et des informations supplémentaires.

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 :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou d'utilisateur root.
- Si la demande a été faite au nom d'un utilisateur du centre d'identité 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 requête a été effectuée par un autre Service AWS.

CloudTrail est actif dans votre compte Compte AWS lorsque vous créez le compte et vous avez automatiquement accès à l'historique des CloudTrail événements. L'historique des CloudTrail événements fournit un enregistrement consultable, consultable, téléchargeable et immuable des 90 derniers jours des événements de gestion enregistrés dans un. Région AWS Pour plus d'informations, consultez la section Utilisation de l'historique des CloudTrail événements dans le guide de AWS CloudTrail l'utilisateur. La consultation de CloudTrail l'historique des événements est gratuite.

Pour un enregistrement continu des événements de vos 90 Compte AWS derniers jours, créez un magasin de données sur les événements de Trail ou CloudTrailLake.

CloudTrail sentiers

Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Tous les sentiers créés à l'aide du AWS Management Console sont multirégionaux. Vous ne pouvez

créer un journal de suivi en une ou plusieurs régions à l'aide de l' AWS CLI. Il est recommandé de créer un parcours multirégional, car vous capturez l'activité dans l'ensemble Régions AWS de votre compte. Si vous créez un journal de suivi pour une seule région, il convient de n'afficher que les événements enregistrés dans le journal de suivi pour une seule région Région AWS. Pour plus d'informations sur les journaux de suivi, consultez Créez un journal de suivi dans vos Compte AWS et Création d'un journal de suivi pour une organisation dans le AWS CloudTrail Guide de l'utilisateur.

Vous pouvez envoyer une copie de vos événements de gestion en cours à votre compartiment Amazon S3 gratuitement CloudTrail en créant un journal. Toutefois, des frais de stockage Amazon S3 sont facturés. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification. Pour obtenir des informations sur la tarification Amazon S3, consultez Tarification Amazon S3.

CloudTrail Stockages de données sur les événements du lac

CloudTrail Lake vous permet d'exécuter des requêtes SQL sur vos événements. CloudTrail Lake convertit les événements existants au format JSON basé sur les lignes au format Apache ORC. ORC est un format de stockage en colonnes qui est optimisé pour une récupération rapide des données. Les événements sont agrégés dans des magasins de données d'événement. Ceux-ci constituent des collections immuables d'événements basées sur des critères que vous sélectionnez en appliquant des sélecteurs d'événements avancés. Les sélecteurs que vous appliquez à un magasin de données d'événement contrôlent les événements qui persistent et que vous pouvez interroger. Pour plus d'informations sur CloudTrail Lake, consultez la section Travailler avec AWS CloudTrail Lake dans le guide de AWS CloudTrail l'utilisateur.

CloudTrail Les stockages et requêtes de données sur les événements de Lake entraînent des coûts. Lorsque vous créez un magasin de données d'événement, vous choisissez l'option de tarification que vous voulez utiliser pour le magasin de données d'événement. L'option de tarification détermine le coût d'ingestion et de stockage des événements, ainsi que les périodes de conservation par défaut et maximale pour le magasin de données d'événement. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification.

CloudWatch Logs prend en charge la journalisation des actions suivantes sous forme d'événements dans des fichiers CloudTrail journaux :

- AssociateKmsKey
- CancelExportTask

- CreateDelivery
- CreateExportTask
- CreateLogAnomalyDetector
- CreateLogGroup
- CreateLogStream
- DeleteAccountPolicy
- DeleteDataProtectionPolicy
- DeleteDelivery
- DeleteDeliveryDestination
- DeleteDeliveryDestinationPolicy
- DeleteDeliverySource
- DeleteDestination
- DeleteIndexPolicy
- DeleteIntegration
- DeleteLogAnomalyDetector
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteQueryDefinition
- DeleteResourcePolicy
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- DeleteTransformer
- DescribeAccountPolicies
- DescribeConfigurationTemplates
- DescribeDeliveries
- DescribeDeliveryDestinations
- DescribeDeliverySources
- DescribeDestinations

- DescribeExportTasks
- DescribeFieldIndexes
- DescribeIndexPolicies
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeQueryDefinitions
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- DisassociateKmsKey
- FilterLogEvents
- GetDataProtectionPolicy
- GetDelivery
- GetDeliveryDestination
- GetDeliveryDestinationPolicy
- GetDeliverySource
- GetIntegration
- GetLogAnomalyDetector
- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults
- GetTransformer
- ListAnomalies
- ListIntegrations
- ListLogAnomalyDetectors
- ListLogGroups
- ListLogGroupsForQuery
- ListTagsForResource

- ListTagsLogGroup
- PutAccountPolicy
- PutDataProtectionPolicy
- PutDeliveryDestination
- PutDeliveryDestinationPolicy
- PutDeliverySource
- PutDestination
- PutDestinationPolicy
- PutIndexPolicy
- PutIntegration
- PutMetricFilter
- PutQueryDefinition
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- PutTransformer
- StartLiveTail
- StartQuery
- StopQuery
- TagResource
- TestMetricFilter
- TestTransformer
- UntagResource
- UpdateAnomaly
- UpdateDeliveryConfiguration
- UpdateLogAnomalyDetector

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 effectuée avec des informations d'identification d'utilisateur root ou IAM.

- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail .

Informations de génération de requêtes dans CloudTrail

CloudTrail la journalisation des événements de la console du générateur de requêtes est également prise en charge. Le générateur de requêtes est actuellement pris en charge pour CloudWatch Logs Insights et CloudWatch Metric Insights. Dans ces CloudTrail événements, le `eventSource` est`monitoring.amazonaws.com`.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre l'GenerateQueryaction dans CloudWatch Logs Insights.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId": "123456789012",
                "userName": "SAMPLE_NAME"
            },
            "attributes": {
                "creationDate": "2020-04-08T21:43:24Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "monitoring.amazonaws.com",
    "eventName": "GenerateQuery",
```

```
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "exampleUserAgent",
    "requestParameters": {
        "query_ask": "***",
        "query_type": "LogsInsights",
        "logs_insights": {
            "fields": "***",
            "log_group_names": ["yourloggroup"]
        },
        "include_description": true
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Présentation des entrées des fichiers journaux

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'entrée de fichier journal suivante indique qu'un utilisateur a appelé l'CreateExportTaskaction CloudWatch Logs.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
```

```
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

CloudWatch Référence de l'agent de journalisation

> **Important**
>
> Cette section est une référence pour ceux qui utilisent l'agent CloudWatch Logs obsolète. Si vous utilisez le service de métadonnées d'instance version 2 (IMDSv2), vous devez utiliser le nouvel CloudWatch agent unifié. Toutefois, même si vous ne l'utilisez pas IMDSv2, nous vous recommandons vivement d'utiliser le nouvel CloudWatch agent unifié au lieu de l'agent CloudWatch Logs obsolète. Pour plus d'informations sur le nouvel CloudWatch agent unifié, consultez la section Collecte de métriques et de journaux à partir d' EC2 instances Amazon et de serveurs locaux avec l' CloudWatch agent. Pour plus d'informations sur la migration de l'agent CloudWatch Logs obsolète vers l'agent unifié, créez le fichier de configuration de l' CloudWatch agent à l'aide de l'assistant.

L'agent CloudWatch Logs fournit un moyen automatique d'envoyer des données de journal à CloudWatch Logs à partir d' EC2instances Amazon. L'agent comprend les éléments suivants :

- Un plug-in AWS CLI qui envoie les données du journal vers CloudWatch Logs.
- Script (daemon) qui lance le processus de transfert de données vers CloudWatch Logs.
- Une tâche cron qui garantit que le programme démon s'exécute en permanence.

Fichier de configuration de l'agent

Le fichier de configuration de l'agent CloudWatch Logs décrit les informations nécessaires à l'agent CloudWatch Logs. La section [general] du fichier de configuration de l'agent définit les configurations courantes qui s'appliquent à tous les flux de journaux. La section [logstream] définit les informations nécessaires pour envoyer un fichier local à un flux de journaux à distance. Vous pouvez avoir plusieurs sections [logstream], mais chacune doit porter un nom unique dans le fichier de configuration, par exemple [logstream1], [logstream2] et ainsi de suite. La valeur [logstream], ainsi que la première ligne de données du fichier journal, définissent l'identités du fichier journal.

```
[general]
state_file = value
logging_config_file = value
use_gzip_http_content_encoding = [true | false]
```

```
[logstream1]
log_group_name = value
log_stream_name = value
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

state_file

Spécifie l'emplacement où est stocké le fichier d'état.

logging_config_file

(Facultatif) Spécifie l'emplacement du fichier de configuration de la journalisation de l'agent. Si vous ne spécifiez pas de fichier de configuration de la journalisation de l'agent ici, le fichier par défaut awslogs.conf est utilisé. L'emplacement du fichier par défaut est `/var/awslogs/etc/awslogs.conf` si vous avez installé l'agent à l'aide d'un script, et `/etc/awslogs/awslogs.conf` si vous avez installé l'agent avec rpm. Le fichier est au format de fichier de configuration Python (https://docs.pylogging-config-fileformatthon.org/2/library/logging.config.html#). Il est possible de personnaliser les enregistreurs suivants.

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

L'exemple ci-dessous fait passer le niveau du lecteur et de l'éditeur à WARNING, alors que la valeur par défaut est INFO.

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

Lorsqu'il est défini sur true (par défaut), active le codage de contenu HTTP gzip pour envoyer des charges utiles compressées à CloudWatch Logs. Cela permet de réduire l'utilisation du processeur, de diminuer NetworkOut et de diminuer la latence de mise en ligne. Pour désactiver cette fonctionnalité, ajoutez use_gzip_http_content_encoding = false dans la section [general] du fichier de configuration de l'agent CloudWatch Logs, puis redémarrez l'agent.

> **Note**
>
> Ce paramètre est uniquement disponible pour la version awscli-cwlogs 1.3.3 et les versions ultérieures.

log_group_name

Spécifie le groupe de journaux de destination. Un groupe de journaux est créé automatiquement s'il n'en existe pas déjà. Les noms des groupes de journaux peuvent comporter entre 1 et 512 caractères. Les caractères autorisés sont : a-z, A-Z, 0-9, « _ » (trait de soulignement), « - » (tiret), « / » (barre oblique) et « . » (point).

log_stream_name

Spécifie le flux de journaux de destination. Vous pouvez définir un nom de flux de journal à l'aide d'une chaîne littérale, de variables prédéfinies ({instance_id}, {hostname} et {ip_address}), ou d'une combinaison de celles-ci. Un flux de journaux est créé automatiquement s'il n'en existe pas déjà.

datetime_format

Spécifie la façon dont l'horodatage est extrait des journaux. L'horodatage est utilisé pour récupérer les événements de journaux et générer des métriques. L'heure actuelle est utilisée pour chaque événement de journal si datetime_format n'est pas fourni. Si la valeur datetime_format fournie n'est pas valide pour un message de journal donné, l'horodatage du dernier événement de journal dont l'horodatage a été analysé avec succès sera utilisé. En cas d'absence d'événements de journaux précédents, l'heure actuelle est utilisée.

Les codes datetime_format courants sont répertoriées ci-dessous. Vous pouvez également utiliser n'importe quel code datetime_format pris en charge par Python, datetime.strptime(). Le décalage horaire (%z) est également pris en charge, même s'il n'est pas pris en charge par les versions

antérieures ou égales à Python 3.2, [+-]HHMM deux points(:). Pour plus d'informations, consultez strftime() and strptime() Behavior.

%y : année sans siècle sous forme de nombre décimal auquel est ajouté un zéro. 00, 01, ..., 99

%Y : année avec siècle sous forme de nombre décimal.1970, 1988, 2001, 2013

%b : mois sous forme du nom abrégé dans la langue locale. jan, fév, ..., déc (fr_FR);

%B : mois sous forme du nom complet dans la langue locale. janvier, février, ..., décembre (fr_FR);

%m : mois sous forme de nombre décimal auquel est ajouté un zéro. 01, 02, ..., 12

%d : jour du mois sous forme de nombre décimal auquel est ajouté un zéro. 01, 02, ..., 31

%H : heure (au format 24 heures) sous forme de nombre décimal auquel est ajouté un zéro. 00, 01, ..., 23

%I : heure (au format 12 heures) sous forme de nombre décimal auquel est ajouté un zéro. 01, 02, ..., 12

%p : équivalent de AM ou PM dans les paramètres régionaux.

%M : minute sous forme de nombre décimal auquel est ajouté un zéro. 00, 01, ..., 59

%S : seconde sous forme de nombre décimal auquel est ajouté un zéro. 00, 01, ..., 59

%f : microseconde sous forme de nombre décimal auquel est ajouté un zéro, à gauche. 000000, ..., 999999

%z : décalage UTC sous la forme +HHMM ou -HHMM. +0000, -0400, +1030

Exemples de formats :

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

Spécifie le fuseau horaire de l'horodatage de l'événement de journal. Les deux valeurs prises en charge sont UTC et LOCAL. La valeur par défaut est LOCAL. Elle est utilisée s'il n'est pas possible de déduire le fuseau horaire d'après datetime_format.

dans le fichier

Spécifie les fichiers journaux que vous souhaitez transférer vers CloudWatch Logs. Le fichier peut pointer vers un fichier spécifique ou plusieurs fichiers (à l'aide de caractères génériques tels que/var/log/system.log*). Seul le dernier fichier est transféré dans CloudWatch Logs en fonction de l'heure de modification du fichier. Nous vous recommandons d'utiliser des caractères génériques pour spécifier une série de fichiers du même type, comme access_log.2014-06-01-01, access_log.2014-06-01-02 et ainsi de suite, mais pas pour différents types de fichiers, comme access_log_80 et access_log_443. Pour spécifier plusieurs types de fichiers, ajoutez une autre entrée de flux de journaux dans le fichier de configuration, afin que chaque type de fichier journal soit envoyé dans un flux de journal différent. Les fichiers zippés ne sont pas pris en charge.

file_fingerprint_lines

Spécifie la plage de lignes pour identifier un fichier. Les valeurs valides sont soit un nombre, soit deux nombres séparés par un tiret, par exemple, « 1 », « 2-5 ». La valeur par défaut est « 1 ». La première ligne est utilisée pour calculer l'empreinte. Les lignes d'empreintes digitales ne sont envoyées à CloudWatch Logs que si toutes les lignes spécifiées sont disponibles.

multi_line_start_pattern

Spécifie le modèle pour identifier le début d'un message de journal. Un message de journal est composé de plusieurs lignes : une ligne correspondant au modèle et les lignes suivantes qui ne correspondent pas au modèle. Les valeurs valides sont les expressions régulières ou {datetime_format}. Lorsque vous utilisez {datetime_format}, l'option datetime_format doit être spécifiée. La valeur par défaut est « ^[^\s] ». Ainsi, toute ligne commençant par un caractère autre qu'un espace ferme le message de journal précédent et commence un nouveau message de journal.

initial_position

Spécifie l'endroit où commencer à lire les données (start_of_file ou end_of_file). La valeur par défaut est start_of_file. Utilisé uniquement si aucun état n'est conservé pour ce flux de journaux.

encoding

Spécifie l'encodage du fichier journal pour que le fichier puisse être lu correctement. La valeur par défaut est utf_8. L'encodage pris en charge par Python codecs.decode() peut être utilisé ici.

> **Warning**
>
> Le fait de spécifier un encodage incorrect peut entraîner une perte de données, car les caractères qui ne peuvent pas être décodés seront remplacés par un autre caractère.

Voici une liste d'encodages courants :

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737,
cp775, cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862,
cp863, cp864, cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950,
cp1006, cp1026, cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255,
cp1256, cp1257, cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr,
gb2312, gbk, gb18030, hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2,
iso2022_jp_2004, iso2022_jp_3, iso2022_jp_ext, iso2022_kr, latin_1,
iso8859_2, iso8859_3, iso8859_4, iso8859_5, iso8859_6, iso8859_7,
iso8859_8, iso8859_9, iso8859_10, iso8859_13, iso8859_14, iso8859_15,
iso8859_16, johab, koi8_r, koi8_u, mac_cyrillic, mac_greek, mac_iceland,
mac_latin2, mac_roman, mac_turkish, ptcp154, shift_jis, shift_jis_2004,
shift_jisx0213, utf_32, utf_32_be, utf_32_le, utf_16, utf_16_be,
utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

Spécifie la durée du regroupement des événements de journaux. La valeur minimale est 5 000 m et la valeur par défaut est 5 000 ms.

batch_count

Spécifie le nombre maximal d'événements de journaux dans un lot. La valeur maximale est 10 000. La valeur par défaut est 10 000.

batch_size

Spécifie la taille maximale des événements de journaux dans un lot, en octets. La taille maximale est 1 048 576 octets. La valeur par défaut est 1 048 576 octets. Cette taille est calculée comme étant la somme de tous les messages d'événement au format UTF-8, plus 26 octets pour chaque événement de journal.

Utilisation de l'agent CloudWatch Logs avec des proxys HTTP

Vous pouvez utiliser l'agent CloudWatch Logs avec des proxys HTTP.

> **Note**
>
> Les proxys HTTP sont pris en charge dans la version 1.3.8 ou ultérieure de awslogs-agent-setup .py.

Pour utiliser l'agent CloudWatch Logs avec des proxys HTTP

1. Effectuez l'une des actions suivantes :

   a. Pour une nouvelle installation de l'agent CloudWatch Logs, exécutez les commandes suivantes :

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
   ```

   ```
   sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
   ```

   Afin de conserver l'accès au service de EC2 métadonnées Amazon sur les EC2 instances, utilisez --no-proxy 169.254.169.254 (recommandé). Pour plus d'informations, consultez la section Métadonnées d'instance et données utilisateur dans le guide de EC2 l'utilisateur Amazon.

   Dans les valeurs pour `http-proxy` et `https-proxy`, vous spécifiez la totalité de l'URL.

   b. Pour une installation existante de l'agent CloudWatch Logs, modifiez/var/awslogs/etc/proxy.conf et ajoutez vos proxies :

   ```
   HTTP_PROXY=
   HTTPS_PROXY=
   NO_PROXY=
   ```

2. Redémarrez l'agent pour que les modifications prennent effet :

```
sudo service awslogs restart
```

Si vous utilisez Amazon Linux 2, utilisez la commande suivante pour redémarrer l'agent :

```
sudo service awslogsd restart
```

Compartimentation des fichiers de configuration de l'agent CloudWatch Logs

Si vous utilisez awslogs-agent-setup .py version 1.3.8 ou ultérieure avec awscli-cwlogs 1.3.3 ou version ultérieure, vous pouvez importer différentes configurations de flux pour différents composants indépendamment les uns des autres en créant des fichiers de configuration supplémentaires dans le répertoire//. var/awslogs/etc/config Lorsque l'agent CloudWatch Logs démarre, il inclut toutes les configurations de flux dans ces fichiers de configuration supplémentaires. Les propriétés de configuration de la section [général] doivent être définies dans le fichier de configuration principal (/var/awslogs/etc/awslogs.conf) and are ignored in any additional configuration files found in /var/awslogs/etc/config/.

Si vous n'avez pas de répertoire/var/awslogs/etc/config/parce que vous avez installé l'agent avec rpm, vous pouvez utiliser le répertoire/etc/awslogs/config/à la place.

Redémarrez l'agent pour que les modifications prennent effet :

```
sudo service awslogs restart
```

Si vous utilisez Amazon Linux 2, utilisez la commande suivante pour redémarrer l'agent :

```
sudo service awslogsd restart
```

CloudWatch FAQ sur les agents de journalisation

Quels sont les types de rotations de fichier pris en charge ?

Les mécanismes de rotation de fichier suivants sont pris en charge :

- Ajout d'un suffixe numérique aux noms des fichiers journaux existants, puis recréation du fichier journal vide d'origine. Par exemple,/var/log/syslog.log is renamed /var/log/syslog.log.1. If /var/log/syslog.log.1 already exists from a previous rotation, it is renamed /var/log/syslog.log.2.
- Troncation du fichier journal d'origine après la création d'une copie. Par exemple,/var/log/syslog.log is copied to /var/log/syslog.log.1 and /var/log/syslog.log est tronqué. Cela peut entraîner une perte de données. Soyez prudent lors de l'utilisation de ce mécanisme de rotation de fichier.
- Création d'un fichier avec un modèle commun à l'ancien. Par exemple,/var/log/syslog.log.2014-01-01 remains and /var/log/syslog.log.2014-01-02 est créé.

L'empreinte (ID de la source) du fichier est calculée en hachant la clé du flux de journal et la première ligne du contenu du fichier. Pour remplacer ce comportement, l'option file_fingerprint_lines peut être utilisée. Lorsque la rotation de fichier se produit, le nouveau fichier doit comporter le nouveau contenu et aucun contenu ne doit être ajouté à l'ancien fichier. L'agent transfère le nouveau fichier une fois qu'il a terminé de lire l'ancien.

Comment déterminer la version de l'agent que j'utilise ?

Si vous avez utilisé un script de configuration pour installer l'agent CloudWatch Logs, vous pouvez utiliser/var/awslogs/bin/awslogs-version.sh pour vérifier la version de l'agent que vous utilisez. Vous pouvez ainsi afficher la version de l'agent et ses principales dépendances. Si vous avez utilisé yum pour installer l'agent CloudWatch Logs, vous pouvez utiliser « yum info awslogs » et « yum info aws-cli-plugin-cloudwatch -logs » pour vérifier la version de l'agent Logs et du plugin. CloudWatch

Comment les entrées de journaux sont-elles converties en événements de journaux ?

Les événements de journaux contiennent deux propriétés : l'horodatage du moment où l'événement s'est produit et le message brut du journal. Par défaut toute ligne commençant par un caractère autre qu'un espace ferme le message de journal précédent, le cas échéant, et commence un nouveau message de journal. Pour remplacer ce comportement, multi_line_start_pattern peut être utilisé et n'importe quelle ligne correspondant au modèle commence un nouveau message de journal. Le modèle peut être n'importe quelle expression régulière ou « {datetime_format} ». Par exemple, si la première ligne de chaque message de journal contient un horodatage comme « 2014-01-02T13:13:01Z », multi_line_start_pattern peut être défini sur « \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z ». Pour simplifier la configuration, la variable « {datetime_format} » peut être utilisée si l'option datetime_format a été spécifiée. Pour le même exemple, si l'option datetime_format est définie sur « %Y-%m-%dT%H:%M:%S%z », alors multi_line_start_pattern peut simplement être « {datetime_format} ».

L'heure actuelle est utilisée pour chaque événement de journal si datetime_format n'est pas fourni. Si la valeur datetime_format fournie n'est pas valide pour un message de journal donné, l'horodatage du dernier événement de journal dont l'horodatage a été analysé avec succès sera utilisé. En cas d'absence d'événements de journaux précédents, l'heure actuelle est utilisée. Un message d'avertissement est enregistré lorsqu'un événement de journal utilise l'heure actuelle ou l'heure de l'événement de journal précédent.

Les horodatages sont utilisés pour récupérer les événements de journaux et générer des métriques. Ainsi, si vous spécifiez un format incorrect, les événements des journaux peuvent ne plus être récupérables et générer des métriques incorrectes.

Comment les événements de journaux sont-ils regroupés par lot ?

Lorsque l'une des conditions suivantes est remplie, un lot devient plein et est publié :

1. La durée buffer_duration s'est écoulée depuis l'ajout du premier événement de journal.
2. Moins de batch_size d'événements de journaux ont été accumulés, mais l'ajout du nouvel événement de journal entraîne le dépassement défini par batch_size.
3. Le nombre d'événements de journaux a atteint la valeur définie dans batch_count.
4. Les événements de journaux du lot ne s'étendent pas sur plus de 24 heures, mais l'ajout du nouvel événement de journal entraîne le dépasse la limite de 24 heures.

Pour quelle raison des entrées de journal, des événements de journaux ou des lots seraient-ils ignorés ou tronqués ?

Pour respecter la contrainte de l'opération `PutLogEvents`, un événement de journal ou un lot peut être ignoré à cause des problèmes suivants.

> **Note**
>
> L'agent CloudWatch Logs écrit un avertissement dans son journal lorsque des données sont ignorées.

1. Si la taille d'un événement de journal dépasse 256 Ko, ce dernier sera complètement ignoré.
2. Si l'horodatage d'un événement de journal est supérieur à 2 heures par rapport à l'heure actuelle, l'événement de journal est ignoré.
3. Si l'horodatage d'un événement de journal est antérieur de plus de 14 jours au jour actuel, l'événement de journal est ignoré.

4. Si un événement de journal est antérieur à la période de conservation du groupe de journaux, l'ensemble du lot est ignoré.
5. Si le lot des événements de journaux d'une seule requête `PutLogEvents` s'étend sur plus de 24 heures, l'opération `PutLogEvents` échoue.

L'arrêt de l'agent entraîne-t-il la perte ou la duplication de données ?

Non, à condition que le fichier d'état soit disponible et qu'aucune rotation de fichier ne se soit produite depuis la dernière exécution. L'agent CloudWatch Logs peut partir de l'endroit où il s'est arrêté et continuer à transmettre les données du journal.

Puis-je pointer différents fichiers journaux depuis un seul ou plusieurs hôtes vers le même flux de journaux ?

La configuration de plusieurs sources de journal pour envoyer des données dans un flux de journal unique n'est pas prise en charge.

Quels appels d'API l'agent effectue-t-il (ou quelles actions dois-je ajouter à ma politique IAM) ?

L'agent CloudWatch Logs nécessite les `PutLogEvents` opérations `CreateLogGroup` `CreateLogStreamDescribeLogStreams`,, et. Si vous utilisez le dernier agent, `DescribeLogStreams` n'est pas nécessaire. Consultez l'exemple de politique IAM ci-dessous.

JSON

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Je ne souhaite pas que l'agent CloudWatch Logs crée automatiquement des groupes de journaux ou des flux de journaux. Comment puis-je empêcher l'agent de recréer des groupes de journaux et des flux de journaux ?

Dans la politique IAM, vous pouvez limiter l'agent afin qu'il puisse uniquement effectuer les opérations suivantes : `DescribeLogStreams`, `PutLogEvents`.

Avant de révoquer les autorisations `CreateLogStream` et `CreateLogGroup` à partir de l'agent, veillez à créer à la fois les groupes de journaux et les flux de journaux que vous souhaitez que l'agent utilise. L'agent de journaux ne peut pas créer de flux de journaux dans un groupe de journaux que vous avez créé, sauf s'il dispose des autorisations `CreateLogGroup` et `CreateLogStream`.

Quels journaux consulter pour résoudre les problèmes ?

Le journal d'installation de l'agent est enregistré à l'adresse `/var/log/awslogs-agent-setup.log` et le journal de l'agent à l'adresse `/var/log/awslogs.log`.

Surveillance à l'aide de CloudWatch métriques

Vous pouvez utiliser les tableaux de cette section pour consulter les statistiques qu'Amazon CloudWatch Logs envoie à Amazon CloudWatch chaque minute.

CloudWatch Métriques des journaux

L'espace de noms `AWS/Logs` inclut les métriques suivantes.

Métrique	Description
`CallCount`	Nombre d'opérations API spécifiées effectuées dans votre compte. `CallCount` est une métrique d'utilisation du service CloudWatch Logs. Pour de plus amples informations, veuillez consulter CloudWatch Statistiques d'utilisation du service de journalisation. Dimensions valides : Classe, Ressource, Service, Type Statistique valide : somme Unités : aucune
`DeliveryErrors`	Nombre d'événements de journal pour lesquels CloudWatch Logs a reçu une erreur lors du transfert des données vers la destination de l'abonnement. Si le service de destination renvoie une erreur réessayable, telle qu'une exception de limitation ou une exception de service réessayable (HTTP 5xx par exemple), CloudWatch Logs continue de réessayer la livraison pendant 24 heures au maximum. CloudWatch Logs n'essaie pas de le renvoyer s'il s'agit d'une erreur non réessayable, telle que ou. `AccessDeniedException` `ResourceNotFoundException` Dimensions valides : LogGroupName DestinationType, FilterName, PolicyLevel Statistique valide : somme Unités : aucune

Métrique	Description
`DeliveryThrottling`	Nombre d'événements de journal pour lesquels CloudWatch Logs a été limité lors du transfert des données vers la destination de l'abonnement. Si le service de destination renvoie une erreur réessayable, telle qu'une exception de limitation ou une exception de service réessayable (HTTP 5xx par exemple), CloudWatch Logs continue de réessayer la livraison pendant 24 heures au maximum. CloudWatch Logs n'essaie pas de le renvoyer s'il s'agit d'une erreur non réessayable, telle que ou. `AccessDeniedException` `ResourceNotFoundException` Dimensions valides : LogGroupName DestinationType, FilterName, PolicyLevel Statistique valide : somme Unités : aucune
`EMFParsingErrors`	Nombre d'erreurs d'analyse rencontrées lors du traitement des journaux du format des métriques intégrées. De telles erreurs se produisent lorsque les journaux sont identifiés comme étant au format des métriques intégrées mais ne suivent pas le format correct. Pour plus d'informations sur le format des métriques intégrées, consulter les Spécification : format des métriques intégrées. Dimensions valides : `LogGroupName` Statistique valide : somme Unités : aucune

Métrique	Description
`EMFValidationErrors`	Nombre d'erreurs de validation rencontrées lors du traitement des journaux sur le format de métriques intégrées. Ces erreurs se produisent lorsque les définitions de métriques dans les journaux ne sont conformes ni au format de métriques intégrées ni aux spécifications de `MetricDatum` . Pour plus d'informations sur le format de métrique CloudWatch intégré, voir Spécification : format de métrique intégré. Pour plus d'informations sur le type de données`MetricDatum` , consultez MetricDatumle Amazon CloudWatch API Reference. **Note** Certaines erreurs de validation peuvent empêcher la publication de plusieurs métriques d'un journal EMF. Par exemple, toutes les métriques définies avec un espace de noms invalide seront supprimées. Dimensions valides : `LogGroupName` Statistique valide : somme Unités : aucune
`ErrorCount`	Le nombre d'opérations API effectuées dans votre compte qui ont donné lieu à des erreurs. `ErrorCount` est une métrique d'utilisation du service CloudWatch Logs. Pour de plus amples informations, veuillez consulter CloudWatch Statistiques d'utilisation du service de journalisation. Dimensions valides : Classe, Ressource, Service, Type Statistique valide : somme Unités : aucune

Métrique	Description
`ForwardedBytes`	Volume des événements du journal en octets compressés transférés vers la destination de l'abonnement. Dimensions valides : LogGroupName, DestinationType, FilterName Statistique valide : somme Unités : octets
`Forwarded LogEvents`	Nombre d'événements du journal envoyés à destination de l'abonnem ent. Dimensions valides : LogGroupName DestinationType, FilterName, PolicyLevel Statistique valide : somme Unités : aucune
`IncomingBytes`	Volume des événements du journal en octets non compressés transféré s dans CloudWatch Logs. Lorsqu'il est utilisé avec la dimension `LogGroupName` , il s'agit du volume des événements du journal en octets décompressés et téléchargés vers le groupe de journaux. Dimensions valides : LogGroupName Statistique valide : somme Unités : octets
`IncomingL ogEvents`	Le nombre d'événements du journal chargés dans CloudWatch Logs. Lorsqu'il est utilisé avec la dimension `LogGroupName` , il s'agit du nombre d'événements du journal téléchargés vers le groupe de journaux. Dimensions valides : LogGroupName Statistique valide : somme Unités : aucune

Métrique	Description
`LogEvents WithFindings`	Nombre d'événements de journal correspondant à une chaîne de données que vous auditez à l'aide de la fonctionnalité de protection des données des CloudWatch journaux. Pour de plus amples informations, veuillez consulter Aider à protéger les données sensibles des journaux grâce au masquage. Dimensions valides : aucune Statistique valide : somme Unités : aucune
`ThrottleCount`	Le nombre d'opérations API effectuées dans votre compte qui ont été limitées en raison de quotas d'utilisation. `ThrottleCount` est une métrique d'utilisation du service CloudWatch Logs. Pour de plus amples informations, veuillez consulter CloudWatch Statistiques d'utilisation du service de journalisation. Dimensions valides : Classe, Ressource, Service, Type Statistique valide : somme Unités : aucune

Dimensions pour les métriques CloudWatch Logs

Les dimensions que vous pouvez utiliser avec la plupart des métriques CloudWatch Logs sont répertoriées dans le tableau suivant.

Dimension	Description
`LogGroupName`	Nom du groupe de CloudWatch journaux pour lequel les métriques doivent être affichées.

Dimension	Description
`DestinationType`	La destination de l'abonnement pour les données CloudWatch Logs, qui peut être AWS Lambda Amazon Kinesis Data Streams ou Amazon Data Firehose.
`FilterName`	Nom du filtre d'abonnement qui transfère des données du groupe de journaux vers la destination. Le nom du filtre d'abonnement est automatiquement converti en CloudWatch ASCII et tous les caractères non pris en charge sont remplacés par un point d'interrogation (?).

Dimensions métriques du filtre d'abonnement

Les dimensions des métriques liées aux filtres d'abonnement au niveau du compte sont répertoriées dans le tableau suivant.

Dimension	Description
`PolicyLevel`	Le niveau auquel la politique s'applique. Actuellement, la seule valeur valide pour cette dimension est `AccountPolicy`
`DestinationType`	La destination de l'abonnement pour les données CloudWatch Logs, qui peut être AWS Lambda Amazon Kinesis Data Streams ou Amazon Data Firehose.
`FilterName`	Nom du filtre d'abonnement qui transfère des données du groupe de journaux vers la destination. Le nom du filtre d'abonnement est automatiquement converti en CloudWatch ASCII et tous les caractères non pris en charge sont remplacés par un point d'interrogation (?).

Métriques et dimensions des transformateurs de log

CloudWatch Logs publie les métriques du transformateur de log suivantes CloudWatch dans l'espace de `AWS/Logs` noms.

Métrique	Description
`TransformationErrors`	Nombre d'erreurs rencontrées lors de la transformation des événements du journal avec le transformateur spécifié. Unité : aucune Statistique valide : somme
`TransformedBytes`	Volume de sortie des événements du journal transformé, en octets non compressés. Unité : octets Statistique valide : somme
`TransformedLogEvents`	Le nombre d'événements du journal transformé. Unité : aucune Statistique valide : somme

Les dimensions suivantes sont utilisées par Transformer Metrics.

Dimension	Description
`LogGroupname`	Cette dimension est utilisée uniquement pour les log-group-level transformateurs.
`PolicyLevel`	Cette dimension est utilisée uniquement pour les transform ateurs au niveau du compte. Actuellement, la seule valeur valide pour cette dimension est `AccountPolicy`

CloudWatch Statistiques d'utilisation du service de journalisation

CloudWatch Logs envoie des métriques CloudWatch permettant de suivre l'utilisation des opérations de l'API CloudWatch Logs. Ces mesures correspondent aux quotas AWS de service. Le suivi de

ces métriques peut vous aider à gérer de manière proactive vos quotas. Pour plus d'informations, consultez Métriques d'intégration et d'utilisation de Service Quotas.

Par exemple, vous pouvez suivre la métrique `ThrottleCount` ou définir une alarme sur cette métrique. Si la valeur de cette métrique augmente, vous devriez envisager de demander une augmentation de quota pour l'opération API qui est limitée. Pour plus d'informations sur CloudWatch les quotas du service Logs, consultezCloudWatch Quotas de journaux.

CloudWatch Logs publie les mesures d'utilisation des quotas de service toutes les minutes dans les `AWS/Logs` espaces de noms `AWS/Usage` et.

Le tableau suivant répertorie les mesures d'utilisation des services publiées par CloudWatch Logs. Ces métriques n'ont pas d'unité spécifique. La statistique la plus utile pour ces métriques est `SUM`, qui représente le nombre total d'opérations pour une période d'une minute.

Chacun de ces métriques est publié avec des valeurs pour toutes les dimensions `Service`, `Class`, `Type` et `Resource`. Ils sont également publiés dans une seule dimension appelée `Account Metrics`. Utilisez la dimension `Account Metrics` pour voir la somme des métriques de toutes les opérations API de votre compte. Utilisez les autres dimensions et spécifiez le nom d'une opération API pour la dimension `Resource` afin de trouver les métriques pour cette API particulière.

Métriques

Métrique	Description
`CallCount`	Nombre d'opérations spécifiées effectuées dans votre compte. `CallCount` est publiée dans les espaces de noms `AWS/Usage` et `AWS/Logs` .
`ErrorCount`	Le nombre d'opérations API effectuées dans votre compte qui ont donné lieu à des erreurs. `ErrorCount` est publiée uniquement dans `AWS/Logs`.
`ThrottleCount`	Le nombre d'opérations API effectuées dans votre compte qui ont été limitées en raison de quotas d'utilisation. `ThrottleCount` est publiée uniquement dans `AWS/Logs`.

Dimensions

Dimension	Description
`Account metrics`	Utilisez cette dimension pour obtenir la somme de la métrique pour tous les CloudWatch journaux APIs. Si vous voulez voir les métriques pour une API particulière, utilisez les autres dimensions répertoriées dans ce tableau et spécifiez le nom de l'API comme valeur de `Resource`.
`Service`	Nom du AWS service contenant la ressource. Pour les métriques d'utilisa tion des CloudWatch journaux, la valeur de cette dimension est`Logs`.
`Class`	Classe de ressource suivie. CloudWatch Les métriques d'utilisation de l'API Logs utilisent cette dimension avec une valeur de`None`.
`Type`	Type de ressource suivi. Actuellement, lorsque la dimension `Service` est `Logs`, la seule valeur valide pour `Type` est `API`.
`Resource`	Nom de l'opération d'API. Les valeurs valides incluent tous les noms d'opérations API répertoriés dans Actions. Par exemple, `PutLogEve nts`

CloudWatch Quotas de journaux

Vous pouvez utiliser le tableau de cette section pour consulter les quotas de service par défaut, également appelés limites, pour un AWS compte dans Amazon CloudWatch Logs. La plupart des quotas de service, mais pas tous, sont répertoriés dans l'espace de noms Amazon CloudWatch Logs dans la console Service Quotas.

> **Note**
>
> Si vous souhaitez demander une augmentation de quota pour l'un de ces quotas, consultez la procédure décrite dans cette section.

Nom	Par défaut	Ajusta	Description
Tâche d'exportation active	Par région prise en charge : 1	Non	Le nombre de tâches d'exportation actives (en cours ou en attente) par compte
AssociateKmsKey limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' associate-kms-keyappels par seconde et par account/per région
Taille de lot	Chaque Région prise en charge : 1 mégaoctet	Non	La taille maximale du lot en Mo d'une put-log-events demande
CancelExportTask limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' cancel-export-task appels par seconde et par account/per région
CreateExportTask limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' create-export-task

Nom	Par défaut	Ajusta	Description
			appels par seconde et par account/per région
CreateLogGroup limite d'accélération des transactions par seconde	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum d' create-log-groupappels par seconde et par account/per région
CreateLogStream limite d'accélération des transactions par seconde	Chaque région prise en charge : 50 par seconde	Oui	Le nombre maximum d' create-log-streamappels par seconde et par account/per région
Archivage des données	Chaque Région prise en charge : 5 giga-octets	Non	La taille maximale en Go d'archivage gratuit des données
DeleteDestination limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels à destination supprimée par seconde et par région account/per
DeleteLogGroup limite d'accélération des transactions par seconde	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum d' delete-log-groupappels par seconde et par account/per région
DeleteLogStream limite d'accélération des transactions par seconde	Chaque région prise en charge : 15 par seconde	Oui	Le nombre maximum d' delete-log-streamappels par seconde et par account/per région
DeleteMetricFilter limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' delete-metric-filterappels par seconde et par account/per région

Nom	Par défaut	Ajusta	Description
DeleteRetentionPolicy limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' delete-retention-policyappels par seconde et par account/per région
DeleteSubscriptionFilter limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' delete-subscription-filterappels par seconde et par account/per région
DeleteTransformer limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels de suppresse-transformateur par seconde et par région account/per
DescribeDestinations limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels de description des destinations par seconde et par région account/per
DescribeExportTasks limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' describe-export-tasksappels par seconde et par account/per région
DescribeLogGroups limite d'accélération des transactions par seconde	Chaque Région prise en charge : 10 par seconde	Oui	Le nombre maximum d' describe-log-groupsappels par seconde et par account/per région
DescribeLogStreams limite d'accélération des transactions par seconde	Chaque région prise en charge : 25 par seconde	Oui	Le nombre maximum d' describe-log-streamsappels par seconde et par account/per région

Nom	Par défaut	Ajusta	Description
DescribeMetricFilters limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' describe-metric-filtersappels par seconde et par account/per région
DescribeSubscriptionFilters limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' describe-subscription-filtersappels par seconde et par account/per région
Taille de l'événement	Chaque région prise en charge : 1 024 kilo-octets	Non	Taille maximale des événements du journal en Ko
FilterLogEvents limite d'accélération des transactions par seconde	us-east-1 : 25 par seconde ap-northeast-3 : 5 par seconde ap-southeast-3 : 5 par seconde eu-central-1 : 5 par seconde Chacune des autres régions prises en charge : 10 par seconde	Non	Le nombre maximum d' filter-log-eventsappels par seconde et par account/per région

Nom	Par défaut	Ajusta	Description
GetLogEvents limite d'accélération des transactions par seconde	us-west-2 : 10 par seconde ap-northeast-3 : 10 par seconde ap-southeast-3 : 10 par seconde eu-central-1 : 10 par seconde eu-west-1 : 10 par seconde eu-west-3 : 30 par seconde Chacune des autres régions prises en charge : 25 par seconde	Non	Le nombre maximum d' get-log-eventsappels par seconde et par account/per région
GetQueryResults limite d'accélération des transactions par seconde	Chaque région prise en charge : 5	Non	Le nombre maximum d' get-query-resultsappels par seconde et par account/per région
GetTransformer limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels get-transformer par seconde et par région account/per
ListTagsForResource limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' list-tags-for-resourceappels par seconde et par account/per région

Nom	Par défaut	Ajusta	Description
ListTagsLogGroup limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' list-tags-log-grou pappels par seconde et par account/per région
Limite de sessions simultanées de Live Tail	Chaque région prise en charge : 15	Oui	Le nombre maximum de sessions Live Tail actives simultanées par compte
Groupes de journaux	Chaque Région prise en charge : 1 000 000	Oui	Le nombre maximum de groupes de journaux qu'un compte peut avoir
Groupes de journaux analysés par session Live Tail	Chaque Région prise en charge : 10	Non	Le nombre maximum de groupes de journaux pouvant être analysés par session Live Tail
Filtres de métriques par groupe de journaux	Chaque Région prise en charge : 100	Non	Le nombre de filtres métriques par groupe de logs
PutDestination limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels à destination par seconde et par région account/per
PutDestinationPolicy limite d'accélér ation des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' put-destination-po licyappels par seconde et par account/per région
PutLogEvents limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 000 par seconde	Oui	Le nombre maximum d' put-log-eventsappels par seconde et par account/p er région

Nom	Par défaut	Ajusta	Description
PutMetricFilter limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' put-metric-filterappels par seconde et par account/per région
PutRetentionPolicy limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' put-retention-policyappels par seconde et par account/per région
PutSubscriptionFilter limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' put-subscription-filterappels par seconde et par account/per région
PutTransformer limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels de put-transformateur par seconde et par région account/per
Politiques basées sur une ressource	Chaque Région prise en charge : 10	Non	Le nombre maximum de politiques de ressources par account/per région
StartLiveTail limite d'accélération des transactions par seconde	Chaque région prise en charge : 5	Non	Le nombre maximum d' start-live-tailappels par seconde et par account/per région. Cette limite s'applique indépendamment à la console et à l'API
StartQuery limite d'accélération des transactions par seconde	Chaque région prise en charge : 5	Non	Le nombre maximum d'appels de requête de démarrage par seconde et par région account/per

Nom	Par défaut	Ajusta	Description
Filtres d'abonnement par groupe de journaux	Chaque région prise en charge : 2	Non	Le nombre de filtres d'abonnement par groupe de journaux
TagLogGroup limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' tag-log-groupappels par seconde et par account/per région
TagResource limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels de ressources de balisage par seconde et par région account/per
TestMetricFilter limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' test-metric-filterappels par seconde et par account/per région
TestTransformer limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels de transform ateurs de test par seconde et par région account/per
UntagLogGroup limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d' untag-log-groupapp els par seconde et par account/per région
UntagResource limite d'accélération des transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Le nombre maximum d'appels untag-resource par seconde et par région account/per

Nom	Par défaut	Ajusta	Description
enregistre les détecteurs d'anomalies	Chaque région prise en charge : 500	Oui	Le nombre maximum de détecteurs d'anomalies logarithmiques actifs

Gestion des quotas de votre service CloudWatch Logs

CloudWatch Logs est intégré à Service Quotas, un AWS service qui vous permet de consulter et de gérer vos quotas à partir d'un emplacement central. Pour plus d'informations, veuillez consulter Qu'est-ce que Service Quotas? dans le Guide de l'utilisateur Service Quotas.

Service Quotas permet de rechercher facilement la valeur de vos quotas de service CloudWatch Logs.

AWS Management Console

Pour afficher les quotas du service CloudWatch Logs à l'aide de la console

1. Ouvrez la console Service Quotas sur https://console.aws.amazon.com/servicequotas/.
2. Dans le volet de navigation, choisissez Services AWS .
3. Dans la liste des AWS services, recherchez et sélectionnez Amazon CloudWatch Logs.

   Dans la liste Service quotas (Quotas de service), vous pouvez voir le nom du Service Quota, la valeur appliquée (le cas échéant), le quota AWS par défaut et si la valeur du quota est réglable.
4. Pour afficher des informations supplémentaires sur un quota de service, notamment la description, choisissez le nom du quota.
5. (Facultatif) Pour demander une augmentation de quota, sélectionnez le quota que vous souhaitez augmenter, sélectionnez Request quota increase (Demander une augmentation de quota), saisissez ou sélectionnez les informations requises, puis sélectionnez Request (Demander).

Pour utiliser davantage les quotas de service à l'aide de la console, consultez le Guide de l'utilisateur Service Quotas. Pour demander une augmentation de quota, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

AWS CLI

Pour consulter les quotas du service CloudWatch Logs à l'aide du AWS CLI

Exécutez la commande suivante pour afficher les quotas de CloudWatch journalisation par défaut.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code logs \
    --output table
```

Pour travailler davantage avec les quotas de service à l'aide du AWS CLI, consultez le Guide de référence des AWS CLI commandes Service Quotas. Pour demander une augmentation de quota, consultez la commande `request-service-quota-increase` dans la référence des commandes AWS CLI.

Historique du document

Le tableau suivant décrit les modifications importantes apportées à chaque version du Guide de l'utilisateur CloudWatch des journaux, à compter de juin 2018. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date
CloudWatch Exemple de politique mise à jour sur la détection des anomalies dans les journaux	CloudWatch Logs a mis à jour l'exemple de politique KMS afin de réduire sa portée et d'améliorer la sécurité en ajoutant des conditions pour `aws:SourceAccount` et`aws:SourceArn` . Pour plus d'informations, voir Chiffrer un détecteur d'anomalie et ses résultats avec KMS AWS	14 juillet 2025
CloudWatch Logs Insights ajoute la prise en charge des journaux du serveur de routage Amazon VPC	CloudWatch Logs Insights ajoute la prise en charge des journaux du serveur de routage Amazon VPC. Pour plus d'informations, voir Activer la journalisation à partir AWS des services. La nouvelle source de journal pour Amazon VPC Route Server `EVENT_LOGS` est documentée dans PutDeliverySource	12 juin 2025
CloudWatch Logs Insights ajoute la prise en charge PCS AWS des journaux	CloudWatch Logs Insights ajoute la prise en charge PCS AWS des journaux. Pour plus d'informations,	12 juin 2025

	voir Activer la journalisation à partir AWS des services. Les nouvelles sources de log pour PCS AWS, `PCS_SCHEDULER_LOGS` et `PCS_JOBCOMP_LOGS` sont documentées dans PutDeliverySource	
CloudWatch Logs Insights ajoute la prise en charge Résolution des entités AWS des journaux	CloudWatch Logs Insights ajoute la prise en charge Résolution des entités AWS des journaux. Pour plus d'informations, voir Activer la journalisation à partir AWS des services. La nouvelle source de journal pour Résolution des entités AWS, `WORKFLOW_LOGS` est documentée dans PutDeliverySource	22 mai 2025
CloudWatch Enregistre les mises à jour des politiques gérées pour prendre en charge les résumés en langage naturel	Des autorisations pour `cloudwatch:GenerateQueryResultsSummary` ont été ajoutées CloudWatchLogsFullAccesset CloudWatchLogsReadOnlyAccesspermettent de générer un résumé en langage naturel des résultats de la requête. Pour consulter le contenu des politiques, consultez CloudWatchLogsFullAccesset CloudWatchLogsReadOnlyAccessdans le Guide de référence des politiques AWS gérées.	20 mai 2025

CloudWatch Logs Insights ajoute la prise en charge de la génération de résumés en langage naturel à partir des résultats des requêtes CloudWatch Logs Insights	Ajout de la prise en charge des résumés en langage naturel dans CloudWatch Logs Insights. Cette fonctionnalité génère des résumés lisibles par l'homme des résultats des requêtes, actuellement disponibles dans l'est des États-Unis (Virginie du Nord). Pour plus d'informations, voir Générer des résumés en langage naturel à partir des résultats des requêtes CloudWatch Logs Insights.	20 mai 2025
CloudWatchOpenSearchDashboardAccesset CloudWatchOpenSearchDashboardsFullAccesssont de nouvelles politiques IAM	CloudWatch Les journaux ont ajouté deux nouvelles politiques IAM, CloudWatchOpenSearchDashboardsFullAccesset CloudWatchOpenSearchDashboardAccess. CloudWatchOpenSearchDashboardsFullAccessaccorde l'autorisation de créer et de gérer des intégrations avec le OpenSearch Service. CloudWatchOpenSearchDashboardAccessautorise l'accès aux tableaux de bord des journaux vendus créés dans ces intégrations. Pour plus d'informations, consultez les tableaux de bord de Vended Log fournis par Amazon OpenSearch Service.	1er décembre 2024

CloudWatchLogsFull Accesspolitique mise à jour	CloudWatch Les journaux ont ajouté des autorisations pour Amazon OpenSearch Service et IAM à la CloudWatchLogsFullAccesspolitique, afin de permettre l'intégration CloudWatch des journaux au OpenSearch service pour certaines fonctionnalités.	1er décembre 2024
CloudWatch Logs Insights ajoute de nouveaux types de structure à la syntaxe des requêtes	CloudWatch Logs Insights ajoute la `unnest` commande et deux fonctions JSON, qui vous permettent d'utiliser des chaînes JSON sous forme de cartes et de listes. Pour plus d'informations, consultez la section Types de structures.	21 novembre 2024
CloudWatch Les journaux prennent en charge la transformation des journaux lors de l'ingestion des journaux	Vous pouvez créer des transformateurs de journaux qui peuvent modifier les événements des journaux au moment de l'ingestion, vous aidant ainsi à normaliser vos journaux dans différents formats et provenant de différentes sources dans des formats cohérents et riches en contexte. Pour plus d'informations, voir Transformer les journaux pendant l'ingestion.	20 novembre 2024

CloudWatch Logs Insights ajoute l'indexation des champs	CloudWatch Logs Insights a ajouté la prise en charge de l'indexation des journaux par champs. Lorsque vous utilisez ensuite un index de champ dans une requête CloudWatch Logs Insights, la requête tente d'ignorer le traitement des événements du journal dont on sait qu'ils n'incluent pas le champ indexé. Pour plus d'informations, voir Création d'index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation.	20 novembre 2024
CloudWatch Le support de Logs Insights pour la génération de requêtes en langage naturel est généralement disponible	CloudWatch Logs Insights prend en charge le langage naturel pour générer et mettre à jour les requêtes. Pour plus d'informations, voir Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Logs Insights.	20 juin 2024
CloudWatchLogsReadOnlyAccesspolitique mise à jour	CloudWatch Logs a ajouté l'`cloudwatch:GenerateQuery` autorisation de CloudWatchLogsReadOnlyAccess, afin que les utilisateurs soumis à cette politique puissent générer une chaîne de requête CloudWatch Logs Insights à partir d'une invite en langage naturel.	26 novembre 2023

CloudWatchLogsFull Accesspolitique mise à jour	CloudWatch Logs a ajouté l'`cloudwatch:GenerateQuery` autorisation de CloudWatchLogsFullAccess, afin que les utilisateurs soumis à cette politique puissent générer une chaîne de requête CloudWatch Logs Insights à partir d'une invite en langage naturel.	26 novembre 2023
CloudWatch Logs ajoute une analyse des modèles de journaux	CloudWatch Logs recherche désormais des modèles dans les événements de journal chaque fois que vous effectuez une requête CloudWatch Logs Insights. Pour plus d'informations, consultez la section Analyse des modèles.	26 novembre 2023
CloudWatch Logs ajoute la détection des anomalies dans les journaux	Vous pouvez créer un détecteur d'anomalies de journal pour un groupe de journaux. Le détecteur d'anomalies analyse les événements du journal ingérés dans le groupe de journaux et détecte les anomalies dans les données du journal. Pour plus d'informations, consultez la section Détection d'anomalies de journal.	26 novembre 2023

CloudWatch Logs ajoute une fonctionnalité de comparaison	Vous pouvez désormais utiliser CloudWatch Logs Insights pour comparer l'évolution des événements de votre journal au fil du temps. Pour plus d'informations, voir Comparer (diff) avec les plages temporelles précédentes.	26 novembre 2023
CloudWatch Logs ajoute une nouvelle classe de journaux	CloudWatch Logs prend en charge deux catégories de groupes de journaux, ce qui vous permet de disposer d'une option rentable pour les journaux auxquels vous accédez rarement, et d'une option complète pour les journaux qui nécessitent une surveillance en temps réel ou d'autres fonctionnalités. Pour plus d'informations, consultez la rubrique Classes de journal.	26 novembre 2023
CloudWatch Logs Insights prend en charge la génération de requêtes en langage naturel	CloudWatch Logs Insights prend en charge le langage naturel pour générer et mettre à jour les requêtes. Pour plus d'informations, voir Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Logs Insights.	26 novembre 2023

CloudWatch Logs ajoute la prise en charge de la syntaxe des modèles de filtres d'expressions régulières pour Live Tail	Vous pouvez désormais personnaliser davantage vos opérations de recherche et de correspondance pour répondre à vos besoins grâce à des expressions régulière s flexibles intégrées à des modèles de filtres Live Tail. Pour plus d'informations, consultez la section Syntaxe du modèle de filtre dans le guide de l'utilisateur Amazon CloudWatch Logs.	13 novembre 2023
CloudWatch Logs ajoute la prise en charge de la syntaxe des modèles de filtres d'expressions régulière s pour les filtres métriques, les filtres d'abonnement et les événements des journaux de filtres	Vous pouvez désormais personnaliser davantage vos opérations de recherche et de correspondance pour répondre à vos besoins grâce à des expressions régulière s flexibles intégrées à des modèles de filtres. Pour plus d'informations, consultez la section Syntaxe du modèle de filtre dans le guide de l'utilisa teur Amazon CloudWatch Logs.	5 septembre 2023

CloudWatch Logs Insights ajoute une commande de modèle	Vous pouvez désormais utiliser des modèles dans vos requêtes CloudWatch Logs Insights pour regrouper automatiquement vos données de journal en modèles. Un modèle est une structure de texte partagée récurrente dans les champs de vos journaux. Pour plus d'informations, consultez le modèle dans le guide de l'utilisateur d'Amazon CloudWatch Logs.	17 juillet 2023
CloudWatch Logs Insights ajoute une commande de déduplication	Vous pouvez désormais utiliser le dédoublement dans vos requêtes CloudWatch Logs Insights pour supprimer les résultats dupliqués en fonction de valeurs spécifiques dans les champs que vous spécifiez. Pour plus d'informations, consultez la section dedup dans le guide de l'utilisateur d'Amazon CloudWatch Logs.	20 juin 2023

Politiques de protection des données au niveau du compte	Vous pouvez désormais définir des politiques de protectio n des données au niveau du compte. Ces politique s au niveau du compte peuvent auditer et masquer les informations sensibles dans les événements du journal dans tous les groupes de journaux du compte. Pour plus d'informations, consultez la section Aider à protéger les données de journal sensibles grâce au masquage dans le guide de l'utilisateur d'Amazon CloudWatch Logs.	8 juin 2023
Ajout de la fonctionnalité Live Tail	CloudWatch Les journaux ont ajouté la fonctionnalité Live Tail, qui vous permet de scanner les journaux au fur et à mesure qu'ils sont ingérés pour faciliter le dépannage . Vous pouvez éventuell ement filtrer le flux d'événeme nts du journal affiché en fonction de termes spécifiés , et également mettre en évidence les événements du journal contenant des termes spécifiés. Pour plus d'informa tions, veuillez consulter Utilisation de Live Tail pour visualiser les journaux en temps quasi réel.	6 juin 2023

CloudWatchLogsRead OnlyAccesspolitique mise à jour	CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsRead OnlyAccess. Les `logs:StopLiveTail` autorisations `logs:StartLiveTail` et ont été ajoutées afin que les utilisateurs soumis à cette politique puissent utiliser la console pour démarrer et arrêter CloudWatch les sessions Logs Live Tail. Pour plus d'informations, veuillez consulter Utilisation de Live Tail pour visualiser les journaux en temps quasi réel.	6 juin 2023
CloudWatch Publication de Logs Insights	Vous pouvez utiliser CloudWatch Logs Insights pour rechercher et analyser les données de vos journaux de manière interactive. Pour plus d'informations, consultez Analyser les données des CloudWatch journaux avec Logs Insights dans le guide de l'utilisateur Amazon CloudWatch Logs.	27 novembre 2018

Prise en charge des points de terminaison Amazon VPC	Vous pouvez désormais établir une connexion privée entre votre VPC et CloudWatch Logs. Pour plus d'informations, consultez la section Utilisation CloudWatch des journaux avec les points de terminaison VPC d'interface dans le guide de l'utilisateur Amazon CloudWatch Logs.	28 juin 2018

Le tableau suivant décrit les modifications importantes apportées au guide de l'utilisateur d'Amazon CloudWatch Logs.

Modification	Description	Date de publication
Points de terminaison d'un VPC d'interface	Dans certaines régions, vous pouvez utiliser un point de terminaison VPC d'interface pour empêcher le trafic entre votre Amazon VPC et les CloudWatch journaux de quitter le réseau Amazon. Pour de plus amples informations, veuillez consulter Utilisation des CloudWatch journaux avec les points de terminaison VPC de l'interface.	7 mars 2018
Journaux de requêtes DNS de Route 53	Vous pouvez utiliser CloudWatch les journaux pour stocker les journaux relatifs aux requêtes DNS reçues par Route 53. Pour plus d'informations, consultez Qu'est-ce qu'Amazon CloudWatch Logs ? ou Journalisation des requêtes DNS dans le Guide du développeur Amazon Route 53.	7 septembre 2017
Étiquetage des groupes de journaux	Vous pouvez utiliser des balises pour classer vos groupes de journaux par catégories. Pour plus d'informations, consultez Étiqueter les groupes de journaux dans Amazon CloudWatch Logs.	13 décembre 2016

Modification	Description	Date de publication
Améliorations apportées à la console	A partir des graphiques de métriques, vous pouvez accéder aux groupes de journaux associés. Pour plus d'informations, consultez Transition des métriques aux journaux.	7 novembre 2016
Amélioration de la facilité d'utilisation de la console	Nous avons amélioré l'environnement afin de faciliter la recherche, les filtres et le dépannage . Par exemple, vous pouvez maintenant filtrer vos données de journal en fonction d'une plage de dates et d'heures. Pour plus d'informations, consultez Afficher les données du journal envoyées à CloudWatch Logs.	29 août 2016
Ajout de la AWS CloudTrail prise en charge d'Amazon CloudWatch Logs et des nouvelles métriques de CloudWatch Logs	Ajout AWS CloudTrail du support pour CloudWatch les journaux. Pour de plus amples informations, veuillez consulter Logging CloudWatch Logs, API et opérations de console dans AWS CloudTrail.	10 mars 2016
Ajout de la prise en charge de l'exportation des CloudWatch journaux vers Amazon S3	Ajout de la prise en charge de l'exportation des données des CloudWatch journaux vers Amazon S3. Pour de plus amples informations, veuillez consulter Exporter les données du journal vers Amazon S3.	7 décembre 2015

Modification	Description	Date de publication
Ajout de la prise en charge des événements AWS CloudTrail enregistrés dans Amazon CloudWatch Logs	Vous pouvez créer des alarmes CloudWatch et recevoir des notifications concernant une activité d'API particulière telle qu'elle est capturée CloudTrail et utiliser la notification pour résoudre les problèmes.	10 novembre 2014
Ajout de la prise en charge d'Amazon CloudWatch Logs	Vous pouvez utiliser Amazon CloudWatch Logs pour surveiller, stocker et accéder à votre système, à votre application et à vos fichiers journaux personnalisés à partir d'instances Amazon Elastic Compute Cloud (Amazon EC2) ou d'autres sources. Vous pouvez ensuite récupérer les données de journal associées dans CloudWatch Logs à l'aide de la CloudWatch console Amazon, CloudWatch des commandes Logs du AWS CLI ou du SDK CloudWatch Logs. Pour de plus amples informations, veuillez consulter Qu'est-ce qu'Amazon CloudWatch Logs ?.	10 juillet 2014

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans la Glossaire AWS référence.

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.