



Guide de l'utilisateur

# Amazon CloudWatch Logs



# Amazon CloudWatch Logs: Guide de l'utilisateur

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

Les marques et la présentation commerciale d'Amazon ne peuvent pas être utilisées en relation avec un produit ou un service qui n'appartient pas à Amazon, de toute manière susceptible de créer une confusion chez les clients ou de toute manière dénigrant ou discréditant 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  |
| Caractéristiques .....                                                                                                        | 1  |
| AWS Services connexes .....                                                                                                   | 4  |
| Tarification .....                                                                                                            | 5  |
| Concepts .....                                                                                                                | 5  |
| Facturation et coûts .....                                                                                                    | 7  |
| Classes de log .....                                                                                                          | 8  |
| Fonctionnalités prises en charge .....                                                                                        | 8  |
| Prise en main .....                                                                                                           | 12 |
| Conditions préalables .....                                                                                                   | 12 |
| Inscrivez-vous pour un Compte AWS .....                                                                                       | 12 |
| Création d'un utilisateur doté d'un accès administratif .....                                                                 | 13 |
| Configuration de l'interface de ligne de commande .....                                                                       | 14 |
| Utilisation de l' CloudWatch agent unifié .....                                                                               | 15 |
| Utilisation de l' CloudWatch agent précédent .....                                                                            | 15 |
| CloudWatch Conditions préalables requises pour l'agent de journalisation .....                                                | 16 |
| Quick Start : Installation de l'agent sur une instance EC2 Linux en cours d'exécution .....                                   | 17 |
| Quick Start : Installation de l'agent sur une instance EC2 Linux au lancement .....                                           | 24 |
| Démarrage rapide : utilisation CloudWatch des journaux avec les instances de Windows Server 2016 .....                        | 28 |
| Démarrage rapide : utilisation CloudWatch des journaux avec les instances de Windows Server 2012 et Windows Server 2008 ..... | 40 |
| Signaler le statut de l'agent CloudWatch Logs .....                                                                           | 51 |
| Démarez l'agent CloudWatch Logs .....                                                                                         | 52 |
| Arrêter l'agent CloudWatch Logs .....                                                                                         | 52 |
| CloudWatch Référence de l'agent de journalisation .....                                                                       | 53 |
| Démarrage rapide avec CloudFormation .....                                                                                    | 65 |
| Ingestion de journaux via des points de terminaison HTTP .....                                                                | 67 |
| Comportement courant .....                                                                                                    | 68 |
| Authentification par jeton porteur .....                                                                                      | 69 |
| Option 1 : démarrage rapide à l'aide de la AWS console .....                                                                  | 69 |
| Option 2 : Configuration manuelle .....                                                                                       | 71 |
| Autorisations de contrôle pour la génération et l'utilisation CloudWatch des clés d'API Logs .....                            | 74 |

|                                                                      |    |
|----------------------------------------------------------------------|----|
| Rotation des clés API .....                                          | 76 |
| Répondre à une clé d'API compromise .....                            | 78 |
| Bonnes pratiques de sécurité pour les clés d'API .....               | 79 |
| Enregistrement de l'utilisation des clés d'API avec CloudTrail ..... | 80 |
| Point de terminaison OTLP .....                                      | 81 |
| Format des demandes .....                                            | 81 |
| Exemple de demande .....                                             | 81 |
| Réponses .....                                                       | 82 |
| Comportements spécifiques à l'OTLP .....                             | 83 |
| Point de terminaison HLC .....                                       | 83 |
| Modes d'entrée .....                                                 | 83 |
| Champ de l'événement (obligatoire) .....                             | 84 |
| Champ horaire (facultatif) .....                                     | 84 |
| Content-Type .....                                                   | 85 |
| Types de valeurs JSON acceptés .....                                 | 85 |
| Format du point de terminaison .....                                 | 85 |
| Format des demandes .....                                            | 86 |
| Exemple de demande .....                                             | 87 |
| Bonnes pratiques .....                                               | 87 |
| Limitations .....                                                    | 88 |
| Point de terminaison NDJSON .....                                    | 88 |
| Format des demandes .....                                            | 89 |
| Types de valeurs JSON acceptés .....                                 | 89 |
| Champ d'horodatage .....                                             | 90 |
| Lignes non valides .....                                             | 91 |
| Exemple de demande .....                                             | 91 |
| Réponses .....                                                       | 91 |
| Bonnes pratiques .....                                               | 92 |
| Limitations .....                                                    | 93 |
| Point de terminaison JSON structuré .....                            | 93 |
| Format des demandes .....                                            | 93 |
| Types de valeurs JSON acceptés .....                                 | 94 |
| Champ d'horodatage .....                                             | 94 |
| Exemple de demande .....                                             | 95 |
| Réponses .....                                                       | 95 |
| Bonnes pratiques .....                                               | 96 |

|                                                                                                                   |     |
|-------------------------------------------------------------------------------------------------------------------|-----|
| Limitations .....                                                                                                 | 96  |
| Comparaison des points de terminaison d'ingestion HTTP .....                                                      | 97  |
| Choix d'un point de terminaison .....                                                                             | 98  |
| Travailler avec AWS SDKs .....                                                                                    | 99  |
| Gestion des journaux .....                                                                                        | 101 |
| Découverte et gestion des sources de données .....                                                                | 101 |
| Qu'est-ce que les sources de données CloudWatch Logs ? .....                                                      | 102 |
| Prise en main .....                                                                                               | 103 |
| Champs du système .....                                                                                           | 104 |
| Accès aux sources de données .....                                                                                | 105 |
| Relation avec les groupes de journaux .....                                                                       | 105 |
| Fonctionnalités activées par les sources de données .....                                                         | 106 |
| Pris en charge Services AWS pour les sources de données .....                                                     | 106 |
| Sources tierces prises en charge pour les sources de données .....                                                | 111 |
| Analyse des données des CloudWatch journaux avec Logs Insights .....                                              | 115 |
| Langages de requête pris en charge .....                                                                          | 118 |
| CloudWatch Langage de requête Logs Insights (Logs Insights QL) .....                                              | 119 |
| OpenSearch Langage de traitement canalisé (PPL) .....                                                             | 188 |
| OpenSearch Langage de requête structuré (SQL) .....                                                               | 201 |
| Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Logs Insights .....             | 216 |
| Exemples de requêtes .....                                                                                        | 217 |
| Refus d'utiliser vos données pour améliorer le service .....                                                      | 220 |
| Journaux pris en charge et champs découverts .....                                                                | 221 |
| Champs des journaux JSON .....                                                                                    | 223 |
| Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation ..... | 225 |
| Syntaxe et quotas d'index de champs .....                                                                         | 229 |
| Création d'une politique d'indexation des champs au niveau du compte .....                                        | 234 |
| Création d'une politique d'index de champs au niveau du groupe de journaux .....                                  | 236 |
| Options de sélection de groupes de journaux lors de la création d'une requête .....                               | 237 |
| Effets de la suppression d'une politique d'index de champs .....                                                  | 238 |
| Utilisez des facettes pour regrouper et explorer les journaux .....                                               | 238 |
| Pour exécuter une requête basée sur les facettes .....                                                            | 239 |
| Pour enregistrer une requête basée sur des facettes .....                                                         | 240 |
| Pour créer une facette au niveau du compte .....                                                                  | 240 |

|                                                                                             |     |
|---------------------------------------------------------------------------------------------|-----|
| Gestion des facettes à l'aide de APIs .....                                                 | 240 |
| Analyse de modèles .....                                                                    | 241 |
| Commencer à utiliser l'analyse de modèles .....                                             | 242 |
| Détails sur la commande pattern .....                                                       | 245 |
| Enregistrer et réexécuter des requêtes .....                                                | 245 |
| Utilisation de requêtes enregistrées avec paramètres .....                                  | 251 |
| Ajouter une requête au tableau de bord ou exporter les résultats de la requête .....        | 254 |
| Afficher les requêtes en cours d'exécution ou l'historique des requêtes .....               | 255 |
| Chiffrez les résultats des requêtes avec AWS Key Management Service .....                   | 256 |
| Restrictions .....                                                                          | 256 |
| Étape 1 : Création d'un AWS KMS key .....                                                   | 257 |
| Étape 2 : Définition des autorisations sur la clé KMS .....                                 | 257 |
| Étape 3 : Associer une clé KMS à vos résultats de requête .....                             | 259 |
| Étape 4 : Dissocier une clé des résultats de requête dans le compte .....                   | 259 |
| Génération d'un résumé en langage naturel à partir des résultats des requêtes de CloudWatch |     |
| Logs Insights .....                                                                         | 259 |
| Comment ça marche .....                                                                     | 260 |
| Disponibilité régionale et traitement des données .....                                     | 260 |
| Prise en main .....                                                                         | 261 |
| Permissions .....                                                                           | 261 |
| Confidentialité des données .....                                                           | 261 |
| Automatisation de l'analyse des journaux avec des requêtes planifiées .....                 | 263 |
| Comprendre les concepts des requêtes planifiées .....                                       | 263 |
| Séparation des rôles IAM .....                                                              | 264 |
| Utilisation entre régions et entre comptes .....                                            | 266 |
| Expressions de planification et gestion des fuseaux horaires .....                          | 267 |
| Choix d'une langue de requête .....                                                         | 268 |
| Sélection de destinations et cas d'utilisation .....                                        | 269 |
| Format et structure des résultats de requête .....                                          | 270 |
| Référence d'expression de planification .....                                               | 271 |
| Bonnes pratiques .....                                                                      | 276 |
| Commencer à utiliser les requêtes planifiées .....                                          | 277 |
| Création d'une requête planifiée .....                                                      | 278 |
| Affichage et gestion des requêtes planifiées .....                                          | 282 |
| Afficher l'historique d'exécution des requêtes planifiées .....                             | 283 |
| Mettre à jour une requête planifiée .....                                                   | 285 |

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Configuration des destinations S3 pour les requêtes planifiées .....                | 287 |
| Transmission des résultats vers un compartiment Amazon S3 dans le même compte ..... | 287 |
| Transmission des résultats vers un compartiment Amazon S3 d'un autre compte .....   | 289 |
| Chiffrer les résultats à l'aide d'une clé gérée par AWS KMS le client .....         | 290 |
| Résolution des problèmes de requêtes planifiées .....                               | 292 |
| L'exécution de la requête échoue en raison d'erreurs d'autorisation .....           | 292 |
| Expiration du délai de requête .....                                                | 293 |
| Le traitement de la destination échoue .....                                        | 294 |
| Erreurs de requête non valides .....                                                | 294 |
| Erreurs de simultanéité des requêtes .....                                          | 295 |
| Détection des anomalies du journal .....                                            | 296 |
| Gravité et priorité des anomalies et des modèles .....                              | 297 |
| Durée de visibilité des anomalies .....                                             | 298 |
| Suppression d'une anomalie .....                                                    | 298 |
| Questions fréquentes (FAQ) .....                                                    | 298 |
| Utilisation de la détection des anomalies dans CloudWatch Logs Insights .....       | 300 |
| Activer la détection des anomalies sur un groupe de journaux .....                  | 300 |
| Afficher les anomalies détectées .....                                              | 302 |
| Créez des alarmes sur les détecteurs d'anomalies du journal .....                   | 305 |
| Métriques publiées par les détecteurs d'anomalies logarithmiques .....              | 307 |
| Chiffrez un détecteur d'anomalie et ses résultats avec AWS KMS .....                | 308 |
| Restrictions .....                                                                  | 308 |
| Résoudre les problèmes avec CloudWatch Logs Live Tail .....                         | 313 |
| Démarez une session Live Tail à l'aide du AWS CLI .....                             | 314 |
| impression uniquement .....                                                         | 314 |
| interactif .....                                                                    | 314 |
| Démarrer une session Live Tail dans la console .....                                | 316 |
| Centralisation des journaux inter-comptes et inter-régions .....                    | 320 |
| Concepts de centralisation des données .....                                        | 320 |
| Configuration de la centralisation des journaux .....                               | 322 |
| Conditions préalables .....                                                         | 322 |
| Personnalisation des noms des groupes de journaux de destination .....              | 323 |
| Création d'une règle de centralisation .....                                        | 325 |
| Modifier une règle de centralisation .....                                          | 328 |
| Afficher une règle de centralisation .....                                          | 328 |
| Supprimer une règle de centralisation .....                                         | 329 |

|                                                                                               |     |
|-----------------------------------------------------------------------------------------------|-----|
| Règles de centralisation de surveillance et de résolution des problèmes .....                 | 329 |
| État de santé des règles de centralisation .....                                              | 330 |
| Surveillance des appels d'API de centralisation avec AWS CloudTrail .....                     | 330 |
| Recommandations de surveillance .....                                                         | 331 |
| Utilisation des groupes de journaux et des flux de journaux .....                             | 334 |
| Création d'un groupe de journaux .....                                                        | 334 |
| Envoi de journaux à un groupe de journaux .....                                               | 335 |
| Affichage des données de journal .....                                                        | 335 |
| Recherche de données journal au moyen de modèles de filtres .....                             | 336 |
| Recherche d'entrées de journal à l'aide de la console .....                                   | 337 |
| Recherchez les entrées du journal à l'aide du AWS CLI .....                                   | 337 |
| Transition des métriques aux journaux .....                                                   | 338 |
| Résolution des problèmes .....                                                                | 339 |
| Modification de la conservation des données de journaux .....                                 | 339 |
| Protéger les groupes de journaux contre la suppression .....                                  | 340 |
| Protéger les groupes de journaux contre la suppression .....                                  | 340 |
| Activer la protection contre les suppressions .....                                           | 340 |
| Étiquetage des groupes de journaux .....                                                      | 341 |
| Principes de base des balises .....                                                           | 342 |
| Suivi des coûts à l'aide d'étiquettes .....                                                   | 343 |
| Restrictions liées aux étiquettes .....                                                       | 343 |
| Marquage de groupes de journaux à l'aide du AWS CLI .....                                     | 344 |
| Marquage de groupes de journaux à l'aide de l'API CloudWatch Logs .....                       | 344 |
| Chiffrez les données du journal à l'aide de AWS KMS .....                                     | 345 |
| Restrictions .....                                                                            | 346 |
| Étape 1 : Création d'une AWS KMS clé .....                                                    | 257 |
| Étape 2 : Définition des autorisations sur la clé KMS .....                                   | 347 |
| Étape 3 : Association d'une clé KMS à un groupe de journaux .....                             | 312 |
| Étape 4 : Dissociation de la clé d'un groupe de journaux .....                                | 312 |
| Clés KMS et contexte de chiffrement .....                                                     | 352 |
| Aider à protéger les données sensibles des journaux grâce au masquage .....                   | 356 |
| Comprendre les politiques de protection des données .....                                     | 360 |
| Autorisations IAM requises pour créer ou utiliser une politique de protection des données ... | 362 |
| Création d'une politique de protection des données à l'échelle du compte .....                | 368 |
| Création d'une politique de protection des données pour un seul groupe de journaux .....      | 371 |
| Affichage de données non masquées .....                                                       | 375 |



|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Rapports de résultats d'audit .....                                                 | 375 |
| Types de données que vous pouvez protéger .....                                     | 377 |
| Transformez les journaux lors de l'ingestion .....                                  | 422 |
| Création et gestion de transformateurs de log .....                                 | 424 |
| Création d'une politique de transformation au niveau du compte .....                | 425 |
| Modifier ou supprimer une politique de transformation au niveau du compte .....     | 427 |
| Créez un transformateur de log-group-level bûches à partir de zéro .....            | 427 |
| Créez un log-group-level transformateur en copiant un transformateur existant ..... | 429 |
| Modifier un log-group-level transformateur .....                                    | 429 |
| Supprimer un log-group-level transformateur .....                                   | 430 |
| Processeurs configurables de type analyseur .....                                   | 431 |
| parseJSON .....                                                                     | 431 |
| grok .....                                                                          | 433 |
| csv .....                                                                           | 465 |
| parseKeyValue .....                                                                 | 468 |
| Processeurs intégrés pour les AWS journaux vendus .....                             | 470 |
| Analyser WAF .....                                                                  | 470 |
| Analyser Postgres .....                                                             | 473 |
| ParseCloud Front .....                                                              | 474 |
| Par Route 53 .....                                                                  | 476 |
| Analyser VPC .....                                                                  | 477 |
| Analyser vers OCSF .....                                                            | 478 |
| Processeurs String Mutate .....                                                     | 479 |
| lowerCaseString .....                                                               | 480 |
| upperCaseString .....                                                               | 481 |
| Chaîne divisée .....                                                                | 482 |
| Chaîne de remplacement .....                                                        | 484 |
| TrimString .....                                                                    | 488 |
| Processeurs de mutation JSON .....                                                  | 489 |
| Ajouter des clés .....                                                              | 489 |
| Supprimer les clés .....                                                            | 491 |
| Touches de déplacement .....                                                        | 492 |
| Renommer les clés .....                                                             | 494 |
| Copier la valeur .....                                                              | 496 |
| listToMap .....                                                                     | 498 |
| Processeurs de conversion de types de données .....                                 | 503 |

|                                                                                                                       |     |
|-----------------------------------------------------------------------------------------------------------------------|-----|
| Type de convertisseur .....                                                                                           | 503 |
| Convertisseur Date/Heure .....                                                                                        | 505 |
| Métriques de transformation et erreurs .....                                                                          | 507 |
| Analyser avec Amazon OpenSearch Service .....                                                                         | 508 |
| Étape 1 : Création de l'intégration avec le OpenSearch service .....                                                  | 510 |
| Autorisations requises .....                                                                                          | 511 |
| Création de l'intégration .....                                                                                       | 518 |
| Étape 2 : Création de tableaux de bord de journaux vendus .....                                                       | 520 |
| Afficher, modifier ou supprimer les tableaux de bord des journaux vendus .....                                        | 521 |
| Afficher les tableaux de bord des journaux vendus dans CloudWatch Logs or Service<br>OpenSearch .....                 | 521 |
| Accorder l'accès à des rôles IAM ou à des utilisateurs IAM supplémentaires pour consulter<br>le tableau de bord ..... | 521 |
| Modifier la configuration du tableau de bord .....                                                                    | 522 |
| Supprimer un tableau de bord de vendue log .....                                                                      | 522 |
| Supprimer toutes les intégrations du tableau de bord des journaux automatiques avec<br>Service OpenSearch .....       | 523 |
| Politiques IAM pour les utilisateurs .....                                                                            | 524 |
| Autorisations nécessaires à l'intégration .....                                                                       | 525 |
| Journaux d'accès avec intégration des tables S3 .....                                                                 | 528 |
| Comprendre l'intégration des tables S3 .....                                                                          | 528 |
| Composants de base .....                                                                                              | 528 |
| Flux de données vers les tables S3 .....                                                                              | 528 |
| Quand utiliser l'intégration des tables S3 .....                                                                      | 529 |
| Conditions préalables .....                                                                                           | 529 |
| Autorisations IAM .....                                                                                               | 530 |
| Politique relative aux clés KMS (pour les données chiffrées) .....                                                    | 531 |
| Démarrage .....                                                                                                       | 532 |
| Intégration des tables Amazon S3 aux services AWS d'analyse - Utilisation de la console<br>Amazon S3 (lien) .....     | 534 |
| Configurer les autorisations de Lake Formation .....                                                                  | 534 |
| Connectez-vous aux outils d'analyse .....                                                                             | 535 |
| Filtres de métriques .....                                                                                            | 537 |
| Concepts .....                                                                                                        | 538 |
| Syntaxe des modèles de filtres pour les filtres de métriques .....                                                    | 540 |
| Configuration des valeurs de métriques pour un filtre de métriques .....                                              | 541 |

|                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------|-----|
| Publication de dimensions avec des métriques à partir des événements du journal .....                        | 542 |
| Utilisation de valeurs dans des événements du journal pour incrémenter la valeur d'une<br>métrique .....     | 545 |
| Création de filtres de métriques .....                                                                       | 547 |
| Créer un filtre de métrique pour un groupe de journaux .....                                                 | 547 |
| Exemple : Comptage des événements du journal .....                                                           | 548 |
| Exemple : Comptage des occurrences d'un terme .....                                                          | 550 |
| Exemple : Comptage du nombre de codes HTTP 404 .....                                                         | 552 |
| Exemple : comptage de codes HTTP 4xx .....                                                                   | 554 |
| Exemple : Extraction des champs d'un journal Apache et attribution de dimensions .....                       | 556 |
| Liste des filtres de métriques .....                                                                         | 558 |
| Suppression d'un filtre de métrique .....                                                                    | 559 |
| Filtres d'abonnements .....                                                                                  | 560 |
| Concepts .....                                                                                               | 562 |
| Filtres d'abonnement au niveau des groupes de journaux .....                                                 | 563 |
| Exemple 1 : filtres d'abonnement avec Amazon Kinesis Data Streams .....                                      | 564 |
| Exemple 2 : filtres d'abonnement avec AWS Lambda .....                                                       | 570 |
| Exemple 3 : filtres d'abonnement avec Amazon Data Firehose .....                                             | 574 |
| Exemple 4 : filtres d'abonnement avec Amazon OpenSearch Service .....                                        | 581 |
| Filtres d'abonnement au niveau du compte .....                                                               | 582 |
| Exemple 1 : filtres d'abonnement avec Amazon Kinesis Data Streams .....                                      | 583 |
| Exemple 2 : filtres d'abonnement avec AWS Lambda .....                                                       | 590 |
| Exemple 3 : filtres d'abonnement avec Amazon Data Firehose .....                                             | 594 |
| Abonnements entre comptes et entre régions .....                                                             | 602 |
| Partage de données de journal entre comptes et entre régions à l'aide d'Amazon Kinesis<br>Data Streams ..... | 603 |
| Partage de données de journal entre comptes et entre régions à l'aide de Firehose .....                      | 624 |
| Abonnements entre comptes et entre régions à l'aide d'Amazon Kinesis Data Streams .....                      | 638 |
| Abonnements entre comptes et entre régions à l'aide de Firehose .....                                        | 658 |
| Prévention du député confus .....                                                                            | 670 |
| Prévention de la récursivité dans les journaux .....                                                         | 672 |
| Syntaxe des modèles de filtres .....                                                                         | 674 |
| Expressions régulières prises en charge .....                                                                | 675 |
| Faire correspondre les termes à l'aide d'expressions régulières .....                                        | 678 |
| Faire correspondre les termes dans les événements du journal non structurés .....                            | 679 |
| Faire correspondre les termes dans les événements du journal JSON .....                                      | 682 |

|                                                                                              |     |
|----------------------------------------------------------------------------------------------|-----|
| Faire correspondre les termes dans les événements du journal délimités par des espaces ..... | 691 |
| Activer la journalisation à partir AWS des services .....                                    | 697 |
| Journalisation nécessitant des autorisations supplémentaires [V1] .....                      | 711 |
| Logs envoyés à CloudWatch Logs .....                                                         | 711 |
| Journaux envoyés à Amazon S3 .....                                                           | 715 |
| Logs envoyés à Firehose .....                                                                | 720 |
| Journalisation nécessitant des autorisations supplémentaires [V2] .....                      | 722 |
| Logs envoyés à CloudWatch Logs .....                                                         | 723 |
| Journaux envoyés à Amazon S3 .....                                                           | 727 |
| Logs envoyés à Firehose .....                                                                | 731 |
| Traces envoyées à X-Ray .....                                                                | 734 |
| Autorisations spécifiques au service .....                                                   | 738 |
| Autorisations spécifiques à la console .....                                                 | 739 |
| Exemple de livraison entre comptes .....                                                     | 740 |
| Création d'une source de livraison .....                                                     | 741 |
| Configuration de la livraison vers un compartiment Amazon S3 .....                           | 741 |
| Configurer la diffusion vers un flux Firehose .....                                          | 745 |
| Prévention du cas de figure de l'adjoint désorienté entre services .....                     | 747 |
| Exporter les données du journal vers Amazon S3 .....                                         | 749 |
| Concepts .....                                                                               | 750 |
| Exporter les données du journal vers Amazon S3 à l'aide de la console .....                  | 751 |
| Exportation sur le même compte (console) .....                                               | 752 |
| Exportation entre comptes (console) .....                                                    | 759 |
| Exportez les données du journal vers Amazon S3 à l'aide du AWS CLI .....                     | 769 |
| Exportation sur le même compte (CLI) .....                                                   | 770 |
| Exportation entre comptes (CLI) .....                                                        | 777 |
| Décrire les tâches d'exportation (CLI) .....                                                 | 787 |
| Annuler une tâche d'exportation (CLI) .....                                                  | 789 |
| Transmission de données vers le OpenSearch service .....                                     | 790 |
| Conditions préalables .....                                                                  | 790 |
| Abonnement d'un groupe de logs au OpenSearch Service .....                                   | 791 |
| Exemples de code .....                                                                       | 793 |
| Principes de base .....                                                                      | 794 |
| Actions .....                                                                                | 795 |
| Scénarios .....                                                                              | 862 |
| Configuration d'Amazon ECS Service Connect .....                                             | 863 |

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| Création de votre première fonction Lambda .....                             | 879 |
| Exécution d'une requête volumineuse .....                                    | 890 |
| Utilisent des événements planifiés pour invoquer une fonction Lambda .....   | 927 |
| Chiffrez les tables de recherche à l'aide de AWS KMS .....                   | 931 |
| Comment CloudWatch Logs l'utilise AWS KMS pour les tables de recherche ..... | 932 |
| Autorisations nécessaires .....                                              | 932 |
| Étape 1 : Création d'une AWS KMS clé .....                                   | 933 |
| Étape 2 : Définition des autorisations sur la clé KMS .....                  | 934 |
| Étape 3 : associer une clé KMS à une table de recherche .....                | 935 |
| Considérations .....                                                         | 936 |
| Sécurité .....                                                               | 937 |
| Protection des données .....                                                 | 938 |
| Chiffrement au repos .....                                                   | 939 |
| Chiffrement en transit .....                                                 | 939 |
| Gestion des identités et des accès .....                                     | 939 |
| Authentification .....                                                       | 940 |
| Contrôle d'accès .....                                                       | 940 |
| Présentation de la gestion des accès .....                                   | 941 |
| Utilisation des politiques basées sur une identité (politiques IAM) .....    | 947 |
| Validation de conformité .....                                               | 975 |
| Résilience .....                                                             | 975 |
| Sécurité de l'infrastructure .....                                           | 976 |
| Points de terminaison de VPC d'Interface .....                               | 976 |
| Disponibilité .....                                                          | 977 |
| Création d'un point de terminaison VPC pour les journaux CloudWatch .....    | 977 |
| Test de la connexion entre votre VPC et Logs CloudWatch .....                | 977 |
| Contrôle de l'accès à votre point de CloudWatch terminaison Logs VPC .....   | 978 |
| Prise en charge des clés de contexte de VPC .....                            | 979 |
| Journalisation des opérations d'API et de console avec AWS CloudTrail .....  | 980 |
| Informations de génération de requêtes dans CloudTrail .....                 | 985 |
| Présentation des entrées des fichiers journaux .....                         | 986 |
| Surveillance de l'utilisation à l'aide de CloudWatch métriques .....         | 988 |
| CloudWatch Métriques des journaux .....                                      | 988 |
| Dimensions pour les métriques CloudWatch Logs .....                          | 993 |
| Métriques et dimensions des transformateurs de log .....                     | 994 |
| Mesures et dimensions de centralisation .....                                | 995 |

|                                                                          |       |
|--------------------------------------------------------------------------|-------|
| CloudWatch Statistiques d'utilisation du service de journalisation ..... | 1000  |
| Quotas de service .....                                                  | 1002  |
| Gestion des quotas de votre service CloudWatch Logs .....                | 1014  |
| Historique du document .....                                             | 1016  |
| AWS Glossaire .....                                                      | 1034  |
| .....                                                                    | mxxxv |

# 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](#).

## Caractéristiques

- Deux classes de CloudWatch journaux pour plus de flexibilité — Logs propose deux classes de journaux afin que vous puissiez 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 `NullPointerException` « ») 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.
- Protection contre la suppression : protection qui empêche la suppression accidentelle de groupes de journaux et de leurs flux de journaux. Lorsqu'elle est activée sur un groupe de journaux, la protection contre la suppression bloque toutes les opérations de suppression jusqu'à ce qu'elle soit explicitement désactivée. Par défaut, la protection contre la suppression n'est pas activée. Cette fonctionnalité optionnelle permet de protéger les données opérationnelles et de conformité critiques contre toute suppression involontaire, telles que les groupes de journaux contenant des données d'audit et les journaux des applications de production à des fins de dépannage et d'analyse.
- 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.
- Centralisation des journaux entre comptes et régions : vous pouvez utiliser la centralisation CloudWatch des journaux pour définir des règles de centralisation entre comptes et entre régions qui répliquent les données de journal ingérées dans des environnements multicomptes et régionaux vers une région et un compte centraux. Vous pouvez configurer une région de sauvegarde au sein du compte central pour une résilience accrue, définir des paramètres de chiffrement pour les groupes de journaux nouvellement créés dans le compte central et créer des filtres métriques et d'abonnement sur les événements des journaux provenant de comptes sources et de régions spécifiques avec des fonctionnalités de filtrage améliorées.

# AWS Services connexes

Les services suivants sont utilisés conjointement avec CloudWatch Logs :

- AWS CloudTrail est 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 un exemple du type de données enregistrées CloudWatch dans les fichiers CloudTrail journaux, consultez [Logging CloudWatch Logs, API et opérations de console dans AWS CloudTrail](#).
- Gestion des identités et des accès AWS (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 des 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 codés 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.

## Protection contre la suppression

La protection contre la suppression est une protection qui empêche la suppression accidentelle de groupes de journaux et de leurs flux de journaux. Lorsqu'elle est activée sur un groupe de journaux, la protection contre la suppression bloque toutes les opérations de suppression jusqu'à ce qu'elle soit explicitement désactivée. Par défaut, la protection contre la suppression n'est pas activée. Cette fonctionnalité optionnelle permet de protéger les données opérationnelles et de conformité critiques contre toute suppression involontaire, telles que les groupes de journaux contenant des données d'audit et les journaux des applications de production à des fins de dépannage et d'analyse.

# 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 ✓                                                                                                                             |  |
| <a href="#">Fonctionnalités multi-comptes</a>                                                                         | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">Chiffrement avec AWS KMS</a>                                                                              | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">CloudWatch Commandes de requête Logs Insights</a>                                                         | Oui ✓    | Oui ✓ (La plupart des commandes, voir <a href="#">Commandes Logs Insights QL prises en charge dans les classes de journaux.</a> ) |  |
| <a href="#">CloudWatch Champs découverts par Logs Insights</a>                                                        | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">Facettes</a>                                                                                              | Oui ✓    | Non                                                                                                                               |  |
| <a href="#">Utiliser CloudWatch Pipeline pour transformer les journaux</a>                                            | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">Exporter vers Amazon S3</a>                                                                               | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">Intégration des tables S3</a>                                                                             | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">Requêtes planifiées</a>                                                                                   | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">Utiliser OpenSearch PPL ou OpenSearch SQL pour effectuer des requêtes dans CloudWatch Logs Insights ;</a> | Oui ✓    | Oui ✓                                                                                                                             |  |
| <a href="#">Assistance aux requêtes en langage naturel</a>                                                            | Oui ✓    | Non                                                                                                                               |  |

| Fonctionnalité                                                                     | Standard | Accès peu fréquent                                                                                                                                                             |  |
|------------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <a href="#">CloudWatch Détection des anomalies dans les journaux</a>               | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">Live Tail</a>                                                          | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">Indexation des champs</a>                                              | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">Comparer avec la plage horaire précédente</a>                          | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">Filtres d'abonnement</a>                                               | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">GetLogEvents</a> set opérations <a href="#">FilterLog Events</a> d'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. |  |
| <a href="#">Filtres métriques</a>                                                  | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">Container Insights enregistre l'ingestion</a>                          | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">Ingestion du journal Lambda Insights</a>                               | Oui ✓    | Non                                                                                                                                                                            |  |
| <a href="#">Protection des données sensibles grâce au masquage</a>                 | Oui ✓    | Oui ✓                                                                                                                                                                          |  |



| Fonctionnalité                              | Standard | Accès peu fréquent |  |
|---------------------------------------------|----------|--------------------|--|
| <a href="#">Format de métriques intégré</a> | 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 des groupes de journaux de la classe Delivery sont conservés CloudWatch Logs pendant deux jours. Cette durée de conservation est fixe et ne peut pas être modifiée. 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 instances Amazon EC2 et de vos serveurs locaux dans CloudWatch Logs, utilisez l'agent unifié. CloudWatch 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

- [Conditions préalables](#)
- [Utilisez l' CloudWatch agent unifié pour démarrer avec CloudWatch Logs](#)
- [Utiliser l' CloudWatch agent précédent pour commencer à utiliser CloudWatch Logs](#)
- [Démarrage rapide : CloudFormation à utiliser pour commencer à utiliser les CloudWatch journaux](#)

## Conditions préalables

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 AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique 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

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisez l'utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, 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 Console](#) tant 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, consultez la section [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'instances Amazon EC2 et de serveurs sur site avec l' CloudWatch agent dans le guide](#) de l'utilisateur Amazon CloudWatch . 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'instances EC2 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'instances Amazon EC2 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'instances Amazon EC2 et de serveurs sur site avec l' CloudWatch agent dans le guide](#) de l'utilisateur Amazon CloudWatch .

## 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 Linux EC2 en cours d'exécution](#)
- [Démarrage rapide : installation et configuration de l'agent CloudWatch Logs sur une instance Linux EC2 au lancement](#)
- [Démarrage rapide : autorisez vos instances Amazon EC2 exécutant Windows Server 2016 à envoyer des journaux à Logs à l'aide de l' CloudWatch agent CloudWatch Logs](#)
- [Démarrage rapide : autorisez vos instances Amazon EC2 exécutant Windows Server 2012 et Windows Server 2008 à envoyer des journaux vers Logs CloudWatch](#)
- [Signaler le statut de l'agent CloudWatch Logs](#)
- [Démarez l'agent CloudWatch Logs](#)
- [Arrêter l'agent CloudWatch Logs](#)
- [CloudWatch Référence de l'agent de journalisation](#)

## 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 Linux EC2 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'instances EC2 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'instances EC2 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 Linux EC2 en cours d'exécution

Vous pouvez utiliser le programme d'installation de l'agent CloudWatch Logs sur une instance EC2 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 sur la ligne de commande ou par le biais de 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 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.

```
{
  "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 instance Amazon EC2 existante

Le processus d'installation de l'agent CloudWatch Logs varie selon que votre instance Amazon EC2 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 empêchant 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 l'utilisateur Amazon EC2.

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

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/awsccli.conf` pointe vers la région `us-east-1`. Pour transmettre vos journaux vers une autre région, modifiez le fichier `awsccli.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 awslogs.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 instance EC2. Pour plus d'informations, consultez [Connect to Your Instance](#) dans le guide de l'utilisateur Amazon EC2.

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

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-south-1`, `ap-northeast-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. Sinon, 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 : installation et configuration de l'agent CloudWatch Logs sur une instance Linux EC2 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 les nouvelles instances EC2. 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 Linux EC2 au lancement

Vous pouvez utiliser les données utilisateur Amazon EC2, une fonctionnalité d'Amazon EC2 qui permet de transmettre des informations paramétriques à l'instance lors du lancement, pour installer et configurer CloudWatch l'agent 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 Politiques, 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'utilisateur Amazon EC2.
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 Select Role Type, choisissez Select 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 console Amazon EC2 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'utilisateur Amazon EC2 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 instances Amazon EC2 exécutant Windows Server 2016 à envoyer des journaux à Logs à l'aide de l'CloudWatch agent CloudWatch Logs

### Tip

CloudWatch inclut un nouvel agent unifié capable de collecter à la fois des journaux et des métriques à partir d'instances EC2 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 instances Amazon EC2 exécutant Windows Server 2016 d'envoyer des journaux à Logs à l'aide de l'ancien CloudWatch 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\\W3SVC1`).

 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'attacher un rôle IAM à une instance existante, consultez [Attacher un rôle IAM à une instance](#) dans le guide de l'utilisateur Amazon EC2.

## 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 console Amazon EC2 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'utilisateur Amazon EC2.

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 d'une commande dans la console Amazon EC2.

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

### Tip

CloudWatch inclut un nouvel agent unifié capable de collecter à la fois des journaux et des métriques à partir d'instances EC2 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 instances Amazon EC2 exécutant Windows Server 2012 et Windows Server 2008 d'envoyer des journaux vers Logs CloudWatch

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\W3SVC1).

 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 instance Amazon EC2 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.

## Signaler le statut de l'agent CloudWatch Logs

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

Pour signaler l'état de l'agent

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

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

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 instance EC2 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 instance EC2. Pour plus d'informations, consultez [Connect to Your Instance](#) dans le guide de l'utilisateur Amazon EC2.

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

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 instance EC2.

Pour arrêter l'agent

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

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

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
```

## 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 de l'instance Amazon EC2 et des serveurs sur site avec l'agent. CloudWatch](#) 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'instances Amazon EC2. 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 awsscli-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
```

Pour conserver l'accès au service de métadonnées Amazon EC2 sur les instances EC2, 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 l'utilisateur Amazon EC2.

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 a besoin d'une autorisation pour effectuer `CreateLogGroup`, `CreateLogStream`, `DescribeLogStreams`, `DescribeLogGroups`, `PutLogEvents` et `PutRetentionPolicy` des actions. 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",
        "logs:DescribeLogGroups",
        "logs:PutRetentionPolicy"
      ],
      "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, veuillez 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`.

## Démarrage rapide : 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 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 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 .



# Ingestion de journaux via des points de terminaison HTTP

Amazon CloudWatch Logs fournit des points de terminaison HTTP qui vous permettent d'envoyer des journaux directement à CloudWatch Logs à l'aide de simples requêtes HTTP POST. Ces points de terminaison prennent en charge à la fois l'authentification SigV4 et l'authentification par jeton porteur.

## Important

Nous recommandons d'utiliser l'authentification SigV4 pour toutes les charges de travail de production où l'intégration du AWS SDK est possible. Le SigV4 utilise des informations d'identification à court terme et fournit la meilleure posture de sécurité. L'authentification par jeton porteur (clé API) est destinée aux scénarios dans lesquels SigV4 n'est pas réalisable, tels que les redirecteurs de journaux tiers qui ne prennent pas en charge AWS l'intégration du SDK. Pour plus d'informations, consultez la section [Alternatives aux clés d'accès longue durée](#) dans le guide de l'utilisateur IAM.

CloudWatch Logs prend en charge les points de terminaison d'ingestion HTTP suivants :

| Endpoint                             | Chemin                    | Content-Type                               | Format                                          |
|--------------------------------------|---------------------------|--------------------------------------------|-------------------------------------------------|
| <a href="#">OpenTelemetry Logs</a>   | /v1/logs                  | application/json ou application/x-protobuf | OTLP JSON ou Protobuf                           |
| <a href="#">HLC Logs</a>             | /services/collector/event | application/json                           | Format HLC                                      |
| <a href="#">ND-JSON Logs</a>         | /ingest/bulk              | application/json ou application/x-ndjson   | JSON délimité par un caractère de saut de ligne |
| <a href="#">Structured JSON Logs</a> | /ingest/json              | application/json                           | Objet ou tableau JSON                           |

# Comportement courant

Tous les points de terminaison d'ingestion HTTP partagent le comportement suivant :

## Authentification

Tous les points de terminaison prennent en charge à la fois l'authentification SigV4 et l'authentification par jeton porteur :

- SigV4 (recommandé) — AWS Signature standard version 4. Utilisez SigV4 chaque fois que votre application ou infrastructure prend en charge le AWS SDK ou peut signer des demandes. SigV4 utilise des informations d'identification à court terme et constitue la méthode d'authentification la plus sécurisée.
- Jeton porteur — Utilisez l'`Authorization: Bearer <ACWL token>` en-tête.
  - Le jeton doit être un jeton porteur ACWL valide. Pour les instructions de configuration, voir [the section called “Authentification par jeton porteur”](#).
  - Nécessite les autorisations `logs:PutLogEvents` et `logs:CallWithBearerToken` IAM.

## Groupe de journaux et flux de journaux

- Fourni via des en-têtes : `x-aws-log-group` et `x-aws-log-stream`
- `?logGroup=<name>&logStream=<name>` Les paramètres de requête sont également pris en charge sur tous les points de terminaison, à l'exception de l'OTLP.
- Vous ne pouvez pas utiliser à la fois les paramètres de requête et les en-têtes pour le même paramètre.
- Le groupe de journaux et le flux de journaux sont requis.

## Réponse

- Succès : HTTP 200 avec le corps `{}`
- Erreurs de validation : HTTP 400
- Défaillances d'authentification : HTTP 401

# Configuration de l'authentification par jeton porteur

Avant de pouvoir envoyer des journaux à l'aide de l'authentification par jeton porteur avec l'un des points de terminaison d'ingestion HTTP, vous devez :

- Création d'un utilisateur IAM avec des autorisations CloudWatch Logs
- Générer des informations d'identification spécifiques au service (jeton porteur)
- Création d'un groupe de journaux et d'un flux de journaux
- Activer l'authentification par jeton porteur sur le groupe de journaux

## Important

Nous recommandons d'utiliser l'authentification SigV4 avec des informations d'identification à court terme pour toutes les charges de travail lorsque cela est possible. Le SigV4 fournit la meilleure posture de sécurité. Limitez l'utilisation de clés d'API (jetons porteurs) aux scénarios dans lesquels l'authentification à court terme basée sur les informations d'identification n'est pas possible. Lorsque vous êtes prêt à intégrer les CloudWatch journaux dans des applications présentant des exigences de sécurité plus strictes, vous devez passer aux informations d'identification à court terme. Pour plus d'informations, consultez la section [Alternatives aux clés d'accès longue durée](#) dans le guide de l'utilisateur IAM.

## Option 1 : démarrage rapide à l'aide de la AWS console

La console AWS de gestion fournit un flux de travail rationalisé pour générer des clés d'API pour l'accès aux points de terminaison HTTP.

Pour configurer l'accès au point de terminaison HTTP à l'aide de la console

1. Connectez-vous à la console AWS de gestion.
2. Accédez à CloudWatch > Paramètres > Journaux.
3. Dans la section Clés d'API, choisissez Générer une clé d'API.
4. Pour Expiration de la clé d'API, effectuez l'une des actions suivantes :
  - Sélectionnez une durée d'expiration de la clé d'API de 1, 5, 30, 90 ou 365 jours.

- Choisissez Durée personnalisée pour spécifier une date d'expiration personnalisée pour la clé d'API.
  - Sélectionnez Ne jamais expirer (non recommandé).
5. Choisissez Générer une clé d'API.

La console :

- Crée un nouvel utilisateur IAM doté des autorisations appropriées
  - Joint la politique [CloudWatchLogsAPIKeyd'accès](#) géré (logs:PutLogEventsinclusions et logs:CallWithBearerToken autorisations)
  - Génère des informations d'identification spécifiques au service (clé API)
6. Copiez et enregistrez en toute sécurité les informations d'identification affichées :
- ID de clé d'API (ID d'identification spécifique au service)
  - Secret de la clé API (jeton porteur)

 Important

Enregistrez immédiatement le secret de la clé API. Vous ne pourrez pas la récupérer ultérieurement. Si vous la perdez, vous devrez générer une nouvelle clé d'API.

7. Créez le groupe de journaux et le flux de journaux dans lesquels vos journaux seront stockés :

```
# Create the log group
aws logs create-log-group \
    --log-group-name /aws/hlc-logs/my-application \
    --region us-east-1

# Create the log stream
aws logs create-log-stream \
    --log-group-name /aws/hlc-logs/my-application \
    --log-stream-name application-stream-001 \
    --region us-east-1
```

8. Activez l'authentification par jeton porteur sur le groupe de journaux :

```
aws logs put-bearer-token-authentication \
    --log-group-identifiant /aws/hlc-logs/my-application \
```

```
--bearer-token-authentication-enabled \  
--region us-east-1
```

Vérifiez la configuration :

```
aws logs describe-log-groups \  
--log-group-name-prefix /aws/hlc-logs/my-application \  
--region us-east-1
```

Autorisations incluses : L'utilisateur IAM créé automatiquement disposera des autorisations suivantes :

- `logs:PutLogEvents`— Envoie les événements du journal à CloudWatch Logs
- `logs:CallWithBearerToken`— Authentifiez-vous à l'aide d'un jeton porteur
- `kms:Describe*`, `kms:GenerateDataKey*`, `kms:Decrypt` — Accédez à des groupes de journaux chiffrés par KMS (avec une condition se limitant au service de journaux)

## Option 2 : Configuration manuelle

Si vous préférez mieux contrôler la configuration IAM ou si vous avez besoin de personnaliser les autorisations, vous pouvez configurer manuellement l'accès au point de terminaison HTTP.

### Étape 1 : créer un utilisateur IAM

Créez un utilisateur IAM qui sera utilisé pour l'ingestion des journaux :

1. Connectez-vous à la console AWS de gestion et accédez à IAM.
2. Dans le volet de navigation de gauche, choisissez Utilisateurs.
3. Choisissez Create user (Créer un utilisateur).
4. Entrez un nom d'utilisateur (par exemple, `cloudwatch-logs-hlc-user`).
5. Choisissez Suivant.
6. Joignez l'une des politiques IAM suivantes :

Option A : utiliser la politique gérée (recommandée)

Joignez la politique de gestion des [CloudWatchLogsAPIKey](#)accès.

## Option B : créer une politique personnalisée

Créez et attachez la politique IAM suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LogsAPIs",
      "Effect": "Allow",
      "Action": [
        "logs:CallWithBearerToken",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "KMSAPIs",
      "Effect": "Allow",
      "Action": [
        "kms:Describe*",
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Condition": {
        "StringEquals": {
          "kms:ViaService": [
            "logs.*.amazonaws.com"
          ]
        }
      },
      "Resource": "arn:aws:kms:*:*:key/*"
    }
  ]
}
```

7. Choisissez Suivant, puis Créer un utilisateur.

 Note

Les autorisations KMS sont requises si vous prévoyez d'envoyer des journaux à des groupes de journaux chiffrés par KMS. La condition limite l'accès KMS aux seules clés utilisées via le service CloudWatch Logs.

## Étape 2 : générer des informations d'identification spécifiques au service (clé API)

Générez la clé CloudWatch d'API Logs à l'aide de l'[CreateServiceSpecificCredential](#) API. Vous pouvez également utiliser la commande [create-service-specific-credential](#) CLI. Pour l'âge des informations d'identification, vous pouvez spécifier une valeur comprise entre 1 et 36 600 jours. Si vous ne spécifiez pas de durée de validité, la clé d'API n'expirera pas.

Pour générer une clé d'API dont l'expiration est de 30 jours, procédez comme suit :

```
aws iam create-service-specific-credential \  
  --user-name cloudwatch-logs-hlc-user \  
  --service-name logs.amazonaws.com \  
  --credential-age-days 30
```

La réponse est un [ServiceSpecificCredential](#) objet. La `ServiceCredentialSecret` valeur est votre clé CloudWatch d'API Logs (jeton porteur).

 Important

Stockez la valeur `ServiceCredentialSecret` en toute sécurité, car vous ne pourrez pas la récupérer ultérieurement. Si vous la perdez, vous devrez générer une nouvelle clé d'API.

## Étape 3 : créer un groupe de journaux et un flux de journaux

Créez le groupe de journaux et le flux de journaux dans lesquels vos journaux seront stockés :

```
# Create the log group  
aws logs create-log-group \  
  --log-group-name /aws/hlc-logs/my-application \  
  --region us-east-1
```

```
# Create the log stream
aws logs create-log-stream \
  --log-group-name /aws/hlc-logs/my-application \
  --log-stream-name application-stream-001 \
  --region us-east-1
```

## Étape 4 : activer l'authentification par jeton porteur

Activez l'authentification par jeton porteur sur le groupe de journaux :

```
aws logs put-bearer-token-authentication \
  --log-group-identifier /aws/hlc-logs/my-application \
  --bearer-token-authentication-enabled \
  --region us-east-1
```

Vérifiez la configuration :

```
aws logs describe-log-groups \
  --log-group-name-prefix /aws/hlc-logs/my-application \
  --region us-east-1
```

## Autorisations de contrôle pour la génération et l'utilisation CloudWatch des clés d'API Logs

La génération et l'utilisation des clés d'API CloudWatch Logs sont contrôlées par des actions et des clés de condition dans les services CloudWatch Logs et IAM.

### Contrôle de la génération des clés d'API CloudWatch Logs

L'action `CreateServiceSpecificCredential` [iam](#) : contrôle la génération d'une clé spécifique au service (telle qu'une clé d'API CloudWatch Logs). Vous pouvez étendre cette action aux utilisateurs IAM en tant que ressource afin de limiter le nombre d'utilisateurs pour lesquels une clé peut être générée.

Vous pouvez utiliser les clés de condition suivantes pour imposer des conditions à l'autorisation pour l'action `iam:CreateServiceSpecificCredential` :

- [iam : ServiceSpecificCredentialAgeDays](#) — Permet de spécifier, dans la condition, le délai d'expiration de la clé en jours. Par exemple, vous pouvez utiliser cette clé de condition pour autoriser uniquement des clés API qui expirent au terme d'un délai de 90 jours.



- [iam : ServiceSpecificCredentialServiceName](#) — Permet de spécifier, dans la condition, le nom d'un service. Par exemple, vous pouvez utiliser cette clé de condition pour autoriser uniquement la création de clés d'API pour les CloudWatch journaux et non pour d'autres services.

## Contrôle de l'utilisation des clés d'API CloudWatch Logs

L'`logs:CallWithBearerToken` action contrôle l'utilisation d'une clé d'API CloudWatch Logs. Pour empêcher une identité d'utiliser CloudWatch les clés de l'API Logs, associez une politique qui refuse l'`logs:CallWithBearerToken` à l'utilisateur IAM associé à la clé.

## Exemples de politiques

Empêcher une identité de générer et d'utiliser CloudWatch les clés d'API Logs

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCWLAPIKeys",
      "Effect": "Deny",
      "Action": [
        "iam:CreateServiceSpecificCredential",
        "logs:CallWithBearerToken"
      ],
      "Resource": "*"
    }
  ]
}
```

### Warning

Cette politique empêchera la création d'informations d'identification pour tous les AWS services qui prennent en charge la création d'informations d'identification spécifiques à un service. Pour plus d'informations, consultez [Informations d'identification spécifiques au service pour les utilisateurs IAM](#).

Empêcher une identité d'utiliser CloudWatch les clés de l'API Logs

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Deny",
    "Action": "logs:CallWithBearerToken",
    "Resource": "*"
  }
]
```

Autoriser la création de clés CloudWatch Logs uniquement si elles expirent dans les 90 jours

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceSpecificCredential",
      "Resource": "arn:aws:iam::123456789012:user/username",
      "Condition": {
        "StringEquals": {
          "iam:ServiceSpecificCredentialServiceName": "logs.amazonaws.com"
        },
        "NumericLessThanEquals": {
          "iam:ServiceSpecificCredentialAgeDays": "90"
        }
      }
    }
  ]
}
```

## Rotation des clés API

La rotation régulière de vos clés d'API réduit le risque d'accès non autorisé. Nous vous recommandons d'établir un calendrier de rotation conforme aux politiques de sécurité de votre organisation.

### Processus de rotation

Pour faire pivoter une clé d'API sans interrompre la livraison du journal, procédez comme suit :

1. Créez un nouvel identifiant (secondaire) pour l'utilisateur IAM :

```
aws iam create-service-specific-credential \  
  --user-name cloudwatch-logs-hlc-user \  
  --service-name logs.amazonaws.com \  
  --credential-age-days 90
```

2. (Facultatif) Stockez les nouvelles informations d'identification AWS Secrets Manager pour une récupération sécurisée et une rotation automatique.
3. Importez les nouvelles informations d'identification dans le portail de votre fournisseur ou mettez à jour la configuration de votre application pour utiliser la nouvelle clé d'API.
4. Définissez les informations d'identification d'origine sur inactive :

```
aws iam update-service-specific-credential \  
  --user-name cloudwatch-logs-hlc-user \  
  --service-specific-credential-id ACCA1234EXAMPLE1234 \  
  --status Inactive
```

5. Vérifiez que la livraison des journaux n'est pas affectée en surveillant la IncomingBytes métrique relative à votre groupe de connexions CloudWatch. Pour plus d'informations, consultez la section [Surveillance à l'aide de CloudWatch métriques](#).
6. Après avoir confirmé la livraison réussie avec la nouvelle clé, supprimez les informations d'identification précédentes :

```
aws iam delete-service-specific-credential \  
  --service-specific-credential-id ACCA1234EXAMPLE1234
```

## Surveillance de l'expiration des clés

Pour vérifier la date de création et le statut de vos clés d'API existantes, utilisez la [list-service-specific-credentials](#) commande :

```
aws iam list-service-specific-credentials \  
  --user-name cloudwatch-logs-hlc-user \  
  --service-name logs.amazonaws.com
```

La réponse inclut CreateDate et Status pour chaque identifiant. Utilisez ces informations pour identifier les clés qui arrivent à expiration ou qui sont actives depuis plus longtemps que ne le permet votre politique de rotation.

## Répondre à une clé d'API compromise

Si vous pensez qu'une clé d'API a été compromise, prenez immédiatement les mesures suivantes :

1. Désactivez immédiatement la clé pour empêcher toute nouvelle utilisation non autorisée :

```
aws iam update-service-specific-credential \  
  --user-name cloudwatch-logs-hlc-user \  
  --service-specific-credential-id ACCA1234EXAMPLE1234 \  
  --status Inactive
```

2. Passez en revue les CloudTrail journaux pour déterminer l'étendue des accès non autorisés. Découvrez [the section called “Enregistrement de l'utilisation des clés d'API avec CloudTrail”](#) comment activer l'audit de l'utilisation des clés d'API.
3. Créez une clé de remplacement en suivant le processus de rotation décrit dans [the section called “Processus de rotation”](#).
4. Supprimez la clé compromise une fois que la clé de remplacement est en place :

```
aws iam delete-service-specific-credential \  
  --service-specific-credential-id ACCA1234EXAMPLE1234
```

5. Ajoutez une politique de refus si vous devez immédiatement bloquer l'accès à tout jeton porteur pour l'utilisateur IAM pendant que vous étudiez :

```
{  
  "Version": "2012-10-17",  
  "Statement": {  
    "Effect": "Deny",  
    "Action": "logs:CallWithBearerToken",  
    "Resource": "*"   
  }  
}
```

### Note

Pour effectuer ces actions via l'API, vous devez vous authentifier à l'aide AWS d'informations d'identification et non à l'aide d'une clé d'API CloudWatch Logs.

Vous pouvez également utiliser les opérations d'API IAM suivantes pour gérer les clés compromises :

- [ResetServiceSpecificCredential](#)— Réinitialisez la clé pour générer un nouveau mot de passe sans supprimer les informations d'identification. La clé ne doit pas avoir expiré.

## Bonnes pratiques de sécurité pour les clés d'API

Suivez ces bonnes pratiques pour protéger vos clés CloudWatch d'API Logs :

- N'intégrez jamais de clés d'API dans le code source. Ne codez pas en dur les clés d'API dans le code de l'application et ne les validez pas dans les systèmes de contrôle de version. Si une clé est accidentellement validée dans un dépôt public, le scan AWS automatique peut la signaler et vous devez immédiatement faire pivoter la clé.
- Utilisez un gestionnaire de secrets. Stockez les clés d'API dans une solution de gestion des secrets [AWS Secrets Manager](#) ou dans une solution équivalente. Cela permet un contrôle d'accès centralisé, un enregistrement des audits et une rotation automatique.
- Définissez une date d'expiration pour toutes les clés. Spécifiez toujours une `credential-age-days` valeur lors de la création de clés d'API. Pour appliquer une durée de vie maximale des clés au sein de votre organisation, utilisez la clé de condition `iam:ServiceSpecificCredentialAgeDays` IAM. Pour obtenir un exemple, consultez [the section called "Autoriser la création de clés CloudWatch Logs uniquement si elles expirent dans les 90 jours"](#).
- Appliquez les autorisations du moindre privilège. Étendez les autorisations de l'utilisateur IAM uniquement aux groupes de journaux et aux actions requises. Utilisez la politique [CloudWatchLogsAPIKeyd'accès](#) géré comme point de départ et limitez davantage selon les besoins.
- Activez la CloudTrail journalisation. Auditez l'utilisation des clés d'API en activant CloudTrail les événements de données pour `AWS::Logs::LogGroupAuthorization`. Consultez [the section called "Enregistrement de l'utilisation des clés d'API avec CloudTrail"](#).
- Surveillez avec IAM Access Analyzer. Utilisez [IAM Access Analyzer](#) pour identifier les informations d'identification non utilisées et les politiques trop permissives associées aux utilisateurs IAM de votre clé d'API.
- Faites pivoter les touches régulièrement. Établissez un calendrier de rotation et suivez le processus décrit dans [the section called "Rotation des clés API"](#).

## Enregistrement de l'utilisation des clés d'API avec CloudTrail

Vous pouvez l'utiliser AWS CloudTrail pour enregistrer les événements de données liés à l'utilisation des clés de l'API CloudWatch Logs. CloudWatch Logs émet `AWS::Logs::LogGroupAuthorization` des événements de données pour les `CallWithBearerToken` appels, ce qui vous permet de vérifier quand et comment les clés d'API sont utilisées pour envoyer des journaux.

Pour activer la CloudTrail journalisation pour l'utilisation des clés de l'API CloudWatch Logs :

### Note

Le compartiment S3 que vous spécifiez pour le suivi doit disposer d'une politique de compartiment qui CloudTrail permet d'y écrire des fichiers journaux. Pour plus d'informations, consultez la [politique relative aux compartiments Amazon S3 pour CloudTrail](#).

### 1. Créez un parcours :

```
aws cloudtrail create-trail \  
  --name cloudwatch-logs-api-key-audit \  
  --s3-bucket-name my-cloudtrail-bucket \  
  --region us-east-1
```

### 2. Configurez des sélecteurs d'événements avancés pour capturer CloudWatch les événements d'autorisation des groupes de journaux journaux :

```
aws cloudtrail put-event-selectors \  
  --region us-east-1 \  
  --trail-name cloudwatch-logs-api-key-audit \  
  --advanced-event-selectors '[{  
    "Name": "CloudWatch Logs API key authorization events",  
    "FieldSelectors": [  
      { "Field": "eventCategory", "Equals": ["Data"] },  
      { "Field": "resources.type", "Equals":  
        ["AWS::Logs::LogGroupAuthorization"] }  
    ]  
  }]'
```

### 3. Commencez à enregistrer les sentiers :

```
aws cloudtrail start-logging \  
  --name cloudwatch-logs-api-key-audit \  
  --region us-east-1
```

## Envoi de journaux à l'aide du point de terminaison OTLP (OpenTelemetry Logs)

Le point de terminaison OpenTelemetry Logs (/v1/logs) accepte les données du journal OpenTelemetry du protocole (OTLP) au format JSON ou Protobuf. Pour obtenir des informations détaillées sur le point de terminaison OTLP, notamment sa configuration et son utilisation, voir [Envoyer des métriques et des traces à CloudWatch with OpenTelemetry](#).

Si vous utilisez l'authentification par jeton porteur, suivez les étapes de configuration décrites [the section called "Authentification par jeton porteur"](#) avant de continuer.

### Format des demandes

- Méthode : POST
- Type de contenu : ou application/json application/x-protobuf
- Groupe de journaux : x-aws-log-group en-tête uniquement (paramètre de requête non pris en charge)
- Flux du journal : x-aws-log-stream en-tête

### Exemple de demande

```
curl -X POST "https://logs.<region>.amazonaws.com/v1/logs" \  
  -H "Authorization: Bearer ACWL<token>" \  
  -H "Content-Type: application/json" \  
  -H "x-aws-log-group: MyLogGroup" \  
  -H "x-aws-log-stream: MyLogStream" \  
  -d '{  
    "resourceLogs": [  
      {  
        "resource": {  
          "attributes": [  
            {
```

```

        "key": "service.name",
        "value": { "stringValue": "my-service" }
      }
    ],
  },
  "scopeLogs": [
    {
      "scope": {
        "name": "my-library",
        "version": "1.0.0"
      },
      "logRecords": [
        {
          "timeUnixNano": "1741900000000000000",
          "severityNumber": 9,
          "severityText": "INFO",
          "body": {
            "stringValue": "User logged in successfully"
          },
          "attributes": [
            {
              "key": "user.id",
              "value": { "stringValue": "12345" }
            }
          ]
        }
      ]
    }
  ]
}
]
}'

```

## Réponses

Succès (tous les événements sont acceptés) :

```

HTTP 200 OK
{}

```

Succès partiel (certains événements rejetés) :

```

{

```



```
"partialSuccess": {
  "rejectedLogRecords": 5,
  "errorMessage": "{\\"tooOldLogEventCount\\": 3, \\"tooNewLogEventCount\\": 1,
\\"expiredLogEventCount\\": 1}"
}
```

Lorsque le type de contenu de la demande est défini `application/x-protobuf`, la réponse est renvoyée sous forme de message `ExportLogsServiceResponse` protobuf sérialisé avec les mêmes champs.

## Comportements spécifiques à l'OTLP

Les comportements suivants sont spécifiques au point de terminaison OTLP et ne sont pas présents sur les autres points de terminaison d'ingestion HTTP :

- En-tête `Retry-After` : inclus dans les réponses 503 et 429 pour indiquer à quel moment le client doit réessayer.

## Envoi de journaux à l'aide du point de terminaison HLC (HLC Logs)

Le point de terminaison HLC Logs (`/services/collector/event`) est basé sur le format HTTP Log Collector (HLC).

Si vous utilisez l'authentification par jeton porteur, suivez les étapes de configuration décrites [the section called “Authentification par jeton porteur”](#) avant de continuer.

## Modes d'entrée

Chaque événement est un objet JSON avec un `"event"` champ obligatoire. Champs de métadonnées facultatifs : `"time"` `"host"` `"source"` `"sourcetype"` `"index"`.

Événement unique :

```
{"event":"Hello world!","time":1486683865.0}
```

Tableau d'événements JSON :

```
[
```

```
{ "event": "msg1", "time": 1486683865.0 },  
{ "event": "msg2", "time": 1486683866.0 }  
]
```

Événements concaténés/groupés (aucun encapsuleur de tableau) :

```
{ "event": "msg1", "time": 1486683865.0 } { "event": "msg2", "time": 1486683866.0 }
```

## Champ de l'événement (obligatoire)

Ce "event" champ est obligatoire. Sa valeur peut être de n'importe quel type JSON :

```
{ "event": "a string message" }  
{ "event": { "message": "structured data", "severity": "INFO" } }  
{ "event": 42 }  
{ "event": true }
```

Les objets sans "event" champ sont ignorés en silence :

```
{ "message": "this is skipped – no event field" }
```

## Champ horaire (facultatif)

Le "time" champ est exprimé en secondes d'époque (et non en millisecondes), avec une décimale facultative pour une précision inférieure à la seconde.

| Format             | Exemple                 | Interprété comme          |
|--------------------|-------------------------|---------------------------|
| Float              | "time":1486683865.500   | 1486683865500 ms          |
| Entier             | "time":1486683865       | 1486683865000 ms          |
| Chaîne (flottante) | "time":"1486683865.500" | 1486683865500 ms          |
| Chaîne (entier)    | "time":"1486683865"     | 1486683865000 ms          |
| Manquant           | (aucun champ horaire)   | Heure actuelle du serveur |
| Non valide         | "time":"invalid"        | Heure actuelle du serveur |

## Content-Type

Seul `application/json` est accepté.

## Types de valeurs JSON acceptés

| Type de haut niveau                     | Comportement                               |
|-----------------------------------------|--------------------------------------------|
| Objet avec "event"                      | Acceptée                                   |
| Objet sans "event"                      | Ignoré                                     |
| Tableau d'objets                        | Chaque élément est traité individuellement |
| Objets concaténés                       | Chaque objet est traité individuellement   |
| Primitif (chaîne, nombre, booléen, nul) | Ignoré                                     |

## Format du point de terminaison

L'URL du point de terminaison HLC suit le format suivant :

```
https://logs.<region>.amazonaws.com/services/collector/event?  
logGroup=<name>&logStream=<name>[&entityName=<name>&entityEnvironment=<environment>]
```

Paramètres requis :

- `<region>`— AWS Région (par exemples `-east-1`, `eu-west-1`)
- `logGroup`— Nom du groupe de logs codé en URL
- `logStream`— Nom du flux de journal codé en URL

Paramètres facultatifs :

Vous pouvez éventuellement associer les événements de votre journal à une Service entité en incluant les paramètres de requête suivants. Les journaux envoyés via le point de terminaison HLC étant de la télémétrie personnalisée, ils ne sont pas automatiquement associés à une entité. En

fournissant ces paramètres, CloudWatch Logs crée une entité `KeyAttributes.Type` définie sur `Service` et l'associe à vos événements de journal. Cela permet à la fonctionnalité associée CloudWatch à Explore de corrélérer ces journaux avec d'autres données de télémétrie (métriques, traces et journaux) du même service, ce qui facilite le dépannage et la surveillance de vos applications sur différents types de signaux. Pour plus d'informations sur les entités et la télémétrie associée, voir [Ajout d'informations connexes à la télémétrie personnalisée](#).

- `entityName`— Nom de l'entité de service à associer aux événements du journal. Cette valeur est stockée en tant qu'entité `KeyAttributes.Name` (par exemple, `my-application-ouapi.my-service.com`).
- `entityEnvironment`— L'environnement dans lequel le service est hébergé ou à quoi il appartient. Cette valeur est stockée en tant qu'entité `KeyAttributes.Environment` (par exemple `production,ec2:default,oueks:my-cluster/default`).

## Format des demandes

Envoyez les journaux à l'aide du protocole HTTP POST avec les en-têtes et le corps suivants :

En-têtes :

- `Authorization: Bearer <your-bearer-token>`
- `Content-Type: application/json`

Format du boîtier :

Le corps de la requête doit être au format JSON avec un tableau d'événements :

```
{
  "event": [
    {
      "time": 1730141374.001,
      "event": "Application started successfully",
      "host": "web-server-1",
      "source": "application.log",
      "severity": "info"
    },
    {
      "time": 1730141374.457,
      "event": "User login successful",
```

```
        "host": "web-server-1",  
        "source": "auth.log",  
        "user": "john.doe"  
    }  
]  
}
```

### Descriptions des champs :

- **time**— Horodatage de l'époque Unix en secondes, avec décimal optionnel pour une précision inférieure à la seconde (facultatif)
- **event**— Le message du journal ou les données de l'événement (obligatoire)
- **host**— Nom d'hôte ou identifiant source (facultatif)
- **source**— Identifiant de la source du journal (facultatif)

Des champs personnalisés supplémentaires peuvent être inclus selon les besoins.

## Exemple de demande

```
curl -X POST "https://logs.<region>.amazonaws.com/services/collector/event?  
logGroup=MyLogGroup&logStream=MyStream" \  
-H "Authorization: Bearer ACWL<token>" \  
-H "Content-Type: application/json" \  
-d '{"event":{"message":"User logged  
in","user_id":"u-123"},"time":1486683865.0,"host":"web-01","source":"auth-service"}'
```

## Bonnes pratiques

### Événements de mise en lots

Pour de meilleures performances et une meilleure efficacité :

- Batch de plusieurs événements en une seule demande lorsque cela est possible
- Taille de lot recommandée : 10 à 100 événements par demande
- Taille maximale de la demande : 1 Mo

## Gestion des erreurs

Implémentez une gestion appropriée des erreurs dans votre application. Codes d'état HTTP courants :

- 200 OK— Logs correctement ingérés
- 400 Bad Request— Format ou paramètres de demande non valides
- 401 Unauthorized— Jeton porteur non valide ou expiré
- 403 Forbidden— Autorisations insuffisantes
- 404 Not Found— Le groupe de journaux ou le flux n'existe pas
- 429 Too Many Requests— Limite de débit dépassée
- 500 Internal Server Error— Erreur de service (nouvelle tentative avec retard exponentiel)

## Limitations

- Taille maximale de l'événement : 256 Ko par événement
- Taille maximale de la demande : 1 Mo
- Nombre maximum d'événements par demande : 10 000
- Les noms des groupes de journaux doivent respecter les conventions de dénomination des CloudWatch journaux
- L'authentification par jeton porteur doit être activée sur le groupe de journaux si l'authentification par jeton porteur est utilisée.

## Envoi de journaux à l'aide du point de terminaison NDJSON (journaux ND-JSON)

Le point de terminaison ND-JSON Logs (/ingest/bulk) accepte les journaux au format [NDJSON \(Newline Delimited JSON\)](#). Chaque ligne contient exactement une valeur JSON, séparée par des caractères de nouvelle ligne.

Si vous utilisez l'authentification par jeton porteur, suivez les étapes de configuration décrites [the section called “Authentification par jeton porteur”](#) avant de continuer.

## Format des demandes

Envoyez une valeur JSON par ligne, séparée par `\n` (LF) ou `\r\n` (CRLF). Les lignes vides sont ignorées en silence.

```
{"timestamp":1771007942000,"message":"event one","level":"INFO"}
{"timestamp":1771007943000,"message":"event two","level":"ERROR"}
{"timestamp":1771007944000,"message":"event three","level":"DEBUG"}
```

Les deux `application/json` et `application/x-ndjson` sont acceptés en tant que type de contenu.

## Types de valeurs JSON acceptés

Selon la spécification NDJSON (RFC 8259), toute valeur JSON valide est acceptée sur chaque ligne.

Objets JSON (les plus courants) :

```
{"timestamp":1771007942000,"message":"User logged in","service":"auth"}
{"timestamp":1771007943000,"error":"Connection timeout","service":"api"}
```

Tableaux JSON (aplatis en événements individuels) :

```
[{"timestamp":1000,"message":"a"}, {"timestamp":2000,"message":"b"}]
```

Cette seule ligne produit 2 événements. Chaque élément du tableau devient un événement de journal distinct.

Valeurs primitives :

```
"a plain string log message"
42
true
null
```

Chaque primitive devient son propre événement avec l'horodatage actuel du serveur.

Types mixtes :

```
{"timestamp":1771007942000,"message":"structured event"}
"unstructured string message"
42
{"timestamp":1771007943000,"error":"something failed"}
```

Les 4 lignes sont acceptées comme des événements valides.

| Contenu de la ligne | Comportement                                          |
|---------------------|-------------------------------------------------------|
| Objet JSON          | Accepté, horodatage extrait s'il est présent          |
| un tableau JSON     | Aplati : chaque élément devient un événement distinct |
| Tableau vide []     | Accepté, produit 0 événements                         |
| chaîne JSON         | Accepté comme message d'événement                     |
| Numéro JSON         | Accepté comme message d'événement                     |
| booléen JSON        | Accepté comme message d'événement                     |
| JSON nul            | Accepté comme message d'événement                     |
| JSON non valide     | Ignoré (compté, le traitement continue)               |
| Ligne vide          | Ignoré (non compté comme ignoré)                      |

## Champ d'horodatage

Le "timestamp" champ est exprimé en millisecondes d'époque (et non en secondes).



| Format             | Exemple                    | Interprété comme          |
|--------------------|----------------------------|---------------------------|
| Numérique (millis) | "timestamp":1771007942000  | 1771007942000 ms          |
| Manquant           | (aucun champ d'horodatage) | Heure actuelle du serveur |
| Non numérique      | "timestamp":"invalid"      | Heure actuelle du serveur |
| Ligne sans objet   | "hello", 42, true          | Heure actuelle du serveur |

## Lignes non valides

Les lignes qui ne sont pas des JSON valides sont ignorées et comptées en silence. Le traitement se poursuit avec la ligne suivante.

```
{"message":"valid event"}
this is not valid json
{"message":"another valid event"}
```

Résultat : 2 événements ingérés, 1 ignoré. Renvoie HTTP 200.

Si toutes les lignes ne sont pas valides, renvoie HTTP 400 avec "All events were invalid".

## Exemple de demande

```
curl -X POST "https://logs.<region>.amazonaws.com/ingest/bulk?
logGroup=MyLogGroup&logStream=MyStream" \
-H "Authorization: Bearer ACWL<token>" \
-H "Content-Type: application/x-ndjson" \
-d '{"timestamp":1771007942000,"message":"User logged in","level":"INFO"}
{"timestamp":1771007943000,"message":"Query took 42ms","level":"DEBUG"}
{"timestamp":1771007944000,"error":"Connection refused","level":"ERROR"}'
```

## Réponses

Succès (tous les événements sont acceptés) :

```
HTTP 200 OK
{}
```

Succès partiel (certains événements rejetés) :

```
{
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\"tooOldLogEventCount\": 3, \"tooNewLogEventCount\": 1, \"expiredLogEventCount\": 1}"
  }
}
```

Le `rejectedLogRecords` champ indique le nombre total d'événements rejetés. Le `errorMessage` champ contient une ventilation codée en JSON par motif de rejet :

- `tooOldLogEventCount`— Événements dont l'horodatage est antérieur à la période de conservation
- `tooNewLogEventCount`— Événements dont l'horodatage est trop lointain dans le futur
- `expiredLogEventCount`— Événements qui ont expiré pendant le traitement

## Bonnes pratiques

### Événements de mise en lots

Pour de meilleures performances et une meilleure efficacité :

- Batch de plusieurs événements en une seule demande lorsque cela est possible
- Taille de lot recommandée : 10 à 100 événements par demande
- Taille maximale de la demande : 1 Mo

### Gestion des erreurs

Mettez en œuvre une gestion appropriée des erreurs dans votre application. Codes d'état HTTP courants :

- **200 OK**— Logs correctement ingérés
- **400 Bad Request**— Format ou paramètres de demande non valides
- **401 Unauthorized**— Jeton porteur non valide ou expiré
- **403 Forbidden**— Autorisations insuffisantes

- 404 Not Found— Le groupe de journaux ou le flux n'existe pas
- 429 Too Many Requests— Limite de débit dépassée
- 500 Internal Server Error— Erreur de service (nouvelle tentative avec retard exponentiel)

## Limitations

- Taille maximale de l'événement : 256 Ko par événement
- Taille maximale de la demande : 1 Mo
- Nombre maximum d'événements par demande : 10 000
- Les noms des groupes de journaux doivent respecter les conventions de dénomination des CloudWatch journaux
- L'authentification par jeton porteur doit être activée sur le groupe de journaux si l'authentification par jeton porteur est utilisée.

## Envoi de journaux à l'aide du point de terminaison JSON structuré (journaux JSON structurés)

Le point de terminaison Structured JSON Logs (/ingest/json) accepte le JSON standard, qu'il s'agisse d'un seul objet JSON ou d'un tableau d'objets JSON. Ce point de terminaison est conçu pour les données de journal structurées où chaque événement est un objet JSON.

Si vous utilisez l'authentification par jeton porteur, suivez les étapes de configuration décrites [the section called “Authentification par jeton porteur”](#) avant de continuer.

## Format des demandes

Seul application/json est accepté comme type de contenu.

Objet JSON unique :

```
{"timestamp":1771007942000,"message":"single event","level":"INFO"}
```

Tableau d'objets JSON :

```
[  
  {"timestamp":1771007942000,"message":"event one","level":"INFO"},  
  ...  
]
```

```
  {"timestamp":1771007943000,"message":"event two","level":"ERROR"}
]
```

## Types de valeurs JSON acceptés

Ce point de terminaison est strict : seuls les objets JSON sont acceptés en tant qu'événements.

| Input                                            | Comportement                               |
|--------------------------------------------------|--------------------------------------------|
| Objet JSON unique                                | Accepté comme un seul événement            |
| Tableau d'objets JSON                            | Chaque objet devient un événement distinct |
| Tableau vide []                                  | Accepté, produit 0 événements              |
| Non-objet dans le tableau (chaîne, nombre, etc.) | Ignoré                                     |
| Primitive de haut niveau ("hello",42)            | Ignoré                                     |
| Objets concaténés {...}{...}                     | Seul le premier objet est analysé          |

Exemple — tableau de types mixtes :

```
[
  {"timestamp":1771007942000,"message":"valid object"},
  "just a string",
  42,
  {"timestamp":1771007943000,"message":"another valid object"}
]
```

Résultat : 2 événements ingérés (les objets), 2 ignorés (la chaîne et le chiffre).

## Champ d'horodatage

Le "timestamp" champ est exprimé en millisecondes d'époque, comme le point de terminaison NDJSON.

| Format             | Exemple                    | Interprété comme          |
|--------------------|----------------------------|---------------------------|
| Numérique (millis) | "timestamp":1771007942000  | 1771007942000 ms          |
| Manquant           | (aucun champ d'horodatage) | Heure actuelle du serveur |
| Non numérique      | "timestamp":"invalid"      | Heure actuelle du serveur |

## Exemple de demande

```
curl -X POST "https://logs.<region>.amazonaws.com/ingest/json?logGroup=MyLogGroup&logStream=MyStream" \
-H "Authorization: Bearer ACWL<token>" \
-H "Content-Type: application/json" \
-d ' [{"timestamp":1771007942000,"message":"User logged in","user_id":"u-123"},
{"timestamp":1771007943000,"message":"Order placed","order_id":"o-456"} ]'
```

## Réponses

Succès (tous les événements sont acceptés) :

```
HTTP 200 OK
{}
```

Succès partiel (certains événements rejetés) :

```
{
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\\"tooOldLogEventCount\\": 3, \\"tooNewLogEventCount\\": 1,
\\"expiredLogEventCount\\": 1}"
  }
}
```

Le `rejectedLogRecords` champ indique le nombre total d'événements rejetés. Le `errorMessage` champ contient une ventilation codée en JSON par motif de rejet :

- `tooOldLogEventCount`— Événements dont l'horodatage est antérieur à la période de conservation

- `tooNewLogEventCount`— Événements dont l'horodatage est trop lointain dans le futur
- `expiredLogEventCount`— Événements qui ont expiré pendant le traitement

## Bonnes pratiques

### Événements de mise en lots

Pour de meilleures performances et une meilleure efficacité :

- Batch de plusieurs événements en une seule demande lorsque cela est possible
- Taille de lot recommandée : 10 à 100 événements par demande
- Taille maximale de la demande : 1 Mo

### Gestion des erreurs

Implémentez une gestion appropriée des erreurs dans votre application. Codes d'état HTTP courants :

- `200 OK`— Logs correctement ingérés
- `400 Bad Request`— Format ou paramètres de demande non valides
- `401 Unauthorized`— Jeton porteur non valide ou expiré
- `403 Forbidden`— Autorisations insuffisantes
- `404 Not Found`— Le groupe de journaux ou le flux n'existe pas
- `429 Too Many Requests`— Limite de débit dépassée
- `500 Internal Server Error`— Erreur de service (nouvelle tentative avec retard exponentiel)

## Limitations

- Taille maximale de l'événement : 256 Ko par événement
- Taille maximale de la demande : 1 Mo
- Nombre maximum d'événements par demande : 10 000
- Les noms des groupes de journaux doivent respecter les conventions de dénomination des CloudWatch journaux

- L'authentification par jeton porteur doit être activée sur le groupe de journaux si l'authentification par jeton porteur est utilisée.

## Comparaison des points de terminaison d'ingestion HTTP

| Fonctionnalité                    | Journaux HLC              | Journaux ND-JSON                         | Journaux JSON structurés    | OpenTelemetry Journaux                     |
|-----------------------------------|---------------------------|------------------------------------------|-----------------------------|--------------------------------------------|
| Chemin                            | /services/collector/event | /ingest/bulk                             | /ingest/json                | /v1/logs                                   |
| Content-Type                      | application/json          | application/json ou application/x-ndjson | application/json            | application/json ou application/x-protobuf |
| Champ d'horodatage                | "time"(secondes)          | "timestamp" (millisecondes)              | "timestamp" (millisecondes) | "timeUnixNano" (nanosecondes)              |
| Champs obligatoires               | "event"                   | Aucune                                   | Aucune                      | Structure OTLP ()<br>"resourceLogs"        |
| Réponse de réussite partielle     | Non                       | Oui                                      | Oui                         | Oui                                        |
| Support des paramètres de requête | Oui                       | Oui                                      | Oui                         | Non (en-têtes uniquement)                  |
| Métadonnées de l'entité           | Oui                       | Oui                                      | Oui                         | Non                                        |
| Accepte les primitives            | Non                       | Oui                                      | Non                         | Non                                        |

| Fonctionnalité                          | Journaux HLC | Journaux ND-JSON | Journaux JSON structurés | OpenTelemetry Journaux |
|-----------------------------------------|--------------|------------------|--------------------------|------------------------|
| Analyse syntaxique basée sur les lignes | Non          | Oui              | Non                      | Non                    |
| Support Protobuf                        | Non          | Non              | Non                      | Oui                    |
| En-tête Retry-After                     | Non          | Non              | Non                      | Oui                    |

## Choix d'un point de terminaison

- Vous utilisez le format HLC ? Utilisez les journaux HLC. Vos charges utiles HLC existantes fonctionnent avec un minimum de modifications.
- Des line-by-line journaux de streaming ? Utilisez les journaux ND-JSON. Idéal pour les pipelines de logs qui émettent un événement par ligne. Très flexible : accepte n'importe quel type de valeur JSON.
- Vous envoyez des charges utiles JSON structurées ? Utilisez des journaux JSON structurés. Idéal pour les applications qui produisent des objets ou des tableaux JSON bien formés.
- Vous l'utilisez déjà OpenTelemetry ? Utilisez OpenTelemetry les journaux. Accepte le format OTLP JSON ou Protobuf et prend en charge les réponses de réussite partielles avec une sémantique de nouvelle tentative.



# 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 kit 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                                             |
|---------------------------------------------|--------------------------------------------------------------|
| <a href="#">AWS SDK pour C++</a>            | <a href="#">AWS SDK pour C++ exemples de code</a>            |
| <a href="#">AWS CLI</a>                     | <a href="#">AWS CLI exemples de code</a>                     |
| <a href="#">AWS SDK pour Go</a>             | <a href="#">AWS SDK pour Go exemples de code</a>             |
| <a href="#">AWS SDK pour Java</a>           | <a href="#">AWS SDK pour Java exemples de code</a>           |
| <a href="#">AWS SDK pour JavaScript</a>     | <a href="#">AWS SDK pour JavaScript exemples de code</a>     |
| <a href="#">AWS SDK pour Kotlin</a>         | <a href="#">AWS SDK pour Kotlin exemples de code</a>         |
| <a href="#">AWS SDK pour .NET</a>           | <a href="#">AWS SDK pour .NET exemples de code</a>           |
| <a href="#">AWS SDK pour PHP</a>            | <a href="#">AWS SDK pour PHP exemples de code</a>            |
| <a href="#">Outils AWS pour PowerShell</a>  | <a href="#">Outils AWS pour PowerShell exemples de code</a>  |
| <a href="#">AWS SDK pour Python (Boto3)</a> | <a href="#">AWS SDK pour Python (Boto3) exemples de code</a> |
| <a href="#">AWS SDK pour Ruby</a>           | <a href="#">AWS SDK pour Ruby exemples de code</a>           |
| <a href="#">AWS SDK pour Rust</a>           | <a href="#">AWS SDK pour Rust exemples de code</a>           |
| <a href="#">AWS SDK pour SAP ABAP</a>       | <a href="#">AWS SDK pour SAP ABAP exemples de code</a>       |
| <a href="#">AWS SDK pour Swift</a>          | <a href="#">AWS SDK pour Swift exemples de code</a>          |

Pour des exemples spécifiques aux CloudWatch journaux, voir [Exemples 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 [Provide feedback](#) (Fournir un commentaire) en bas de cette page.

# Gestion des journaux

CloudWatch Logs fournit des fonctionnalités avancées de gestion des journaux qui vous aident à organiser, transformer et analyser les données de vos journaux de manière plus efficace. Ces fonctionnalités incluent la [centralisation des données entre comptes et entre régions](#), la [découverte automatique des données et la gestion des schémas](#), la [transformation des journaux pendant les ingestions](#) et des [analyses améliorées avec des facettes](#) pour une exploration interactive des journaux.

## Rubriques

- [Découverte et gestion des sources de données](#)
- [Fonctionnalités activées par les sources de données](#)
- [Pris en charge Services AWS pour les sources de données](#)
- [Sources tierces prises en charge pour les sources de données](#)

## Découverte et gestion des sources de données

CloudWatch Logs découvre et classe automatiquement les données de vos journaux par source de données et par type, ce qui facilite la compréhension et la gestion de vos journaux à grande échelle. Cette fonctionnalité permet de découvrir des schémas pour des AWS sources vendues telles qu'Amazon VPC Flow Logs et Route 53 CloudTrail, ainsi que pour des outils de sécurité tiers.

La console de gestion des journaux fournit une vue de haut niveau de vos journaux organisés par source de données et par type, plutôt que simplement par groupes de journaux. Cette organisation vous aide à :

- Afficher les journaux classés par AWS services, sources tierces (telles que Okta ou CrowdStrike) et sources personnalisées
- Comprenez automatiquement le schéma et la structure de vos données de journal
- Créez des politiques d'index de champs basées sur les champs de schéma découverts
- Gérez les journaux plus efficacement dans les différentes sources de données
- Journaux de requêtes par différentes sources de données

Lorsque vous [activez la journalisation CloudWatch des journaux pour les AWS services pris en charge](#), CloudWatch Logs applique automatiquement le schéma approprié à vos journaux. Cette

application de schéma automatique permet de maintenir la cohérence et fournit des informations immédiates sur la structure de vos journaux.

## Qu'est-ce que les sources de données CloudWatch Logs ?

CloudWatch Les sources de données des journaux sont une fonctionnalité qui fournit une nouvelle façon d'organiser et de classer les données de vos journaux en fonction de la source qui les génère. Alors que CloudWatch Logs utilise traditionnellement des groupes de journaux pour organiser les journaux, les sources de données offrent une couche d'organisation supplémentaire qui regroupe les journaux en fonction de leur service d'origine et de leur type de journal.

### Comment fonctionnent les sources de données

Les sources de données fournissent une organisation des journaux basée sur les services et une découverte simplifiée au sein de votre AWS infrastructure. Vous pouvez facilement localiser les journaux provenant de services spécifiques et les filtrer par type de journal sans avoir à connaître les noms ou les structures des groupes de journaux individuels.

Pour les sources tierces et éventuellement pour les sources de journaux d'applications, les sources de données fonctionnent avec des CloudWatch pipelines pour classer vos journaux. Lorsque vous configurez un pipeline pour ingérer et transformer des journaux, vous spécifiez le nom et le type de la source de données. CloudWatch Logs classe ensuite automatiquement tous les journaux traités par le pipeline. Pour plus d'informations, consultez la section sur les [CloudWatch pipelines](#) dans le guide de CloudWatch l'utilisateur Amazon.

Les sources de données classent les journaux à l'aide de deux identifiants clés :

- Nom de la source de données : le AWS service, la source tierce ou l'application qui génère les journaux (par exemple, Route 53, Amazon VPC CloudTrail, Okta SSO ou Falçon). CrowdStrike
- Type de source de données : type spécifique de journal généré par ce service.

Un schéma définit la structure des données du journal, notamment les champs présents et la manière dont les informations sont organisées. Une source de données unique peut produire plusieurs types de journaux avec des schémas et des objectifs différents. Par exemple, la source de AWS CloudTrail données est de deux types : les événements de gestion (qui suivent les opérations du plan de contrôle telles que la création ou la suppression de ressources) et les événements de données (qui suivent les opérations du plan de données telles que l'accès aux objets S3). Chaque type possède un schéma différent car ils capturent différents types d'informations.

## Prise en main

CloudWatch Logs classe vos journaux dans des sources de données en fonction de leur origine. La méthode dépend du type de journaux que vous utilisez :

### Service AWS journaux

Les journaux [pris en charge Services AWS](#) sont automatiquement regroupés par source de données sans aucune configuration requise. CloudWatch Logs reconnaît ces journaux et applique le nom et le type de source de données appropriés en fonction du service d'origine.

### Logs tiers

Les journaux tiers nécessitent des pipelines pour la catégorisation des sources de données. Lorsque vous configurez un pipeline pour ingérer des journaux provenant de sources tierces prises en charge telles que Microsoft Office 365, Okta ou Palo Alto Networks CrowdStrike, vous spécifiez le [nom de la source de données et saisissez](#) la configuration du pipeline. CloudWatch Logs classe automatiquement tous les journaux traités par le pipeline à l'aide de ces identifiants.

Les pipelines peuvent éventuellement transformer des journaux tiers au format Open Cybersecurity Schema Framework (OCSF) pour une analyse standardisée des événements de sécurité. Lorsque la transformation OCSF est activée, le nom et le type de la source de données sont automatiquement déterminés en fonction du mappage du schéma OCSF. Sans transformation OCSF, vous spécifiez le nom et le type de la source de données dans la configuration du pipeline.

### Journaux d'application

Pour les journaux d'applications personnalisés, vous pouvez les classer par source de données à l'aide de l'une des méthodes suivantes :

- **Balises de groupes de journaux** : ajoutez des balises à vos groupes de journaux à l'aide des clés `cw:datasource:name` et `cw:datasource:type` pour spécifier respectivement le nom et le type de la source de données pour tous les journaux ingérés dans le groupe de journaux. Les valeurs des balises peuvent comporter jusqu'à 64 caractères et ne contenir que des lettres minuscules, des chiffres et des traits de soulignement. Ils doivent commencer par une lettre ou un chiffre et ne doivent pas contenir de traits de soulignement doubles (`__`).
- **Configuration du pipeline** : configurez les informations des sources de données via des pipelines de traitement des journaux lors de l'ingestion des journaux de votre application.

 Note

Les noms des sources de données ne peuvent pas commencer par « aws » ou « amazon » pour éviter tout conflit avec les journaux AWS de service.

## Champs du système

CloudWatch Logs ajoute automatiquement trois champs système aux journaux classés par source de données. Ces champs servent de facettes par défaut :

- `@data_source_name`- Contient le nom de la source de données, ou « Inconnu » s'il n'est pas déterminé
- `@data_source_type`- Contient le type de source de données, ou « Inconnu » s'il n'est pas déterminé
- `@data_format`- Indique le format des données du journal

Lorsque le nom ou le type de la source de données ne peuvent pas être déterminés, ces champs sont définis sur « Inconnu ». Les sources de données dont les valeurs sont « inconnues » sont toujours visibles par facettes et dans le tableau des sources de données sous « Gestion des journaux » dans la console, ce qui vous permet d'identifier les journaux non classés et de quel groupe de journaux ils proviennent.

Le `@data_format` champ peut contenir l'une des valeurs suivantes :

- `Default`- Logs ingérés sans modification.
- `Custom`- Journaux traités par le biais de processeurs de pipeline ou journaux ingérés dans un groupe de journaux avec des `name/type` balises de source de données.
- `OCSF-<version>`- Logs traités avec les processeurs OCSF (Open Cybersecurity Schema Framework) dans des pipelines.
- `AWS-OTEL-LOG-V<version>`- OpenTelemetry journaux ingérés via le point de terminaison CloudWatch OTLP.
- `AWS-OTEL-TRACE-V<version>`- OpenTelemetry traces ingérées via le point de terminaison CloudWatch OTLP.

Ces champs système vous permettent de filtrer et d'interroger vos journaux en fonction de leur source et de leur format, ce qui facilite l'utilisation de journaux provenant de différentes origines et de différents pipelines de traitement.

## Accès aux sources de données

### Console

Dans la console CloudWatch Logs, vous utilisez l'onglet Log Management pour accéder à vos sources de données. CloudWatch Logs consolide automatiquement vos données de journal par source et par type de données, en découvrant en permanence les données nouvellement ingérées. Dans la liste des sources de données, vous pouvez créer des pipelines, définir des index de champs et des facettes.

### AWS CLI

Utilisez la commande suivante pour répertorier les sources de données et les types de journaux distincts de votre compte :

```
aws logs list-aggregate-log-group-summaries --group-by DATA_SOURCE_NAME_AND_TYPE
```

## Relation avec les groupes de journaux

Les sources de données complètent les groupes de journaux au lieu de les remplacer. Vos journaux continuent d'être stockés dans des groupes de journaux comme par le passé, mais ils sont désormais automatiquement balisés avec les informations relatives aux sources de données. Cette double organisation vous permet de :

- Utilisez des groupes de journaux pour définir des politiques précises de contrôle d'accès et de rétention
- Utilisez des sources de données pour la découverte et l'analyse des journaux basées sur les services
- Journaux de requêtes utilisant l'une ou l'autre méthode d'organisation en fonction de vos besoins

Les sources de données facilitent l'utilisation des journaux à grande échelle en fournissant une vue centrée sur le service des données de vos journaux dans l'ensemble de votre AWS infrastructure.

## Fonctionnalités activées par les sources de données

Les sources de données offrent des fonctionnalités avancées de traitement des journaux et d'analyse grâce à la découverte sur le terrain et à des structures de données cohérentes.

- **Facettes** : les facettes sont des champs de log indexés qui fournissent un filtrage et une analyse interactifs sans écrire de requêtes. CloudWatch Logs crée automatiquement des facettes pour le nom et le type de source de données, et vous pouvez créer des politiques de facettes sur les champs découverts afin d'accélérer le dépannage. Les facettes affichent les distributions de valeurs et les dénombrements dans CloudWatch Logs Insights, ce qui permet d'identifier facilement les modèles par le biais de point-and-click l'exploration.
- **Pipelines** : créez des pipelines de transformation qui s'appliquent à tous les journaux provenant d'un nom et d'un type de source de données spécifiques. Cela vous permet de définir des règles de traitement cohérentes pour les journaux provenant de la même source.
- **Découverte de champs** : CloudWatch les journaux découvrent automatiquement les champs et leurs types de données pour chaque combinaison de nom et de type de source de données en fonction des processeurs de pipeline. Pour les journaux AWS gérés, les structures de champs sont prédéfinies. Pour les journaux d'applications, nous recommandons de conserver des formats de journaux cohérents afin de maximiser la compatibilité avec les outils d'analyse tels que les tables Amazon S3 qui nécessitent des structures de champs bien définies.

Vous pouvez consulter la liste complète des champs et leurs types pour n'importe quelle source de données à l'aide de l'GetLogFieldsAPI :

```
aws logs get-log-fields --data-source-name <name> --data-source-type <type>
```

Cette découverte et cette cohérence des champs permettent des analyses et des intégrations avancées, car les outils externes peuvent fonctionner avec des structures de champs prévisibles lors du traitement des données de vos journaux.

## Pris en charge Services AWS pour les sources de données

Le tableau suivant répertorie Services AWS les sources de données automatiquement classées par CloudWatch Logs :



| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| amazon_api_gateway                                    | access                                              |
| amazon_bedrock_agentcore                              | browser_usage                                       |
| amazon_bedrock_agentcore                              | code_interpreter_application                        |
| amazon_bedrock_agentcore                              | code_interpreter_usage                              |
| amazon_bedrock_agentcore                              | gateway_application                                 |
| amazon_bedrock_agentcore                              | identity_workload_application                       |
| amazon_bedrock_agentcore                              | memory_application                                  |
| amazon_bedrock_agentcore                              | online_evaluation_config                            |
| amazon_bedrock_agentcore                              | runtime_application                                 |
| amazon_bedrock_agentcore                              | runtime_usage                                       |
| amazon_bedrock_agents                                 | application                                         |
| amazon_bedrock_agents                                 | event                                               |
| amazon_bedrock_knowledge_bases                        | application                                         |
| amazon_cloudfront                                     | access                                              |
| amazon_cloudfront                                     | connection                                          |
| amazon_cloudwatch                                     | rum_app_monitor                                     |
| amazon_cognito                                        | user_pool                                           |
| amazon_ec2                                            | verified_access                                     |
| amazon_eks                                            | api_server                                          |

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| amazon_eks                                            | audit                                               |
| amazon_eks                                            | authenticator                                       |
| amazon_eks                                            | controller_manager                                  |
| amazon_eks                                            | scheduler                                           |
| amazon_elasticache                                    | cluster                                             |
| amazon_eventbridge                                    | eventbus_error                                      |
| amazon_eventbridge                                    | eventbus_info                                       |
| amazon_eventbridge                                    | pipes_execution                                     |
| amazon_interactive_video_service                      | chat                                                |
| amazon_managed_prometheus                             | scraper                                             |
| amazon_managed_prometheus                             | workspace                                           |
| amazon_msk                                            | broker                                              |
| amazon_msk                                            | connect                                             |
| amazon_opensearch_service                             | pipeline                                            |
| amazon_q_business                                     | events                                              |
| amazon_q_business                                     | sync_job                                            |
| amazon_q_connect                                      | events                                              |
| amazon_route53                                        | global_resolver_query                               |
| amazon_route53                                        | hosted_zones                                        |
| amazon_route53                                        | profiles_resolver_query                             |

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| amazon_route53                                        | resolver_query                                      |
| amazon_sagemaker                                      | workteam_activity                                   |
| amazon_ses                                            | ingress_endpoints                                   |
| amazon_ses                                            | rule_sets                                           |
| amazon_ses                                            | traffic_policy                                      |
| amazon_vpc                                            | flow                                                |
| amazon_vpc                                            | route_server_peer                                   |
| amazon_vpc_lattice                                    | access                                              |
| amazon_vpc_lattice                                    | resource_access                                     |
| amazon_workmail                                       | access_control                                      |
| amazon_workmail                                       | authentication                                      |
| amazon_workmail                                       | personal_access                                     |
| amazon_workmail                                       | workmail_access                                     |
| amazon_workmail                                       | workmail_availability                               |
| aws_b2b_data_interchange                              | execution                                           |
| aws_backup                                            | data_access                                         |
| aws_backup                                            | hypervisor                                          |
| aws_clean_rooms                                       | analysis                                            |
| aws_client_vpn                                        | connection                                          |
| aws_client_vpn                                        | event                                               |

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| aws_cloudtrail                                        | data                                                |
| aws_cloudtrail                                        | management                                          |
| aws_elemental_mediapackage                            | egress_access                                       |
| aws_elemental_mediapackage                            | ingress_access                                      |
| aws_elemental_mediatailor                             | ad_decision                                         |
| aws_elemental_mediatailor                             | manifest                                            |
| aws_elemental_mediatailor                             | transcode                                           |
| aws_entity_resolution                                 | id_mapping_workflow                                 |
| aws_entity_resolution                                 | matching_workflow                                   |
| aws_iot_fleetwise                                     | error                                               |
| aws_mainframe_modernization                           | batch_job                                           |
| aws_mainframe_modernization                           | config                                              |
| aws_mainframe_modernization                           | console                                             |
| aws_mainframe_modernization                           | dataset_import                                      |
| aws_network_firewall                                  | alert                                               |
| aws_network_firewall                                  | flow                                                |
| aws_network_firewall                                  | tls                                                 |
| aws_nlb                                               | access                                              |
| aws_pcs                                               | job_completion                                      |
| aws_pcs                                               | scheduler                                           |

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| aws_security_hub_cspm                                 | asff_finding                                        |
| aws_shield                                            | protection_flow                                     |
| aws_step_functions                                    | express                                             |
| aws_step_functions                                    | standard                                            |
| aws_transfer_family                                   | server                                              |
| aws_waf                                               | access                                              |

## Sources tierces prises en charge pour les sources de données

Le tableau suivant répertorie les sources tierces qui sont automatiquement classées par CloudWatch Logs en tant que sources de données lorsqu'elles sont ingérées via des pipelines :

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| crowdstrike_falcon                                    | detection_finding                                   |
| crowdstrike_falcon                                    | process_activity                                    |
| github_auditlogs                                      | account_change                                      |
| github_auditlogs                                      | api_activity                                        |
| github_auditlogs                                      | entity_management                                   |
| microsoft_entraid                                     | account_change                                      |
| microsoft_entraid                                     | authentication                                      |
| microsoft_entraid                                     | entity_management                                   |

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| microsoft_entraid                                     | user_access_management                              |
| microsoft_office365                                   | account_change                                      |
| microsoft_office365                                   | application_lifecycle                               |
| microsoft_office365                                   | authentication                                      |
| microsoft_office365                                   | compliance_finding                                  |
| microsoft_office365                                   | detection_finding                                   |
| microsoft_office365                                   | email_activity                                      |
| microsoft_office365                                   | file_hosting_activity                               |
| microsoft_office365                                   | group_management                                    |
| microsoft_office365                                   | incident_finding                                    |
| microsoft_office365                                   | user_access_management                              |
| microsoft_office365                                   | vulnerability_finding                               |
| microsoft_office365                                   | web_resources_activity                              |
| microsoft_windows                                     | account_change                                      |
| microsoft_windows                                     | authentication                                      |
| microsoft_windows                                     | entity_management                                   |
| microsoft_windows                                     | event_log_activity                                  |
| microsoft_windows                                     | file_system_activity                                |
| microsoft_windows                                     | group_management                                    |
| microsoft_windows                                     | kernel_activity                                     |

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| okta_auth0                                            | api_activity                                        |
| okta_auth0                                            | authentication                                      |
| okta_sso                                              | api_activity                                        |
| okta_sso                                              | authentication                                      |
| okta_sso                                              | detection_finding                                   |
| okta_sso                                              | entity_management                                   |
| paloaltonetworks_nextgenerationfirewall               | authentication                                      |
| paloaltonetworks_nextgenerationfirewall               | detection_finding                                   |
| paloaltonetworks_nextgenerationfirewall               | network_activity                                    |
| paloaltonetworks_nextgenerationfirewall               | process_activity                                    |
| sentinelone_endpointsecurity                          | dns_activity                                        |
| sentinelone_endpointsecurity                          | file_system_activity                                |
| sentinelone_endpointsecurity                          | http_activity                                       |
| sentinelone_endpointsecurity                          | process_activity                                    |
| servicenow_cmdb                                       | api_activity                                        |
| servicenow_cmdb                                       | datastore_activity                                  |
| servicenow_cmdb                                       | entity_management                                   |

| Nom de la source de données (champ @data_source_name) | Type de source de données (champ @data_source_type) |
|-------------------------------------------------------|-----------------------------------------------------|
| wiz_cnapp                                             | api_activity                                        |
| wiz_cnapp                                             | authentication                                      |
| wiz_cnapp                                             | compliance_finding                                  |
| wiz_cnapp                                             | detection_finding                                   |
| wiz_cnapp                                             | vulnerability_finding                               |
| zscaler_internetaccess                                | authentication                                      |
| zscaler_internetaccess                                | dns_activity                                        |
| zscaler_internetaccess                                | http_activity                                       |
| zscaler_internetaccess                                | network_activity                                    |



# 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 effectuer des requêtes pour vous aider à répondre de manière plus efficace aux problèmes opérationnels. Outre les requêtes à l'aide de groupes de journaux, vous pouvez effectuer des requêtes à l'aide de facettes, de sources de données et de types de données. En cas de problème, vous pouvez utiliser CloudWatch Logs Insights pour identifier les causes potentielles et valider les correctifs déployés. Vous êtes limité à 100 CloudWatch Logs Insights QL simultanés par compte, y compris les requêtes ajoutées aux tableaux de bord. En outre, vous pouvez exécuter 15 requêtes simultanées pour OpenSearch Service PPL ou OpenSearch Service SQL.

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 que les

requêtes soient correctement demandées. 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.

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 les événements de votre journal, vous pouvez les utiliser dans une requête. La requête ignore les événements du journal de traitement dont on sait qu'ils n'incluent pas 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, réexécuter des requêtes enregistrées et [utiliser des requêtes enregistrées avec des paramètres](#).
- [Ajouter des requêtes aux tableaux de bord](#).
- [Chiffrer les résultats des requêtes avec AWS Key Management Service](#).
- [La génération de requêtes en langage naturel](#) vous permet d'utiliser le langage naturel pour créer des requêtes CloudWatch Logs Insights. Vous pouvez poser des questions ou décrire les données que vous recherchez, puis l'IA génère une requête en fonction de votre demande et line-by-line explique le fonctionnement de la requête.
- [Utilisez des facettes pour regrouper, filtrer et explorer vos journaux de manière interactive](#).

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

- [Requêtes de comparaison](#) qui comparent les événements du journal d'un groupe de journaux aux événements du journal d'une période précédente.

 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 [StartQuery](#) le 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](#)
- [Utilisez des facettes pour regrouper et explorer les journaux](#)
- [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](#)
- [Chiffrez les résultats des requêtes 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 : didacticiels 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 le délai le plus court 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 QLpattern, à l'exception de diff, et. unmask

|                                       |                                                                                                                                                                                                                                     |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><b><u>anomaly</u></b></a> | Identifie les modèles inhabituels dans les données de vos journaux à l'aide de l'apprentissage automatique.                                                                                                                         |
| <a href="#"><b><u>display</u></b></a> | Affiche un ou plusieurs champs spécifiques dans les résultats de la requête.                                                                                                                                                        |
| <a href="#"><b><u>fields</u></b></a>  | 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. |
| <a href="#"><b><u>filter</u></b></a>  | Filtre la requête pour ne renvoyer que les événements du journal qui satisfont à une ou plusieurs conditions.                                                                                                                       |

|                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><b>filterIndex</b></a> | <p>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.</p> <p>Cette commande n'est pas prise en charge pour les groupes de journaux de la classe de journaux Infrequent Access.</p> |
| <a href="#"><b>pattern</b></a>     | <p>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'analyser les modèles trouvés dans vos événements de journal. Pour de plus amples informations, veuillez consulter <a href="#">Analyse de modèles</a>.</p>                                                                                                                                                                                 |
| <a href="#"><b>diff</b></a>        | <p>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.</p>                                                                                                                                                                                                                                                                              |
| <a href="#"><b>parse</b></a>       | <p>Extrait les données d'un champ de journal afin de créer un champ extrait que vous pouvez traiter dans votre requête. <b>parse</b> prend en charge à la fois le mode glob utilisant des caractères génériques et des expressions régulières.</p>                                                                                                                                                                                                                                                                                              |
| <a href="#"><b>sort</b></a>        | <p>Affiche les événements du journal renvoyés dans l'ordre croissant (asc) ou décroissant (desc).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#"><b>SOURCE</b></a>      | <p>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.</p>                                                                                                                               |
| <a href="#"><b>stats</b></a>       | <p>Calcule les statistiques agrégées en utilisant les valeurs des champs du journal.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

|                                                       |                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>limit</u></a>                          | Spécifie le nombre maximum d'événements du journal que votre requête doit renvoyer. Utile avec <b>sort</b> pour renvoyer les « 20 meilleurs » résultats ou les « 20 plus récents ».                                                                                                                                                                               |
| <a href="#"><u>dedup</u></a>                          | Supprime les résultats dupliqués en fonction de valeurs spécifiques dans les champs que vous indiquez.                                                                                                                                                                                                                                                            |
| <a href="#"><u>unmask</u></a>                         | 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 <a href="#"><u>Aider à protéger les données sensibles des journaux grâce au masquage</u></a> (français non garanti). |
| <a href="#"><u>unnest</u></a>                         | Aplatit une liste prise en entrée pour produire plusieurs enregistrements avec un seul enregistrement pour chaque élément de la liste.                                                                                                                                                                                                                            |
| <a href="#"><u>lookup</u></a>                         | Enrichit les événements du journal avec les données d'une table de recherche en faisant correspondre les valeurs des champs. Utilisez les tables de recherche pour ajouter des données de référence telles que des informations sur les utilisateurs, les noms des applications ou des informations sur les produits aux résultats de vos requêtes.               |
| <a href="#"><u>Autres opérations et fonctions</u></a> | CloudWatch Logs Insights prend également en charge de nombreuses 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 de CloudWatch Logs Insights.

## Rubriques

- [Commandes Logs Insights QL prises en charge dans les classes de journaux](#)
- [anomalie](#)
- [display](#)
- [fields](#)
- [filtre](#)



- [Index des filtres](#)
- [SOURCE](#)
- [pattern](#)
- [diff](#)
- [parse](#)
- [sort](#)
- [stats](#)
- [limite](#)
- [dedup](#)
- [unmask](#)
- [unnest](#)
- [chercher](#)
- [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`, `diff`, `filterIndex`, et `unmask`.

anomaly

Utilisez-le `anomaly` pour identifier automatiquement les modèles inhabituels et les problèmes potentiels dans les données de vos journaux à l'aide de l'apprentissage automatique.

La `anomaly` commande étend les `pattern` fonctionnalités existantes et utilise des analyses avancées pour aider à identifier les anomalies potentielles dans les données du journal. Vous pouvez l'utiliser `anomaly` pour réduire le temps nécessaire à l'identification et à la résolution des problèmes opérationnels en faisant apparaître automatiquement des modèles ou des comportements inhabituels dans vos journaux.

La `anomaly` commande fonctionne avec la [pattern](#) commande pour d'abord identifier les modèles de journalisation, puis détecter les anomalies au sein de ces modèles. Vous pouvez également

les combiner `anomaly` avec les [sort](#) commandes [filter](#) ou pour concentrer la détection des anomalies sur des sous-ensembles spécifiques de vos données.

## Entrée de commande anormale

La `anomaly` commande est généralement utilisée après la [pattern](#) commande pour analyser les modèles identifiés dans les données de votre journal. La commande ne nécessite aucun paramètre supplémentaire et analyse le résultat des commandes précédentes de votre requête.

## Types d'anomalies identifiées

La `anomaly` commande identifie cinq types d'anomalies distincts :

- Anomalies de fréquence des modèles : fréquences inhabituelles de modèles de log spécifiques, par exemple lorsqu'une application commence à générer plus de messages d'erreur que d'habitude.
- Nouvelles anomalies de schéma : modèles de journal inédits qui peuvent indiquer de nouveaux types d'erreurs ou de nouveaux messages apparaissant dans vos journaux.
- Anomalies de variation des jetons : modifications inattendues du contenu des messages de journal susceptibles d'indiquer des variations inhabituelles des formats de journal attendus.
- Anomalies numériques des jetons : modifications inhabituelles des valeurs numériques dans les journaux qui peuvent aider à détecter des problèmes de performance potentiels ou des variations inattendues des mesures.
- Anomalies du code d'erreur HTTP : modèles liés aux réponses d'erreur HTTP, particulièrement utiles lors de la surveillance des applications Web et APIs.

## Sortie de commande Anomaly

La `anomaly` commande préserve tous les champs des données d'entrée et ajoute les résultats de détection des anomalies pour aider à identifier les modèles inhabituels dans les données de votre journal.

## Exemples

La commande suivante identifie les modèles dans les données de votre journal, puis détecte les anomalies au sein de ces modèles :

```
fields @timestamp, @message
```

```
| pattern @message  
| anomaly
```

La `anomaly` commande peut être utilisée avec le filtrage pour se concentrer sur des types de journaux spécifiques :

```
fields @timestamp, @message  
| filter @type = "REPORT"  
| pattern @message  
| anomaly
```

La `anomaly` commande peut être combinée avec le tri pour organiser les résultats :

```
fields @timestamp, @message  
| filter @type = "ERROR"  
| pattern @message  
| anomaly  
| sort @timestamp desc
```

## 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.

## filter

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 `surrequestId`. 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-chaînes 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 :

- `@logStream`
- `@aws.region`
- `@aws.account`
- `@source.log`
- `@data_source_name`
- `@data_source_type`
- `@data_format`

- `traceId`
- `severityText`
- `attributes.session.id`

CloudWatch Logs fournit également des index de champs par défaut pour certaines combinaisons de noms et de types de sources de données. Les index de champs par défaut sont automatiquement disponibles pour les combinaisons de nom et de type de source de données suivantes :

| Nom et type de source de données                                           | Index de champs par défaut                                                                                                                                 |
|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>amazon_vpc.flow</code>                                               | <code>action</code><br><code>logStatus</code><br><code>region</code><br><code>flowDirection</code><br><code>type</code>                                    |
| <code>amazon_route53.resolver_query</code>                                 | <code>query_type</code><br><code>transport</code><br><code>rcode</code>                                                                                    |
| <code>aws_waf.access</code>                                                | <code>action</code><br><code>httpRequest.country</code>                                                                                                    |
| <code>aws_cloudtrail.data</code><br><code>aws_cloudtrail.management</code> | <code>eventSource</code><br><code>eventName</code><br><code>awsRegion</code><br><code>userAgent</code><br><code>errorCode</code><br><code>eventType</code> |

| Nom et type de source de données | Index de champs par défaut |
|----------------------------------|----------------------------|
|                                  | managementEvent            |
|                                  | readOnly                   |
|                                  | eventCategory              |
|                                  | requestId                  |

Les index de champs par défaut s'ajoutent aux index de champs personnalisés que vous définissez dans 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 sources de and/or données des 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.

### Groupes de journaux de requêtes

`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.

- `accountIdentifier` 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, consultez [Classes 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'classifiant, 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"
```

## Sources de données d'interrogation

**SOURCE** Pour spécifier les sources de données à interroger, vous pouvez utiliser le `dataSource` mot clé. Vous pouvez inclure jusqu'à dix sources de données dans la liste.

L'exemple suivant sélectionne la source `amazon_vpc.flow` de données.

```
SOURCE dataSource(['amazon_vpc.flow'])
```

L'exemple suivant sélectionne la source de `amazon_vpc.flow` données et limite les groupes de journaux en fonction d'un préfixe de nom de groupe de journaux.

```
SOURCE dataSource(['amazon_vpc.flow']) logGroups(namePrefix: ['namePrefix1'])
```

## 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 :

- `diffcompare` les événements du journal dans la plage de temps actuellement sélectionnée aux événements du journal de la plage de temps immédiatement précédente.
- `diff previousDaycompare` les événements du journal dans la plage horaire actuellement sélectionnée aux événements du journal survenus à la même heure le jour précédent.
- `diff previousWeekcompare` les événements du journal dans la plage horaire actuellement sélectionnée aux événements du journal survenus à la même période de la semaine précédente.
- `diff previousMonthcompare` 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, consultez [Syntaxe 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 dans des 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.

[illegible]

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 des 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                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>avg(fieldName: NumericLogField)</code>                        | nombre           | Moyenne des valeurs dans le champ spécifié.                                                                                                                                                                                                               |
| <code>count()</code><br><br><code>count(fieldName: LogField)</code> | number           | Compte les événements du journal.<br><code>count()</code> (ou <code>count(*)</code> ) compte tous les événements renvoyés par la requête, tandis que <code>count(fieldName)</code> compte tous les enregistrements qui incluent le nom du champ spécifié. |
| <code>count_distinct(fieldName: LogField)</code>                    | number           | 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 <code>count_distinct</code> n'est qu'une approximation.                                    |
| <code>max(fieldName: LogField)</code>                               | LogFieldValeur   | Valeur maximale des valeurs pour ce journal dans le champ interrogé.                                                                                                                                                                                      |
| <code>min(fieldName: LogField)</code>                               | LogFieldValeur   | Valeur minimale des valeurs pour ce journal dans le champ interrogé.                                                                                                                                                                                      |
| <code>pct(fieldName: LogFieldValue, percent: number)</code>         | LogFieldValeur   | Un centile indique la position relative d'une valeur dans un ensemble de données. Par exemple, <code>pct(@duration, 95)</code> renvoie la valeur <code>@duration</code> à laquelle 95 % des valeurs de <code>@duration</code> sont inférieures            |

| Fonction                                        | Type de résultat | Description                                             |
|-------------------------------------------------|------------------|---------------------------------------------------------|
|                                                 |                  | à cette valeur et 5 % des valeurs lui sont supérieures. |
| <code>stddev(fieldName: NumericLogField)</code> | number           | Déviation standard des valeurs dans le champ spécifié.  |
| <code>sum(fieldName: NumericLogField)</code>    | number           | 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                                                                                                                                    |
|----------------------------------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>earliest(fieldName: LogField)</code>   | LogField         | Renvoie la valeur de <code>fieldName</code> à partir de l'événement de journal qui a l'horodatage le plus récent dans les journaux interrogés. |
| <code>latest(fieldName: LogField)</code>     | LogField         | Renvoie la valeur de <code>fieldName</code> à partir de l'événement de journal qui a l'horodatage le plus ancien dans les journaux interrogés. |
| <code>sortsFirst(fieldName: LogField)</code> | LogField         | Renvoie la valeur de <code>fieldName</code> triée la première dans les journaux interrogés.                                                    |
| <code>sortsLast(fieldName: LogField)</code>  | LogField         | Renvoie la valeur de <code>fieldName</code> 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`.

Lorsque vous l'utilisez `dedup` dans une requête, la console affiche un message tel que Afficher X des Y enregistrements, où X représente le nombre de résultats dédoublés et Y le nombre total d'enregistrements mis en correspondance avant la déduplication. Cela indique que les doublons ont été supprimés et ne signifie pas que des données sont manquantes.

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, consultez [Requêtes 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 `MAP`, `numbers`, `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
```

### chercher

**lookup** À utiliser pour enrichir les résultats de votre requête avec des données de référence issues d'une table de recherche. Une table de recherche contient des données CSV que vous chargez

sur Amazon CloudWatch Logs. Lorsqu'une requête est exécutée, la `lookup` commande fait correspondre un champ de votre journal des événements à un champ de la table de recherche et ajoute les champs de sortie spécifiés aux résultats.

Utilisez les tables de recherche pour les scénarios d'enrichissement des données, tels que le mappage des informations utilisateur IDs à utilisateur, les codes de produit aux informations sur le produit ou les codes d'erreur aux descriptions d'erreurs.

## Création et gestion de tables de recherche

Avant de pouvoir utiliser la `lookup` commande dans une requête, vous devez créer une table de recherche. Vous pouvez créer et gérer des tables de recherche depuis la CloudWatch console ou à l'aide de l'API Amazon CloudWatch Logs.

### Pour créer une table de recherche (console)

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, choisissez Paramètres, puis l'onglet Journaux.
3. Accédez aux tables de recherche, puis sélectionnez Gérer.
4. Choisissez Créer une table de recherche.
5. Entrez un nom pour la table de recherche. Le nom ne peut contenir que des caractères alphanumériques, des traits d'union et des traits de soulignement.
6. (Facultatif) Entrez une description.
7. Téléchargez un fichier CSV. Le fichier doit inclure une ligne d'en-tête avec les noms de colonnes, utiliser le codage UTF-8 et ne pas dépasser 10 Mo.
8. (Facultatif) Spécifiez une AWS KMS clé pour chiffrer les données de la table.
9. Choisissez Créer.

Après avoir créé une table de recherche, vous pouvez l'afficher dans l'éditeur de requêtes CloudWatch Logs Insights. Cliquez sur l'onglet Tables de recherche pour parcourir les tables disponibles et leurs champs.

Pour mettre à jour une table de recherche, sélectionnez-la et choisissez Actions, Mettre à jour. Téléchargez un nouveau fichier CSV pour remplacer tout le contenu existant. Pour supprimer une table de recherche, choisissez Actions, Supprimer.

**Note**

Vous pouvez créer jusqu'à 100 tables de recherche par compte et par compte Région AWS. Les fichiers CSV peuvent peser jusqu'à 10 Mo. Vous pouvez également gérer les tables de recherche à l'aide de l'API Amazon CloudWatch Logs. Pour plus d'informations, consultez [CreateLookupTable](#) le manuel Amazon CloudWatch Logs API Reference.

**Note**

Si la table de recherche est chiffrée à l'aide d'une clé KMS, l'appelant doit avoir l'`kms:Decrypt` autorisation sur la clé (la clé KMS utilisée pour chiffrer la table de recherche) d'utiliser l'`StartQueryAPI` avec une requête faisant référence à cette table de recherche. Pour de plus amples informations, veuillez consulter [Chiffrez les tables de recherche dans les CloudWatch journaux à l'aide de AWS Key Management Service](#).

Syntaxe de requête pour la recherche

Structure de commande

Le format de cette commande est illustré ci-dessous.

```
lookup table lookup-field as log-field [,...] output-mode output-field [,...]
```

La commande utilise les arguments suivants :

- *table*— Nom de la table de recherche à utiliser.
- *lookup-field*— Le champ de la table de recherche à comparer.
- *log-field*— Le champ correspondant dans votre journal des événements. La correspondance est exacte et distingue les majuscules et minuscules.
- *output-mode*— Spécifiez OUTPUT d'ajouter les champs de sortie aux résultats. Si un champ portant le même nom existe déjà dans le journal des événements, il est remplacé.
- *output-field*— Un ou plusieurs champs de la table de recherche à ajouter aux résultats.

Exemple : enrichir les événements du journal avec les détails de l'utilisateur

Supposons que vous disposiez d'un groupe de journaux contenant des événements contenant un `id` champ et d'une table de recherche nommée `user_data` par des colonnes `id`, `name`, `email`, et `department`. La requête suivante enrichit chaque événement du journal avec le nom, l'adresse e-mail et le service de l'utilisateur à partir de la table de recherche.

```
fields action, status, name, email, department
| lookup user_data id OUTPUT name, email, department
```

Exemple : utilisation de la recherche avec agrégation

Vous pouvez utiliser des champs de sortie de recherche avec des fonctions d'agrégation. La requête suivante enrichit les événements du journal avec les détails de l'utilisateur, puis compte les événements regroupés par adresse e-mail.

```
fields user_id, action, username, email, department
| lookup user_data user_id OUTPUT username, email, department
| stats count(*) by email
```

Exemple : utiliser la recherche avec filtre

Vous pouvez filtrer les résultats en fonction des champs renvoyés par la recherche. La requête suivante enrichit les événements du journal, puis filtre pour n'afficher que les événements d'un département spécifique.

```
fields user_id, action
| lookup user_data user_id OUTPUT username, email, department
| filter department = "Engineering"
```

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                                           |
| $a - b$   | 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         |
|--------------------|---------------------|
| <code>=</code>     | Égal à              |
| <code>!=</code>    | Non égal à          |
| <code>&lt;</code>  | Inférieur à         |
| <code>&gt;</code>  | Supérieur à         |
| <code>&lt;=</code> | Inférieur ou égal à |
| <code>&gt;=</code> | 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                                                                                                |
|-------------------------------|------------------|------------------------------------------------------------------------------------------------------------|
| <code>abs(a: number)</code>   | nombre           | Valeur absolue                                                                                             |
| <code>ceil(a: number)</code>  | number           | Arrondir jusqu'au nombre entier supérieur suivant (le plus petit nombre entier supérieur à la valeur de a) |
| <code>floor(a: number)</code> | number           | Arrondir jusqu'au nombre entier inférieur suivant (le plus grand nombre entier inférieur à la valeur de a) |

| Opération                                              | Type de résultat | Description                      |
|--------------------------------------------------------|------------------|----------------------------------|
| <code>greatest(a: number, ...numbers: number[])</code> | number           | Renvoie la valeur la plus grande |
| <code>least(a: number, ...numbers: number[])</code>    | number           | Renvoie la valeur la plus petite |
| <code>log(a: number)</code>                            | number           | Journal naturel                  |
| <code>sqrt(a: number)</code>                           | number           | 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 tout autre champ donne `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 :

- `m` pendant des millisecondes
- `s` pendant quelques secondes
- `mp` 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 utilisant `s`. 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 avec `s`. 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                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>bin(period: Period)</code> | Horodatage       | <p>Arrondit la valeur de <code>@timestamp</code> à la période donnée, puis la tronque. Par exemple, <code>bin(5m)</code> arrondit la valeur de <code>@timestamp</code> aux 5 minutes les plus proches.</p> <p>Vous pouvez l'utiliser pour regrouper plusieurs entrées de journal dans une requête. L'exemple suivant indique le nombre d'exceptions par heure :</p> |

| Fonction                                                     | Type de résultat | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                              |                  | <pre>filter @message like /Exception/     stats count(*) as exceptionCount   by bin(1h)     sort exceptionCount desc</pre> <p>les unités de temps et les abréviations suivantes sont prises en charge par la fonction <code>bin</code>. Pour toutes les unités et abréviations qui incluent plus d'un caractère, l'ajout de <code>s</code> au pluriel est pris en charge. Donc, les deux <code>hr</code> et <code>hrs</code> travaillent pour spécifier les heures.</p> <ul style="list-style-type: none"> <li>• <code>millisecond ms msec</code></li> <li>• <code>second s sec</code></li> <li>• <code>minute m min</code></li> <li>• <code>hour h hr</code></li> <li>• <code>day d</code></li> <li>• <code>week w</code></li> <li>• <code>month mo mon</code></li> <li>• <code>quarter q qtr</code></li> <li>• <code>year y yr</code></li> </ul> |
| <code>datefloor(timestamp: Timestamp, period: Period)</code> | Horodatage       | Tronque l'horodatage pour la période donnée. Par exemple, <code>datefloor(@timestamp, 1h)</code> tronque toutes les valeurs de <code>@timestamp</code> vers la valeur la plus basse de l'heure.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>dateceil(timestamp: Timestamp, period: Period)</code>  | Horodatage       | Arrondit l'horodatage pour la période donnée, puis la tronque. Par exemple, <code>dateceil(@timestamp, 1h)</code> tronque toutes les valeurs de <code>@timestamp</code> vers la valeur la plus élevée de l'heure.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Fonction                                    | Type de résultat | Description                                                                                                                                                                                                                                                          |
|---------------------------------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>fromMillis(fieldName: number)</code>  | Horodatage       | Interprète le champ en entrée comme le nombre de millisecondes depuis l'époque Unix et le convertit en horodatage.                                                                                                                                                   |
| <code>toMillis(fieldName: Timestamp)</code> | number           | Convertit l'horodatage trouvé dans le champ nommé en un nombre représentant les millisecondes depuis l'époque Unix. Par exemple, <code>toMillis(@timestamp)</code> convertit l'horodatage <code>2022-01-14T13:18:031.000-08:00</code> à <code>1642195111000</code> . |

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

 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                                      |
|-----------------------------------------------------------------------|------------------|--------------------------------------------------|
| <code>ispresent(fieldName: LogField)</code>                           | Booléen          | Renvoie <code>true</code> si le champ existe     |
| <code>coalesce(fieldName: LogField, ...fieldNames: LogField[])</code> | LogField         | Renvoie la première valeur non nulle de la liste |

### Fonctions JSON

#### Fonctions JSON

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

| Fonction                                  | Type de résultat     | Description                                                                                                                                                                                                    |
|-------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>jsonParse(fieldName: string)</code> | 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 représentation. |

| Fonction                                          | Type de résultat | Description                                                              |
|---------------------------------------------------|------------------|--------------------------------------------------------------------------|
| <code>jsonStringify(fieldName: Map   List)</code> | String           | 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                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>isValidIp(fieldName: string)</code>                      | booléen          | Renvoie <code>true</code> si le champ est une IPv6 adresse IPv4 ou une adresse valide.                                                                                                                                                                                                                                                       |
| <code>isValidIPv4(fieldName: string)</code>                    | boolean          | Renvoie <code>true</code> si le champ est une IPv4 adresse valide.                                                                                                                                                                                                                                                                           |
| <code>isValidIPv6(fieldName: string)</code>                    | boolean          | Renvoie <code>true</code> si le champ est une IPv6 adresse valide.                                                                                                                                                                                                                                                                           |
| <code>isIpInSubnet(fieldName: string, subnet: string)</code>   | boolean          | Renvoie <code>true</code> 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 <code>192.0.2.0/24</code> ou <code>2001:db8::/32</code> , où <code>192.0.2.0</code> ou <code>2001:db8::</code> est le début du bloc d'adresse CIDR. |
| <code>isIPv4InSubnet(fieldName: string, subnet: string)</code> | boolean          | Renvoie <code>true</code> 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 <code>192.0.2.0/24</code> où                                                                                                                                   |

| Fonction                                                       | Type de résultat | Description                                                                                                                                                                                                                       |
|----------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                |                  | 192.0.2.0 est le début du bloc d'adresse CIDR.                                                                                                                                                                                    |
| <code>isIpv6InSubnet(fieldName: string, subnet: string)</code> | 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                                                                               |
|---------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------------|
| <code>isempty(fieldName: string)</code>                                               | Number           | Renvoie 1 si le champ est manquant ou est une chaîne vide.                                |
| <code>isblank(fieldName: string)</code>                                               | Number           | Renvoie 1 si le champ est manquant, est une chaîne vide ou contient uniquement un espace. |
| <code>concat(str: string, ...strings: string[])</code>                                | chaîne           | Concatène les chaînes.                                                                    |
| <code>ltrim(str: string)</code><br><code>ltrim(str: string, trimChars: string)</code> | 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 <code>trimChars</code> à gauche de <code>str</code> . Par exemple, <code>ltrim("xy ZxyfooxyZ", "xyZ")</code> renvoie <code>"fooxyZ"</code> .                                                                                      |
| <code>rtrim(str: string)</code><br><br><code>rtrim(str: string, trimChars: string)</code> | 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 <code>trimChars</code> à droite de <code>str</code> . Par exemple, <code>rtrim("xy ZfooxyxyZ", "xyZ")</code> renvoie <code>"xyZfoo"</code> . |

| Fonction                                                                            | Type de résultat | Description                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>trim(str: string)</code><br><code>trim(str: string, trimChars: string)</code> | 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 caractères de <code>trimChars</code> des deux côtés de <code>str</code> . Par exemple, <code>trim("xyZxyfooxyxyZ", "xyZ")</code> renvoie <code>"foo"</code> . |
| <code>strlen(str: string)</code>                                                    | number           | Renvoie la longueur de la chaîne en points de code Unicode.                                                                                                                                                                                                                                                                                                                                                       |
| <code>toupper(str: string)</code>                                                   | chaîne           | Convertit la chaîne en majuscules.                                                                                                                                                                                                                                                                                                                                                                                |
| <code>tolower(str: string)</code>                                                   | chaîne           | Convertit la chaîne en minuscules.                                                                                                                                                                                                                                                                                                                                                                                |

| Fonction                                                                                                              | Type de résultat | Description                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>substr(str: string, startIndex: number)</pre> <pre>substr(str: string, startIndex: number, length: number)</pre> | 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, <code>substr("xyZfooxyZ", 3, 3)</code> renvoie "foo".                                                                                                                       |
| <pre>replace(fieldName: string, searchValue: string, replaceValue: string)</pre>                                      | chaîne           | <p>Remplace toutes les instances de <code>searchValue</code> dans <code>fieldName: string</code> par <code>replaceValue</code>.</p> <p>Par exemple, la fonction <code>replace(logGroup, "smoke_test", "Smoke")</code> recherche les événements du journal où le champ <code>logGroup</code> contient la valeur de chaîne <code>smoke_test</code> et remplace la valeur par la chaîne <code>Smoke</code>.</p> |
| <pre>strcontains(str: string, searchValue: string)</pre>                                                              | number           | Renvoie 1 si <code>str</code> contient <code>searchValue</code> 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 antiochés (). ` 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 : didacticiels 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, consultez [Commencer 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, consultez [Classes 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'événements 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 événement 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 guillemets 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, consultez [CloudWatch 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 provisionedMemoryMB,
```



```
min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,  
avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,  
max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,  
provisionedMemoryMB - 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 hôtes Amazon EC2 qui ont été démarrés ou arrêtés dans une région donnée AWS .

```
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 d'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, consultez [Analyse 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, choisissez Journaux, Informations sur les journaux.

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.

L'inclusion `SOURCE` dans une requête PPL est un moyen utile de spécifier les groupes de journaux, les index de champs et les sources de données 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 ainsi que le nom et le type de source de données.

`aws:fieldIndex` 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. Les groupes de journaux pertinents sont automatiquement sélectionnés en fonction des champs spécifiés dans la `filterIndex` commande. Cela réduit le volume numérisé en ignorant les groupes de journaux qui ne contiennent aucun événement de journal contenant le champ spécifié dans la requête, et en analysant uniquement les groupes de journaux correspondant à la valeur spécifiée dans la requête pour cet index de champ. `aws:fieldIndex` À utiliser pour spécifier le nom du champ, ainsi que le nom et la valeur du champ dans la commande source afin d'interroger uniquement les données indexées contenant le champ et la valeur spécifiés. Pour de plus amples informations, consultez [Créez des index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation](#).

Vous pouvez utiliser OpenSearch PPL 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                 | <code>fields field1, field2</code>                                                                                                     | Affiche un ensemble de champs qui doivent être projetés.                              |
| joindre                | <code>LEFT JOIN left=l, right=r on l.id = r.id<br/>`join_right_lg`   fields l.field_1,<br/>r.field_2</code>                            | Joint deux ensembles de données.                                                      |
| où                     | <code>where field1="success"   where field2 !=<br/>"i-023fe0a90929d8822"   fields field3,<br/>field4, field5,field6   head 1000</code> | Filtre les données en fonction des conditions que vous spécifiez.                     |
| AWS : Index des champs | <code>source = [`aws:fieldIndex`="region",<br/>`region` = "us-west-2"]   where status =<br/>200   head 10</code>                       | Renvoie uniquement les données indexées, en forçant une requête à analyser uniquement |

| Commande ou fonction | Exemple de requête                                                                                                                            | Description                                                                                                                                                                                     |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      |                                                                                                                                               | les groupes de journaux indexés sur un champ que vous spécifiez dans la requête.                                                                                                                |
| stats                | <code>stats count(), count(field1), min(field1), max(field1), avg(field1) by field2   head 1000</code>                                        | Effectue des agrégations et des calculs                                                                                                                                                         |
| parse                | <code>parse field1 ".*/(?&lt;field2&gt;[^\r\n]+\$)"   where field2 = "requestId"   fields field1, field2   head 1000</code>                   | 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. |
| sort                 | <code>stats count(), count(field1), min(field1) as field1Alias, max(`field1`), avg(`field1`) by field2   sort -field1Alias   head 1000</code> | Triez les résultats affichés par nom de champ. Utilisez le tri - FieldName pour trier par ordre décroissant.                                                                                    |

| Commande ou fonction | Exemple de requête                                                    | Description                                                                                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eval                 | <pre>eval field2 = field1 * 2   fields field1, field2   head 20</pre> | 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             | <pre>rename field2 as field1   fields field1;</pre>                   | Renomme un ou plusieurs champs dans les résultats de recherche.                                                                                                                                                                       |
| head                 | <pre>fields `@message`   head 20</pre>                                | Limite les résultats de requête affichés aux N premières lignes.                                                                                                                                                                      |
| top                  | <pre>top 2 field1 by field2</pre>                                     | Recherche les valeurs les plus fréquentes pour un champ.                                                                                                                                                                              |

| Commande ou fonction | Exemple de requête                                                                                  | Description                                                                          |
|----------------------|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| dedup                | <code>dedup field1   fields field1, field2, field3</code>                                           | Supprime les entrées dupliquées en fonction des champs que vous spécifiez.           |
| rare                 | <code>rare field1 by field2</code>                                                                  | Recherche les valeurs les moins fréquentes de tous les champs de la liste de champs. |
| sous-requête         | <code>where field_1 IN [ search source=`subquery_lg`   fields field_2 ]   fields id, field_1</code> | Exécute des requêtes complexes et imbriquées dans vos instructions PPL.              |
| ligne de tendance    | <code>trendline sma(2, field1) as field1Alias</code>                                                | Calcule les moyennes mobiles des champs.                                             |

| Commande ou fonction        | Exemple de requête                                                                                                                                                                                               | Description                                                                                                                                                                                                                                                                           |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Statistiques de l'événement | <code>eventstats sum(field1) by field2</code>                                                                                                                                                                    | Enrichissez les données de vos événements grâce à des statistiques récapitulatives calculées. Il analyse les champs spécifiés au sein de vos événements, calcule diverses mesures statistiques, puis ajoute ces résultats à chaque événement d'origine sous forme de nouveaux champs. |
| expand                      | <code>eval tags_array_string = json_extract(`@message`, '\$.tags')  eval tags_array = json_array(json_extract(tags_string, '\$[0]'), json_extract(tags_string, '\$[1]'))  expand tags_array as color_tags</code> | Décompose un champ contenant plusieurs valeurs en lignes distinctes, en créant une nouvelle ligne pour chaque valeur du champ spécifié.                                                                                                                                               |

| Commande ou fonction | Exemple de requête                                                                                                                                                                                       | Description                                                                                                     |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| fillnull             | <pre>fields `@timestamp`, error_code, status_code   fillnull using status_code = "UNKNOWN", error_code = "UNKNOWN"</pre>                                                                                 | Remplit les champs nuls avec la valeur que vous fournissez. Il peut être utilisé dans un ou plusieurs domaines. |
| flatten              | <pre>eval metadata_struct = json_object('size', json_extract(metadata_string, '\$.size'), 'color', json_extract(metadata_string, '\$.color'))   flatten metadata_struct as (meta_size, meta_color)</pre> | Aplatit un champ. Le champ doit être de ce type : struct<?,?> ou array<struct<?,?>> .                           |
| cidrmatch            | <pre>where cidrmatch(ip, '2003:db8::/32')   fields ip</pre>                                                                                                                                              | Vérifie si l'adresse IP spécifiée se situe dans la plage CIDR donnée.                                           |
| résumé du champ      | <pre>where field1 != 200   fieldsummary includefields= field1 nulls=true</pre>                                                                                                                           | Calcule les statistiques de base pour chaque champ (nombre, nombre distinct, min, max, avg, stddev et moyenne). |



| Commande ou fonction | Exemple de requête                                                   | Description                                                                                                                                                                                                                          |
|----------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| grok                 | <pre>grok email '.*@%{HOSTNAME:host}'  <br/>fields email, host</pre> | Analyse un champ de texte à l'aide d'un motif grok et ajoute les résultats au résultat de recherche.                                                                                                                                 |
| Fonctions de chaîne  | <pre>eval field1Len = LENGTH(field1)   fields<br/>field1Len</pre>    | 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   | <pre>eval newDate = ADDDATE(DATE('2020-08-26'), 1)   fields newDate</pre> | Fonctions intégrées pour gérer et transformer les données de date et d'horodatage dans les requêtes PPL. Par exemple, <code>date_add</code> , <code>date_format</code> , <code>datediff</code> , <code>date_sub</code> , <code>timestamp add</code> , <code>timestamp diff</code> , <code>current_timezone</code> , <code>utc_timestamp</code> et <code>current_date</code> . |
| Fonctions de condition | <pre>eval field2 = isnull(field1)   fields field2, field1, field3</pre>   | 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 | <code>eval field2 = ACOS(field1)   fields field1</code> | Fonctions intégrées pour effectuer des calculs mathématiques et des transformations dans les requêtes PPL. Par exemple, <code>abs</code> (valeur absolue), <code>round</code> (arrondit les nombres), <code>sqrt</code> (racine carrée), <code>pow</code> (calcul de puissance) et <code>ceil</code> (arrondit au nombre entier le plus proche). |
| CryptoGraphic fonctions | <code>eval crypto = MD5(field)  head 1000</code>        | Pour calculer le hachage d'un champ donné                                                                                                                                                                                                                                                                                                        |

| Commande ou fonction | Exemple de requête                                                                     | Description                                                                                                                                                                                                                                                                                                     |
|----------------------|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fonctions JSON       | <pre>eval valid_json = json('[1,2,3,{"f1":1,"f2":[5,6]},4]')   fields valid_json</pre> | Fonctions intégrées pour gérer le JSON, notamment les tableaux, l'extraction et la validation. Par exemple, <code>json_object</code> , <code>json_array</code> , <code>to_json_string</code> , <code>json_array_length</code> , <code>json_extract</code> , <code>json_keys</code> et <code>json_valid</code> . |

## Étendue de la requête

L'inclusion de `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 ou pour créer une requête. La commande `SOURCE` n'est prise en charge que dans l'API AWS CLI et, 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 ainsi que le nom et le type de source de données.

La commande source de PPL permet désormais de les spécifier de plusieurs manières :

1. Groupe de journaux
2. Index des champs - Nouveau
3. Source et type de données - Nouveau

## Groupe de journaux

La sélection de la source du groupe de journaux peut être utilisée lorsque les clients savent exactement quel ou quels groupes de journaux doivent être recherchés

```
source = [lg:`/aws/lambda/my-function`] | where status = 200 | head 10
```

## Index des champs

La sélection de sources basée sur l'index des champs réduit la quantité de données demandées en limitant les résultats aux seules données indexées lorsque vos filtres ciblent les champs qui ont été indexés. Les groupes de journaux pertinents sont automatiquement sélectionnés en fonction des champs spécifiés dans la `filterIndex` commande. Pour plus d'informations sur les index de champs et sur la façon de les créer, voir [Création d'index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation](#).

`aws:fieldIndex` 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, voir [Création d'index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation](#).

En PPL, `aws:fieldIndex` est utilisé pour spécifier quelles paires clé-valeur doivent être traitées comme des index. La syntaxe est la suivante

```
source = [aws:fieldIndex="region", `region` = "us-west-2"] | where status = 200 | head 10
```

Où :

1. ``aws:fieldIndex`="region"` identifie la région en tant qu'index de champ.
  - a. Remarque : Au lieu de `=`, les clients peuvent utiliser `IN` pour spécifier plusieurs index (exemple ci-dessous)
2. ``region`="us-west-2"` identifie la condition de filtre à appliquer

- a. Remarque : Au lieu de =, les clients peuvent utiliser IN pour spécifier plusieurs valeurs (exemple ci-dessous)

Les clients peuvent spécifier plusieurs FieldIndexes comme suit

```
source = [`aws:fieldIndex` IN ("status", "region"), `status` = 200, `region` IN ("us-west-2", "us-east-1")] | head 10
```

## Source et type de données

La sélection basée sur la source de données et le type de source peut être utilisée lorsque les clients savent quelles sources de données exactes doivent être interrogées. Cette requête est exécutée sur un ou plusieurs groupes de journaux contenant la source de données et le type spécifiés.

```
source = [ds:`data_source.type`] | where status = 200 | head 10
```

## PPL pris en charge pour les requêtes de sources de données

Pour prendre en charge le cas d'utilisation de l'interrogation de sources de données dans PPL, vous pouvez utiliser la clause de sélection de source dynamique. À l'aide de cette syntaxe, vous pouvez interroger des sources de données en les spécifiant dans la commande de recherche. Vous pouvez spécifier jusqu'à 10 sources de données.

## Syntaxe

```
source=[ds:`DataSource1.Type1`, ds:`DataSource2.Type2`, ...ds:`DataSourceN.TypeN`]
```

## Exemple de requête

```
search source=[ds:`DataSource1.Type1`, ds:`DataSource2.Type2`] | fields field1, field2
```

## Exemple combiné

Les clients peuvent spécifier tous les opérateurs de sélection de source dans n'importe quel ordre et les résultats seront l'intersection de toutes les conditions appliquées.

Par exemple, `/aws/lambda/my-function-1` peut contenir plusieurs sources et types de données, y compris une grande variété d'index. Lorsque la requête suivante a été exécutée, les résultats renvoyés ne comporteront que des événements de source et de type `DataSource 1.Type1` répondant aux critères « `status` » = 200.

```
search source=[
  ds:`DataSource1.Type1`,
  lg:`/aws/lambda/my-function-1`,
  `aws:fieldIndex` IN ("status"), `status` = 200
]
```

## Restrictions

Les restrictions suivantes s'appliquent lorsque vous utilisez OpenSearch PPL pour effectuer des requêtes dans CloudWatch Logs Insights.

- Vous ne pouvez pas utiliser de commandes de jointure ou de sous-requête avec des requêtes de source de données.

## 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.

**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. Réduisez le volume numérisé en ignorant les groupes de journaux qui ne contiennent aucun événement de journal contenant le champ spécifié dans la requête, et en analysant uniquement les groupes de journaux correspondant à la valeur spécifiée dans la requête pour cet index de champ.

**filterIndex** À utiliser pour spécifier le nom du champ, ainsi que le nom et la valeur du champ pour interroger uniquement les données indexées contenant le champ et la valeur spécifiés.

Vous pouvez utiliser le OpenSearch SQL pour les requêtes de groupes de journaux dans la classe de journal standard. SQL prend également en charge les requêtes à l'aide du nom et du type de source de données.

**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                | <pre>SELECT * FROM `LogGroupA` WHERE Operation = 'x'</pre>                                   | Les filtres enregistrent les événements en fonction des critères de champ fournis.                                                                                     |
| Index des filtres    | <pre>SELECT * FROM `filterIndex('region' = 'us-east-1')` WHERE status = 200 LIMIT 10;</pre>  | Renvoie uniquement les données indexées, en forçant une requête à analyser uniquement les groupes de journaux indexés sur un champ que vous spécifiez dans la requête. |
| GROUP BY             | <pre>SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream`</pre> | Les groupes enregistrent les événements par catégorie et trouvent la moyenne en fonction des statistiques.                                                             |

| Commande ou fonction | Exemple de requête                                                                                                                | Description                                                                                                                                 |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| HAVING               | <pre>SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream` HAVING log_count &gt; 100</pre>            | Filtre les résultats en fonction des conditions de regroupement.                                                                            |
| ORDER BY             | <pre>SELECT * FROM `LogGroupA` ORDER BY `@timestamp` DESC</pre>                                                                   | Trie les résultats en fonction des champs de la clause ORDER BY. Vous pouvez trier par ordre décroissant ou croissant.                      |
| JOIN                 | <pre>SELECT A.`@message`, B.`@timestamp` FROM `LogGroupA` as A INNER JOIN `LogGroupB` as B ON A.`requestId` = B.`requestId`</pre> | Joint les résultats de deux tables en fonction de champs communs. La jointure interne ou la jointure externe gauche doivent être spécifiées |
| LIMIT                | <pre>Select * from `LogGroupA` limit 10</pre>                                                                                     | Limite les résultats de requête affichés aux N premières lignes.                                                                            |

| Commande ou fonction | Exemple de requête                                                                                                                                   | Description                                                                                                                                                                                                                                               |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fonctions de chaîne  | <pre>SELECT upper(Operation) , lower(Operation), Operation FROM `LogGroupA`</pre>                                                                    | Fonctions intégrées dans SQL permettant de manipuler et de transformer des chaînes de caractères et des données textuelles dans des requêtes SQL. Par exemple, convertir des majuscules, combiner des chaînes, extraire des parties et nettoyer du texte. |
| Fonctions de date    | <pre>SELECT current_date() as today, date_add(current_date(), 30) as thirty_days_later, last_day(current_date()) as month_end FROM `LogGroupA`</pre> | Fonctions intégrées pour gérer et transformer les données de date et d'horodatage dans les requêtes SQL. Par exemple, <code>date_add</code> , <code>date_format</code> , <code>datediff</code> et <code>current_date</code> .                             |

| Commande ou fonction      | Exemple de requête                                                                                | Description                                                                                                                                                            |
|---------------------------|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fonctions conditionnelles | <pre>SELECT Operation, IF(Error &gt; 0, 'High', 'Low') as error_category FROM `LogGroup A`;</pre> | Fonctions intégrées qui exécutent des actions en fonction de conditions spécifiées ou qui évaluent les expressions de manière conditionnelle. Par exemple, CASE et IF. |
| Fonctions d'agrégation    | <pre>SELECT AVG(bytes) as bytesWritten FROM `LogGroupA`</pre>                                     | Fonctions intégrées qui effectuent des calculs sur plusieurs lignes pour produire une seule valeur résumée. Par exemple, SUM, COUNT, AVG, MAX et MIN.                  |

| Commande ou fonction | Exemple de requête                                                                 | Description                                                                                                                                                                                                                                                                                                            |
|----------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fonctions JSON       | <pre>SELECT get_json_object(json_column, '\$.name') as name FROM `LogGroupA`</pre> | Fonctions intégrées pour analyser, extraire, modifier et interroger des données au format JSON dans des requêtes SQL (par exemple, <code>from_json</code> , <code>to_json</code> , <code>get_json_object</code> , <code>json_tuple</code> ) permettant de manipuler les structures JSON dans les ensembles de données. |

| Commande ou fonction | Exemple de requête                                                                                               | Description                                                                                                                                                                                                                                          |
|----------------------|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fonctions de tableau | <pre>SELECT scores, size(scores) as length,<br/>array_contains(scores, 90) as has_90<br/>FROM `LogGroupA`;</pre> | Fonctions intégrées permettant de travailler avec des colonnes de type tableau dans les requêtes SQL, permettant des opérations telles que l'accès, la modification et l'analyse de données de tableau (par exemple, size, explode, array_contains). |

| Commande ou fonction   | Exemple de requête                                                                                   | Description                                                                                                                                                                                                                                                         |
|------------------------|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fonctions de fenêtrage | <pre>SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) as field2Rank FROM `LogGroupA`;</pre> | Fonctions intégrées qui effectuent des calculs sur un ensemble spécifique de lignes liées à la ligne actuelle (fenêtre), permettant des opérations 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 | <pre>SELECT CAST('123' AS INT) as converted_number, CAST(123 AS STRING) as converted_string FROM `LogGroupA`</pre> | 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                      | <pre>SELECT scores, size(scores) as length,<br/>array_contains(scores, 90) as has_90<br/>FROM `LogGroupA`;</pre>                                          | 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. |
| Sélectionnez plusieurs groupes de journaux | <pre>SELECT lg1.field1, lg1.field2 from<br/>`logGroups( logGroupIdentifier:<br/>['LogGroup1', 'LogGroup2'])` as lg1<br/>where lg1.field3= "Success"</pre> | Vous permet de spécifier plusieurs groupes de journaux dans une instruction SELECT                                                                                                                  |
| Sélectionnez plusieurs sources de données  | <pre>SELECT ds1.field1, ds1.field2 from<br/>`dataSource(['DataSource1', 'DataSou<br/>ce2'])` as ds1 where ds1.field3=<br/>"Success"</pre>                 | Vous permet de spécifier plusieurs sources de données dans une instruction SELECT                                                                                                                   |

## SQL pris en charge pour les multi-log-group requêtes

Pour prendre en charge le cas d'utilisation permettant d'interroger 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'
```

## SQL pris en charge pour les requêtes de source de données

Pour faciliter l'utilisation des requêtes de sources de données dans SQL, vous pouvez utiliser la commande `DataSource`. À l'aide de cette syntaxe, vous pouvez interroger des sources de données en les spécifiant dans la `FROM` commande. Vous pouvez spécifier jusqu'à 10 sources de données.

Syntaxe

```
`dataSource(  
    ['DataSource1', 'DataSource2', ... 'DataSourcen']  
)`
```

## Exemple de requête

```
SELECT DS1.Column1, DS1.Column2 from `dataSource(  
    ['DataSource1', 'DataSource2']  
)` as DS1 WHERE DS1.Column1 = 'ABC'
```

## Étendue de la requête

Dans l'API AWS CLI and, vous pouvez spécifier les journaux à interroger en utilisant le groupe de journaux, la source et le type de données, ainsi que les index des champs.

### Groupe de journaux

La sélection de la source du groupe de journaux peut être utilisée lorsque les clients savent exactement quel ou quels groupes de journaux doivent être recherchés

```
SELECT * FROM `logGroups(logGroupIdentifier: ['/aws/lambda/my-function'])`;
```

### Source et type de données

Les clients peuvent interroger leurs journaux en utilisant le nom et le type de source de données.

La sélection basée sur la source de données et le type de source peut être utilisée lorsque les clients savent quelles sources de données exactes doivent être interrogées. Cette requête est exécutée sur un ou plusieurs groupes de journaux contenant la source de données et le type spécifiés.

Pour faciliter l'utilisation des requêtes de sources de données dans SQL, vous pouvez utiliser la commande DataSource. À l'aide de cette syntaxe, vous pouvez interroger des sources de données en les spécifiant dans la commande FROM. Vous pouvez spécifier jusqu'à 10 sources de données.

Syntaxe :

```
`dataSource(  
    ['DataSource1.Type1', 'DataSource2.Type2', ... 'DataSourcen.TypeN']  
)`
```

Exemple de requête :

```
SELECT DS1.Column1, DS1.Column2 from `dataSource(  
    ['DataSource1.Type1', 'DataSource2.Type2']
```

```
)` as DS1 WHERE DS1.Column1 = 'ABC'
```

Pour plus d'informations sur les requêtes par sources de données, consultez [Utilisez des facettes pour regrouper et explorer les journaux](#).

### Exemple combiné

Les clients peuvent spécifier tous les opérateurs de sélection de source dans les backticks dans n'importe quel ordre et les résultats seront basés sur l'intersection de toutes les conditions appliquées.

Par exemple, `/aws/lambda/my-function-1` peut contenir plusieurs sources et types de données, y compris une grande variété d'index. Lorsque la requête suivante a été exécutée, les résultats renvoyés ne comporteront que des événements de source et de type `DataSource 1.Type1` répondant aux critères « `status` » = 200.

```
SELECT * FROM `
  logGroups(logGroupIdentifier: ['/aws/lambda/my-function'])
  filterIndex('status' = 200)
  dataSource(['DataSource1.Type1'])
`;
```

### Index des champs

La sélection de source basée sur l'index des champs identifie automatiquement les groupes de journaux pertinents lorsque vos filtres ciblent les champs indexés, réduisant ainsi le volume de numérisation et le temps d'exécution des requêtes.

`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, voir [Création d'index de champs pour améliorer les performances des requêtes et réduire le volume de numérisation](#).

En SQL, `FilterIndex` est utilisé pour spécifier quelles paires clé-valeur doivent être traitées comme des index. La syntaxe est la suivante

```
SELECT * FROM `filterIndex('region' = 'us-east-1')`;
```

Où :

1. FilterIndex (...) indique de traiter les valeurs clés qu'elles contiennent comme des index de champs. Chaque paire clé-valeur est séparée par une virgule (exemple ci-dessous)
2. 'region' = 'us-east-1' indique la condition réelle à appliquer
  - a. Remarque : Au lieu de =, les clients peuvent utiliser IN pour spécifier plusieurs valeurs (exemple ci-dessous)

L'utilisation de plusieurs FilterIndex combinerait les conditions en utilisant « AND ». Dans l'exemple, les journaux correspondant à status=200 et à la région dans us-east-1 ou us-west-2 seraient interrogés.

```
SELECT * FROM `filterIndex('status' = 200, 'region' IN ['us-east-1', 'us-west-2'])`;
```

## 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.
- Vous ne pouvez pas utiliser JOIN ou des sous-requêtes avec des requêtes de source de données.
- 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 ne différant que dans certains cas (tels que 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 fonction effectue des appels interrégionaux vers des régions des États-Unis pour traiter les invites de requête. 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 [ReadOnlyAccess](#) IAM, 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
```

### Query

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
```



## Query

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
```

## Query

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, consultez [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, consultez [Classes 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, consultez [Créez 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          | <p>@timestamp , @logStream , @message, @requestId , @duration, @billedDuration , @type, @maxMemoryUsed , @memorySize</p> <p>Si une ligne de journal Lambda contient un ID de suivi X-Ray, elle inclut également les champs suivants : @xrayTraceId et @xraySegmentId .</p> <p>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 <b>parse</b>. Pour de plus amples informations, veuillez consulter <a href="#">Champs des journaux JSON</a>.</p> |
| CloudTrail journaux      | Pour plus d'informations, consultez <a href="#">Champs des journaux JSON</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Journaux au format 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 elle-même, 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 choisissiez 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 distinguent les majuscules et minuscules. 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 :

- `@logStream`
- `@aws.region`
- `@aws.account`



- @source.log
- @data\_source\_name
- @data\_source\_type
- @data\_format
- traceId
- severityText
- attributes.session.id

CloudWatch Logs fournit également des index de champs par défaut pour certaines combinaisons de noms et de types de sources de données. Les index de champs par défaut sont automatiquement disponibles pour les combinaisons de nom et de type de source de données suivantes :

| Nom et type de source de données | Index de champs par défaut                                             |
|----------------------------------|------------------------------------------------------------------------|
| amazon_vpc.flow                  | action<br><br>logStatus<br><br>region<br><br>flowDirection<br><br>type |
| amazon_route53_resolver_query    | query_type<br><br>transport<br><br>rcode                               |
| aws_waf.access                   | action<br><br>httpRequest.country                                      |
| aws_cloudtrail.data              | eventSource                                                            |
| aws_cloudtrail.management        | eventName<br><br>awsRegion                                             |

| Nom et type de source de données | Index de champs par défaut   |
|----------------------------------|------------------------------|
|                                  | <code>userAgent</code>       |
|                                  | <code>errorCode</code>       |
|                                  | <code>eventType</code>       |
|                                  | <code>managementEvent</code> |
|                                  | <code>readOnly</code>        |
|                                  | <code>eventCategory</code>   |
|                                  | <code>requestId</code>       |

Les index de champs par défaut s'ajoutent aux index de champs personnalisés que vous définissez dans 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](#) et [filtre](#).

## 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. De même, vous pouvez créer des politiques d'index au niveau du compte qui s'appliquent à une combinaison de nom et de type de source de données spécifique. Une seule politique de compte peut être créée par combinaison de nom et de type de source de données.

Les politiques d'index de champs au niveau du groupe de journaux remplacent les politiques d'index de champs au niveau du compte : elles s'appliquent au groupe de journaux dans son ensemble (par exemple, les politiques au niveau du compte sans critères de sélection ou avec des critères de sélection basés sur le préfixe du nom du groupe de journaux). Les politiques au niveau du compte qui correspondent au niveau des événements du journal (par exemple, pour une combinaison de nom et de type de source de données donnée) s'appliqueront en plus des politiques correspondant au groupe de journaux dans son ensemble. Si vous créez une politique d'index au niveau du groupe de journaux, ce groupe de journaux n'utilise pas de politiques au niveau du compte qui correspondent au niveau du groupe de journaux.

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 40 politiques d'index au niveau du compte, dont 20 peuvent utiliser des critères de sélection du préfixe du nom du groupe de journaux et 20 peuvent utiliser des critères de sélection basés sur les sources de données. 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`. De même, si plusieurs politiques d'index au niveau du compte sont filtrées en fonction des combinaisons de nom et de type de source de données, aucune d'entre elles ne peut utiliser le même nom et le même type de source de données. Par exemple, si vous avez une politique filtrée selon le nom `amazon_vpc` et le type de source de données de la source de données, `flow` vous ne pouvez pas créer une autre politique avec cette combinaison.

Si votre politique d'index au niveau du compte ne comporte aucun préfixe de nom et s'applique à tous les groupes de journaux, aucune autre politique d'index au niveau du compte avec des filtres de préfixes de nom de groupe de journaux ne peut être créée ; vous pouvez créer des politiques d'index au niveau du compte qui utilisent des filtres de nom et de type de source de données.

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.

Pour les politiques d'indexation au niveau du compte avec le nom de la source de données et les critères de sélection basés sur le type, une restriction supplémentaire s'applique : tous les champs doivent être des types de données primitifs, les primitives imbriquées ne sont prises en charge que pour les structures.

### Champs générés et champs réservés

CloudWatch Logs Insights génère automatiquement des champs système pour chaque événement du 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
- @xraySegmentId

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 :

- @logStream
- @aws.region
- @aws.account
- @source.log
- @data\_source\_name
- @data\_source\_type
- @data\_format
- traceId
- severityText
- attributes.session.id

CloudWatch Logs fournit également des index de champs par défaut pour certaines combinaisons de noms et de types de sources de données. Les index de champs par défaut sont automatiquement disponibles pour les combinaisons de nom et de type de source de données suivantes :

| Nom et type de source de données                                           | Index de champs par défaut                                                                                                                                                                                          |
|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>amazon_vpc.flow</code>                                               | <code>action</code><br><code>logStatus</code><br><code>region</code><br><code>flowDirection</code><br><code>type</code>                                                                                             |
| <code>amazon_route53.resolver_query</code>                                 | <code>query_type</code><br><code>transport</code><br><code>rcode</code>                                                                                                                                             |
| <code>aws_waf.access</code>                                                | <code>action</code><br><code>httpRequest.country</code>                                                                                                                                                             |
| <code>aws_cloudtrail.data</code><br><code>aws_cloudtrail.management</code> | <code>eventSource</code><br><code>eventName</code><br><code>awsRegion</code><br><code>userAgent</code><br><code>errorCode</code><br><code>eventType</code><br><code>managementEvent</code><br><code>readOnly</code> |

| Nom et type de source de données | Index de champs par défaut |
|----------------------------------|----------------------------|
|                                  | eventCategory              |
|                                  | requestId                  |

Les index de champs par défaut s'ajoutent aux index de champs personnalisés que vous définissez dans 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 le champ d'application de la politique, 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 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 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.

Choisissez Enregistrer les données par source de données pour appliquer la politique à une combinaison de nom et de type de source de données spécifique. Vous pouvez ensuite sélectionner la source de données et le type de données dans le menu déroulant.

Après avoir sélectionné le nom et le type de la source de données, vous pouvez sélectionner Obtenir des champs pour renseigner la section Configurer les index de champs et les facettes avec des informations pertinentes telles que les champs disponibles, les groupes de journaux inclus, ainsi que les index de champs par défaut et personnalisés.

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 ou sélectionnez un champ dans le menu déroulant. 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. (Facultatif) En plus de spécifier le chemin du champ, vous pouvez sélectionner Définir en tant que facette pour créer le champ en tant que facette.
9. Répétez l'étape précédente pour ajouter jusqu'à 20 index de champs.
10. 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. (Facultatif) En plus de spécifier le chemin du champ, vous pouvez sélectionner Définir en tant que facette pour créer le champ en tant que facette.
8. Répétez l'étape précédente pour ajouter jusqu'à 20 index de champs.
9. 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, choisissez Journaux, Informations sur les journaux.
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.

En utilisant le 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.

## Utilisez des facettes pour regrouper et explorer les journaux

Les facettes sont utiles pour analyser les journaux car elles vous permettent de filtrer et de regrouper vos données de manière interactive sans exécuter de requêtes. Une facette est un champ de vos journaux (tel que `ServiceName` ou `StatusCode`) qui permet le filtrage, l'agrégation et l'analyse entre les groupes de journaux. Vous pouvez consulter la liste des champs à facettes dans la console CloudWatch Logs Insights, ainsi que le nombre d'événements de journal pour chaque valeur de facette en fonction de la plage de temps que vous avez sélectionnée. Lorsque vous sélectionnez différentes facettes et valeurs, les valeurs et les nombres de facettes sont mis à jour en temps réel, ce qui vous permet d'explorer vos journaux de manière interactive.

Chaque facette affiche les valeurs disponibles et les nombres automatiquement extraits des champs de vos journaux en fonction de la plage de temps et de l'étendue de la requête sélectionnés, et conservés pendant 30 jours. Le nombre de facettes indiqué est approximatif. Vous pouvez utiliser les facettes par défaut telles que le nom ou le type de source de données pour explorer vos journaux,

ou créer des facettes personnalisées dans n'importe quel champ de vos journaux. Le nom de la source de données est un service ou une application AWS qui génère les journaux (par exemple, Route 53, Amazon VPC ou CloudTrail) et le type de source de données est le type spécifique de journal généré par ce service. Les facettes par défaut sont créées par CloudWatch et incluent `@aws.region@data_source_name,@data_source_type`, et `@data_format`. Pour de plus amples informations, veuillez consulter [Gestion des journaux](#). Les facettes ne sont disponibles que pour les journaux ingérés dans le compte. Si vous avez configuré l'observabilité entre comptes, le compte de surveillance ne peut pas afficher les facettes sur la base des journaux des comptes sources.

Pour créer des facettes supplémentaires, sélectionnez les champs de vos journaux qui sont pertinents pour votre résolution des problèmes et configurez-les à l'aide des politiques d'index. Pour les facettes personnalisées, nous vous recommandons de les créer sur des champs à faible cardinalité (champs contenant moins de 100 valeurs uniques par jour, tels que Status et ApplicationName). Les facettes contenant plus de 100 valeurs uniques par jour sont considérées comme présentant une cardinalité élevée et les valeurs de ces facettes ne sont pas affichées. Sélectionnez une ou plusieurs facettes et cliquez pour exécuter des requêtes dans vos journaux.

Pour commencer à utiliser les facettes dans CloudWatch Logs Insights :

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, choisissez Journaux, Informations sur les journaux.
3. (Facultatif) Utilisez le sélecteur de plage de temps pour sélectionner la période que vous souhaitez analyser. Pour la plage de temps sélectionnée, les facettes et valeurs disponibles sont affichées dans le panneau.
4. Sélectionnez des facettes pour explorer vos données et consulter les mises à jour en temps réel des distributions de valeur entre les facettes.

Les facettes comportant plus de 100 valeurs uniques ne sont pas affichées. Pour rechercher des valeurs spécifiques, utilisez plutôt des filtres dans votre requête.

## Pour exécuter une requête basée sur les facettes

1. Sélectionnez une ou plusieurs valeurs sur les facettes.
2. Le nombre d'événements sera mis à jour en fonction des facettes et des valeurs sélectionnées.
3. Au fur et à mesure que les valeurs des facettes sont sélectionnées, l'étendue de la requête est mise à jour pour refléter la sélection.

4. Après avoir sélectionné les valeurs des facettes, cliquez sur Exécuter pour exécuter votre requête.
5. Le nombre maximum de valeurs uniques prises en charge par facette est de 100. Par exemple, s'il existe plus de 100 valeurs pour une facette, tous les nombres sont affichés sous la forme « - », ce qui indique que les valeurs sont inconnues.

## Pour enregistrer une requête basée sur des facettes

1. Créez votre requête à l'aide d'une ou de plusieurs valeurs de facette.
2. Les autres étapes sont identiques à celles de l'enregistrement d'une requête Logs Insights. Voir les [requêtes Saving CloudWatch Logs Insights](#).
3. Vos requêtes enregistrées sont disponibles dans la section Requêtes enregistrées. Lorsque vous récupérez une requête enregistrée, elle inclut automatiquement les facettes et les valeurs utilisées pour la requête, ce qui facilite l'analyse de vos journaux.

## Pour créer une facette au niveau du compte

1. Pour créer des facettes, vous devez d'abord créer le champ sous forme d'index et le configurer en tant que facette. Dans le volet de navigation, sélectionnez Paramètres, Journaux, Politiques d'index au niveau du compte. Vous pouvez également sélectionner Gérer les facettes dans le panneau des facettes.
2. Choisissez Créer une nouvelle politique d'index. Pour plus de détails sur la création de politiques d'index, consultez [Création d'une politique d'indexation des champs au niveau du compte](#).
3. Pour créer une facette, cochez la case Définir comme facette pour le champ sélectionné sur la page de création de la politique d'index.

## Gestion des facettes à l'aide de APIs

La gestion des facettes peut être effectuée à l'aide de la politique d'index des champs. Consultez [field index](#) APIs pour plus de détails.

## Index des champs APIs

| Non. | Nom                 | Description                                                                                                                                                                                                                                                                                                               |
|------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | PutIndexPolicy      | Crée ou met à jour une politique d'index de champs pour le groupe de journaux spécifique                                                                                                                                                                                                                                  |
| 2    | PutAccountPolicy    | Crée une politique de protection des données au niveau du compte, une politique de filtrage des abonnements, une politique d'index de champs, une politique de transformation ou une politique d'extraction métrique qui s'applique à tous les groupes de journaux ou à un sous-ensemble de groupes de journaux du compte |
| 3    | DeleteIndexPolicy   | Supprime une politique d'index de champs au niveau du groupe de journaux qui a été appliquée à un seul groupe de journaux                                                                                                                                                                                                 |
| 4    | DeleteAccountPolicy | Supprime une politique de compte CloudWatch Logs                                                                                                                                                                                                                                                                          |

## 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, choisissez Journaux, Informations sur les journaux.



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* 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`.

 Note

Vous pouvez créer et enregistrer des requêtes avec des paramètres, c'est-à-dire des modèles réutilisables avec des espaces réservés nommés. Au lieu d'enregistrer plusieurs variantes de la même requête avec des valeurs différentes, créez un modèle et fournissez différentes valeurs de paramètres lorsque vous l'exécutez. Cette fonctionnalité est actuellement prise en charge uniquement pour les requêtes utilisant le langage de requête Logs Insights. Pour plus d'informations, consultez la section [Utilisation de requêtes enregistrées avec des paramètres](#).

## Console

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.
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. (Facultatif) Pour utiliser des paramètres dans votre requête, procédez comme suit :
  - a. Ajoutez des paramètres à votre requête. Remplacez les valeurs statiques par des espaces réservés en utilisant la `{{parameter}}` syntaxe (doubles accolades avant et après le nom du paramètre).

Exemple : requête originale avec des valeurs statiques :

```
fields @timestamp, @message
| filter level = "Error"
| filter applicationName = "OrderService"
```

Requête mise à jour avec paramètres :

```
fields @timestamp, @message
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
```

b. Définissez les paramètres utilisés dans votre requête. Pour chaque paramètre d'espace réservé, spécifiez :

- Nom : doit correspondre exactement au nom de l'espace réservé (par exemple, `logLevel`, `applicationName`).
- Valeur par défaut (facultatif) : valeur à utiliser si aucune valeur de paramètre n'est fournie.
- Description (facultatif) : explique l'objectif du paramètre.

c. Les requêtes avec paramètres peuvent être exécutées en utilisant le nom de la requête avec un \$ préfixe et en transmettant les noms des paramètres sous forme de paires clé-valeur.

Pour plus de détails, reportez-vous à la section [Pour exécuter une requête enregistrée](#).

9. Choisissez Enregistrer.

## AWS CLI

Pour enregistrer une requête, utilisez `put-query-definition` :

```
aws logs put-query-definition \
  --name "ErrorsByLevel" \
  --query-string "fields @timestamp, @message | filter level = \"ERROR\" \" \" \
  --log-group-names "/aws/lambda/my-function" \
  --region us-east-1
```

(Facultatif) Pour enregistrer une requête avec des paramètres, ajoutez l'option `--parameters` et utilisez des `{{parameterName}}` espaces réservés dans la chaîne de requête :

```
aws logs put-query-definition \
```

```
--name "ErrorsByLevel" \
--query-string "fields @timestamp, @message | filter level = {{logLevel}} | filter
applicationName = {{applicationName}}" \
--parameters '[{"name":"logLevel","defaultValue":"ERROR","description":"Log level
to filter"},
{"name":"applicationName","defaultValue":"OrderService","description":"Application
name to filter"}]' \
--log-group-names "/aws/lambda/my-function" \
--region us-east-1
```

Pour enregistrer une requête dans un dossier, préfixez le nom de la requête par le chemin du dossier :

```
aws logs put-query-definition \
--name "my-folder/ErrorsByLevel" \
--query-string "fields @timestamp, @message | filter level = {{logLevel}}" \
--parameters '[{"name":"logLevel","defaultValue":"ERROR","description":"Log level
to filter"}]' \
--log-group-names "/aws/lambda/my-function" \
--region us-east-1
```

## API

Pour enregistrer une requête, appelez [PutQueryDefinition](#):

```
{
  "name": "ErrorsByLevel",
  "queryString": "fields @timestamp, @message | filter level = \"ERROR\"",
  "logGroupNames": ["/aws/lambda/my-function"]
}
```

(Facultatif) Pour enregistrer une requête avec des paramètres, incluez le `parameters` champ et utilisez des `{{parameterName}}` espaces réservés dans la chaîne de requête :

```
{
  "name": "ErrorsByLevel",
  "queryString": "fields @timestamp, @message | filter level = {{logLevel}} | filter
applicationName = {{applicationName}}",
  "logGroupNames": ["/aws/lambda/my-function"],
  "parameters": [
    {
```

```
{
  "name": "logLevel",
  "defaultValue": "ERROR",
  "description": "Log level to filter"
},
{
  "name": "applicationName",
  "defaultValue": "OrderService",
  "description": "Application name to filter"
}
]
```

### 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](#).

## Console

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 des requêtes enregistrées. Le texte de la requête apparaît dans l'éditeur de requêtes.
5. (Facultatif) Pour utiliser une requête paramétrée :
  - a. Cliquez sur l'icône + à côté du nom de la requête dans le panneau latéral des requêtes enregistrées.
  - b. La requête paramétrée apparaît dans l'éditeur de requêtes. Par exemple, si vous choisissez l'icône + à côté `ErrorsByLevel`, l'éditeur de requêtes est renseigné avec :  
`$ErrorsByLevel(level=, applicationName=)`

- c. Fournissez les valeurs des paramètres (level, ApplicationName) et exécutez la requête. Par exemple : `$ErrorsByLevel(level= "ERROR", applicationName= "OrderService")`
6. Cliquez sur Exécuter.

## AWS CLI

Pour exécuter une requête enregistrée avec des paramètres

À utiliser `start-query` avec la `$QueryName()` syntaxe suivante :

```
aws logs start-query \  
  --log-group-names "/aws/lambda/my-function" \  
  --start-time 1707566400 --end-time 1707570000 \  
  --query-string '$ErrorsByLevel(level= "ERROR", applicationName= "OrderService")' \  
  --region us-east-1
```

## API

Pour exécuter une requête enregistrée avec des paramètres

Appelez [StartQuery](#) avec la `$QueryName()` syntaxe dans le `queryString` champ :

```
{  
  "logGroupNames": ["/aws/lambda/my-function"],  
  "startTime": 1707566400,  
  "endTime": 1707570000,  
  "queryString": "$ErrorsByLevel(level=\"ERROR\", applicationName= \"OrderService  
  \")\"  
}
```

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).

## Utilisation de requêtes enregistrées avec paramètres

Les requêtes enregistrées avec paramètres sont des modèles de requêtes réutilisables avec des espaces réservés nommés. Au lieu de conserver plusieurs copies de requêtes presque identiques, vous pouvez enregistrer un modèle et fournir différentes valeurs de paramètres lors de l'exécution de la requête. Les paramètres ne sont pris en charge que dans le langage de requête CloudWatch Logs Insights.

### Fonctionnement

Lorsque vous enregistrez une requête, les espaces réservés identifient les valeurs que vous pouvez fournir au moment de l'exécution de la requête. Les espaces réservés utilisent la `{{parameterName}}` syntaxe. Voici un exemple de requête enregistrée nommée `ErrorsByLevel` avec deux paramètres `logLevel` et `applicationName`.

```
fields @timestamp, @message
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
```

Pour exécuter une requête enregistrée, vous pouvez l'invoquer en utilisant le nom de la requête préfixé par `$` et en transmettant les valeurs des paramètres. Le moteur de requête CloudWatch Logs Insights remplace chaque espace réservé. Si un paramètre contient des valeurs par défaut, ces valeurs sont utilisées si aucune autre valeur n'est fournie.

```
# Run query by using query name and passing parameter values explicitly
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService")

# Run query without specifying parameter values - default values are used in this case.
$ErrorsByLevel()
```

Les noms de requêtes enregistrés contenant des espaces ou des caractères spéciaux doivent être suivis de backticks :

```
$`Errors By Level`(logLevel = "WARN")
```

## Exemples de requêtes enregistrées avec paramètres

Ajouter une limite de résultats en tant que paramètre

Nom de la requête : `ErrorsByLevel` avec paramètres `logLevel` (par défaut : `"ERROR"`), `applicationName` (par défaut : `"OrderService"`) et `maxResults` (par défaut : `50`)

```
fields @timestamp, @message, @logStream
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
| sort @timestamp desc
| limit {{maxResults}}
```

```
# Run the query using the query name and passing parameter values
```

```
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService", maxResults = 100)
```

## Utilisation de plusieurs requêtes enregistrées avec paramètres

L'exemple ci-dessous utilise `ErrorsByLevel` une seconde requête `RecentN` enregistrée définie comme `sort @timestamp desc | limit {{count}}` (avec paramètre `count`, par défaut 20). Le moteur de requêtes CloudWatch Logs Insights développe chaque requête avant de l'exécuter.

```
# Using multiple queries with parameters in sequence
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService")
| $RecentN(count = 10)

# Each of the queries is expanded, resulting in the following query when it is run.
fields @timestamp, @message
| filter level = "WARN"
| filter applicationName = "OrderService"
| sort @timestamp desc
| limit 10
```

## Quotas et gestion des erreurs

### Note

Chaque requête enregistrée peut comporter un maximum de 20 paramètres.

La chaîne de requête étendue ne peut pas dépasser 10 000 caractères. Les noms des paramètres doivent commencer par une lettre ou un trait de soulignement. Une requête enregistrée ne peut pas faire référence à une autre requête enregistrée (les invocations imbriquées ne sont pas prises en charge).

### Erreurs courantes

| Erreur                                                                    | Cause                                                                                          |
|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| Les paramètres ne sont pris en charge que pour le langage de requête CWLI | Les paramètres ne sont pris en charge que dans le langage de requête CloudWatch Logs Insights. |

| Erreur                                                | Cause                                                                                                                       |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| Paramètres obligatoires introuvables dans QueryString | Le nom d'un paramètre dans <code>--parameters</code> ne correspond pas <code>{{placeholder}}</code> à la chaîne de requête. |
| Le nombre de paramètres dépasse le maximum de 20      | Les requêtes enregistrées ne prennent actuellement en charge que 20 paramètres.                                             |
| Nom du paramètre dupliqué                             | La définition de requête contient des paramètres dupliqués <code>parameters</code> .                                        |

 Note

Pour créer ou mettre à jour une requête enregistrée avec des paramètres, vous devez disposer de cette `logs:PutQueryDefinition` autorisation. Pour en exécuter un, il vous faut `logs:StartQuery` et `logs:DescribeQueryDefinitions`.

## 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 100 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 100 requêtes CloudWatch Logs Insights simultanées par compte, y compris les requêtes ajoutées aux tableaux de bord. En outre, vous pouvez exécuter 15 requêtes simultanées pour OpenSearch Service PPL ou 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.

# Chiffrez les résultats des requêtes 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](#).

## Restrictions

- 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` à l'aide de la [get-key-policy](#) commande 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-policy](#) commande 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-identifiant "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-key](#) commande suivante :

```
aws logs disassociate-kms-key --resource-identifiant "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)                                   | US East (N. Virginia) Region  |
|                                                   | US East (Ohio) Region         |
|                                                   | US West (Oregon) Region       |
| Europe                                            | Région Europe (Frankfurt)     |
|                                                   | Europe (Ireland) Region       |
|                                                   | Région Europe (Paris)         |
|                                                   | Région Europe (Stockholm)     |

| Géographie des CloudWatch journaux pris en charge | Région de traitement possible                                                    |
|---------------------------------------------------|----------------------------------------------------------------------------------|
|                                                   | Région Europe (Londres)                                                          |
| Asie-Pacifique                                    | US East (N. Virginia) Region<br>US East (Ohio) Region<br>US West (Oregon) Region |
| Amérique du Sud                                   | US East (N. Virginia) Region<br>US East (Ohio) Region<br>US West (Oregon) Region |

## Prise en main

Pour générer un résumé en langage naturel

1. Exécutez votre requête CloudWatch Logs Insights.
2. Une fois la requête terminée, sélectionnez Résumer les résultats.

## Permissions

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.

# Automatisation de l'analyse des journaux avec des requêtes planifiées

Les requêtes planifiées vous permettent d'automatiser l'exécution des requêtes CloudWatch Logs Insights selon un calendrier régulier. Au lieu d'exécuter manuellement des requêtes pour analyser les données de vos journaux, vous pouvez configurer des requêtes planifiées pour qu'elles s'exécutent automatiquement et fournissent des résultats à des destinations telles que les buckets Amazon S3 ou les bus d' EventBridge événements Amazon. Cette automatisation est idéale pour générer des rapports réguliers, surveiller les tendances ou déclencher des processus en aval sur la base des résultats de l'analyse des journaux.

Les requêtes planifiées prennent en charge les trois langages de requête disponibles dans CloudWatch Logs Insights :

- [Langage de requête Logs Insights \(Logs Insights QL\)](#)
- [OpenSearch Langage de traitement par tuyauterie de service \(PPL\)](#)
- [OpenSearch Langage de requête structuré de service \(SQL\)](#)

## Table des matières

- [Comprendre les concepts des requêtes planifiées](#)
- [Référence d'expression de planification](#)
- [Bonnes pratiques](#)
- [Commencer à utiliser les requêtes planifiées](#)
- [Configuration des destinations S3 pour les requêtes planifiées](#)
- [Résolution des problèmes de requêtes planifiées](#)

## Comprendre les concepts des requêtes planifiées

Avant de créer des requêtes planifiées, comprenez ces concepts clés qui affectent le mode d'exécution de vos requêtes et l'endroit où les résultats sont fournis.

## Séparation des rôles IAM

Les requêtes planifiées nécessitent deux rôles IAM distincts : l'un pour exécuter les requêtes et l'autre pour fournir des résultats à Amazon S3 ou aux bus d' EventBridge événements Amazon. Comprendre pourquoi cette séparation existe vous aide à configurer correctement les autorisations et à tirer parti des avantages opérationnels et de sécurité qu'elle apporte.

L'architecture à deux rôles répartit les responsabilités entre l'accès aux données et leur livraison. Le rôle d'exécution des requêtes accède aux données de votre journal et exécute des requêtes, tandis que le rôle de livraison de destination écrit les résultats vers la destination que vous avez choisie. Cette séparation suit le principe du moindre privilège : chaque rôle ne dispose que des autorisations dont il a besoin pour sa fonction spécifique.

### Rôle d'exécution des requêtes

Permet à CloudWatch Logs d'exécuter CloudWatch des requêtes Logs Insights en votre nom. Ce rôle a besoin d'autorisations pour accéder à vos groupes de journaux et exécuter des requêtes, mais il n'a pas besoin d'accéder aux ressources de destination. Autorisations requises :

- `logs:StartQuery`
- `logs:StopQuery`
- `logs:GetQueryResults`
- `logs:DescribeLogGroups`
- `logs:Unmask` si des données de démasquage sont requises

Pour les groupes de journaux chiffrés par KMS : `kms:Decrypt` et `kms:DescribeKey` autorisations pour la clé KMS utilisée pour chiffrer les groupes de journaux. Ces autorisations doivent également être ajoutées.

Exigence de relation de confiance : le rôle d'exécution des requêtes doit inclure une politique de confiance qui permet au service CloudWatch Logs (`logs.amazonaws.com`) d'assumer ce rôle. Sans cette relation de confiance, les requêtes planifiées échoueront avec des erreurs d'autorisation.

Exemple de politique de confiance pour le rôle d'exécution de requêtes :

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

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "logs.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
```

Exemple de politique d'autorisation pour le rôle d'exécution de requêtes :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:GetQueryResults",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

## Rôle de livraison à destination

Permet à CloudWatch Logs de fournir les résultats des requêtes à la destination que vous avez choisie. Ce rôle ne nécessite des autorisations que pour le service de destination spécifique, conformément au principe du moindre privilège. Les autorisations requises varient selon le type de destination.

Exigence de relation de confiance : le rôle de livraison de destination doit également inclure une politique de confiance permettant au service CloudWatch Logs (`logs.amazonaws.com`) d'assumer ce rôle.

Exemple de politique d'autorisation pour le rôle de livraison de destination S3 :

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": "arn:aws:s3:::your-scheduled-query-results-bucket/*"
  }
]
```

Cette séparation offre des avantages pratiques pour vos opérations. Du point de vue de la sécurité, si vous devez modifier l'endroit où les résultats sont fournis, vous ne modifiez que le rôle de livraison de destination sans modifier les autorisations d'exécution des requêtes. À des fins de conformité et d'audit, vous pouvez clairement savoir quel rôle accède aux données de journal sensibles et quel rôle écrit sur des systèmes externes. Cela permet de démontrer plus facilement que votre infrastructure d'analyse des journaux suit les meilleures pratiques en matière de sécurité.

## Utilisation entre régions et entre comptes

Une requête planifiée est créée dans une région spécifique et s'exécute dans cette région. Cependant, vous pouvez interroger des groupes de journaux et fournir des résultats par région et par compte. Vous devez configurer un ou plusieurs AWS comptes en tant que comptes de surveillance et les associer à plusieurs comptes sources. Un compte de surveillance est un AWS compte central qui permet de consulter et d'interagir avec les données d'observabilité générées à partir des comptes sources. Un compte source est un AWS compte individuel qui génère des données d'observabilité pour les ressources qu'il contient. Les comptes sources partagent leurs données d'observabilité avec le compte de surveillance. Vous pouvez donc configurer des requêtes planifiées à partir du compte de surveillance à l'aide des groupes de journaux de tous les comptes associés.

### Interrogation de groupes de journaux interrégionaux

Votre requête planifiée peut accéder à des groupes de journaux dans n'importe quelle région. Spécifiez les groupes de journaux en utilisant leur format ARN complet :`arn:aws:logs:region:account-id:log-group:log-group-name`. Le rôle d'exécution des requêtes nécessite `logs:StartQuery` et `logs:GetQueryResults` autorise les groupes de journaux dans toutes les régions cibles.



### Important

Lorsque vous interrogez des groupes de journaux ou que vous fournissez des résultats entre régions, les données du journal dépassent les frontières régionales. Éléments à prendre en compte :

- Exigences relatives à la résidence des données : assurez-vous que le transfert de données entre régions est conforme aux politiques de gouvernance des données et aux exigences réglementaires de votre organisation
- Coûts de transfert de données - Le transfert de données entre régions entraîne des frais supplémentaires
- Latence du réseau : les requêtes accédant à des groupes de journaux situés dans des régions éloignées peuvent connaître une latence plus élevée

Pour des performances et une rentabilité optimales, créez des requêtes planifiées dans la même région que vos principaux groupes de journaux.

Approche alternative : utilisez [la centralisation CloudWatch des journaux](#) pour répliquer les données des journaux de plusieurs comptes et régions vers un compte de surveillance central. Cela vous permet de créer des requêtes planifiées dans une seule région qui accèdent à tous vos journaux centralisés, d'éviter les requêtes interrégionales et de simplifier la gestion des autorisations IAM.

## Expressions de planification et gestion des fuseaux horaires

Le calendrier que vous définissez détermine le moment et la fréquence d'exécution de votre requête. Le choix de la bonne expression de planification influe sur le moment où vous recevez les résultats et sur la quantité de données que vous interrogez. La compréhension des types d'expressions vous permet de choisir entre simplicité et précision.

Les expressions Cron fournissent un contrôle précis du calendrier, vous permettant de spécifier les heures, les jours de la semaine ou les jours du mois exacts. Utilisez des expressions cron lorsque vous devez exécuter des requêtes à des heures ouvrées spécifiques ou s'aligner sur les plannings opérationnels. Dans la console, vous pouvez également planifier des requêtes à l'aide d'options de calendrier simples.

## Expressions Cron

Exécutez des requêtes à des moments précis. Format :`cron(minute hour day-of-month month day-of-week year)`. Exemples :

- `cron(0 9 * * ? *)`- Tous les jours à 9h00 UTC
- `cron(0 18 ? * MON-FRI *)`- En semaine à 18h00 UTC
- `cron(0 0 1 * ? *)`- Le premier jour de chaque mois à minuit UTC
- `cron(0 12 ? * SUN *)`- Tous les dimanches à midi UTC
- `cron(30 8 1 1 ? *)`- Le 1er janvier à 8 h 30 UTC

Toutes les requêtes planifiées sont exécutées en UTC, quel que soit votre fuseau horaire local ou l'emplacement de vos AWS ressources. Cela est particulièrement important lorsque vous planifiez des requêtes pour les heures ouvrables ou pour des analyses urgentes. Par exemple, si votre entreprise exerce ses activités à l'heure de l'Est des États-Unis et que vous souhaitez obtenir un rapport quotidien à 9 h 00 ET, vous devez tenir compte du décalage UTC (14 h UTC pendant l'heure d'été, 13 h 00 UTC dans le cas contraire). Planifiez vos expressions de planification en tenant compte de l'UTC afin de garantir que les requêtes soient exécutées aux heures prévues.

## Choix d'une langue de requête

Les requêtes planifiées prennent en charge trois langages de requête différents, et votre choix influe à la fois sur la façon dont vous rédigez les requêtes et sur la facilité avec laquelle votre équipe peut les gérer. Le langage approprié dépend de vos exigences en matière d'analyse et des compétences existantes de votre équipe.

Si vous filtrez et agrégez principalement des données de journal, le langage de requête CloudWatch Logs Insights propose la syntaxe la plus simple. Pour les transformations de données complexes où vous devez remodeler ou enrichir les données en plusieurs étapes, l'approche de pipeline de PPL facilite le suivi de la logique. Lorsque vous devez effectuer des jointures ou des agrégations complexes similaires à des opérations de base de données, SQL fournit une syntaxe familière que les équipes expérimentées peuvent adopter rapidement.

### CloudWatch Langage de requête Logs Insights (CWLI)

Spécialement conçu pour l'analyse des journaux avec une syntaxe intuitive. Idéal pour :

- Analyse et filtrage des journaux basés sur le texte

- Agrégations de séries chronologiques et statistiques
- Équipes novices en matière d'analyse des logs

### OpenSearch Langage de traitement par tuyaux de service (PPL)

Langage de requête basé sur un pipeline doté de puissantes fonctionnalités de transformation des données. Idéal pour :

- Transformations et enrichissement complexes des données
- Flux de travail de traitement des données en plusieurs étapes
- Équipes familiarisées avec le traitement basé sur les pipelines

### OpenSearch Langage de requête structuré de service (SQL)

Syntaxe SQL standard pour les requêtes habituelles de type base de données. Idéal pour :

- Jointures et agrégations complexes
- Veille économique et production de rapports
- Des équipes ayant une solide expérience SQL

## Sélection de destinations et cas d'utilisation

L'endroit où vous envoyez les résultats de la requête détermine ce que vous pouvez en faire. Ce choix façonne l'ensemble de votre flux de travail en aval, qu'il s'agisse de créer des analyses à long terme, de déclencher des réponses automatisées, ou les deux. Comprendre les points forts de chaque type de destination vous aide à concevoir l'architecture adaptée à votre cas d'utilisation.

Les destinations Amazon S3 sont optimisées pour le stockage et le traitement par lots. Lorsque vous devez conserver les résultats de vos requêtes pendant des mois ou des années, analyser les tendances au fil du temps ou alimenter des plateformes d'analyse, Amazon S3 fournit un stockage rentable avec une rétention illimitée. EventBridge les destinations sont optimisées pour une automatisation en temps réel. Lorsque les résultats d'une requête doivent déclencher des actions immédiates, comme l'envoi d'alertes, le démarrage de flux de travail ou la mise à jour EventBridge de systèmes, les résultats sont des événements auxquels vos applications peuvent réagir instantanément. Par défaut, tous les événements de fin de requête sont automatiquement envoyés sous forme d'événements au bus d'événements par défaut, ce qui permet l'intégration avec les systèmes de traitement en aval, les fonctions Lambda ou d'autres architectures pilotées par les événements. Les résultats ne sont publiés vers les destinations que lorsque la requête est exécutée avec succès.

## Destinations Amazon S3

Stockez les résultats des requêtes sous forme de fichiers JSON pour une conservation à long terme et un traitement par lots. Idéal pour :

- Analyse historique et archivage des données
- Intégration aux lacs de données et aux plateformes d'analyse
- Exigences en matière de conformité et d'audit
- Stockage rentable de grands ensembles de résultats

## EventBridge destinations

Envoyez les résultats des requêtes sous forme d'événements pour un traitement et une automatisation en temps réel. Vous pouvez récupérer les résultats de la requête en utilisant le QueryID envoyé dans l'événement pendant 30 jours uniquement, car nous conservons les résultats pendant 30 jours. Idéal pour :

- Déclenchement de réponses automatisées aux résultats des requêtes
- Intégration aux flux de travail sans serveur et aux fonctions Lambda
- Systèmes d'alerte et de notification en temps réel
- Architectures et microservices pilotés par les événements

## Format et structure des résultats de requête

Pour les destinations Amazon S3, les résultats des requêtes sont fournis au format JSON avec la même structure que la réponse de l' `GetQueryResults` API. Pour Amazon, EventBridge comprendre le format des résultats des requêtes planifiées vous permet de concevoir des flux de travail de traitement et d'intégration en aval.

Les résultats des requêtes sont fournis au format JSON avec la structure suivante :

```
{
  "version": "0",
  "id": "be72061b-eca2-e068-a7e1-83e01d6fe807",
  "detail-type": "Scheduled Query Completed",
  "source": "aws.logs",
  "account": "123456789012",
  "time": "2025-11-18T11:31:48Z",
  "region": "us-east-1",
```

```
"resources": [
  "arn:aws:logs:us-east-1:123456789012:scheduled-query:477b4380-b098-474e-9c5e-
e10a8cc2e6e7"
],
"detail": {
  "queryId": "2038fd57-ab4f-4018-bb2f-61d363f4a004",
  "queryString": "fields @timestamp, @message, @logStream, @log\n| sort
@timestamp desc\n| limit 10000",
  "logGroupIdentifiers": [
    "/aws/lambda/my-function"
  ],
  "status": "Complete",
  "startTime": 1763465460,
  "statistics": {
    "recordsMatched": 0,
    "recordsScanned": 0,
    "estimatedRecordsSkipped": 0,
    "bytesScanned": 0,
    "estimatedBytesSkipped": 0,
    "logGroupsScanned": 1
  }
}
```

Les principaux éléments sont les suivants :

- **statistics**- Indicateurs de performance des requêtes, notamment les enregistrements mis en correspondance, les enregistrements numérisés, les octets traités et l'estimation des données ignorées
- **startTime**- Quand l'exécution de la requête a commencé (horodatage Unix)
- **queryString**- La requête réelle qui a été exécutée
- **queryId**- ID de requête de la requête à l'aide duquel les résultats peuvent être récupérés
- **logGroupIdentifiers**- Liste des groupes de journaux interrogés
- **status**- État d'exécution de la requête (terminé, échec, etc.)

## Référence d'expression de planification

Utilisez ces tables de référence pour créer des expressions de planification pour vos requêtes planifiées. Toutes les heures sont indiquées en heure universelle coordonnée (UTC).

## Syntaxe des expressions Cron

Format : `cron(minute hour day-of-month month day-of-week year)`

| Cas d'utilisation      | Expression cron                            | Description                                               | À utiliser quand                         |
|------------------------|--------------------------------------------|-----------------------------------------------------------|------------------------------------------|
| Horaires quotidiens    | <code>cron(0 9 * * ? *)</code>             | Tous les jours à 9h00 UTC                                 | Reportages quotidiens                    |
|                        | <code>cron(0 */6 * * ? *)</code>           | Toutes les 6 heures (00:00, 06:00, 12:00, 18:00 UTC)      | Surveillance fréquente                   |
|                        | <code>cron(30 2 * * ? *)</code>            | Tous les jours à 2 h 30 UTC                               | Analyse en dehors des périodes de pointe |
| Heures d'ouverture     | <code>cron(0 9-17 ? * MON-FRI *)</code>    | Toutes les heures de 9 h à 17 h, du lundi au vendredi UTC | Surveillance des activités               |
|                        | <code>cron(0 18 ? * MON-FRI *)</code>      | En semaine à 18h00 UTC                                    | Fin du jour ouvrable                     |
|                        | <code>cron(0 8,12,17 ? * MON-FRI *)</code> | 8 h, midi et 17 h en semaine UTC                          | Moments clés de l'activité               |
| Horaires hebdomadaires | <code>cron(0 12 ? * SUN *)</code>          | Tous les dimanches à midi UTC                             | Résumés hebdomadaires                    |
|                        | <code>cron(0 9 ? * MON *)</code>           | Tous les lundis à 9h00 UTC                                | Rapports de                              |

| Cas d'utilisation | Expression cron                        | Description                                    | À utiliser quand               |
|-------------------|----------------------------------------|------------------------------------------------|--------------------------------|
|                   |                                        |                                                | début de semaine               |
|                   | <code>cron(0 23 ? * FRI *)</code>      | Tous les vendredis à 23 h 00 UTC               | Nettoyage de fin de semaine    |
| Horaires mensuels | <code>cron(0 0 1 * ? *)</code>         | Le premier jour de chaque mois à minuit UTC    | Rapports mensuels              |
|                   | <code>cron(0 9 L * ? *)</code>         | Le dernier jour de chaque mois à 9 h 00 UTC    | Traitement de fin de mois      |
|                   | <code>cron(0 10 1 1,4,7,10 ? *)</code> | Premier jour de chaque trimestre à 10 h 00 UTC | Analyse trimestrielle          |
| Haute fréquence   | <code>cron(* /15 * * * ? *)</code>     | Toutes les 15 minutes                          | Surveillance en temps réel     |
|                   | <code>cron(0,30 * * * ? *)</code>      | Toutes les 30 minutes (à 00 h et 30 h 30)      | Contrôles fréquents            |
|                   | <code>cron(0 */2 * * * ? *)</code>     | Toutes les 2 heures                            | Intervalles réguliers          |
| Cas spéciaux      | <code>cron(30 8 1 1 ? *)</code>        | Le 1er janvier à 8 h 30 UTC                    | Rapports annuels               |
|                   | <code>cron(0 6 * * SAT,SUN *)</code>   | Week-ends à 6 h 00 UTC                         | Traitement des fins de semaine |

| Cas d'utilisation | Expression cron                    | Description                                  | À utiliser quand        |
|-------------------|------------------------------------|----------------------------------------------|-------------------------|
|                   | <code>cron(0 0 ? * MON#1 *)</code> | Le premier lundi de chaque mois à minuit UTC | Planification mensuelle |

## Référence du champ d'expression Cron

| Champ             | Valeurs         | Caractères génériques | Exemples                                                                         |
|-------------------|-----------------|-----------------------|----------------------------------------------------------------------------------|
| Minute (1re)      | 0-59            | * , - /               | 0(en haut de l'heure), */15 (toutes les 15 minutes), 0, 30 (deux fois par heure) |
| Heure (2e)        | 0-23            | * , - /               | 9(9 h), */2 (toutes les 2 heures), 9-17 (heures de bureau)                       |
| Day-of-month (3e) | 1-31, L, W      | * , - / ?             | 1(1er jour), L (dernier jour), ? (lors de l'utilisation day-of-week)             |
| Mois (4e)         | 1-12 ou JAN-DEC | * , - /               | 1(janvier)JAN, 1, 4, 7, 10 (trimestriel)                                         |
| Day-of-week (5e)  | 1-7 ou DIM-SAM  | * , - / ? # L         | MON-FRI(en semaine)SUN, MON#1 (1er lundi)                                        |
| Année (6e)        | 1970-2199       | * , - /               | *(chaque année), 2024 (année spécifique), 2024-2026 (gamme)                      |

## Caractères génériques et expressions spéciales



**\* (astérisque)**

Correspond à toutes les valeurs du champ. Exemple : \* dans le champ des heures, cela signifie toutes les heures.

**?(point d'interrogation)**

Aucune valeur spécifique. À utiliser dans day-of-month ou day-of-week lorsque l'autre est spécifié. Exemple : utilisez ? in day-of-month lorsque vous spécifiez MON-FRI dans day-of-week.

**- (tiret)**

Plage de valeurs. Exemple : MON-FRI (du lundi au vendredi), 9-17 (de 9 h à 17 h).

**, (virgule)**

Plusieurs valeurs spécifiques. Exemple : MON, WED, FRI (lundi, mercredi, vendredi), 8, 12, 17 (8 h, midi, 17 h).

**/ (barre oblique)**

Valeurs d'étape ou incréments. Exemple : 0/15 en minutes signifie toutes les 15 minutes à partir de la minute 0 (0, 15, 30, 45). \*/2 en heures signifie toutes les 2 heures.

**L (dernier)**

Dernier jour du mois ou dernière occurrence du jour de la semaine. Exemple : L in day-of-month signifie le dernier jour du mois. FRIL signifie le dernier vendredi du mois.

**W (jour de semaine)**

Le jour de semaine le plus proche. Exemple : 15W signifie le jour de la semaine le plus proche du 15 du mois.

**# (nième occurrence)**

Neuvième occurrence d'un jour de semaine en un mois. Exemple : MON#1 signifie le premier lundi du mois, FRI#2 le deuxième vendredi du mois.

**Modèles courants et meilleures pratiques**

- Pour les applications professionnelles : utilisation MON-FRI et heures de bureau (par exemple 9-17) pour éviter d'exécuter des requêtes le week-end ou en dehors des heures de bureau.

- Pour la surveillance à haute fréquence : utilisez des incréments \*/15 (toutes les 15 minutes, par exemple), mais tenez compte des limites de simultanéité des requêtes.
- Pour une utilisation efficace des ressources : planifiez des requêtes gourmandes en ressources pendant les heures creuses, en utilisant des heures tôt le matin, comme UTC. 2-6
- Pour les rapports mensuels : à utiliser L pour le dernier jour du mois ou 1 pour des dates spécifiques, comme le premier jour, afin de garantir un calendrier cohérent.

## Bonnes pratiques

Suivez ces bonnes pratiques pour garantir la fiabilité et l'efficacité des opérations de requête planifiées :

### Optimisation des requêtes

- Testez les requêtes manuellement avant de les planifier pour vérifier les performances et les résultats
- Utilisez des index de filtre au début de votre requête pour réduire le traitement des données
- Limitez les plages de temps pour éviter les délais d'expiration liés aux groupes de journaux à volume élevé
- Tenez compte de la complexité des requêtes et des délais d'exécution

### Planification des horaires

- Évitez les chevauchements d'exécution en veillant à ce que les requêtes soient terminées avant la prochaine exécution planifiée
- Tenez compte des délais d'ingestion des journaux lors de la définition des plages de temps
- Utilisez des expressions cron pour des moments spécifiques
- Répartissez les plannings pour vous assurer de ne pas atteindre la limite de simultanéité des requêtes

### Surveillance et maintenance

- Surveillez régulièrement l'historique des exécutions pour identifier les défaillances ou les problèmes de performance
- Passez en revue et mettez à jour régulièrement les rôles IAM pour garantir la sécurité
- Testez l'accessibilité des destinations avant le déploiement en production

## Autorisation

- Toutes les autorisations APIs pour les requêtes planifiées portent sur la ressource de requête planifiée et non sur les ressources qu'elle utilise en entrée, comme les groupes de journaux. Configurez les politiques IAM en conséquence
- Gérez l'autorisation des groupes de journaux à l'aide du rôle d'exécution transmis dans APIs

## Commencer à utiliser les requêtes planifiées

Lorsque vous créez une requête planifiée, vous configurez plusieurs composants clés qui définissent le mode d'exécution de votre requête et l'endroit où les résultats sont fournis. La compréhension de ces composants vous aidera à configurer une analyse automatique efficace des journaux.

Chaque requête planifiée comprend les éléments clés suivants :

### Configuration des requêtes

La chaîne de requête CloudWatch Logs Insights, les groupes de journaux cibles et le langage de requête à utiliser pour l'analyse.

### Expression de planification

Expression cron ou calendrier de fréquence qui définit le moment où la requête est exécutée. Vous pouvez définir des paramètres de fuseau horaire pour garantir que les requêtes sont exécutées à l'heure locale correcte. La console affiche une description lisible par l'homme de votre emploi du temps, telle que « Exécutez la requête tous les mardis à 15h10 pendant une période de 5 minutes, avec effet immédiat, en UTC, jusqu'à une durée indéfinie ».

### Plage horaire

Période rétrospective pour chaque exécution de requête, définie par un décalage entre l'heure de début et l'heure d'exécution. Cela détermine la quantité de données historiques que chaque exécution de requête analysera.

### Aperçu du calendrier d'exécution

La console affiche les trois prochaines exécutions de requêtes planifiées avec les dates et heures exactes (par exemple, 2025/10/28 15:10, UTC ; 2025/11/04 15:10, UTC ; 2025/11/11 15:10, UTC), ce qui vous aide à vérifier que le calendrier est correctement configuré.

## Destinations

Où les résultats de la requête sont fournis après une exécution réussie. Les destinations prises en charge incluent les compartiments Amazon S3 et, par défaut, les métadonnées des résultats sont envoyées au bus d'événements par défaut.

## Rôle d'exécution

Rôle IAM que CloudWatch Logs suppose d'exécuter la requête et de fournir les résultats aux destinations spécifiées.

Avant de créer des requêtes planifiées, assurez-vous que les autorisations et les ressources nécessaires sont configurées.

## Création d'une requête planifiée

Créez une requête planifiée qui exécute automatiquement les requêtes CloudWatch Logs Insights et fournit les résultats aux destinations que vous avez choisies.

## Conditions préalables

Avant de créer une requête planifiée, assurez-vous de disposer des éléments suivants :

- Groupes de journaux : un ou plusieurs groupes de journaux contenant les données que vous souhaitez analyser
- Rôle IAM d'exécution : rôle IAM doté des autorisations suivantes :
  - `logs:StartQuery`- Autorisation de lancer CloudWatch des requêtes Logs Insights
  - `logs:GetQueryResults`- Autorisation de récupérer les résultats de la requête
  - `logs:DescribeLogGroups`- Autorisation d'accéder aux informations du groupe de journaux. Cela n'est requis que pour les groupes de journaux basés sur des préfixes pour la découverte des groupes de journaux
- Autorisations de destination : autorisations IAM supplémentaires pour la destination que vous avez choisie :
  - Pour les destinations Amazon S3 : `s3:PutObject`
- Pour AWS CLI et utilisation de l'API - AWS Informations d'identification configurées avec autorisations pour appeler CloudWatch les journaux APIs

Pour des exemples de politique IAM détaillés, voir [Gestion des identités et des accès pour Amazon CloudWatch Logs](#). Il convient également de noter que vous ne pouvez avoir que 1000 requêtes planifiées par compte.

## Console

Pour créer une requête planifiée (console)

1. Vous voulez ouvrir la console CloudWatch Logs à [https://us-east-1.console.aws.amazon.com/cloudwatch/la\\_maison?region=us-east-1#LogsV2:Logs-Insights](https://us-east-1.console.aws.amazon.com/cloudwatch/la_maison?region=us-east-1#LogsV2:Logs-Insights).
2. Dans le volet de navigation, sélectionnez Logs Insights.
3. Choisissez Créer une requête planifiée.
4. Dans la section Définition de la requête :
  - a. Pour Langue de requête, choisissez la langue de requête à utiliser dans la liste.
  - b. Pour la chaîne de requête, entrez votre requête CloudWatch Logs Insights dans le champ.
  - c. Pour les groupes de journaux, sélectionnez les groupes de journaux à interroger dans la liste.
5. Dans la section Configuration du calendrier :
  - a. Pour l'expression Schedule, configurez le moment où la requête s'exécute. Choisissez parmi les options prédéfinies ou entrez une expression cron personnalisée.
  - b. Pour Effectif à la création, spécifiez à quel moment le calendrier devient actif. Choisissez de commencer immédiatement ou à une date et à une heure spécifiques en utilisant le YYYY/MM/DD format.
  - c. Pour Plage de temps, spécifiez la période rétrospective pour chaque exécution de requête. Entrez la durée en minutes qui définit la distance qui s'est écoulée depuis l'heure d'exécution de la requête.
  - d. Pour Continuer indéfiniment, spécifiez la date de fin du calendrier. Choisissez de l'exécuter indéfiniment ou jusqu'à une date et une heure spécifiques en utilisant le YYYY/MM/DD format.
6. La console affiche les trois prochaines exécutions de requête planifiées en fonction de votre configuration, en indiquant les dates et heures exactes en UTC auxquelles la requête sera exécutée.

7. Dans la section Publier les résultats de la requête dans S3 - facultatif (si vous utilisez la destination S3) :
  - a. Pour le compartiment S3, sélectionnez Ce compte si le compartiment de destination se trouve dans le même AWS compte, ou sélectionnez Un autre compte si le compartiment se trouve dans un autre AWS compte et saisissez l'ID du compte propriétaire du compartiment en entrée.
  - b. Pour l'URI Amazon S3, entrez le compartiment Amazon S3 et le préfixe dans lesquels les résultats seront stockés (par exemple, `s3://my-bucket/query-results/`). Si vous avez sélectionné Ce compte, vous pouvez choisir Parcourir Amazon S3 pour naviguer et sélectionner un emplacement Amazon S3 existant.
  - c. (Facultatif) Pour l'ARN de la clé KMS, entrez l'ARN d'une AWS KMS clé gérée par le client pour chiffrer les résultats de la requête à l'aide de SSE-KMS. La clé doit se trouver dans la même AWS région que le compartiment Amazon S3 de destination.
8. Dans la section Rôle IAM pour publier les résultats des requêtes sur Amazon S3, choisissez l'une des options suivantes :
  - a. Choisissez Créer automatiquement un nouveau rôle avec des autorisations par défaut pour configurer automatiquement un rôle IAM avec les autorisations requises pour que CloudWatch Logs fournisse les résultats des requêtes à Amazon S3.
  - b. Choisissez Utiliser un rôle existant pour sélectionner un rôle IAM existant avec les politiques requises pour que CloudWatch Logs fournisse les résultats des requêtes à Amazon S3. Utilisez le champ de recherche pour trouver et sélectionner le rôle IAM approprié dans la liste.
9. Dans la section Rôle IAM pour l'exécution planifiée des requêtes, choisissez l'une des options suivantes :
  - a. Choisissez Créer automatiquement un nouveau rôle avec des autorisations par défaut pour configurer automatiquement un rôle IAM avec les autorisations requises pour que CloudWatch Logs exécute des requêtes planifiées.
  - b. Choisissez Utiliser un rôle existant pour sélectionner un rôle IAM existant avec les politiques requises pour que les CloudWatch journaux exécutent des requêtes planifiées. Utilisez le champ de recherche pour trouver et sélectionner le rôle IAM approprié dans la liste.
10. Choisissez Créer un calendrier pour créer la requête planifiée.

## AWS CLI

Pour créer une requête planifiée (AWS CLI)

- Utilisez la `create-scheduled-query` commande pour créer une nouvelle requête planifiée :

```
aws logs create-scheduled-query \
  --name "ErrorAnalysisQuery" \
  --query-language "CWL" \
  --query-string "fields @timestamp, @message | filter @message like /ERROR/ |
stats count() by bin(5m)" \
  --schedule-expression "cron(8 * * * ? *)" \
  --execution-role-arn "arn:aws:iam::123456789012:role/
CloudWatchLogsScheduledQueryRole" \
  --log-group-identifiers "/aws/lambda/my-function" "/aws/apigateway/my-api" \
  --state "ENABLED"
```

## API

Pour créer une requête planifiée (API)

- Utilisez cette `CreateScheduledQuery` action pour créer une nouvelle requête planifiée. L'exemple suivant crée une requête planifiée qui s'exécute toutes les heures :

```
{
  "name": "ErrorAnalysisQuery",
  "queryLanguage": "CWL",
  "queryString": "fields @timestamp, @message | filter @message like /ERROR/ |
stats count() by bin(5m)",
  "scheduleExpression": "cron(8 * * * ? *)",
  "executionRoleArn": "arn:aws:iam::123456789012:role/
CloudWatchLogsScheduledQueryRole",
  "logGroupIdentifiers": ["/aws/lambda/my-function", "/aws/apigateway/my-
api"],
  "state": "ENABLED"
}
```

Après avoir créé la requête planifiée, vous pouvez la consulter et la gérer à partir de la page des requêtes planifiées et à l'aide de l' `ListScheduledQueries` API, qui affiche toutes vos requêtes

planifiées avec leurs noms, dates de création, statut de la dernière exécution, heure du dernier déclenchement et fréquence de répétition.

## Affichage et gestion des requêtes planifiées

Les informations suivantes sont disponibles pour chaque requête :

### Nom

Le nom unique que vous avez attribué à la requête planifiée. Sélectionnez le nom pour afficher l'historique détaillé de la configuration et de l'exécution.

### Date de création

Date à laquelle la requête planifiée a été créée, affichée au YYYY-MM-DD format.

### État de la dernière course

État d'exécution de la dernière requête exécutée. Les valeurs possibles incluent :

- Terminé : la requête a été exécutée avec succès et les résultats ont été envoyés à toutes les destinations configurées.
- Échec : échec de l'exécution de la requête ou de la livraison des résultats. Consultez l'historique des exécutions pour obtenir des informations détaillées sur les erreurs.
- Requête non valide : la requête n'est pas valide et présente des problèmes de syntaxe
- Délai d'expiration : le délai imparti à la requête a expiré. Une requête expire automatiquement au bout de 60 minutes

### Heure du dernier déclenchement

Date et heure de dernière exécution de la requête, affichées au format YYYY-MM-DD HH:MM:SS. Affiche Jamais si la requête n'a pas encore été exécutée.

### Répéter chaque

Fréquence de planification de la requête. Affiche Custom pour les requêtes utilisant des expressions cron ou des descriptions de fréquence spécifiques pour des plannings plus simples.

La page Requetes planifiées fournit une vue d'ensemble de toutes vos requêtes planifiées, indiquant leur statut actuel et leur historique d'exécution afin que vous puissiez consulter, surveiller et gérer toutes vos requêtes planifiées à partir d'un emplacement centralisé. Utilisez ces informations pour surveiller les performances des requêtes, identifier les problèmes et gérer vos flux de travail d'analyse automatique des journaux.



## Console

Pour consulter les requêtes planifiées (console)

1. Vous voulez ouvrir la console CloudWatch Logs à [https://us-east-1.console.aws.amazon.com/cloudwatch/la\\_maison?region=us-east-1#LogsV2:Logs-Insights](https://us-east-1.console.aws.amazon.com/cloudwatch/la_maison?region=us-east-1#LogsV2:Logs-Insights).
2. Dans la console CloudWatch Logs, choisissez Requête planifiée, Afficher les requêtes planifiées.

## AWS CLI

Pour répertorier les requêtes planifiées (AWS CLI)

- Utilisez la `list-scheduled-queries` commande pour répertorier toutes les requêtes planifiées :

```
aws logs list-scheduled-queries --max-results 10
```

## API

Pour répertorier les requêtes planifiées (API)

- Utilisez l'`ListScheduledQueries` action pour récupérer toutes les requêtes planifiées :

```
{
  "maxResults": 10
}
```

L'en-tête de la page Requêtes planifiées indique le nombre total de requêtes planifiées dans votre compte, ce qui vous permet de suivre votre utilisation et de gérer efficacement vos flux de travail d'analyse automatique des journaux.

## Afficher l'historique d'exécution des requêtes planifiées

Utilisez l'historique d'exécution pour surveiller les performances de vos requêtes planifiées et résoudre les problèmes liés à l'exécution des requêtes ou à la livraison des résultats.

L'historique des exécutions indique le statut de chaque requête exécutée, y compris les exécutions réussies, les échecs et les résultats du traitement de destination. Vous pouvez utiliser ces informations pour identifier des modèles, diagnostiquer des problèmes et vérifier que vos requêtes s'exécutent comme prévu.

## Console

Pour consulter l'historique des exécutions (console)

1. Dans la console CloudWatch Logs, choisissez Requête planifiée, Afficher les requêtes planifiées.
2. Sélectionnez la requête planifiée que vous souhaitez examiner.
3. Sélectionnez l'onglet Historique d'exécution.

## AWS CLI

Pour consulter l'historique des exécutions (AWS CLI)

1. Utilisez la `get-scheduled-query-history` commande pour récupérer l'historique d'exécution d'une requête planifiée :

```
aws logs get-scheduled-query-history \
  --identifiant "DailyErrorMonitoring" \
  --start-time 1743379200 \
  --end-time 1743465600 \
  --max-results 10
```

2. Pour filtrer par statut d'exécution, ajoutez le `--execution-statuses` paramètre :

```
aws logs get-scheduled-query-history \
  --identifiant "DailyErrorMonitoring" \
  --start-time 1743379200 \
  --end-time 1743465600 \
  --max-results 1 \
  --execution-statuses "SUCCEEDED"
```

## API

Pour consulter l'historique des exécutions (API)

- Utilisez l'GetScheduledQueryHistoryaction pour récupérer l'historique des exécutions :

```
{
  "identifiant": "DailyErrorMonitoring",
  "startTime": 1743379200,
  "endTime": 1743465600,
  "maxResults": 10,
  "executionStatuses": ["SUCCEEDED", "FAILED"]
}
```

L'historique d'exécution affiche :

- État de l'exécution : en cours, terminée, échec, expiration du délai ou InvalidQuery
- Heure de déclenchement : date à laquelle la requête a été exécutée
- Destinations - État du traitement pour chaque destination configurée, y compris S3 et EventBridge
- Messages d'erreur : détails sur les échecs éventuels lors de l'exécution des requêtes ou du traitement des destinations

## Mettre à jour une requête planifiée

Modifiez la configuration de votre requête planifiée pour modifier la chaîne de requête, le calendrier, les destinations ou le rôle d'exécution en fonction de l'évolution de vos besoins.

Vous pouvez mettre à jour n'importe quel aspect d'une requête planifiée, notamment la chaîne de requête, l'expression de planification, les destinations et le rôle d'exécution. Les modifications prennent effet immédiatement pour les futures exécutions.

## Console

Pour mettre à jour une requête planifiée (console)

1. Dans la console CloudWatch Logs, choisissez Requête planifiée, Afficher les requêtes planifiées.
2. Sélectionnez la requête planifiée que vous souhaitez mettre à jour.

3. Choisissez Modifier.
4. Modifiez la configuration selon vos besoins.
5. Sélectionnez Enregistrer les modifications.

## AWS CLI

Pour mettre à jour une requête planifiée (AWS CLI)

- Utilisez la `update-scheduled-query` commande pour modifier une requête planifiée existante :

```
aws logs update-scheduled-query \
  --identifiant "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f" \
  --description "Monitor for ERROR level logs daily" \
  --query-language "LogsQL" \
  --query-string "fields @timestamp, @message | filter @message like /ERROR/"
\
  --log-group-identifiers "/aws/lambda/my-function-1" "/aws/lambda/my-
function-2"
```

## API

Pour mettre à jour une requête planifiée (API)

1. Utilisez l'`UpdateScheduledQuery` action pour modifier la configuration des requêtes planifiées :

```
{
  "identifiant": "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f",
  "queryString": "fields @timestamp, @message | filter @message like /WARNING|
ERROR/ | stats count() by bin(5m)",
  "scheduleExpression": "cron(0 */2 * * ? *)",
  "state": "ENABLED"
}
```

2. Pour mettre à jour plusieurs paramètres de configuration à la fois :

```
{
```

```
"identifiant": "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f",
  "queryString": "fields @timestamp, @message, @level | filter @level =
'ERROR'",
  "scheduleExpression": "cron(0 8,12,16 * * ? *)",
  "executionRoleArn": "arn:aws:iam::111122223333:role/
UpdatedScheduledQueryRole",
  "logGroupIdentifiers": ["/aws/lambda/my-function", "/aws/lambda/another-
function"],
  "destinationConfiguration": {
    "s3Configuration": {
      "destinationIdentifier": "s3://111122223333-sqn-results-bucket/
processed-results",
      "roleArn": "arn:aws:iam::111122223333:role/Admin"
    }
  }
}
```

## Configuration des destinations S3 pour les requêtes planifiées

Configurez Amazon S3 comme destination pour stocker les résultats de vos requêtes planifiées sous forme de fichiers JSON pour une conservation et une analyse à long terme.

Lorsque vous utilisez Amazon S3 comme destination, les résultats des requêtes sont stockés sous forme de fichiers JSON dans le compartiment et le préfixe spécifiés. Cette option est idéale pour archiver les résultats, effectuer des analyses par lots ou intégrer d'autres AWS services traitant des données S3.

Vous pouvez envoyer les résultats de la requête à un compartiment Amazon S3 du même AWS compte que la requête planifiée ou à un compartiment d'un autre AWS compte. Vous pouvez également éventuellement chiffrer les résultats des requêtes à l'aide d'une AWS KMS clé gérée par le client (SSE-KMS).

## Transmission des résultats vers un compartiment Amazon S3 dans le même compte

Lorsque le compartiment Amazon S3 de destination se trouve dans le même AWS compte que la requête planifiée, vous pouvez parcourir et sélectionner le compartiment directement depuis la console.

## Pour configurer une destination Amazon S3 pour le même compte (console)

1. Dans la section Publier les résultats de la requête dans S3, pour le compartiment S3, sélectionnez Ce compte.
2. Pour l'URI Amazon S3, entrez le compartiment Amazon S3 et le préfixe dans lesquels les résultats seront stockés (par exemple, `s3://my-bucket/query-results/`), ou choisissez Parcourir Amazon S3 pour naviguer et sélectionner un emplacement Amazon S3 existant.
3. (Facultatif) Pour chiffrer les résultats avec une AWS KMS clé gérée par le client, entrez l'ARN de la AWS KMS clé dans le champ ARN de la clé KMS. La clé doit se trouver dans la même AWS région que le compartiment Amazon S3 de destination. Si vous ne spécifiez aucune AWS KMS clé, les paramètres de chiffrement par défaut du compartiment s'appliquent.
4. Dans la section Rôle IAM pour publier les résultats des requêtes sur Amazon S3, choisissez Créer automatiquement un nouveau rôle avec les autorisations par défaut pour configurer automatiquement les autorisations requises, ou choisissez Utiliser un rôle existant pour sélectionner un rôle IAM existant avec les politiques requises.

Le rôle IAM de livraison de destination nécessite les autorisations suivantes :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-bucket/prefix/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

## Transmission des résultats vers un compartiment Amazon S3 d'un autre compte

Vous pouvez envoyer les résultats des requêtes planifiées à un compartiment Amazon S3 d'un autre AWS compte. Lorsque vous utilisez un compartiment multi-comptes, vous devez fournir l'URI Amazon S3 et l'ID de compte du compte propriétaire du compartiment.

Pour configurer une destination Amazon S3 entre comptes (console)

1. Dans la section Publier les résultats de la requête dans S3, pour le compartiment S3, sélectionnez Un autre compte et saisissez l'ID du compte propriétaire du compartiment en entrée.
2. Pour l'URI Amazon S3, entrez l'URI Amazon S3 complet du compartiment de destination et le préfixe de l'autre compte (par exemple, `s3://cross-account-bucket/query-results/`).
3. (Facultatif) Pour chiffrer les résultats avec une AWS KMS clé gérée par le client, entrez l'ARN de la AWS KMS clé dans le champ ARN de la clé KMS. La clé doit se trouver dans la même AWS région que le compartiment Amazon S3 de destination.
4. Dans la section Rôle IAM pour publier les résultats des requêtes sur Amazon S3, choisissez Créer automatiquement un nouveau rôle avec les autorisations par défaut pour configurer automatiquement les autorisations requises, ou choisissez Utiliser un rôle existant pour sélectionner un rôle IAM existant avec les politiques requises.

La livraison entre comptes nécessite des autorisations des deux côtés. Le rôle IAM de livraison de destination dans le compte source nécessite les autorisations suivantes :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::cross-account-bucket/prefix/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

```
]
}
```

La politique du compartiment Amazon S3 du compte de destination doit accorder au rôle IAM du compte source l'autorisation d'écrire des objets :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowScheduledQueryRolePutObject",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/my-s3-delivery-role"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::cross-account-bucket/prefix/*"
    }
  ]
}
```

## Chiffrer les résultats à l'aide d'une clé gérée par AWS KMS le client

Vous pouvez éventuellement spécifier une AWS KMS clé gérée par le client pour chiffrer les résultats des requêtes transmis à Amazon S3 à l'aide de SSE-KMS. La AWS KMS clé peut se trouver dans le même compte que la requête planifiée ou dans un autre compte.

Lorsque vous spécifiez une AWS KMS clé, la requête planifiée utilise cette clé pour chiffrer les résultats via SSE-KMS. Lorsque vous ne spécifiez aucune AWS KMS clé, les paramètres de chiffrement par défaut du compartiment s'appliquent. Si le compartiment est configuré avec le chiffrement SSE-KMS par défaut à l'aide d'une clé gérée par le client, le rôle IAM de livraison de destination doit toujours disposer d'une `kms:GenerateDataKey` autorisation sur cette clé.

Le rôle IAM de livraison de destination nécessite une `kms:GenerateDataKey` autorisation sur la AWS KMS clé. L'exemple suivant montre les autorisations requises pour une destination Amazon S3 avec une AWS KMS clé gérée par le client :

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



```

    "Effect": "Allow",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3::my-bucket/prefix/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceAccount": "111122223333"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "kms:GenerateDataKey",
    "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "s3.us-east-1.amazonaws.com"
      },
      "StringLike": {
        "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::my-bucket*"
      }
    }
  }
]
}

```

Lorsque la AWS KMS clé se trouve dans un compte différent du rôle IAM de livraison de destination, la politique AWS KMS clé du compte propriétaire de la clé doit explicitement accorder l'accès au rôle :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowScheduledQueryRoleToEncrypt",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/my-s3-delivery-role"
      },
      "Action": "kms:GenerateDataKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "s3.us-east-1.amazonaws.com"
        }
      },
    }
  ]
}

```

```
    "StringLike": {  
      "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::my-bucket*"  
    }  
  }  
}
```

#### Note

Lorsque la AWS KMS clé et le rôle IAM de livraison de destination se trouvent dans le même compte, la politique d'identité IAM à elle seule suffit si la politique AWS KMS clé inclut l'instruction racine par défaut « Activer les politiques IAM ». Une attribution explicite de politique AWS KMS clé n'est requise que si la politique clé ne délègue pas à IAM.

Le rôle IAM pour publier les résultats des requêtes sur Amazon S3 doit être configuré séparément du rôle IAM pour l'exécution planifiée des requêtes. Cette séparation permet un contrôle d'accès précis, où le rôle d'exécution peut exécuter des requêtes tandis que le rôle Amazon S3 gère spécifiquement la livraison des résultats. Les deux rôles doivent inclure une politique de confiance qui autorise le service CloudWatch Logs (logs.amazonaws.com) à assumer le rôle.

## Résolution des problèmes de requêtes planifiées

Utilisez ces rubriques de résolution des problèmes pour résoudre les problèmes courants liés aux requêtes planifiées.

### L'exécution de la requête échoue en raison d'erreurs d'autorisation

Résolvez les erreurs d'autorisation qui empêchent les requêtes planifiées d'exécuter ou de fournir des résultats aux destinations.

Des erreurs d'autorisation se produisent lorsque le rôle d'exécution ne dispose pas des autorisations nécessaires pour lire des informations dans des groupes de journaux ou écrire sur des ressources de destination.

## Pour résoudre les erreurs d'autorisation

1. Vérifiez que le rôle d'exécution dispose `logs:StartQuery` d'`logs:GetQueryResults` `logs:DescribeLogGroups` autorisations et d'autorisations pour les groupes de journaux cibles.
2. Assurez-vous que le rôle d'exécution dispose d'autorisations d'écriture pour les ressources de destination (par exemple `s3:PutObject` pour les compartiments S3).
3. Vérifiez que la politique de confiance autorise CloudWatch Logs à assumer le rôle d'exécution. Le rôle doit faire confiance au principal du service de logs (`logs.amazonaws.com`) dans sa politique de confiance.

Les causes courantes incluent des autorisations IAM manquantes, une ressource incorrecte ARNs dans la politique ou des problèmes de configuration de la politique de confiance.

Pour éviter les erreurs d'autorisation, appliquez le principe du moindre privilège lors de la création de rôles d'exécution et testez les autorisations avant de déployer des requêtes planifiées en production.

## Expiration du délai de requête

Résolvez les erreurs de temporisation qui se produisent lorsque les requêtes planifiées dépassent le délai d'exécution maximal.

Les délais d'attente des requêtes se produisent lorsque la requête met plus de 60 minutes à traiter la plage de données spécifiée, souvent en raison de grands ensembles de données ou d'une logique de requête complexe.

### Pour résoudre les erreurs de délai

1. Réduisez l'intervalle de temps en diminuant le décalage de l'heure de début afin de traiter moins de données par exécution.
2. Optimisez la requête en ajoutant des filtres au début de la requête afin de réduire la quantité de données traitées. Utilisez des index de filtre pour réduire la taille de numérisation des données.
3. Envisagez de diviser les requêtes complexes en requêtes plus simples et plus ciblées.

Les causes courantes incluent l'interrogation de longues plages de temps, le traitement de groupes de journaux volumineux ou l'utilisation d'agrégations complexes sans filtrage approprié.

Pour éviter les délais, testez les requêtes manuellement dans CloudWatch Logs Insights avec le volume de données attendu et optimisez les performances avant de les planifier.

## Le traitement de la destination échoue

Résolvez les défaillances qui se produisent lorsque les résultats des requêtes planifiées ne peuvent pas être transmis aux destinations configurées.

Les échecs de traitement de destination se produisent lorsque le compartiment ou le bus d'EventBridge événements Amazon S3 cible est inaccessible ou mal configuré.

Pour résoudre les problèmes lorsque les résultats des requêtes ne sont pas publiés vers la destination

1. Vérifiez que le compartiment Amazon S3 spécifié existe et qu'il est accessible.
2. Vérifiez que la configuration de destination est correcte URIs.
3. Assurez-vous que le rôle d'exécution dispose des autorisations nécessaires pour écrire sur la destination.

Les causes courantes incluent des ressources de destination supprimées ou renommées, une destination URIs incorrecte ou des problèmes de connectivité réseau.

Pour éviter les défaillances de destination, validez régulièrement les configurations de destination et surveillez la disponibilité des ressources de destination.

## Erreurs de requête non valides

Résolvez les erreurs de syntaxe et de logique dans les chaînes de requête planifiées qui empêchent une exécution réussie.

Des erreurs de requête non valides se produisent lorsque la chaîne de requête contient des erreurs de syntaxe, fait référence à des champs inexistants ou utilise des fonctionnalités de langage de requête non prises en charge.

Pour résoudre les erreurs de requête non valides

1. Testez votre requête manuellement dans CloudWatch Logs Insights pour vérifier la syntaxe et la logique.
2. Vérifiez que tous les champs de journal référencés existent dans vos groupes de journaux cibles.

3. Vérifiez que les fonctionnalités du langage de requête que vous utilisez sont prises en charge pour les requêtes planifiées.

Les causes courantes incluent des fautes de frappe dans les noms de champs, une syntaxe de requête incorrecte ou l'utilisation de fonctionnalités de requête non prises en charge dans l'environnement d'exécution planifiée.

Pour éviter les erreurs de requête non valides, testez toujours les requêtes de manière interactive avant de les planifier et utilisez les fonctionnalités de découverte des champs pour vérifier les noms des champs.

## Erreurs de simultanéité des requêtes

Certains points importants mentionnés ci-dessous doivent être pris en compte lorsque des erreurs de simultanéité sont détectées car les requêtes planifiées utilisent le même quota que les requêtes d'informations de Cloudwatch Logs. Il est recommandé d'étaler vos plannings pour éviter d'atteindre la limite de simultanéité.

- Quota : vous pouvez exécuter jusqu'à 100 requêtes CloudWatch Logs Insights simultanées par AWS compte.
- Tableaux de bord : les requêtes ajoutées aux CloudWatch tableaux de bord sont également prises en compte dans cette limite de simultanéité, car elles sont exécutées lorsque le tableau de bord est chargé ou actualisé.
- OpenSearch Service PPL/SQL : vous pouvez exécuter jusqu'à 15 requêtes OpenSearch PPL ou OpenSearch SQL simultanées par compte. AWS
- Requêtes entre comptes : le quota de simultanéité s'applique à la fois aux requêtes uniques et aux requêtes multicomptes. Lorsque vous utilisez l' CloudWatch observabilité entre comptes, les requêtes initiées dans un compte de surveillance par rapport à un compte source associé sont également prises en compte dans le calcul de la limite de simultanéité du compte de surveillance.
- Groupes de journaux d'accès peu fréquents : pour les groupes de journaux de la classe de journaux d'accès peu fréquent, le nombre maximum de requêtes Logs Insights simultanées est limité à cinq.

# Détection des anomalies du journal

Vous pouvez détecter les anomalies dans les données de votre journal de deux manières : en créant un détecteur d'anomalies de journal pour une surveillance continue, ou en utilisant la [anomaly detection](#) commande dans les requêtes CloudWatch Logs Insights pour une analyse à la demande.

Un détecteur d'anomalies du journal analyse les événements du journal ingérés dans un groupe de journaux et détecte automatiquement 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. Pour une analyse à la demande, vous pouvez utiliser la `anomaly detection` commande dans les requêtes CloudWatch Logs Insights pour identifier des modèles inhabituels dans les données de séries chronologiques. Pour plus d'informations sur la détection des anomalies basée sur les requêtes, consultez [Utilisation de la détection des anomalies dans CloudWatch Logs Insights](#)

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 l'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 FATALERROR, etWARN. Si aucun de ces mots clés n'est trouvé, la gravité d'un modèle est marquée commeNONE.

## 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 [ListAnomalies](#) d'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.

## Suppression d'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, consultez [Analyse 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](#).

# Utilisation de la détection des anomalies dans CloudWatch Logs Insights

Outre la création de détecteurs d'anomalies de journal pour une surveillance continue, vous pouvez également utiliser la `anomaly` commande dans les requêtes CloudWatch Logs Insights pour identifier à la demande des modèles inhabituels dans les données de vos journaux. Cette commande étend les `pattern` fonctionnalités existantes et utilise l'apprentissage automatique pour détecter cinq types d'anomalies, notamment les changements de fréquence des modèles, les nouveaux modèles et les variations symboliques.

La `anomaly` commande est particulièrement utile pour :

- Analyser ad hoc des données historiques du journal pour identifier les modèles inhabituels
- Enquêter sur des périodes spécifiques pour détecter un comportement anormal
- Surveiller les applications telles que les fonctions Lambda pour détecter les problèmes d'exécution

Pour plus d'informations sur l'utilisation de la `anomaly` commande dans vos requêtes, consultez [anomalie](#).

Cette détection d'anomalies basée sur des requêtes complète les détecteurs d'anomalies continus décrits dans les sections suivantes, vous offrant à la fois des fonctionnalités de surveillance en temps réel et d'analyse à la demande.

## 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, 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.
  - 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'ALARMÉtat 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, consultez[Gravité 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 s'affiche 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](#).

## Restrictions

- 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. Au cours de 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` à l'aide de la [get-key-policy](#) commande 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-policy](#) commande 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, consultez [Classes 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'`StartLiveTailAPI` achemine les demandes à l'aide de l'injection du préfixe hôte du SDK. Les versions du SDK publiées avant le 1er avril 2026 sont acheminées vers `streaming-logs.Region.amazonaws.com`, ce qui ne prend pas en charge les points de terminaison VPC. Les versions du SDK publiées le 1er avril 2026 ou après cette date sont acheminées vers `stream-logs.Region.amazonaws.com`, qui prend en charge les points de terminaison VPC. Pour configurer un point de terminaison VPC pour cette API, consultez [Création d'un point de terminaison VPC](#) pour les journaux. CloudWatch

# 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 en continu 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 avec 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 le format JSON et le texte brut. Le mode interactif affiche également

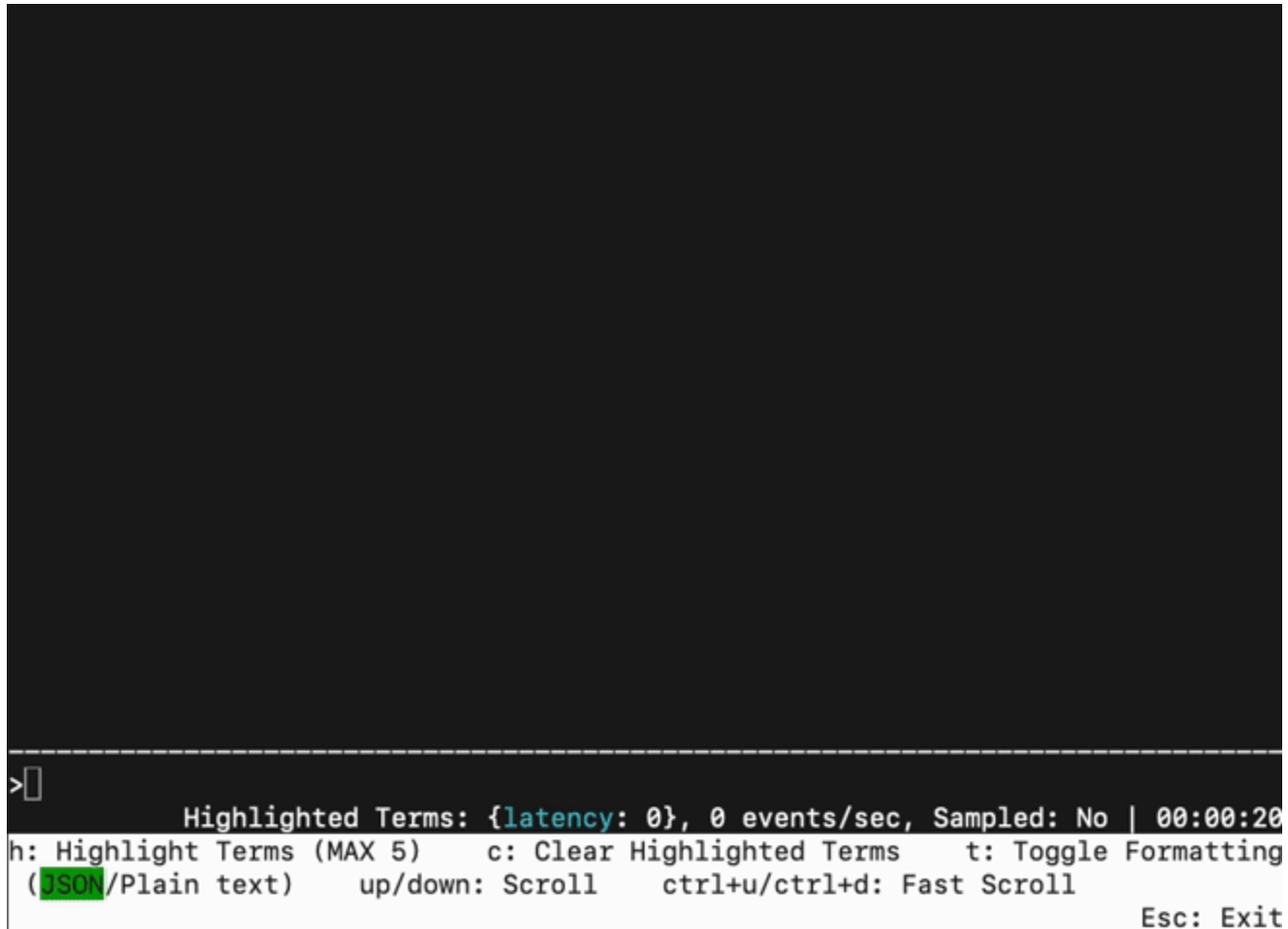


des informations sur la session Live Tail, telles que la durée de la session, si la session est échantillonnée, les termes actuellement surlignés et le nombre de fois qu'ils ont été rencontrés.

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.



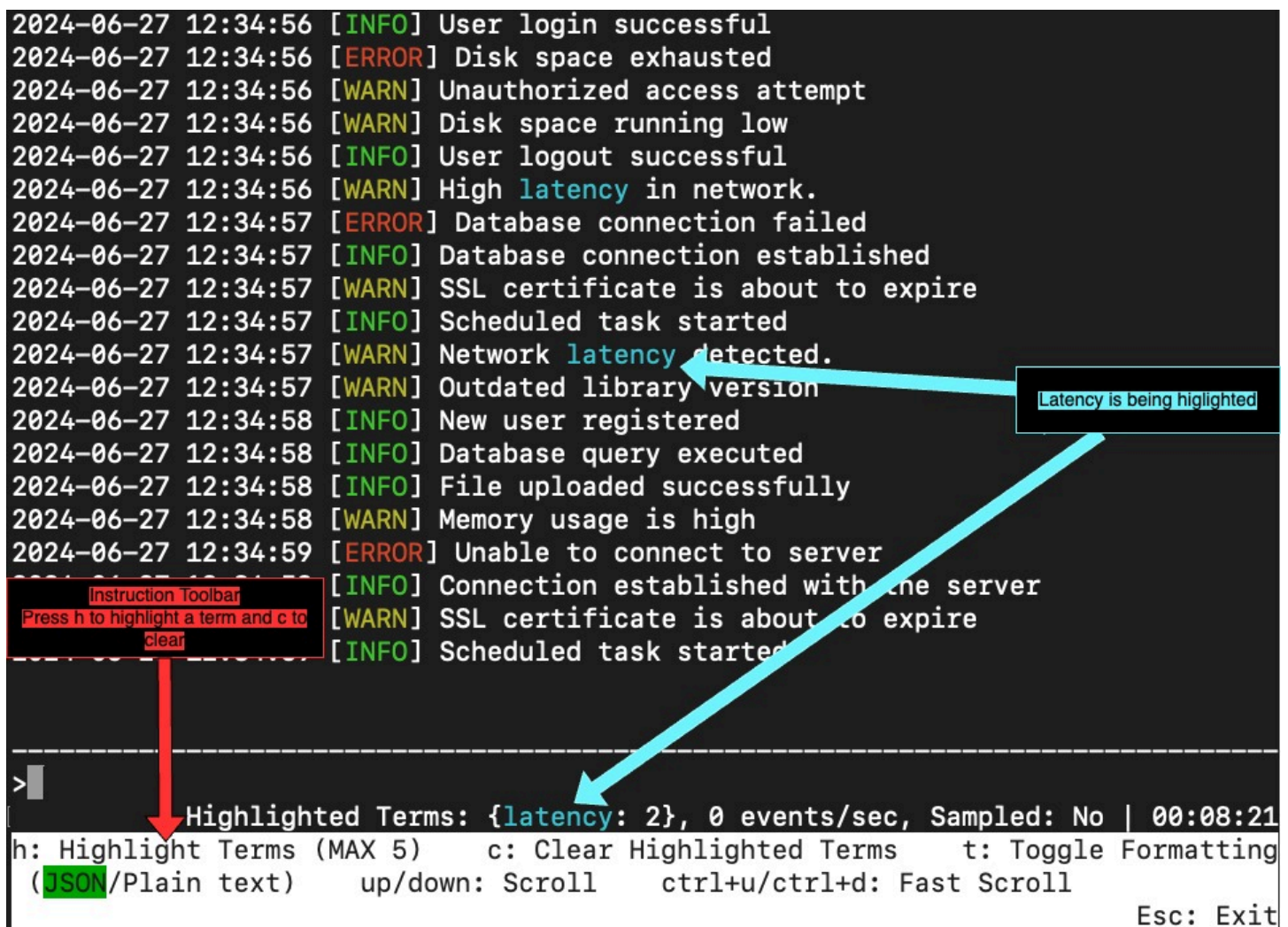
Pour surligner un terme dans les journaux de diffusion, appuyez sur h, puis saisissez-le. Ce qui suit montre l'écran une fois que le terme latency a été surligné.

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.



```
2024-06-27 12:34:56 [INFO] User login successful
2024-06-27 12:34:56 [ERROR] Disk space exhausted
2024-06-27 12:34:56 [WARN] Unauthorized access attempt
2024-06-27 12:34:56 [WARN] Disk space running low
2024-06-27 12:34:56 [INFO] User logout successful
2024-06-27 12:34:56 [WARN] High latency in network.
2024-06-27 12:34:57 [ERROR] Database connection failed
2024-06-27 12:34:57 [INFO] Database connection established
2024-06-27 12:34:57 [WARN] SSL certificate is about to expire
2024-06-27 12:34:57 [INFO] Scheduled task started
2024-06-27 12:34:57 [WARN] Network latency detected.
2024-06-27 12:34:57 [WARN] Outdated library version
2024-06-27 12:34:58 [INFO] New user registered
2024-06-27 12:34:58 [INFO] Database query executed
2024-06-27 12:34:58 [INFO] File uploaded successfully
2024-06-27 12:34:58 [WARN] Memory usage is high
2024-06-27 12:34:59 [ERROR] Unable to connect to server
2024-06-27 12:34:59 [INFO] Connection established with the server
2024-06-27 12:34:59 [WARN] SSL certificate is about to expire
2024-06-27 12:34:59 [INFO] Scheduled task started
```

Instruction Toolbar  
Press h to highlight a term and c to clear

Highlighted Terms: {latency: 2}, 0 events/sec, Sampled: No | 00:08:21  
h: Highlight Terms (MAX 5) c: Clear Highlighted Terms t: Toggle Formatting  
(JSON/Plain text) up/down: Scroll ctrl+u/ctrl+d: Fast Scroll Esc: Exit

## 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 [Démarez 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).

# Centralisation des journaux inter-comptes et inter-régions

La centralisation des données Amazon CloudWatch Logs permet de AWS Organizations collecter les données de journal de plusieurs comptes membres dans un seul référentiel de données en utilisant des règles de centralisation entre comptes et entre régions. Vous définissez les règles qui répliquent automatiquement les données de journal de plusieurs comptes et Régions AWS dans un compte centralisé au sein de votre organisation. Cette fonctionnalité rationalise la consolidation des journaux pour une surveillance, une analyse et une conformité centralisées améliorées sur l'ensemble de votre AWS infrastructure.

CloudWatch La centralisation des données des journaux offre une flexibilité de configuration répondant aux exigences opérationnelles et de sécurité, telles que la possibilité de configurer une région de sauvegarde lors de la configuration des règles dans le compte de destination afin de garantir une résilience accrue. En outre, vous avez un contrôle total sur le comportement de chiffrement des groupes de journaux copiés à partir de comptes sources afin de gérer les données initialement chiffrées à l'aide de clés KMS gérées par le client.

## Note

La fonctionnalité de centralisation CloudWatch des journaux traite uniquement les nouvelles données de journal qui arrivent dans les comptes sources une fois que vous avez créé la règle de centralisation. Les données des journaux historiques (journaux qui existaient avant la création des règles) ne sont pas centralisées.

## Concepts de centralisation des données

Avant de commencer à utiliser la centralisation des données des CloudWatch journaux, familiarisez-vous avec les concepts suivants :

### Règle de centralisation

Configuration qui définit la manière dont les données de journal des comptes et des régions sources sont répliquées vers un compte et une région de destination. Les règles spécifient les critères de source et les paramètres de destination.

## Compte source

Le AWS compte d'où proviennent les données du journal. Les événements du journal des comptes sources sont répliqués sur le compte de destination en fonction des règles de centralisation que vous définissez.

## Compte de destination

Le AWS compte de destination où les données du journal répliquées sont stockées. Ce compte sert d'emplacement centralisé pour l'analyse et la surveillance des journaux.

## Région de sauvegarde

Région secondaire facultative du compte de destination dans laquelle les données du journal peuvent être répliquées à des fins de résilience et de reprise après sinistre accrues.

## Chiffrement dans CloudWatch les journaux

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. Vous pouvez également utiliser le service de gestion des AWS clés pour ce chiffrement. Pour plus d'informations, consultez la [documentation sur le chiffrement CloudWatch des journaux](#).

- Fonctionnement du chiffrement lors de la centralisation : La centralisation CloudWatch des journaux copie activement les données des journaux au moment de l'ingestion des comptes source vers les comptes de destination. Au cours de ce processus, vos données restent cryptées en transit à l'aide d'une clé de service AWS détenue. Les données au repos dans les groupes de journaux source et de destination sont chiffrées à l'aide de la méthode de chiffrement que vous avez choisie (clés KMS gérées ou AWS détenues par le client). Si vous utilisez une clé KMS gérée par le client dans vos groupes de journaux de destination, ajoutez la balise `LogsManaged = true` à la clé KMS pour que le service de centralisation puisse y accéder.
- Lorsque des autorisations KMS sont requises :
  - Si vous utilisez des clés KMS gérées par le client dans vos comptes sources, CloudWatch Logs nécessite des [autorisations KMS](#) dans les exemples de scénarios suivants :
    - Gestion du débit : Lorsque les limites de débit de centralisation sont atteintes, les données du journal sont temporairement stockées cryptées avec la clé KMS gérée par le client jusqu'à ce que la bande passante soit disponible.



- **Protection et rédaction des données** : lorsque les politiques de protection des données sont activées dans les groupes de CloudWatch journaux sources, Logs a besoin d'autorisations de déchiffrement pour accéder aux données brutes des journaux afin de les centraliser.

 **Important**

Les règles de centralisation sont gérées par le compte de gestion AWS des Organizations ou par l'administrateur délégué. Pour exclure de la centralisation les groupes de journaux chiffrés par KMS gérés par le client, configurez les paramètres de la règle sur « Ne pas centraliser les groupes de journaux chiffrés avec une AWS clé KMS ».

## Configuration de la centralisation des journaux

Pour configurer la centralisation CloudWatch des journaux, vous devez configurer des règles de centralisation qui définissent la manière dont les données des journaux circulent des groupes de journaux des comptes source vers les groupes de journaux de votre compte de destination.

Une fois que la règle de centralisation est activée et que les événements du journal sont répliqués sur le compte de destination, vous pouvez créer des filtres de métriques, d'abonnement et de compte sur des groupes de journaux centralisés dotés de fonctionnalités de filtrage améliorées. Ces filtres peuvent cibler les événements du journal provenant de comptes source et de régions spécifiques, et peuvent émettre des informations sur le compte source et la région sous forme de dimensions métriques. Pour de plus amples informations, veuillez consulter [Création de métriques à partir d'événements du journal à l'aide de filtres](#).

## Conditions préalables

- AWS Organizations doit être configuré et les comptes source et de destination doivent tous deux appartenir à l'organisation.
- L'accès sécurisé doit être activé pour CloudWatch le compte de gestion et le compte de destination afin de fournir un accès aux données du journal.



**Note**

Il est recommandé d'activer l'accès sécurisé via la console, qui crée automatiquement le rôle lié au service (SLR) requis. Si l'accès sécurisé est activé par d'autres méthodes, le rôle lié au service devra être créé séparément.

## Personnalisation des noms des groupes de journaux de destination

Lorsque vous créez une règle de centralisation, vous pouvez personnaliser la façon dont les noms des groupes de journaux de destination sont structurés à l'aide d'attributs. Ces attributs sont automatiquement remplacés par des valeurs réelles lors de la création de groupes de journaux, ce qui vous permet d'organiser les journaux de manière hiérarchique dans votre compte de destination. Par défaut, seul l'`${source.logGroup}` attribut est utilisé, qui fusionne tous les groupes de journaux portant le même nom dans le compte de destination. Si une variable ne peut pas être résolue, elle hérite de la valeur de sa variable parent dans la hiérarchie.

### Attributs disponibles

Vous pouvez utiliser les attributs suivants dans le modèle de nom de votre groupe de journaux de destination :

Attributs du nom du groupe de journaux de destination

| Attribut                           | Description                                                           |
|------------------------------------|-----------------------------------------------------------------------|
| <code>\${source.accountId}</code>  | L'ID du AWS compte d'où provient le journal.                          |
| <code>\${source.region}</code>     | L' Région AWS origine du journal.                                     |
| <code>\${source.logGroup}</code>   | Le nom du groupe de journaux d'origine indiqué dans le compte source. |
| <code>\${source.org.id}</code>     | Votre AWS Organizations identifiant du compte source.                 |
| <code>\${source.org.ouId}</code>   | L'ID de l'unité organisationnelle du compte source                    |
| <code>\${source.org.rootId}</code> | L'ID racine de l'organisation                                         |

| Attribut                         | Description                                                |
|----------------------------------|------------------------------------------------------------|
| <code>\${source.org.path}</code> | Le parcours organisationnel complet, du compte à la racine |

## Exemples

### Préserver la structure d'origine du groupe de logs

Modèle : `/centralized/${source.accountId}${source.logGroup}`

Résultat: `/centralized/123456789012/aws/lambda/my-function`

### Organiser par compte et par région

Modèle : `/centralized/${source.accountId}/${source.region}`

Résultat: `/centralized/123456789012/us-east-1`

### Organiser par structure organisationnelle

Modèle : `/logs/${source.org.id}/${source.org.ouId}/${source.accountId}`

Résultat: `/logs/o-abc123/ou-xyz-12345678/123456789012`

### Structure plate simple

Modèle : `/centralized-logs`

Résultat: `/centralized-logs`

## Bonnes pratiques

- Incluez l'identifiant du compte source pour identifier facilement d'où proviennent les journaux de compte.
- Incluez la région source si vous effectuez une centralisation à partir de plusieurs régions.
- Structurez les noms des groupes de journaux de destination de manière à ce qu'ils comportent moins de 512 caractères. CloudWatch Logs impose une longueur maximale de 512 caractères pour les noms de groupes de journaux.

## Création d'une règle de centralisation

Utilisez la procédure suivante pour créer une règle de centralisation qui réplique les données de journal des comptes source vers votre compte de destination.

Pour créer une règle de centralisation

1. Accédez à la CloudWatch console dans le compte de gestion ou d'administrateur délégué de l'organisation.
2. Cliquez sur Paramètres.
3. Accédez à l'onglet Organisation.
4. Choisissez Configurer la règle.
5. Spécifiez les détails de la source en définissant les champs suivants, puis choisissez Next :
  - a. Nom de la règle de centralisation : entrez un nom unique pour la règle de centralisation.
  - b. Comptes sources : définissez les critères de sélection des sources pour sélectionner les comptes à partir desquels les données de télémétrie seront centralisées. Les critères de sélection peuvent inclure :
    - Liste des comptes des membres de l'organisation
    - Liste des unités organisationnelles de l'organisation
    - L'ensemble de l'organisation

Vous pouvez fournir les critères de sélection selon deux modes :

- Builder : une expérience basée sur les clics pour générer les critères de sélection des sources
- Éditeur : zone de texte libre pour fournir les critères de sélection de la source

Syntaxe prise en charge pour les critères de sélection des sources :

- Clés prises en charge : `OrganizationId` `OrganizationUnitId` | `AccountId` | | \*
  - Opérateurs pris en charge : `=` | `IN` | `OR`
- c. Régions sources : sélectionnez une liste de régions pour rechercher les données de télémétrie à centraliser.
6. Spécifiez les détails de la destination en définissant les champs suivants, puis choisissez Next :

- a. Compte de destination : sélectionnez un compte dans l'organisation qui sert de destination centrale pour les données de télémétrie.
  - b. Région de destination : sélectionnez une région principale qui stocke une copie des données de télémétrie centralisées.
  - c. Région de sauvegarde : sélectionnez éventuellement une région qui stocke une deuxième copie des données de télémétrie centralisées.
7. Spécifiez les données de télémétrie en définissant les champs suivants, puis choisissez Next :

- a. Groupes de journaux : choisissez l'une des options suivantes :
  - Tous les groupes de journaux : centralisez les journaux de tous les groupes de journaux dans les comptes sources.
  - Filtrer les groupes de journaux : centralisez les journaux à partir d'un sous-ensemble de groupes de journaux dans les comptes sources, en fonction des critères de sélection. Vous pouvez fournir les critères de sélection selon deux modes :
    - Builder : une expérience basée sur les clics pour générer les critères de sélection
    - Editeur : une zone de texte libre pour fournir les critères de sélection

Vous pouvez utiliser deux critères de sélection pour filtrer les journaux :

- Critères de sélection des groupes de journaux : critères de sélection qui spécifient les groupes de journaux sources à centraliser.
  - Clés prises en charge : LogGroupName | \*
  - Opérateurs pris en charge : = | != | DANS | PAS DANS | ET | OU | COMME | PAS COMME
- Critères de sélection des sources de données : critères de sélection qui indiquent les sources de données à centraliser.
  - Clés prises en charge : DataSourceName | DataSourceType
  - Opérateurs pris en charge : = | != | DANS | PAS DANS | ET | OU | COMME | PAS COMME

Lorsque les critères de sélection des groupes de journaux et les critères de sélection des sources de données sont spécifiés, un événement de journal doit répondre aux deux critères pour être centralisé.

- b. Groupe de journaux chiffrés KMS

**⚠ Important**

CloudWatch les règles de centralisation ne pourront pas transmettre les journaux du compte source aux groupes de journaux de destination si la clé KMS fournie dans la règle de centralisation n'autorise pas CloudWatch les journaux à l'utiliser. Si vous utilisez une clé KMS gérée par le client dans vos groupes de journaux de destination, ajoutez le tag `LogsManaged = true` à la clé KMS. Pour de plus amples informations, veuillez consulter [Étape 2 : Définition des autorisations sur la clé KMS](#).

Choisissez l'une des options suivantes :

- Centralisez les groupes de journaux sources chiffrés avec des clés KMS gérées par le client à l'aide d'une clé KMS gérée par le client spécifique à une destination : centralisez les événements des journaux des groupes de journaux sources chiffrés avec des clés KMS gérées par le client dans les groupes de journaux de destination chiffrés avec une clé KMS gérée par le client dans le compte de destination.

Lorsque ce paramètre est sélectionné, vous devez également définir les paramètres suivants :

- ARN de la clé de chiffrement de destination : ARN de la clé KMS gérée par le client dans le compte de destination et la région de destination principale, à associer aux groupes de journaux de destination nouvellement créés.
- ARN de la clé de chiffrement de destination de sauvegarde (si la région de sauvegarde est sélectionnée) : ARN de la clé KMS gérée par le client dans le compte de destination et la région de destination de sauvegarde, à associer aux groupes de journaux de destination nouvellement créés.
- Ignorer la centralisation vers des groupes de journaux de destination non chiffrés (facultatif) : si un groupe de journaux existe déjà sans clé KMS gérée par le client, CloudWatch impossible de mettre à jour son chiffrement. Choisissez cette option pour ignorer la centralisation des événements de journal des groupes de journaux source chiffrés à l'aide de clés KMS gérées par le client vers des groupes de journaux de destination qui ne sont pas associés à une clé KMS gérée par le client.
- Centralisez les groupes de journaux chiffrés à l'aide de clés KMS gérées par le client dans le compte de destination avec la clé KMS AWS détenue : centralisez les événements des journaux des groupes de journaux source chiffrés avec des clés KMS gérées par le client

dans les groupes de journaux de destination nouvellement créés et chiffrés à l'aide d'une clé KMS AWS détenue.

- Ne centralisez pas les groupes de journaux chiffrés à l'aide de clés KMS gérées par le client : ignorez la centralisation des événements de journal à partir des groupes de journaux source chiffrés à l'aide de clés KMS gérées par le client.

8. Passez en revue la règle de centralisation, apportez éventuellement des modifications de dernière minute et choisissez Créer une politique de centralisation.

## Modifier une règle de centralisation

Utilisez la procédure suivante pour modifier une règle de centralisation existante.

Pour modifier une règle de centralisation

1. Accédez à la CloudWatch console dans le compte de gestion ou d'administrateur délégué de l'organisation.
2. Cliquez sur Paramètres.
3. Accédez à l'onglet Organisation.
4. Choisissez Gérer les règles.
5. Sélectionnez la règle à mettre à jour, puis choisissez Modifier.
6. Mettez à jour la configuration des règles selon vos besoins, en choisissant Next pour passer à chaque étape.
7. À l'étape 4, Révision et configuration, choisissez Mettre à jour la politique de centralisation.

## Afficher une règle de centralisation

Utilisez la procédure suivante pour afficher les détails d'une règle de centralisation existante.

Pour consulter une règle de centralisation

1. Accédez à la CloudWatch console dans le compte de gestion ou d'administrateur délégué de l'organisation.
2. Cliquez sur Paramètres.
3. Accédez à l'onglet Organisation.
4. Choisissez Gérer les règles.

5. Consultez la liste de toutes les règles de centralisation existantes et choisissez un nom de règle spécifique pour en afficher les détails.

## Supprimer une règle de centralisation

Pour supprimer une règle de centralisation existante, procédez comme suit.

Pour supprimer une règle de centralisation

1. Accédez à la CloudWatch console dans le compte de gestion ou d'administrateur délégué de l'organisation.
2. Cliquez sur Paramètres.
3. Accédez à l'onglet Organisation.
4. Choisissez Gérer les règles.
5. Sélectionnez la règle à supprimer, puis cliquez sur Supprimer.
6. Pour confirmer la suppression, choisissez Supprimer.

## Règles de centralisation de surveillance et de résolution des problèmes

Vous pouvez surveiller l'état et les performances de vos règles de centralisation à l'aide CloudWatch des métriques, de la console CloudWatch Logs et AWS CloudTrail des journaux. Cela vous permet de vous assurer que les données du journal sont correctement répliquées et d'identifier tout problème lié à votre configuration de centralisation.

CloudWatch Logs fournit :

1. État de santé des règles par règle de centralisation
  - a. Cliquez sur Paramètres.
  - b. Accédez à l'onglet Organisation.
  - c. Choisissez Gérer les règles.
2. Enregistre les appels d'API avec AWS CloudTrail
3. CloudWatch publie également des métriques pour la centralisation, notamment les événements de journal répliqués, les erreurs et la régulation. Pour plus d'informations sur ces mesures et leurs dimensions, consultez [Mesures et dimensions de centralisation](#).

## État de santé des règles de centralisation

Chaque règle de centralisation possède un état de santé qui indique si elle fonctionne correctement. Vous pouvez vérifier l'état des règles via la console ou par programmation à l'aide de l'API.

Les états d'intégrité des règles incluent :

- **HEALTHY**: La règle fonctionne normalement et réplique les données du journal telles que configurées
- **UNHEALTHY**: la règle a rencontré des problèmes et ne reproduit peut-être pas correctement les données
- **PROVISIONING**: La centralisation de l'organisation est en cours de mise en place.

Lorsqu'une règle est marquée comme **INSTABLE**, le `FailureReason` champ fournit des détails sur le problème spécifique à résoudre.

## Surveillance des appels d'API de centralisation avec AWS CloudTrail

AWS CloudTrail enregistre les appels d'API effectués vers le service de centralisation, ce qui vous permet de suivre les modifications de configuration et de résoudre les problèmes liés aux comptes membres de votre compte. AWS Organizations

Les principaux CloudTrail événements liés à la centralisation sont les suivants :

- **CreateCentralizationRuleForOrganization**: Lorsqu'une nouvelle règle de centralisation est créée
- **UpdateCentralizationRuleForOrganization**: Lorsqu'une règle existante est modifiée
- **DeleteCentralizationRuleForOrganization**: Lorsqu'une règle est supprimée
- **GetCentralizationRuleForOrganization**: Lorsque les détails des règles sont récupérés
- **ListCentralizationRulesForOrganization**: Lorsque les règles sont répertoriées

Vous pouvez utiliser CloudTrail les journaux pour auditer les modifications de configuration de la centralisation et les corréliser aux problèmes de performances ou aux échecs de réplication.



## Recommandations de surveillance

Pour garantir le bon fonctionnement de la centralisation, nous vous recommandons de configurer des CloudWatch alarmes sur les principaux indicateurs de centralisation que nous vendons à CloudWatch Metrics. Cette surveillance proactive vous aide à détecter les problèmes à un stade précoce et à maintenir une centralisation fiable des journaux au sein de votre organisation.

Les principaux indicateurs à surveiller sont les suivants :

- **IncomingCopiedBytes**: Surveillez le volume de données de journal correctement répliquées sur votre compte de destination. Une baisse soudaine ou l'absence de cette métrique peut indiquer des problèmes de centralisation.
- **CentralizationError**: Configurez des alarmes en cas d'erreur dans le processus de centralisation afin d'identifier et de résoudre rapidement les problèmes.
- **CentralizationThrottled**: surveillez les événements de limitation susceptibles d'avoir un impact sur les performances de réplication des journaux.

Pour une liste complète des mesures de centralisation disponibles et de leurs dimensions, voir [Mesures et dimensions de centralisation](#).

Si les journaux ne sont pas centralisés comme prévu, passez en revue les scénarios courants suivants qui peuvent empêcher la centralisation des journaux.

### Données du journal historique

La fonctionnalité de centralisation CloudWatch des journaux traite uniquement les nouvelles données de journal qui arrivent dans les comptes sources une fois que vous avez créé la règle de centralisation. Les données des journaux historiques (journaux qui existaient avant la création des règles) ne sont pas centralisées.

### Autorisations clés KMS

Les règles de centralisation ne parviennent pas à transmettre les journaux du compte source aux groupes de journaux de destination si la clé KMS fournie dans la règle de centralisation n'autorise pas CloudWatch Logs à l'utiliser. Assurez-vous que la politique des clés KMS accorde les autorisations nécessaires aux CloudWatch journaux. Pour de plus amples informations, veuillez consulter [Étape 2 : Définition des autorisations sur la clé KMS](#).

## Configuration de la clé KMS gérée par le client

Si vous avez sélectionné **Ne pas centraliser les groupes de journaux chiffrés avec une clé KMS gérée par le client** lors de la création de la règle, les événements de journal provenant des groupes de journaux source chiffrés avec une clé KMS gérée par le client seront ignorés et non centralisés.

## Non-concordance du chiffrement de destination

Si le groupe de journaux de destination existe déjà avec une configuration de chiffrement KMS différente de celle spécifiée par la règle de centralisation et que la résolution du conflit est définie sur **SKIP**, les enregistrements seront supprimés et une `DestinationEncryptionMismatch` erreur sera émise. Par exemple, cela se produit lorsque la destination dispose d'un chiffrement par défaut mais que la règle spécifie une clé KMS gérée par le client.

## Accès sécurisé non activé

L'accès sécurisé doit être activé AWS Organizations pour CloudWatch le compte de gestion et le compte de destination afin de fournir un accès aux données du journal.

## Critères de sélection de la source

Vérifiez que les critères de sélection des sources de votre règle de centralisation sont correctement configurés :

- **Comptes et régions** : assurez-vous que les comptes sources et les régions d'où proviennent les journaux sont inclus dans la règle. Les groupes de journaux provenant de comptes ou de régions non spécifiés dans la règle ne seront pas centralisés.
- **Filtres de groupes de journaux** : si vous avez configuré des filtres de groupes de journaux, seuls les groupes de journaux correspondant aux critères spécifiés seront centralisés. Vérifiez que vos critères de sélection des groupes de journaux incluent les groupes de journaux que vous souhaitez centraliser.
- **Appartenance à l'organisation** : les comptes source et de destination doivent appartenir à la même AWS Organizations organisation. Les comptes extérieurs à l'organisation ne peuvent pas participer à la centralisation.

## Limite de quota de groupes de journaux atteinte

Si le compte de destination a atteint son quota de groupes de journaux, il est impossible de créer de nouveaux groupes de journaux à des fins de centralisation. Vérifiez que le compte de destination dispose d'un quota suffisant pour accueillir les groupes de journaux centralisés.

provenant de tous les comptes sources. Vous pouvez demander une augmentation de quota si nécessaire.

### La limite de longueur du nom du flux de log est dépassée

Les noms de flux de journaux sont soumis à des restrictions de longueur maximale. Lorsque la centralisation réplique les flux de journal vers le compte de destination, un suffixe est ajouté au nom du flux de journal. Si le nom du flux de journal obtenu dépasse la longueur maximale autorisée, les enregistrements seront supprimés et une `InvalidLogStream` erreur sera envoyée au compte client.

### État de santé des règles

Vérifiez l'état de santé des règles de centralisation dans la console ou à l'aide de `GetCentralizationRuleForOrganizationAPI`. Si la règle est marquée comme étant `INSTABLE`, consultez le `FailureReason` champ pour obtenir des informations spécifiques sur le problème.

Pour diagnostiquer les problèmes de centralisation, vérifiez l'état de santé des règles de centralisation dans la console, vérifiez les CloudWatch métriques pour détecter les erreurs et les limitations, et examinez les AWS CloudTrail journaux pour détecter les échecs d'appels d'API. Pour plus d'informations sur les métriques de centralisation, consultez [Mesures et dimensions de centralisation](#).

# 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.

Pour les entreprises qui ont besoin de consolider les données de journaux provenant de plusieurs comptes et régions, vous pouvez utiliser la centralisation CloudWatch des journaux pour répliquer automatiquement les groupes de journaux sur un compte central. Pour de plus amples informations, veuillez consulter [Centralisation des journaux inter-comptes et inter-régions](#).

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 instance Amazon EC2 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 volet de navigation, choisissez Log Management.
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 Recently 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'instances Amazon EC2 et de serveurs sur site avec l' CloudWatch agent dans le guide](#) de l'utilisateur Amazon CloudWatch .
- AWS CLI: [put-log-events](#) télécharge des lots d'événements du journal dans Logs. CloudWatch
- Par programmation : l'[PutLogEvents](#) API 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 volet de navigation, choisissez Log Management.
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 volet de navigation, choisissez Log Management.
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 volet de navigation, choisissez Log Management.
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 effectuer une recherche dans les entrées du 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-events](#) commande 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.

## Protection des groupes de journaux contre la suppression

Vous pouvez éventuellement activer la protection contre la suppression pour empêcher la suppression accidentelle de groupes de journaux importants. Pour obtenir des informations détaillées sur la protection contre la suppression, consultez [Protection des groupes de journaux contre la suppression](#).

## Protection des groupes de journaux contre la suppression

### Activer la protection contre les suppressions

Vous pouvez activer la protection contre la suppression lors de la création d'un nouveau groupe de journaux ou sur des groupes de journaux existants. Lors de la création du groupe de journaux, sélectionnez « Protection contre la suppression activée » ou en transmettant le paramètre - -

deletion-protection-enabled. Par défaut, la protection contre la suppression n'est pas activée.

Pour activer ou désactiver la protection contre la suppression sur un groupe de journaux existant (console)

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation, choisissez Log Management.
3. Sélectionnez le groupe de journaux que vous souhaitez protéger.
4. Choisissez Actions, puis Modifier la protection contre la suppression.
5. Dans la boîte de dialogue, passez en revue puis soumettez les modifications.

Si vous utilisez le AWS CLI, pour activer la protection contre la suppression sur un groupe de journaux existant :

```
aws logs put-log-group-deletion-protection \  
--log-group-identifiant "/my-application/logs" \  
--deletion-protection-enabled
```

Pour supprimer la protection contre la suppression d'un groupe de journaux existant, procédez comme suit :

```
aws logs put-log-group-deletion-protection \  
--log-group-identifiant "/my-application/logs" \  
--no-deletion-protection-enabled
```

## Gestion des erreurs

Si vous tentez de supprimer un groupe de journaux avec la protection contre la suppression activée, vous recevez le message suivant : « Impossible de supprimer le groupe de journaux avec la protection contre la suppression activée. ValidationException Désactivez d'abord la protection contre la suppression. »

## 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 les valeurs des balises distinguent les majuscules et minuscules.
- 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 remplace 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](#).

## Restrictions

- 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-policy](#) commande 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-policy](#) commande 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-groups](#) commande 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-key](#) commande 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": "*"br/>    },  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "logs.region.amazonaws.com"  
      },  
      "Action": [  
        "kms:Decrypt"  
      ],  
      "Resource": "*"br/>    }br/>  ]  
}
```

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 est pris en charge pour les groupes de journaux dans les classes de journaux d'accès standard et peu fréquent. Pour plus d'informations sur les classes de log, consultez [Classes 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 `LogEventsWithFindings` métrique 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, consultez [Surveillance 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 \(PII\)](#)
  - [Identifiants de 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 des journaux détecte ensuite 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 `LogEventsWithFindings` métrique 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, consultez [Surveillance 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 | <code>logs:PutAccountPolicy</code> | *         |



| Opération                                                                                            | Autorisation IAM requise      | Ressource                                                  |
|------------------------------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------|
| Créez une politique de protection des données avec CloudWatch les journaux comme destination d'audit | logs:PutDataProtectionPolicy  | *                                                          |
|                                                                                                      | logs:PutAccountPolicy         | *                                                          |
|                                                                                                      | logs:PutDataProtectionPolicy  | *                                                          |
|                                                                                                      | logs:CreateLogDelivery        | *                                                          |
|                                                                                                      | logs:PutResourcePolicy        | *                                                          |
|                                                                                                      | logs:DescribeResourcePolicies | *                                                          |
| Créez une politique de protection des données avec Firehose comme destination d'audit                | logs:DescribeLogGroups        | *                                                          |
|                                                                                                      | logs:PutAccountPolicy         | *                                                          |
|                                                                                                      | logs:PutDataProtectionPolicy  | *                                                          |
|                                                                                                      | logs:CreateLogDelivery        | *                                                          |
|                                                                                                      | firehose:TagDeliveryStream    | arn:aws:logs:::deliverystream/ <i>YOUR_DELIVERY_STREAM</i> |

| Opération                                                                              | Autorisation IAM requise                        | Ressource                                     |
|----------------------------------------------------------------------------------------|-------------------------------------------------|-----------------------------------------------|
| Créer une politique de protection des données avec Amazon S3 comme destination d'audit | <code>logs:PutAccountPolicy</code>              | *                                             |
|                                                                                        | <code>logs:PutDataProtectionPolicy</code>       | *                                             |
|                                                                                        | <code>logs:CreateLogDelivery</code>             | *                                             |
|                                                                                        | <code>s3:GetBucketPolicy</code>                 | <code>arn:aws:s3::: <i>YOUR_BUCKET</i></code> |
|                                                                                        | <code>s3:PutBucketPolicy</code>                 | <code>arn:aws:s3::: <i>YOUR_BUCKET</i></code> |
| Démasquer les événements du journal masqués dans un groupe de journaux spécifié        | <code>logs:Unmask</code>                        | <code>arn:aws:logs::log-group:*</code>        |
| Afficher une politique de protection des données existante                             | <code>logs:GetDataProtectionPolicy</code>       | *                                             |
| Supprimer une politique de protection des données                                      | <code>logs&gt;DeleteAccountPolicy</code>        | *                                             |
|                                                                                        | <code>logs&gt;DeleteDataProtectionPolicy</code> | *                                             |

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: <b>YOUR_LOG_GROUP</b> :*                                      |
| Créez une politique de protection des données avec CloudWatch les journaux comme destination d'audit | logs:PutDataProtectionPolicy<br>logs:CreateLogDelivery<br>logs:PutResourcePolicy<br>logs:DescribeResourcePolicies<br>logs:DescribeLogGroups | arn:aws:logs::log-group: <b>YOUR_LOG_GROUP</b> :*<br>*<br>*<br>*<br>*                  |
| Créez une politique de protection des données avec Firehose comme destination d'audit                | logs:PutDataProtectionPolicy<br>logs:CreateLogDelivery<br>firehose:TagDeliveryStream                                                        | arn:aws:logs::log-group: <b>YOUR_LOG_GROUP</b> :*<br>*<br>arn:aws:logs::deliverystream |

| Opération                                                                                   | Autorisation IAM requise                                                                                       | Ressource                                                                                                                                  |
|---------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                             |                                                                                                                | arn:aws:logs::log-group: <i>YOUR_LOG_GROUP</i> :*                                                                                          |
| Création d'une politique de protection des données avec Amazon S3 comme destination d'audit | logs:PutDataProtectionPolicy<br><br>logs:CreateLogDelivery<br><br>s3:GetBucketPolicy<br><br>s3:PutBucketPolicy | arn:aws:logs::log-group: <i>YOUR_LOG_GROUP</i> :*<br><br>*<br><br>arn:aws:s3::: <i>YOUR_BUCKET</i><br><br>arn:aws:s3::: <i>YOUR_BUCKET</i> |
| Démasquage des événements du journal masqués                                                | logs:Unmask                                                                                                    | arn:aws:logs::log-group: <i>YOUR_LOG_GROUP</i> :*                                                                                          |
| Affichage d'une politique de protection des données existante                               | logs:GetDataProtectionPolicy                                                                                   | arn:aws:logs::log-group: <i>YOUR_LOG_GROUP</i> :*                                                                                          |
| Supprimer une politique de protection des données                                           | logs>DeleteDataProtectionPolicy                                                                                | arn:aws:logs::log-group: <i>YOUR_LOG_GROUP</i> :*                                                                                          |

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:us-east-1:111122223333:deliverystream/delivery-  
stream-name",
        "arn:aws:s3:::amzn-s3-demo-destination-bucket",
        "arn:aws:logs:us-east-1:111122223333: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 [Identifiants de 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 Activer la protection des données (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 `DataIdentifier` et une propriété `Operation` avec une action `Audit`. Le tableau `DataIdentifier` 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 propriété `Operation` associée à une action `Audit` 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 `DataIdentifier` et une propriété `Operation` avec une action `Deidentify`. Le tableau `DataIdentifier` doit correspondre exactement au tableau `DataIdentifier` 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, consultez [Utilisation 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 audité 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, consultez [Types 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 [Identifiants de 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 `DataIdentifier` et une `Operation` propriété avec une `Audit` action. Le `DataIdentifier` 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 `DataIdentifier` et une propriété `Operation` avec une action `Deidentify`. Le tableau `DataIdentifier` doit correspondre exactement au `DataIdentifier` 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 13<sup>e</sup> caractère d'événement du journal et se termine au 26<sup>e</sup> caractère. La deuxième adresse e-mail est comprise entre le 30<sup>e</sup> et le 43<sup>e</sup> 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 \(PII\)](#)
- [Identifiants de 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               | ÉTATS-UNIS                                                                                                                 |
| ElectoralRollNumber               | Personnel           | Go                                                                                                                         |
| HealthInsuranceCardNumber         | Santé               | UE                                                                                                                         |
| HealthInsuranceClaimNumber        | Santé               | ÉTATS-UNIS                                                                                                                 |
| HealthInsuranceNumber             | Santé               | FR                                                                                                                         |
| HealthcareProcedureCode           | Santé               | ÉTATS-UNIS                                                                                                                 |
| IndividualTaxIdentificationNumber | Personnel           | ÉTATS-UNIS                                                                                                                 |
| InseeCode                         | Personnel           | FR                                                                                                                         |
| MedicareBeneficiaryNumber         | Santé               | ÉTATS-UNIS                                                                                                                 |
| NationalDrugCode                  | Santé               | ÉTATS-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-awscr<br>edential | Tous            |
| Clé privée OpenSSH      | OpenSSHPrivateKey               | Aucune                                                                                     | Tous            |
| Clé privée PGP          | PgpPrivateKey                   | Aucune                                                                                     | Tous            |
| Clés privées PKCS       | PkcsPrivateKey                  | Aucune                                                                                     | Tous            |
| Clé privée PuTTY        | PuttyPrivateKey                 | Aucune                                                                                     | 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                       | Aucune         | 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, Allemagne, 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'Amérique | de compte bancaire international (IBANs) composés d'un maximum de 34 caractères alphanumériques, y compris des éléments tels que les codes de pays. |
| Date d'expiration de la carte de crédit | CreditCardExpiration            | expiry, expiration                                                                                     | 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 suivantes : American Express, Dankort, Diner's Club, Discover, Electron, Japanese Card Bureau (JCB), Mastercard et Visa. UnionPay |
| Code de vérification de carte de crédit | CreditCardSecurity Code         | 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.

| Country     | 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.

```
1220000000000003, 2222405343248877, 2222990905257051, 2223007648726984,
2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
36148900647913,
36700102000000, 371449635398431, 378282246310005, 378734493671000, 38520000023237,
4012888888881881,
4111111111111111, 4222222222222, 4444333322221111, 4462030000000000, 4484070000000000,
49118300000000,
4917300800000000, 4917610000000000, 4917610000000000003, 5019717010103742,
5105105105105100,
5111010030175156, 5185540810000019, 5200828282828210, 5204230080000017,
5204740009900014, 5420923878724339,
5454545454545454, 5455330760000018, 55069004900000436, 55069004900000444,
55069005100000234, 5506920809243667,
5506922400634930, 5506927427317625, 5553042241984105, 555553753048194,
555555555554444, 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 , krankenversicherungsnnummer , 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 , sjukförsäkringsnummer, sjukförsäkringskort , suomi-ehic-numero , tarjeta de salud, terveystkortti , 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#, hcn, hcn#, 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 lequel | 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 físicas, 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 nombreux pays. Pour plus de détails, consultez 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. | Allemagne, 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, Allemagne, Italie, Espagne, Royaume-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                                                                                                                                                                                              |
|---------------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Numéro de téléphone | PhoneNumber                     | <p>Brésil : les mots clés incluent également : cel, celular, fone, móvel, número residencial , numero residencial , telefone</p> <p>Autres : cell, contact, fax, fax number, mobile, phone, phone number, tel, telephone , telephone number</p> | Brésil, Canada, France, Allemagne, Italie, Espagne, Royaume-Uni, États-Unis | <p>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 nécessaire que le numéro inclue un code de pays. Si un mot</p> |

| 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                      | Aucune                                                                                                                                                                                        | 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                             | <p>Espagne : número de la seguridad social, social security no., social security no. número de la seguridad social, social security number, social securityno# , ssn, ssn#</p> <p>États-Unis : social security, ss#, ssn</p> | Espagne<br>États-Unis                   |                                                                                                                      |
| Numéro d'identification ou de référence du contribuable | TaxId                           | <p>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.</p> <p>.</p>                | France, Allemagne, Espagne, Royaume-Uni | <p>Cela inclut TIN (France) ; Steueridentifikationsnummer (Allemagne) ; CIF (Espagne) et TRN, UTR (Royaume-Uni).</p> |



| 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                         | Aucune          | Australie, Canada, France, Allemagne, 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 | EmailAddress                    | Aucune          | N'importe lequel                                                               |                                                                                                                                               |

| 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 ,<br>coordinates ,lat<br>long, latitude<br>longitude ,location,<br>position | N'importe lequel | CloudWatch Logs 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°56.916 8'N 87°39.318 7'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                            | Aucune         | N'importe lequel | CloudWatch Logs ne peut détecter que les noms complets. Le support est limité aux jeux de caractère s latins. |

| Type de données                           | ID de l'identifiant des données     | Mot-clé requis                                                                                                                                                                                                                                                                                                   | Pays et régions     | Remarques                                                                                                                                                                                                                                                                        |
|-------------------------------------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Numéro d'identification de véhicule (VIN) | VehicleId<br>entificat<br>ionNumber | Fahrgestellnummer ,<br>niv, numarul de<br>identificare ,<br>numarul seriei de<br>sasiu, serie sasiu,<br>numer VIN, Número de<br>Identificação do<br>Veículo, Número de<br>Identificación de<br>Automóviles , numéro<br>d'identification<br>du véhicule, vehicle<br>identification<br>number, vin, VIN<br>numerus | N'importe<br>lequel | CloudWatc<br>h Les<br>journaux<br>peuvent<br>détecter<br>VINs<br>une<br>séquence<br>de 17<br>caractère<br>s et<br>respecter<br>les<br>normes<br>ISO<br>3779<br>et<br>3780.<br>Ces<br>normes<br>ont été<br>conçues<br>pour<br>être<br>utilisées<br>dans<br>le<br>monde<br>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 průkaz, 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, fuhrerschein, fuhrerschein- nr, fuhrerschein- nr, fuhrersch einnummer, fuhrerscheinnnummer    |
| 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, führungsschein                                                                                                                                                                                                  |
| Malte          | licenzja 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 conducir, el carnet de conducir, licencia conducir, licencia de manejo, número carnet conducir, número de carnet de conducir, número de permiso conducir, número de permiso de conducir, número licencia conducir, número permiso conducir, permiso conducción, permiso conducir, 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 fiscale, 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#, nationali dno#, 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         | passport, passport#, 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 física, cadastro de pessoas físicas, cadastro de pessoas físicas, 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, steueridentifikationsnummer, 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 contribuable, 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-identifiant/Address
```

```
arn:aws:dataprotection::aws:data-identifiant/CepCode-BR
```

```
arn:aws:dataprotection::aws:data-identifiant/Cnpj-BR
```

```
arn:aws:dataprotection::aws:data-identifiant/CpfCode-BR
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-AT
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-AU
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-BE
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-BG
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-CA
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-CY
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-CZ
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-DE
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-DK
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-EE
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-ES
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-FI
```

```
arn:aws:dataprotection::aws:data-identifiant/DriversLicense-FR
```

```
arn:aws:dataprotection::aws:data-identifiant/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
```



## Identifiants de 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": {
            "Mask": "REDACT"
          }
        }
      }
    }
  ]
}

```

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 [FilterLogEvents](#)actions [GetLogEvents](#)et renvoient uniquement les versions d'origine des événements du journal, avant leur transformation.

 Important

Bien qu'un seul événement de journal PutLogEvents autorise jusqu'à 1 Mo, la transformation des journaux ne peut gérer que les événements de journal d'une taille inférieure à 512 Ko. Tout événement de journal supérieur à 512 Ko échouera lors de la transformation et émettra une erreur. La taille totale de PutLogEvents peut toujours dépasser 512 Ko.

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 configurables de type analyseur](#)

- [Processeurs intégrés pour les AWS journaux vendus](#)
- [Processeurs String Mutate](#)
- [Processeurs de mutation JSON](#)
- [Processeurs de conversion de types de données](#)
- [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.
- Un transformateur doit comporter au moins un processeur de type analyseur. 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.
- Chaque événement du journal DOIT être inférieur à 512 Ko. La taille totale des événements du journal peut toujours dépasser 512 Ko.

## 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 dans [Processeurs configurables de type analyseur](#).

7. Pour ajouter un autre processeur, choisissez Sélectionner un processeur. Sélectionnez ensuite le processeur que vous souhaitez dans la zone Processeur et renseignez les paramètres de configuration.

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 dans [Processeurs 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.

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 tâches 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 transformateur de log

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 que vous souhaitez dans la zone Processeur et renseignez les paramètres de configuration.

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 configurables de type analyseur

Cette section contient des informations sur les processeurs d'analyse de données configurables que vous pouvez utiliser dans un transformateur d'événements de journalisation.

## Table des matières

- [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](#)

## 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       | Restrictions                                                                  |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------------|-------------------------------------------------------------------------------|
| 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, <code>store.book</code> | Non           | @message         | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |
| destination | Le champ de destination du JSON analysé                                                                                                                                       | Non           | Parent JSON node | Longueur maximale : 128<br><br>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 | Restrictions                                                                                                                                                                                                  | Remarques                                                 |
|--------|-------------------------------------------------------------|---------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| source | Chemin du champ sur lequel appliquer la correspondance Grok | Non           | @message   | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3                                                                                                                                 |                                                           |
| match  | Le modèle de grok à comparer à l'événement du journal       | Oui           |            | Longueur maximale : 512<br><br>Motifs de rainure maximum : 20<br><br>Certains types de modèles de grok ont des limites d'utilisation individuelles. Toute combinaison des modèles suivants peut être utilisée | <a href="#">Voir tous les modèles Grok pris en charge</a> |

| Champ | Description | Obligatoire ? | Par défaut | Restrictions                                                                                                                                                                                                                                                                                                                                                                                                                           | Remarques |
|-------|-------------|---------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
|       |             |               |            | <p>jusqu'à cinq fois :<br/> {URI, URIPARAM,<br/> URIPATHPARAM,<br/> SPACE, DATA,<br/> GREEDYDATA,<br/> GREEDYDATA_MULTILINE}</p> <p>Les modèles Grok ne prennent pas en charge les conversions de type.</p> <p>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.</p> |           |

## 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 grok](#) prédéfinis 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<br><br>Modèle : %{USERNAME:name}<br><br>Sortie : {"name": "user123.name-TEST"} |
| INT                              | Correspond à un signe plus ou moins facultatif suivi d'un ou de plusieurs chiffres.                                                                                                                              | 20                        | Entrée : -456<br><br>Modèle : %{INT:num}                                                                  |

| Motif Grok | Description                                                                                                                                 | Limite maximale de motifs | Exemple                                                                                                          |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------|
| BASE10NUM  | Correspond à un entier ou à un nombre à virgule flottante avec un signe et une virgule décimale facultatifs                                 | 20                        | Sortie : {"num": "-456"}<br><br>Entrée : -0.67<br><br>Modèle : %{BASE10NUM:num}<br><br>Sortie : {"num": "-0.67"} |
| BASE16NUM  | Fait correspondre les nombres décimaux et hexadécimaux avec un signe facultatif (+ ou -) et un préfixe 0x facultatif                        | 20                        | Entrée : +0xA1B2<br><br>Modèle : %{BASE16NUM:num}<br><br>Sortie : {"num": "+0xA1B2"}                             |
| POINTANT   | 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<br><br>Modèle : %{POSINT:num}<br><br>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<br><br>Modèle : %{NONNEGINT:num}<br><br>Sortie : {"num": "007"}                                     |

| Motif Grok   | Description                                                                                                                                    | Limite maximale de motifs | Exemple                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------|
| 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<br><br>Modèle : %{WORD:user}<br><br>Sortie : {"user": "user_123"}                      |
| PAS D'ESPACE | Correspond à un ou plusieurs caractères autres que des espaces blancs.                                                                         | 5                         | Entrée : hello_world123<br><br>Modèle : %{NOTSPACE:msg}<br><br>Sortie : {"msg": "hello_world123"}        |
| SPACE        | Correspond à zéro ou plusieurs caractères d'espacement.                                                                                        | 5                         | Entrée : " "<br><br>Modèle : %{SPACE:extra}<br><br>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<br><br>Modèle : %{DATA:x}<br>%{DATA:y}<br><br>Sortie : {"x": "abc", "y": "def ghi"} |

| Motif Grok                        | Description                                                                                             | Limite maximale de motifs | Exemple                                                                                                                                           |
|-----------------------------------|---------------------------------------------------------------------------------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| DONNÉES<br>GOURMANDE<br>S         | Correspond à n'importe quel caractère (sauf la nouvelle ligne) zéro fois ou plus, gourmand.             | 5                         | <p>Entrée : abc def ghi</p> <p>Modèle : %{GREEDYD<br/>ATA:x} %{GREEDYD<br/>ATA:y}</p> <p>Sortie : {"x": "abc def",<br/>"y": "ghi"}</p>            |
| GREEDYDAT<br>A_MULTILI<br>NE      | Correspond à n'importe quel personnage (y compris une nouvelle ligne) zéro fois ou plus, gourmand.      | 1                         | <p>Entrée :</p> <p>abc</p> <p>def</p> <p>ghi</p> <p>Modèle : %{GREEDYD<br/>ATA_MULTILINE:data}</p> <p>Sortie : {"data": "abc<br/>\ndef\nghi"}</p> |
| CHAÎNE<br>ENTRE<br>GUILLEMET<br>S | Fait correspondre les chaînes entre guillemets (guillemets simples ou doubles) aux caractères échappés. | 20                        | <p>Entrée : "Hello, world!"</p> <p>Modèle : %{QUOTEDS<br/>TRING:msg}</p> <p>Sortie : {"msg": "Hello,<br/>world!"}</p>                             |

| Motif Grok | Description                                                                                                                                                                                         | Limite maximale de motifs | Exemple                                                                                                                                        |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 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                        | <p>Entrée : 550e8400-e29b-41d4-a716-446655440000</p> <p>Modèle : %{UUID:id}</p> <p>Sortie : {"id": "550e8400-e29b-41d4-a716-446655440000"}</p> |
| URNE       | Correspond à la syntaxe URN (Uniform Resource Name).                                                                                                                                                | 20                        | <p>Entrée : urn:isbn:0451450523</p> <p>Modèle : %{URN:urn}</p> <p>Sortie : {"urn": "urn:isbn:0451450523"}</p>                                  |

### AWS motifs de grok

| Modèle | Description                                                                                                                                                                                                                   | Limite maximale de motifs | Exemple                                                                                                                                        |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| ARN    | Correspond aux noms de ressources AWS Amazon (ARNs), en capturant la partition (awsaws-cn, ouaws-us-gov ), le service, la région, l'ID de compte et jusqu'à 5 identifiants de ressources hiérarchiques séparés par des barres | 5                         | <p>Entrée : arn:aws:iam:us-east-1:123456789012:user/johndoe</p> <p>Modèle : %{ARN:arn}</p> <p>Sortie : {"arn": "arn:aws:iam:us-east-1:1234</p> |



| Modèle | Description                                                                            | Limite maximale de motifs | Exemple                    |
|--------|----------------------------------------------------------------------------------------|---------------------------|----------------------------|
|        | obliques. Cela ne correspondra pas ARNs aux informations manquantes entre deux points. |                           | 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<br><br>Modèle : %{CISCOMAC:MacAddress}<br><br>Sortie : {"MacAddress": "0123.4567.89AB"}             |
| WINDOWSMA<br>C | Correspond à une adresse MAC au format hexadécimal avec des tirets   | 20                        | Entrée : 01-23-45-67-89-AB<br><br>Modèle : %{WINDOWS<br>MAC:MacAddress}<br><br>Sortie : {"MacAddress": "01-23-45-67-89-AB"} |
| MAC<br>COMMUN  | Correspond à une adresse MAC au format hexadécimal avec deux points. | 20                        | Entrée : 01:23:45:67:89:AB<br><br>Modèle : %{COMMONMAC:MacAddress}                                                          |

| Motif Grok | Description                                                                                                           | Limite maximale de motifs | Exemple                                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
|            |                                                                                                                       |                           | 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<br><br>Modèle : %{MAC:m1}<br><br>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<br><br>Modèle : %{IPV6:ip}<br><br>Sortie : {"ip": "2001:db8:3333:4444:5555:6666:7777:8888"} |
| IPV4       | Correspond à une IPv4 adresse.                                                                                        | 20                        | Entrée : 192.168.0.1<br><br>Modèle : %{IPV4:ip}<br><br>Sortie : {"ip": "192.168.0.1"}                                                       |
| 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<br><br>Modèle : %{IP:ip}<br><br>Sortie : {"ip": "192.168.0.1"}                                                         |

| Motif Grok       | Description                                                                                                                                                                                           | Limite maximale de motifs | Exemple                                                                                                                                                |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| HOSTNAME ou HOST | Correspond aux noms de domaine, y compris les sous-domaines                                                                                                                                           | 5                         | <p>Entrée : server-01.internal-network.local</p> <p>Modèle : %{HOST:host}</p> <p>Sortie : {"host": "server-01.internal-network.local"}</p>             |
| IP OU HÔTE       | Correspond à un nom d'hôte ou à une adresse IP                                                                                                                                                        | 5                         | <p>Entrée : 2001:db8:3333:4444:5555:6666:7777:8888</p> <p>Modèle : %{IPORHOST:ip}</p> <p>Sortie : {"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}</p> |
| 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                         | <p>Entrée : 192.168.1.1:8080</p> <p>Modèle : %{HOSTPORT:ip}</p> <p>Sortie : {"ip": "192.168.1.1:8080", "PORT": "8080"}</p>                             |

| 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 %<br>{IPORHOST}, éventuellement suivi de deux points et d'un numéro de port, capturant le port en tant que « port » s'il est présent. | 5                         | <p>Entrée : <code>example.com:443<br/>10.0.0.1</code></p> <p>Modèle : <code>%{URIHOST<br/>:host} %{URIHOST:ip}</code></p> <p>Sortie : <code>{"host": "<br/>example.com:443", "<br/>port": "443", "ip": "<br/>10.0.0.1"}</code></p> |

### 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                        | <p>Entrée : <code>/search?q=regex</code></p> <p>Modèle : <code>%{UNIXPATH:path}</code></p> <p>Sortie : <code>{"path": "/<br/>search?q=regex"}</code></p>                   |
| WINPATH    | Correspond aux chemins de fichiers Windows.                                       | 5                         | <p>Entrée : <code>C:\Users\John<br/>\Documents\file.txt</code></p> <p>Modèle : <code>%{WINPATH:path}</code></p> <p>Sortie : <code>{"path": "C:<br/>\Users\John\</code></p> |

| Motif Grok | Description                                                                                                                    | Limite maximale de motifs | Exemple                                                                                                                                     |
|------------|--------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
|            |                                                                                                                                |                           | Documents\\file.txt"}<br>}                                                                                                                  |
| PATH       | Correspond à une URL ou à un chemin de fichier Windows                                                                         | 5                         | Entrée : /search?q=regex<br><br>Modèle : %{PATH:path}<br><br>Sortie : {"path":"/search?q=regex"}                                            |
| TTY        | Correspond aux chemins des périphériques Unix pour les terminaux et les pseudo-terminaux.                                      | 20                        | Entrée : /dev/tty1<br><br>Modèle : %{TTY:path}<br><br>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<br><br>Modèle : %{URIPROTO:protocol}<br><br>Sortie : {"protocol":"web+transformer"}                                |
| URIPATH    | Correspond au composant de chemin d'un URI                                                                                     | 20                        | Entrée : /category/sub-category/product_name<br><br>Modèle : %{URIPATH:path}<br><br>Sortie : {"path":"/category/sub-category/product_name"} |

| Motif Grok        | Description                                                                | Limite maximale de motifs | Exemple                                                                                                                                                                                     |
|-------------------|----------------------------------------------------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| URIPARAM          | Correspond aux paramètres de requête d'URL                                 | 5                         | <p>Entrée : ?param1=value1&amp;param2=value2</p> <p>Modèle : %{URIPARAM:url}</p> <p>Sortie : {"url":"?param1=value1&amp;param2=value2"}</p>                                                 |
| PARAMÈTRE URIPATH | Correspond à un chemin d'URI éventuellement suivi de paramètres de requête | 5                         | <p>Entrée : /category/sub-category/product?id=12345&amp;color=red</p> <p>Modèle : %{URIPATHPARAM:path}</p> <p>Sortie : {"path":"/category/sub-category/product?id=12345&amp;color=red"}</p> |

| Motif Grok | Description                 | Limite maximale de motifs | Exemple                                                                                                                                                                                                                                                                             |
|------------|-----------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| URI        | Correspond à un URI complet | 5                         | <p>Entrée : <code>https://user:password@example.com/path/to/resource?param1=value1&amp;param2=value2</code></p> <p>Modèle : <code>%{URI:uri}</code></p> <p>Sortie : <code>{"path": "https://user:password@example.com/path/to/resource?param1=value1&amp;param2=value2"}</code></p> |

### 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                        | <p>Entrée : Jan</p> <p>Modèle : <code>%{MONTH:month}</code></p> <p>Sortie : <code>{"month": "Jan"}</code></p> <p>Entrée : January</p> <p>Modèle : <code>%{MONTH:month}</code></p> <p>Sortie : <code>{"month": "January"}</code></p> |

| 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                        | <p>Entrée : 5</p> <p>Modèle : %{MONTHNUM:month}</p> <p>Sortie : {"month": "5"}</p> <p>Entrée : 05</p> <p>Modèle : %{MONTHNUM:month}</p> <p>Sortie : {"month": "05"}</p> |
| MONTHNUM     | Correspond aux numéros de mois à deux chiffres compris entre 01 et 12.                                            | 20                        | <p>Entrée : 05</p> <p>Modèle : %{MONTHNUM2:month}</p> <p>Sortie : {"month": "05"}</p>                                                                                   |
| JOUR DU MOIS | Correspond au jour du mois compris entre 1 et 31, avec un zéro en tête optionnel.                                 | 20                        | <p>Entrée : 31</p> <p>Modèle : %{MONTHDAY:monthDay}</p> <p>Sortie : {"monthDay": "31"}</p>                                                                              |



| Motif Grok | Description                                                                       | Limite maximale de motifs | Exemple                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| YEAR       | Correspond à l'année en deux ou quatre chiffres                                   | 20                        | <p>Entrée : 2024</p> <p>Modèle : %{YEAR:year}</p> <p>Sortie : {"year": "2024"}</p> <p>Entrée : 24</p> <p>Modèle : %{YEAR:year}</p> <p>Sortie : {"year": "24"}</p> |
| DAY        | Correspond aux noms de jour complets ou abrégés.                                  | 20                        | <p>Entrée : Tuesday</p> <p>Modèle : %{DAY:day}</p> <p>Sortie : {"day": "Tuesday"}</p>                                                                             |
| HOUR       | Correspond à l'heure au format 24 heures avec un zéro (0) 0-23 en tête en option. | 20                        | <p>Entrée : 22</p> <p>Modèle : %{HOUR:hour}</p> <p>Sortie : {"hour": "22"}</p>                                                                                    |
| MINUTE     | Correspond aux minutes (00-59).                                                   | 20                        | <p>Entrée : 59</p> <p>Modèle : %{MINUTE:min}</p> <p>Sortie : {"min": "59"}</p>                                                                                    |

| 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                        | <p>Entrée : 3</p> <p>Modèle : %{SECOND:second}</p> <p>Sortie : {"second": "3"}</p> <p>Entrée : 30.5</p> <p>Modèle : %{SECOND:minSec}</p> <p>Sortie : {"minSec": "30.5"}</p> <p>Entrée : 30:5</p> <p>Modèle : %{SECOND:minSec}</p> <p>Sortie : {"minSec": "30:5"}</p> |
| 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                        | <p>Entrée : 09:45:32</p> <p>Modèle : %{TIME:time}</p> <p>Sortie : {"time": "09:45:32"}</p>                                                                                                                                                                           |

| 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                        | <p>Entrée : 11/23/2024</p> <p>Modèle : %{DATE_US :date}</p> <p>Sortie : {"date": "11/23/2024"}</p> <p>Entrée : 1-01-24</p> <p>Modèle : %{DATE_US :date}</p> <p>Sortie : {"date": "1-01-24"}</p> |
| 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                        | <p>Entrée : 23/11/2024</p> <p>Modèle : %{DATE_EU :date}</p> <p>Sortie : {"date": "23/11/2024"}</p> <p>Entrée : 1.01.24</p> <p>Modèle : %{DATE_EU :date}</p> <p>Sortie : {"date": "1.01.24"}</p> |

| Motif Grok                    | Description                                                                                                                                                                            | Limite maximale de motifs | Exemple                                                                                                                                                                                                                                                                     |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ISO8601_F<br>USEAU<br>HORAIRE | Correspond au décalage UTC « Z » ou au décalage de fuseau horaire avec deux points optionnels au format [+-(H) H (: ) mm.                                                              | 20                        | <p>Entrée : +05:30</p> <p>Modèle : %{ISO8601_TIMEZONE:tz}</p> <p>Sortie : {"tz":"+05:30"}</p> <p>Entrée : -530</p> <p>Modèle : %{ISO8601_TIMEZONE:tz}</p> <p>Sortie : {"tz":"-530"}</p> <p>Entrée : Z</p> <p>Modèle : %{ISO8601_TIMEZONE:tz}</p> <p>Sortie : {"tz":"Z"}</p> |
| ISO8601_S<br>ECONDE           | 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                        | <p>Entrée : 60</p> <p>Modèle : %{ISO8601_SECOND:second}</p> <p>Sortie : {"second":"60"}</p>                                                                                                                                                                                 |

| Motif Grok        | Description                                                                                                                                               | Limite maximale de motifs | Exemple                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HORODATAE_ISO8601 | Correspond au format ISO8601 date/heure (yy) yy- (M) M- (d) dT (H) H:mm : ((s) s) (Z  [+ -] (H) H:mm) avec les secondes et le fuseau horaire facultatifs. | 20                        | <p>Entrée : 2023-05-15T14:30:00+05:30</p> <p>Modèle : %{TIMESTAMP_ISO8601:timestamp}</p> <p>Sortie : {"timestamp": "2023-05-15T14:30:00+05:30"}</p> <p>Entrée : 23-5-1T1:25+5:30</p> <p>Modèle : %{TIMESTAMP_ISO8601:timestamp}</p> <p>Sortie : {"timestamp": "23-5-1T1:25+5:30"}</p> <p>Entrée : 23-5-1T1:25Z</p> <p>Modèle : %{TIMESTAMP_ISO8601:timestamp}</p> <p>Sortie : {"timestamp": "23-5-1T1:25Z"}</p> |

| 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<br><br>Modèle : %{DATE:date}<br><br>Sortie : {"date": "11/29/2024"}<br><br>Entrée : 29.11.2024<br><br>Modèle : %{DATE:date}<br><br>Sortie : {"date": "29.11.2024"} |
| HORODATE   | Correspond à % {DATE} suivi du modèle % {TIME}, séparé par un espace ou un trait d'union.                         | 20                        | Entrée : 29-11-2024 14:30:00<br><br>Modèle : %{DATE:ESTAMP:dateTime}<br><br>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<br><br>Modèle : %{TZ:tz}<br><br>Sortie : {"tz": "PDT"}                                                                                                                    |

| Motif Grok       | Description                                                                                                      | Limite maximale de motifs | Exemple                                                                                                                                                                                                                                                                                                               |
|------------------|------------------------------------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DATESTAMP_RFC822 | Correspond à la date et à l'heure au format :<br>Jour MonthName (D) D (YY) YY (H) H:mm :( s) s<br>Fuseau horaire | 20                        | <p>Entrée : Monday Jan 5 23 1:30:00 CDT</p> <p>Modèle : %{DATESTAMP_RFC822 :dateTime}</p> <p>Sortie : {"dateTime": "Monday Jan 5 23 1:30:00 CDT"}</p> <p>Entrée : Mon January 15 2023 14:30:00 PST</p> <p>Modèle : %{DATESTAMP_RFC822 :dateTime}</p> <p>Sortie : {"dateTime": "Mon January 15 2023 14:30:00 PST"}</p> |

| 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                        | <p>Entrée : Mon, 15 May 2023 14:30:00 +0530</p> <p>Modèle : %{DATESTAMP_RFC2822:dateTime}</p> <p>Sortie : {"dateTime":"Mon, 15 May 2023 14:30:00 +0530"}</p> <p>Entrée : Monday, 15 Jan 23 14:30:00 Z</p> <p>Modèle : %{DATESTAMP_RFC2822:dateTime}</p> <p>Sortie : {"dateTime":"Monday, 15 Jan 23 14:30:00 Z"}</p> |
| DATESTAMP_OTHER   | Correspond à la date et à l'heure au format : Jour MonthName (d) d (H) H:mm : (s) s Fuseau horaire (yy) yy | 20                        | <p>Entrée : Mon May 15 14:30:00 PST 2023</p> <p>Modèle : %{DATESTAMP_OTHER:dateTime}</p> <p>Sortie : {"dateTime":"Mon May 15 14:30:00 PST 2023"}</p>                                                                                                                                                                |



| 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                        | <p>Entrée : 20230515143000</p> <p>Modèle : %{DATESTAMP_EVENTLOG:dateTime}</p> <p>Sortie : {"dateTime":"20230515143000"}</p> |

## Motifs en bois

| Motif Grok        | Description                                                                                                                                                                                                                                                                                                                     | Limite maximale de motifs | Exemple                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------|
| NIVEAU DU JOURNAL | Correspond aux niveaux de log standard dans différentes capitalisations 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/CRITICAL /CRITICAL Fatal/FATAL Severe/SEVERE Emerg/Emergency/EMERG/EMERGENCY | 20                        | <p>Entrée : INFO</p> <p>Modèle : %{LOGLEVEL:logLevel}</p> <p>Sortie : {"logLevel":"INFO"}</p> |

| 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                        | <p>Entrée : 23/Nov/2024:14:30:00 +0640</p> <p>Modèle : %{HTTPDATE:date}</p> <p>Sortie : {"date":"23/Nov/2024:14:30:00+0640"}</p>    |
| HORODATE<br>E 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                        | <p>Entrée : Nov 29 14:30:00</p> <p>Modèle : %{SYSLOGTIME:timestamp:date}</p> <p>Sortie : {"date":"Nov 29 14:30:00"}</p>             |
| 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                        | <p>Entrée : user.profile/settings-page</p> <p>Modèle : %{PROG:program}</p> <p>Sortie : {"program":"user.profile/settings-page"}</p> |

| Motif Grok    | Description                                                                                       | Limite maximale de motifs | Exemple                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SYSLOGPROGRAM | Correspond au modèle PROG grok éventuellement suivi d'un identifiant de processus entre crochets. | 20                        | <p>Entrée : user.profile/settings-page[1234]</p> <p>Modèle : %{SYSLOGPROGRAM:programWithId}</p> <p>Sortie : {"programWithId": "user.profile/settings-page[1234]", "program": "user.profile/settings-page", "pid": "1234"}</p> |
| HÔTE SYSLOG   | Correspond à un modèle % {HOST} ou % {IP}                                                         | 5                         | <p>Entrée : 2001:db8:3333:4444:5555:6666:7777:8888</p> <p>Modèle : %{SYSLOGHOST:ip}</p> <p>Sortie : {"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}</p>                                                                      |

| Motif Grok             | Description                                                                                           | Limite maximale de motifs | Exemple                                                                                                                          |
|------------------------|-------------------------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| INSTALLATION<br>SYSLOG | Correspond à la priorité Syslog au format décimal.<br>La valeur doit être placée entre crochets (<>). | 20                        | Entrée : <13.6><br><br>Modèle : %{SYSLOGFACILITY:syslog}<br><br>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 | Restrictions                                                                  |
|--------|--------------------------------------------------------------------|---------------|------------|-------------------------------------------------------------------------------|
| source | Chemin d'accès au champ du journal des événements qui sera analysé | Non           | @message   | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |

| Champ                               | Description                                                                                                      | Obligatoire ? | Par défaut                        | Restrictions                                                                                                      |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------|---------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>delimiter</code>              | 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 <code>\t</code> ou <code>\s</code>                                    |
| <code>Personnage de citation</code> | Caractère utilisé comme qualificatif de texte pour une seule colonne de données                                  | Non           | "                                 | Longueur maximale : 1                                                                                             |
| <code>columns</code>                | Liste des noms à utiliser pour les colonnes de l'événement du journal transformé.                                | Non           | <code>[column_1, column_2]</code> | Nombre maximum de colonnes CSV : 100<br>Longueur maximale : 128<br>Profondeur maximale des touches imbriquées : 3 |
| <code>destination</code>            | Le champ parent sous lequel placer les paires clé-valeur transformées                                            | Non           | Root node                         | Longueur maximale : 128<br>Profondeur maximale des touches imbriquées : 3                                         |

Le réglage `delimiter` sur « » `\t` séparera chaque colonne sur un caractère de tabulation et `\s` 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"
}
```

## Exemple 2

Supposons qu'un événement de journal ingéré ressemble à ceci :

```
{
  "timestamp": "2024-11-23T16:03:12Z",
  "type": "user_data",
  "logMsg": "'Akua Mansa':28:'New York: USA'"
}
```

Supposons que nous analysons l'événement au format JSON, ils analysent un champ JSON avec le processeur csv, en spécifiant les noms des colonnes et leur destination :

```
[
  {
    "parseJSON": {}
  },
  {
    "csv": {
      "source": "logMsg",
      "delimiter": ":",
      "quoteCharacter": "",
      "columns": ["name", "age", "location"],
      "destination": "msg"
    }
  }
]
```

```
    }  
  }  
]
```

L'événement du journal transformé serait le suivant.

```
{  
  "timestamp": "2024-11-23T16:03:12Z",  
  "logMsg": "'Akua Mansa':28:'New York: USA'",  
  "type": "user_data",  
  "msg": {  
    "name": "Akua Mansa",  
    "age": "28",  
    "location": "New York: USA"  
  }  
}
```

## parseKeyValue

Utilisez le `parseKeyValue` 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 | Restrictions                                                                  |
|----------------------|----------------------------------------------------------------------------|---------------|------------|-------------------------------------------------------------------------------|
| source               | Chemin d'accès au champ du journal des événements qui sera analysé         | Non           | @message   | Longueur maximale : 128<br><br>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      | Non           | &          | Longueur maximale : 128                                                       |

| Champ             | Description                                                                                                           | Obligatoire ? | Par défaut | Restrictions            |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|---------------|------------|-------------------------|
|                   | r dans les événements du journal d'origine                                                                            |               |            |                         |
| keyValueDelimiter | Chaîne de délimitation à utiliser entre la clé et la valeur dans chaque paire dans l'événement du journal transformé  | Non           | =          | Longueur maximale : 128 |
| 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

Cette section contient des informations sur les processeurs intégrés que vous pouvez utiliser avec les AWS services de vente de journaux.

### Table des matières

- [Analyser WAF](#)
- [Analyser Postgres](#)
- [ParseCloud Front](#)
- [Par Route 53](#)
- [Analyser VPC](#)
- [Analyser vers OCSF](#)

## 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/STMTTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
  "terminatingRuleId": "STMTTest_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
SOX4xwn4XV6Q4rgb7XiVG0Hms_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": "S0X4xwn4XV6Q4rgb7XiVG0Hms_BG1TAC4KyHmureZmBNrjGdRLiNIQ==",
  "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 vendus, 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"  
}
```

## Analyser vers OCSF

Le `parseToOCSF` processeur convertit les journaux en événements OCSF (Open Cybersecurity Schema Framework). L'OCSF est un standard ouvert qui fournit un schéma commun pour les données de sécurité, permettant une meilleure interopérabilité et une meilleure analyse entre différents outils et plateformes de sécurité.

Ce processeur est particulièrement utile pour les flux de travail d'analyse de sécurité dans lesquels vous devez standardiser les formats de journaux de différents AWS services dans un schéma cohérent pour une analyse en aval.

### Paramètres

#### `eventSource` (obligatoire)

Spécifie le AWS service ou le processus qui produit les événements du journal à convertir. Les valeurs valides sont :

- `CloudTrail`- CloudTrail journaux
- `Route53Resolver`- Journaux du résolveur Route 53
- `VPCFlow`- Journaux de flux Amazon VPC
- `EKSAudit`- Journaux d'audit Amazon EKS
- `AWSWAF`- AWS WAF journaux

#### `ocsfVersion` (obligatoire)

Spécifie la version du schéma OCSF à utiliser pour les événements du journal transformé.

Versions actuellement prises en charge : `V1.1`, `V1.5`

#### `mappingVersion` (facultatif)

Spécifie la version du mappage de transformation OCSF. Contrôlez la logique de transformation appliquée lors de la conversion des journaux au format OCSF. Si ce n'est pas spécifié, utilisez la dernière version disponible au moment de la création de la politique. Les politiques existantes ne sont pas automatiquement mises à niveau lorsque de nouvelles versions de mappage sont publiées. Dernière version actuelle : `v1.5.0`.

Remarque : Non pris en charge quand `ocsfVersion` c'est le cas `V1.1`.

source (facultatif)

Le chemin d'accès au champ du journal des événements que vous souhaitez analyser. En cas d'omission, l'intégralité du message du journal est analysée.

## Exemple

L'exemple suivant montre comment convertir les journaux `parseToOCSF` de flux VPC au format OCSF :

```
{
  "parseToOCSF": {
    "eventSource": "VPCFlow",
    "ocsfVersion": "V1.1"
  }
}
```

L'exemple suivant montre comment spécifier une version de mappage particulière pour un comportement de transformation cohérent :

```
{
  "parseToOCSF": {
    "eventSource": "CloudTrail",
    "ocsfVersion": "V1.5",
    "mappingVersion": "v1.5.0"
  }
}
```

## Processeurs String Mutate

Cette section contient des informations sur les processeurs de mutation de chaîne que vous pouvez utiliser avec un transformateur d'événements de journalisation.

### Table des matières

- [lowerCaseString](#)
- [upperCaseString](#)
- [Chaîne divisée](#)

- [Chaîne de remplacement](#)
- [TrimString](#)

## lowerCaseString

Le `lowerCaseString` processeur convertit une chaîne en sa version minuscule.

| Champ     | Description                                 | Obligatoire ? | Par défaut | Restrictions                  |
|-----------|---------------------------------------------|---------------|------------|-------------------------------|
| 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 | Restrictions                  |
|-----------|---------------------------------------------|---------------|------------|-------------------------------|
| 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 | Restrictions                  |
|-----------|-----------------------------------------------------------------------------------------|---------------|------------|-------------------------------|
| entries   | Tableau d'entrées. Chaque élément du tableau doit contenir source des delimiter champs. | Oui           |            | Nombre maximum d'entrées : 10 |
| 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 :

```
{
  "outer_key": {
    "inner_key": "inner_value"
  }
}
```

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 | Restrictions                  |
|---------|-------------------------------------------------------------------------------------------------------------------------|---------------|------------|-------------------------------|
| entries | Tableau d'entrées. Chaque élément du tableau doit contenir des to champs <code>source</code> et <code>from</code> , et. | Oui           |            | Nombre maximum d'entrées : 10 |
| source  | La clé du champ à modifier                                                                                              | Oui           |            | Longueur maximale : 128       |

| Champ | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Obligatoire ? | Par défaut | Restrictions                                                                                                                                 |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |            | Profondeur maximale des touches imbriquées : 3                                                                                               |
| from  | <p>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 <a href="#">Class Pattern</a> sur le site Web d'Oracle.</p> <p>Vous pouvez intégrer un motif ( . . . ) pour créer un groupe de capture numéroté et créer des groupes de capture ( ?P&lt;group_name&gt; . . . ) nommés qui peuvent être référencés to sur le terrain.</p> | Oui           |            | Longueur maximale : 128                                                                                                                      |
| to    | <p>La chaîne à substituer pour chaque correspondance de from. Les références arrières à des groupes de capture peuvent être utilisées. Utilisez le formulaire \$n pour les groupes numérotés tels que \$1 et utilisez-le \${group_name} pour les groupes nommés tels que \${my_group} . &gt;</p>                                                                                                                                                                                                                                                                                                                    | Oui           |            | <p>Longueur maximale : 128</p> <p>Nombre maximum de références arrières : 10</p> <p>Nombre maximum de références arrières dupliquées : 2</p> |

## 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 | Restrictions                  |
|-----------|------------------------------|---------------|------------|-------------------------------|
| 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

Cette section contient des informations sur les processeurs de mutation JSON que vous pouvez utiliser avec un transformateur d'événements de journal.

### Table des matières

- [Ajouter des clés](#)
- [Supprimer les clés](#)
- [Touches de déplacement](#)
- [Renommer les clés](#)
- [Copier la valeur](#)
- [listToMap](#)

### Ajouter des clés

Utilisez le addKeys processeur pour ajouter de nouvelles paires clé-valeur à l'événement du journal.

| Champ   | Description                                                | Obligatoire ? | Par défaut | Restrictions                 |
|---------|------------------------------------------------------------|---------------|------------|------------------------------|
| entries | Tableau d'entrées. Chaque élément du tableau peut contenir | Oui           |            | Nombre maximum d'entrées : 5 |

| Champ             | Description                                                                                                                                                                   | Obligatoire ? | Par défaut | Restrictions                                                                  |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------|-------------------------------------------------------------------------------|
|                   | des <code>overwriteIfExists</code> champs <code>keyvalue</code> , et.                                                                                                         |               |            |                                                                               |
| clé               | La clé de la nouvelle entrée à ajouter                                                                                                                                        | Oui           |            | Longueur maximale : 128<br><br>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 <code>true</code> , la valeur existante est remplacée si elle existe déjà dans l'événement. La valeur par défaut est <code>false</code> . | 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": [
          {
            "key": "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    | Restrictions                 |
|-----------|-----------------------------|---------------|---------------|------------------------------|
| Avec clés | 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 | Restrictions                                                                  |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------|-------------------------------------------------------------------------------|
| entries | Tableau d'entrées. Chaque élément du tableau peut contenir des <code>overwriteIfExists</code> champs <code>source</code> <code>target</code> , et. | Oui           |            | Nombre maximum d'entrées : 5                                                  |
| source  | La clé pour bouger                                                                                                                                 | Oui           |            | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |
| cible   | La clé pour passer à                                                                                                                               | Oui           |            | Longueur maximale : 128                                                       |

| Champ             | Description                                                                                                                                                                   | Obligatoire ? | Par défaut         | Restrictions                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------------|------------------------------------------------|
|                   |                                                                                                                                                                               |               |                    | Profondeur maximale des touches imbriquées : 3 |
| overwriteIfExists | Si vous définissez ce paramètre sur <code>true</code> , la valeur existante est remplacée si elle existe déjà dans l'événement. La valeur par défaut est <code>false</code> . | Non           | <code>false</code> | 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    | Restrictions                                                                  |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|---------------|---------------|-------------------------------------------------------------------------------|
| entries           | Tableau d'entrées. Chaque élément du tableau peut contenir des <code>overwriteIfExists</code> champs <code>keytarget</code> , et. | Oui           | Aucune limite | Nombre maximum d'entrées : 5                                                  |
| clé               | La clé pour renommer                                                                                                              | Oui           | Aucune limite | Longueur maximale : 128                                                       |
| cible             | Le nouveau nom de clé                                                                                                             | Oui           | Aucune limite | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |
| overwriteIfExists | Si vous définissez ce paramètre sur <code>true</code> , la valeur existante est remplacée si elle existe key déjà                 | Non           | false         | Aucune limite                                                                 |

| Champ | Description                                       | Obligatoire ? | Par défaut | Restrictions |
|-------|---------------------------------------------------|---------------|------------|--------------|
|       | dans l'événement. La valeur par défaut est false. |               |            |              |

## 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    | Restrictions                                                                  |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|---------------|-------------------------------------------------------------------------------|
| entries           | Tableau d'entrées. Chaque élément du tableau peut contenir des <code>overwriteIfExists</code> champs <code>source</code> et <code>target</code> , et.                         | Oui           |               | Nombre maximum d'entrées : 5                                                  |
| source            | La clé à copier                                                                                                                                                               | Oui           |               | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |
| cible             | La clé vers laquelle copier la valeur                                                                                                                                         | Oui           | Aucune limite | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |
| overwriteIfExists | Si vous définissez ce paramètre sur <code>true</code> , la valeur existante est remplacée si elle existe déjà dans l'événement. La valeur par défaut est <code>false</code> . | 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": [
        {
          "key": "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 | Restrictions                                                                  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------|-------------------------------------------------------------------------------|
| source        | La clé ProcessingEvent avec une liste d'objets qui seront convertis en carte                                                                                                                                                                                                   | Oui           |            | Longueur maximale : 128<br><br>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                                                       |



| Champ          | Description                                                                                                                                                                                                                                                                                                                                                                                               | Obligatoire ?                                                                          | Par défaut  | Restrictions                                                                  |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|-------------|-------------------------------------------------------------------------------|
| cible          | La clé du champ qui contiendra la carte générée                                                                                                                                                                                                                                                                                                                                                           | Non                                                                                    | Nœud racine | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |
| flatten        | Valeur booléenne indiquant si la liste sera aplatie en éléments individuels ou si les valeurs de la carte générée seront des listes.<br><br>Par défaut, les valeurs des clés correspondantes seront représentées dans un tableau. Définissez <code>flatten</code> cette valeur sur <code>true</code> pour convertir le tableau en une valeur unique basée sur la valeur <code>deflattenedElement</code> . | Non                                                                                    | false       |                                                                               |
| Élément aplati | Si vous définissez <code>flatten</code> sur <code>true</code> , <code>flattenedElement</code> utilisez-le pour spécifier quel élément, <code>first</code> ou <code>last</code> , à conserver.                                                                                                                                                                                                             | Obligatoire lorsque <code>flatten</code> le paramètre est défini sur <code>true</code> |             | La valeur ne peut être que <code>first</code> ou <code>last</code>            |

## 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

Cette section contient des informations sur les processeurs de conversion de types de données que vous pouvez utiliser avec un transformateur d'événements de journalisation.

### Table des matières

- [Type de convertisseur](#)
- [Convertisseur Date/Heure](#)

## 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 | Restrictions                                                                  |
|---------|-----------------------------------------------------------------------------------------------------|---------------|------------|-------------------------------------------------------------------------------|
| entries | Tableau d'entrées. Chaque élément du tableau doit contenir key des type champs.                     | Oui           |            | Nombre maximum d'entrées : 10                                                 |
| clé     | La clé avec la valeur à convertir en un autre type                                                  | Oui           |            | Longueur maximale : 128<br><br>Profondeur maximale des touches imbriquées : 3 |
| type    | Type vers lequel effectuer la conversion. Les valeurs valides sont integerdouble, string etboolean. | 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           | Restrictions                                                                  |
|-----------------|----------------------------------------------------------------------------------|---------------|----------------------|-------------------------------------------------------------------------------|
| 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 <code>source</code> champ                     | Oui           |                      | Nombre maximum d'entrées : 5                                                  |
| cible           | Le champ JSON dans lequel stocker le résultat.                                   | Oui           |                      | Longueur maximale : 128<br><br>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 | Longueur maximale : 64                                                        |

| Champ                       | Description                                                                                                                                                                                 | Obligatoire ? | Par défaut | Restrictions          |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------|-----------------------|
|                             |                                                                                                                                                                                             |               | :ss.SSS'Z  |                       |
| Fuseau horaire de la source | <p>Fuseau horaire du champ source.</p> <p>Pour obtenir la liste des valeurs possibles, consultez la section <a href="#">Identifiants et décalages de zone pris en charge par Java</a>.</p>  | Non           | UTC        | Longueur minimale : 1 |
| Fuseau horaire cible        | <p>Fuseau horaire du champ cible.</p> <p>Pour obtenir la liste des valeurs possibles, consultez la section <a href="#">Identifiants et décalages de zone pris en charge par Java</a>.</p>   | Non           | UTC        | Longueur minimale : 1 |
| locale                      | <p>Les paramètres régionaux du champ source.</p> <p>Pour une liste des valeurs possibles, consultez la <a href="#">méthode locale getAvailableLocales () en Java avec des exemples</a>.</p> | 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 `TransformedLogEvents`, `TransformedBytes`, 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 par 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 Firewall tableau 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 `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 dans 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 collecte des OpenSearch services.

7. Pour le rôle IAM permettant d'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 dans [Autorisations 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 créer le tableau de bord AWS WAF, les journaux de flux Amazon VPC CloudTrail, ou. 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, et 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 de la collection pour 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, consultez [CloudWatchOpenSearchDashboardAccess](#).

## 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 vendu 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, `CloudWatchOpenSearchDashboardsFullAccess` et `CloudWatchOpenSearchDashboardAccess`. Le tableau suivant répertorie les actions autorisées par chacune de ces politiques.

| Action                                                                  | Politique IAM                                                                                             | Autorisations supplémentaires nécessaires                                                                                                                                                                |
|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Création d'une intégration                                              | <code>CloudWatchOpenSearchDashboardsFullAccess</code>                                                     |                                                                                                                                                                                                          |
| Supprimer l'intégration                                                 | <code>CloudWatchOpenSearchDashboardsFullAccess</code>                                                     |                                                                                                                                                                                                          |
| Création d'un tableau de bord                                           | <code>CloudWatchOpenSearchDashboardsFullAccess</code>                                                     |                                                                                                                                                                                                          |
| Modifier le tableau de bord                                             | <code>CloudWatchOpenSearchDashboardsFullAccess</code>                                                     |                                                                                                                                                                                                          |
| Supprimer le tableau de bord                                            | <code>CloudWatchOpenSearchDashboardsFullAccess</code>                                                     |                                                                                                                                                                                                          |
| Actualiser le tableau de bord en utilisant <code>Synchronize now</code> | <code>CloudWatchOpenSearchDashboardsFullAccess</code>                                                     |                                                                                                                                                                                                          |
| Afficher l'intégration dans les paramètres                              | <code>CloudWatchOpenSearchDashboardAccess</code> ou <code>CloudWatchOpenSearchDashboardsFullAccess</code> |                                                                                                                                                                                                          |
| Afficher le tableau de bord                                             | <code>CloudWatchOpenSearchDashboardAccess</code> ou <code>CloudWatchOpenSearchDashboardsFullAccess</code> | 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 <a href="#">Contrôle d'accès aux données pour Amazon OpenSearch Service Serverless</a> 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 <a href="#">Contrôle d'accès aux données pour Amazon OpenSearch Service Serverless</a> 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"
  ]
}
```

# Journaux d'accès avec intégration des tables S3

L'intégration de S3 Tables vous CloudWatch permet d'accéder aux données des journaux ingérées à CloudWatch l'aide de moteurs d'analyse tels qu'Amazon Athena, Amazon Redshift et d'outils tiers prenant en charge la connexion à des boutiques compatibles avec Apache Iceberg. Cette intégration vous permet d'effectuer une analyse complète des journaux à l'aide des outils de votre choix et de corréler les données des CloudWatch journaux avec celles qui ne le sont pas CloudWatch.

## Comprendre l'intégration des tables S3

Amazon S3 Tables Integration est une solution entièrement gérée qui rend vos journaux de CloudWatch connexion disponibles sous forme de tables Amazon S3 gérées. Grâce à cette intégration, vous bénéficiez d'une plus grande flexibilité quant à la manière dont vous analysez vos journaux, en plus des fonctionnalités CloudWatch des journaux.

L'intégration fonctionne en créant un compartiment de table Amazon S3 géré (`aws-cloudwatch`) et en associant des sources de journaux spécifiques aux tables Amazon S3 en fonction du nom et du type de source de données (qui peuvent être gérés depuis l'onglet Gestion des CloudWatch journaux > Sources de données de la console Logs). Une fois associées, CloudWatch les données des journaux sont accessibles via les tables Amazon S3 au format Apache Iceberg. Ce format fournit un moyen standardisé aux différents moteurs d'analyse d'interroger les données de manière efficace.

## Composants de base

### Association de sources de données

Processus consistant à lier des sources de CloudWatch journaux spécifiques à l'intégration des tables S3 en fonction de critères de source et de type de données.

### Tables Apache Iceberg

Format de table sous-jacent utilisé par S3 Tables qui fournit un stockage de données structuré et assure la compatibilité avec plusieurs moteurs d'analyse.

## Flux de données vers les tables S3

Comprendre comment les données circulent entre les CloudWatch journaux et les tables S3 vous aide à planifier votre intégration et à gérer efficacement les données de vos journaux.

Lorsque vous créez une association, CloudWatch Logs envoie automatiquement les nouveaux événements de journal correspondant au nom et au type de source de données associés à un compartiment de table S3 CloudWatch géré. Vous pouvez trouver ces événements dans l'espace de noms des journaux sous le tableau correspondant à cette source de données. Les processus d'intégration enregistrent uniquement les événements ajoutés après la création de l'association et ne remplacent pas les journaux antérieurs à la création de l'association.

La conservation des données dans le compartiment de table S3 correspond à la politique de rétention définie pour le groupe de journaux. Par exemple, si vous définissez un groupe de journaux pour une durée de conservation d'un jour, CloudWatch Logs supprime les données des CloudWatch journaux et de la table S3 au bout d'un jour. Lorsque vous supprimez un groupe de journaux ou un flux de CloudWatch journaux, Logs supprime également les données du compartiment de table S3.

## Quand utiliser l'intégration des tables S3

Envisagez d'utiliser l'intégration des tables S3 pour corréler les données des journaux avec d'autres CloudWatch données externes ou non, ou lorsque vous préférez utiliser d'autres outils d'analyse tels qu'Amazon Athena pour effectuer des analyses CloudWatch sur les données des journaux. Utilisez cette intégration lorsque vous avez besoin de fonctionnalités allant au-delà de ce qui est disponible dans CloudWatch Logs. Cette intégration est particulièrement utile lorsque :

- Vous devez exécuter des requêtes complexes de type SQL sur de grands volumes de données de journal
- Vous souhaitez intégrer l'analyse des journaux aux flux de travail et outils d'analyse existants
- Vous avez besoin de fonctionnalités complètes d'analyse des journaux couvrant plusieurs sources de données

Il n'y a aucun frais de stockage ou de maintenance des tables supplémentaires pour les tables S3 créées grâce à cette intégration, au-delà des tarifs CloudWatch d'ingestion et de stockage existants.

## Conditions préalables

Avant de mettre en œuvre l'intégration, assurez-vous de disposer des éléments suivants :

- Données CloudWatch des journaux existants
- Autorisations IAM appropriées pour l'accès multiservice entre les CloudWatch journaux et les tables S3, comme décrit dans la section suivante

## Autorisations IAM

Pour intégrer CloudWatch les journaux aux tables S3, vous devez configurer les autorisations IAM pour deux entités distinctes : l'utilisateur ou le rôle qui configure l'intégration et le rôle de service que CloudWatch Logs assume pour écrire des données dans les tables S3.

### Pour le rôle ou l'utilisateur qui crée l'intégration

L'utilisateur ou le rôle qui configure l'intégration nécessite les autorisations suivantes :

- `observabilityadmin:CreateS3TableIntegration` pour créer l'intégration et `logs:AssociateSourceToS3TableIntegration` pour ajouter des sources
- `s3tables:CreateTableBuckets`, `s3tables:PutTableBucketEncryption`, et `s3tables:PutTableBucketPolicy` pour configurer le compartiment de table S3

### Pour le rôle de service

Associez la politique IAM suivante au rôle de service IAM que CloudWatch Logs utilise pour écrire des données dans le bucket de table. Cette politique autorise l'écriture dans les tables. Remplacez *aws-region123456789012*, et *log-group-name* par votre AWS région, votre identifiant de compte et le nom de votre groupe de journaux.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:integrateWithS3Table"
      ],
      "Resource": ["arn:aws:logs:aws-region:123456789012:log-group:log-group-name"],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "123456789012"
        }
      }
    }
  ]
}
```



Associez la politique de confiance suivante au rôle de service IAM que CloudWatch Logs assumera pour écrire les données des journaux dans les tables S3. Vous créez ou sélectionnez ce rôle lors de la configuration de l'intégration. Les conditions limitent le rôle afin que CloudWatch Logs ne puisse l'assumer que pour le compte et le groupe de journaux spécifiés. Remplacez *aws-region123456789012*, et *log-group-name* par votre AWS région, votre identifiant de compte et le nom de votre groupe de journaux.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:aws-region:123456789012:log-
group:log-group-name"]
        }
      }
    }
  ]
}
```

## Politique relative aux clés KMS (pour les données chiffrées)

Si vous utilisez une clé gérée par le client pour chiffrer les données de vos journaux, vous devez accorder au principal de CloudWatch service et au service de maintenance des tables S3 l'accès principal à la clé. Ajoutez les instructions suivantes à votre politique de clés KMS. Remplacez les valeurs de l'espace réservé par votre Compte AWS ID, votre région, l'ID de clé KMS et l'ARN de la table ou du bucket de table S3.

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

```

        "Sid": "EnableSystemTablesKeyUsage",
        "Effect": "Allow",
        "Principal": {
            "Service": "systemtables.cloudwatch.amazonaws.com"
        },
        "Action": [
            "kms:DescribeKey",
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "arn:aws:kms:aws-region:123456789012:key/key-id",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "123456789012"
            }
        }
    },
    {
        "Sid": "EnableKeyUsage",
        "Effect": "Allow",
        "Principal": {
            "Service": "maintenance.s3tables.amazonaws.com"
        },
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "arn:aws:kms:aws-region:123456789012:key/key-id",
        "Condition": {
            "StringLike": {
                "kms:EncryptionContext:aws:s3:arn": "<table-or-table-bucket-arn>/*"
            }
        }
    }
}

```

## Démarrage

Pour commencer à intégrer les tables S3, vous devez configurer l'intégration entre vos CloudWatch journaux et vos tables S3. Ce processus implique de configurer les associations de sources de données et de configurer les autorisations IAM appropriées.

## Pour créer une intégration de tables S3

1. Ouvrez la console CloudWatch Logs à l'<https://console.aws.amazon.com/cloudwatch/>adresse ».
2. Choisissez Paramètres, Global, Créer une intégration de table S3.
3. Personnalisez la manière dont les journaux seront chiffrés dans les tables S3 et le rôle que CloudWatch les journaux utiliseront pour écrire vos journaux dans les tables S3.
4. Choisissez Create S3 Table Integration.

## Pour associer des sources à une intégration de table S3

1. Ouvrez la console CloudWatch Logs à l'<https://console.aws.amazon.com/cloudwatch/>adresse ».
2. Choisissez Paramètres, Global, Gérer l'intégration des tables S3.
3. Choisissez Associer une source de données.
4. Sélectionnez le nom et le type de source de données pour lesquels vous souhaitez activer l'intégration.
5. Choisissez Associer une source de données.

## Pour associer des sources à une intégration de table S3 depuis la page de gestion des journaux

1. Ouvrez la console CloudWatch Logs à l'<https://console.aws.amazon.com/cloudwatch/>adresse ».
2. Choisissez Log Management dans le volet de navigation.
3. Sélectionnez l'onglet Sources de données.
4. Choisissez le nom et le type de source de données que vous souhaitez intégrer.
5. Choisissez Actions relatives à la source de données.
6. Sélectionnez Associer à l'intégration des tables S3.
7. Passez en revue les sources de données, puis choisissez Associer une source de données.

Avant de pouvoir utiliser les données, vous devez suivre les 3 étapes suivantes :

1. Intégration des tables Amazon S3 aux services AWS d'analyse - Utilisation de la console Amazon S3
2. Configurer les autorisations de Lake Formation
3. Connectez-vous aux outils d'analyse

## Intégration des tables Amazon S3 aux services AWS d'analyse - Utilisation de la console Amazon S3 ([lien](#))

Pour activer l'intégration des tables S3 à l'aide de la console S3

1. Ouvrez la console Amazon S3 à l'adresse <https://console.aws.amazon.com/s3/>.
2. Dans le volet de navigation de gauche, choisissez Compartiments de tables.
3. Cliquez sur le bouton Activer l'intégration en haut de la page.
4. La première fois que vous intégrez des compartiments de table dans une région, Amazon S3 crée un nouveau rôle de service IAM en votre nom. Ce rôle permet à Lake Formation d'accéder à tous les compartiments de tables de votre compte et de fédérer l'accès à vos tables dans AWS Glue Data Catalog.

## Configurer les autorisations de Lake Formation

Bien que CloudWatch Logs soit autorisé à écrire dans la table (configuré dans les étapes précédentes), les utilisateurs et les rôles d'analyse ne sont pas automatiquement autorisés à lire les données. Vous devez explicitement accorder l'accès en utilisant AWS Lake Formation. Vous devez le faire pour chaque principal IAM auquel vous souhaitez donner accès à la table.

Pour accorder l'accès aux requêtes à des utilisateurs ou à des rôles

Vous devez accorder les autorisations SELECT et DESCRIBE aux principaux IAM (utilisateurs ou rôles) qui exécuteront des requêtes dans Athena ou Redshift.

1. Ouvrez la console AWS Lake Formation.
2. Dans le volet de navigation, sous Autorisations, sélectionnez Autorisations du lac de données.
3. Choisissez Accorder.
4. Principes : Sélectionnez les utilisateurs ou les rôles IAM qui nécessitent un accès (par exemple, vos analystes de données ou le rôle d'administrateur que vous utilisez actuellement).
5. Balises LF ou ressources du catalogue : sélectionnez les ressources du catalogue de données nommées.
6. Bases de données et tables :
  - Sélectionnez le compartiment S3 Table créé par l' CloudWatch intégration (aws-cloudwatch).

- Sélectionnez la table spécifique associée à votre source de données (facultatif).
7. Autorisations du tableau : sélectionnez Sélectionner et décrire.
  8. Choisissez Accorder.

#### Note

Si vous rencontrez des erreurs « Accès refusé » lorsque vous interrogez les journaux dans Athena, assurez-vous que l'utilisateur exécutant la requête possède à la fois les autorisations IAM appropriées pour Athéna et les autorisations Lake Formation définies ci-dessus.

Pour en savoir plus sur les autorisations de Lake Formation, rendez-vous sur <https://docs.aws.amazon.com/lake-formation/latest/dg/granting-catalog-permissions.html>.

## Connectez-vous aux outils d'analyse

Une fois les autorisations accordées, vous pouvez configurer votre service d'analyse préféré pour interroger les tables S3. Les tables S3 utilisent le format Apache Iceberg, qui est pris en charge de manière native par Amazon Athena, Amazon Redshift et Amazon EMR.

### Pour interroger les données du journal dans Amazon Athena

Amazon Athena interagit avec les tables S3 via le catalogue de tables Amazon S3.

Pour configurer Athena afin qu'elle interroge les données de votre journal

1. Ouvrez la console Amazon Athena à l'adresse. <https://console.aws.amazon.com/athena/>
2. Dans l'éditeur de requêtes, sélectionnez le catalogue Amazon S3 Tables dans le menu déroulant des sources de données.
3. Si le catalogue ne s'affiche pas, assurez-vous d'avoir suivi les étapes d'autorisation de Lake Formation ci-dessus pour votre rôle d'utilisateur spécifique.
4. Une fois le catalogue sélectionné, vos tables de log apparaîtront dans la liste des bases de données. Vous pouvez désormais exécuter des requêtes SQL standard sur les données de vos journaux.

Exemple de requête : `SELECT * FROM "amazon_vpc__flow" LIMIT 100;`

Pour en savoir plus sur la connexion des services Analytics aux tables S3, rendez-vous sur <https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-tables-integrating-aws.html>.

# 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 vous avez configuré AWS Organizations et utilisez des comptes membres, vous pouvez utiliser la centralisation des journaux pour collecter les données des journaux des comptes sources vers un compte de surveillance central.

Lorsque vous travaillez avec des groupes de journaux centralisés, vous pouvez utiliser les dimensions de ces champs système lors de la création de filtres métriques.

- `@aws.account`- Cette dimension représente l'identifiant du AWS compte à l'origine de l'événement du journal.
- `@aws.region`- Cette dimension représente la AWS région dans laquelle l'événement du journal a été généré.

Ces dimensions aident à identifier la source des données des journaux, ce qui permet un filtrage et une analyse plus précis des métriques dérivées des journaux centralisés. Pour de plus amples informations, veuillez consulter [Centralisation des journaux inter-comptes et inter-régions](#).

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, consultez [Classes 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. Lorsque vous testez un modèle de filtre, l'aperçu des résultats du filtre affiche les 50 premières lignes de journal correspondantes à des fins de validation. Si l'horodatage des résultats filtrés est antérieur à l'heure de création de la métrique, aucun journal n'est affiché.

## 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. Les dimensions faisant partie de l'identifiant unique d'une métrique, chaque fois qu'une name/value paire unique est extraite de vos journaux, vous créez une nouvelle variante 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, consultez [CloudWatch 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 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 [MetricDatum](#) le 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 name/value valeurs uniquement disponibles pour les modèles de filtre JSON et 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 vous éviter de percevoir des frais élevés accidentels, Amazon peut désactiver votre filtre métrique s'il génère 1 000 name/value paires différentes pour des dimensions spécifiées sur une période donnée.

Vous pouvez créer des alarmes de facturation qui vous informent de vos frais estimés. Pour plus d'informations, consultez [Création d'une alarme de facturation pour surveiller vos frais AWS 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.

### Exemple: 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.

### Exemple: 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.

Exemple: 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
```

Exemple: 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"`.

Exemple: 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) Si vous utilisez des groupes de journaux centralisés, sous Critères de sélection des filtres, vous pouvez spécifier des filtres en fonction du compte source (@aws.account), de la région source (@aws.region) ou des deux conditions.
7. (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.
8. Sélectionnez Next (Suivant), puis saisissez un nom pour le filtre de métrique.

9. 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.
10. Pour Metric name (Nom de la métrique), saisissez un nom pour la nouvelle métrique.
11. 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.

12. (Facultatif) Pour Unit (Unité), sélectionnez une unité à affecter à la métrique. Si vous ne spécifiez pas d'unité, elle est définie comme None.
13. (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.

14. 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. Lorsque vous tracez ces données dans la CloudWatch console, 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

A 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 Next, 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 **ApacheNotFoundErrorCode**.
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=ApacheNotFoundErrorCode,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": "ApacheNotFoundErrorCode"
```

```
    }  
  ],  
  "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 `ApacheNotFoundErrorCode`.

## 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,dimension
$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": "ApacheNotFoundErrorCode"
        }
      ],
      "creationTime": 1399277571078,
      "filterPattern": "[ip, id, user, timestamp, request, status_code=404,"
    }
  ]
}
```

# 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.

Vous pouvez également utiliser la centralisation CloudWatch des journaux pour répliquer les données des journaux de plusieurs comptes et régions vers un emplacement central. Pour de plus amples informations, veuillez consulter [Centralisation des journaux inter-comptes et inter-régions](#).

Pour commencer à vous abonner pour enregistrer les événements, créez la ressource de réception, telle qu'un flux Amazon Kinesis Data Streams, dans lequel les événements seront diffusés. 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 `AccessDeniedException` ou `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, consultez [Classes 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.

Pour le traitement par lots et l'analyse des données du journal selon un calendrier, pensez à utiliser [Automatisation de l'analyse des journaux avec des requêtes planifiées](#). Les requêtes planifiées exécutent automatiquement les requêtes CloudWatch Logs Insights et fournissent des résultats à des destinations telles que les buckets Amazon S3 ou les bus d' EventBridge événements Amazon.

 Note

Les filtres d'abonnement garantissent la diffusion d'événements au moins une fois, tandis que des événements dupliqués peuvent parfois se produire.

## 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

Le nom de ressource Amazon (ARN) du flux Amazon 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, consultez [Prévention de la récursivité dans les journaux](#).

Les critères de sélection ont une limite de taille de 25 Ko.

Pour les groupes de journaux centralisés, les éléments clés suivants sont également inclus. Ces éléments peuvent être utilisés comme critères de sélection de champs pour aider à identifier la source des données des journaux, ce qui permet un filtrage plus granulaire et une analyse des métriques dérivées des journaux centralisés.

## @aws .account

Ce champ identifie l'identifiant du AWS compte à l'origine de l'événement du journal.

## @aws .région

Ce champ identifie la AWS région dans laquelle l'événement du journal a été généré.

# 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. Si vous utilisez des journaux centralisés avec votre AWS Organizations, vous pouvez choisir d'émettre le champ

@aws.region système @aws.account et pour identifier quelles données proviennent de quels comptes et régions de votre organisation. 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, Amazon Kinesis Data Streams et Service. OpenSearch

#### 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 Amazon 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 Amazon Kinesis Data Streams

L'exemple suivant associe un filtre d'abonnement à un groupe de journaux contenant des événements AWS CloudTrail . Le filtre d'abonnement transmet toutes les activités enregistrées à l'aide des AWS informations d'identification « root » à un flux appelé « RootAccess » dans Amazon 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 [PutSubscriptionFilter](#) ou `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 Amazon 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 Amazon 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 Amazon 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 Amazon Kinesis Data [Streams](#) `describe-stream` suivante pour vérifier le `StreamDescription` `StreamStatus` propriété. 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": "AA0IIAH450GAB4HC5F431",
    "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 la commande [put-role-policy](#) 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 c'est bien le cas en accédant à un itérateur de partition Amazon Kinesis Data Streams et en utilisant la commande Amazon Kinesis Data Streams get-records pour récupérer certains enregistrements Amazon Kinesis Data Streams :

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id
shardId-0000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
  "ShardIterator":
  "AAAAAAAAAAFGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRgB9v4scv+3vaq+f+0IK8zM5My8ID
+g6rMo7UKWeI4+IWiK20Sh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
```

```
+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGB9v4scv+3vaq+f+0IK8zM5My8ID
+g6rMo7UKWeI4+IWIK20Sh0uP"
```

Notez que vous devrez peut-être effectuer cet appel plusieurs fois avant qu'Amazon Kinesis Data Streams ne commence à renvoyer des données.

Vous devriez obtenir une réponse constituée d'un tableau d'enregistrements. L'attribut Data d'un enregistrement Amazon 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 Amazon 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\":\n\"Root\"}}",
    },
    {
      "id": "31953106606966983378809025079804211143289615424298221569",
      "timestamp": 1432826855000,
      "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\n\"Root\"}}",
    },
    {
      "id": "31953106606966983378809025079804211143289615424298221570",
      "timestamp": 1432826855000,
      "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\n\"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": "AA0IIAH450GAB4HC5F431",
    "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 la commande `put-role-policy` 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-stream` commande Firehose pour vérifier le `DeliveryStreamDescription` `DeliveryStreamStatus` propriété. 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-0000000000001",
        "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": "AA01IAH450GAB4HC5F431",
    "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 cluster Amazon OpenSearch Service, choisissez le nom du cluster dans lequel les données du groupe de journaux seront livrées.
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, consultez [Prévention 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 Amazon 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 Amazon 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 Amazon Kinesis Data Streams

Avant de créer un flux de données Amazon 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 Amazon 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 Amazon Kinesis Data Streams à l'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 Amazon 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 Amazon 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 Amazon Kinesis Data Streams. Le modèle de filtre associe tous les événements du journal au texte `Test` et les transmet au flux dans Amazon Kinesis Data Streams.

Pour créer une politique d'abonnement au niveau du compte pour Amazon 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` `StreamStatus` propriété.

```
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": "EXAMPLE8463463374607431768211455",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
            "EXAMPLE688818456679503831981458784591352702181572610"
        }
      }
    ]
  }
}

```

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 la commande [put-role-policy](#) 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 que le flux des événements du journal en utilisant un itérateur de partition Amazon Kinesis Data Streams et en utilisant la commande Amazon Kinesis `get-records` Data Streams pour récupérer certains enregistrements Amazon Kinesis Data Streams :

```
aws kinesis get-shard-iterator --stream-name TestStream --shard-id
shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
  "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRgB9v4scv+3vaq+f+0IK8zM5My8ID
+g6rMo7UKWeI4+IWIK20Sh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRgB9v4scv+3vaq+f+0IK8zM5My8ID
+g6rMo7UKWeI4+IWIK20Sh0uP"
```

Vous devrez peut-être utiliser cette commande plusieurs fois avant qu'Amazon Kinesis Data Streams ne commence à renvoyer des données.

Vous devriez obtenir une réponse constituée d'un tableau d'enregistrements. L'attribut Data d'un enregistrement Amazon 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 Amazon 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 corresponderaient 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 Amazon 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 la commande `put-role-policy` 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"}'

```

Amazon 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. Patientez quelques minutes pour que le stream soit activé. Vous pouvez utiliser la `describe-delivery-stream` commande Firehose pour vérifier le `DeliveryStreamDescription` `DeliveryStreamStatus` propriété. 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": "AA0IIAH450GAB4HC5F431",
    "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, consultez [Prévention 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 Amazon 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 en temps réel de données d'événements sur ces comptes peut être assemblé et transmis aux groupes de sécurité de l'information, qui peuvent utiliser Amazon 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. 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 une région sont créées dans l'est des États-Unis (Virginie du Nord).

Si vous avez configuré AWS Organizations et utilisez des comptes membres, vous pouvez utiliser la centralisation des journaux pour collecter les données des journaux à partir des comptes sources vers un compte de surveillance central.

Lorsque vous travaillez avec des groupes de journaux centralisés, vous pouvez utiliser ces dimensions de champs système pour créer des filtres d'abonnement :

- `@aws.account-` Cette dimension représente l'identifiant du AWS compte à l'origine de l'événement du journal.
- `@aws.region-` Cette dimension représente la AWS région dans laquelle l'événement du journal a été généré.



Ces dimensions aident à identifier la source des données des journaux, ce qui permet un filtrage plus granulaire et une analyse des métriques dérivées des journaux centralisés.

## Rubriques

- [Partage de données de journal entre comptes et entre régions à l'aide d'Amazon 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 d'Amazon 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 d'Amazon 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 de journal : définit une destination qui encapsule un flux Amazon Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir des données de 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.

#### Rôle ARN (ARN de rôle)

Rôle Gestion des identités et des accès AWS (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. 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 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 Amazon 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 destination dans Amazon 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.StreamStatus` propriété. Prenez également note de la valeur `StreamDescription.StreamArn`, car vous la transmettez 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": "34028236692093846346337460743176EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
"4955113521868881845667950383198145878459135270218EXAMPLE"
        }
      }
    ]
  }
}
```

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": "AA0IIAH450GAB4HC5F431",
    "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 `~/PermissionsForCWL.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 `DestinationArn` 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'PutSubscriptionFilter action 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::99999999999999999999: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 dans [Étape 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 pour les 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. Entrez la commande suivante pour vérifier quel rôle ou utilisateur IAM est utilisé pour exécuter les commandes de AWS journalisation.

```
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"
}
```

Notez 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.

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 999999999999 :

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": "AllowSubscriptionFilteronAccountResource",
      "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 Amazon 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 Amazon Kinesis Data Streams, et en utilisant la commande `aws kinesis get-records` pour récupérer certains enregistrements Amazon Kinesis Data Streams :

```
aws kinesis get-shard-iterator \
  --stream-name RecipientStream \
  --shard-id shardId-000000000000 \
  --shard-iterator-type TRIM_HORIZON

{
  "ShardIterator":
  "AAAAAAAAAAGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+0IK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
  --limit 10 \
  --shard-iterator
  "AAAAAAAAAAGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+0IK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

#### Note

Il se peut que vous deviez réexécuter la commande `get-records` à quelques reprises avant qu'Amazon Kinesis Data Streams ne commence à renvoyer des données.

Vous devriez voir une réponse contenant un ensemble d'enregistrements Amazon Kinesis Data Streams. L'attribut de données de l'enregistrement Amazon 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": "3195310660696698337880902507980421114328961542429EXAMPLE",
        "timestamp": 1432826855000,
        "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
      }\",
      {
        "id": "3195310660696698337880902507980421114328961542429EXAMPLE",
        "timestamp": 1432826855000,
        "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
      }\",
      {
        "id": "3195310660696698337880902507980421114328961542429EXAMPLE",
        "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 Amazon 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": "{\n\"Version\": \"2012-10-17\", \"Statement\":
        [ {\n\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\n\"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 `PutDestinationPolicy` pour 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 dans [Activer 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 dans [Étape 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 de journal** : définit une destination qui encapsule un flux Amazon Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir des données de 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. Cependant, la AWS ressource pointée par 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": "AR0AR3BXASEKW7K635M53",
    "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 actif [Étape 1 : créer un flux de diffusion Firehose](#). Vous pouvez utiliser la commande suivante pour vérifier le StreamDescription.StreamStatus propriété.

```
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 pour 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. Entrez la commande suivante pour vérifier quel rôle ou utilisateur IAM est utilisé pour exécuter les commandes de AWS journalisation.

```
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"  
}
```

Notez 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.

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 999999999999 :

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

Après avoir 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'`PutDestinationPolicy` action 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    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n      \"Principal\" : {\n        \"AWS\" : \"111111111111\"\n      },\n      \"Action\n\" : \"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 d'Amazon 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 de journal** : définit une destination qui encapsule un flux Amazon Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir des données de 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.

#### Rôle ARN (ARN de rôle)

Rôle Gestion des identités et des accès AWS (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. 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 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 : créer 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 Amazon 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 destination dans Amazon 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` `StreamStatus` propriété. Prenez également note de la valeur `StreamDescription.StreamArn`, car vous la transmettez 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": "34028236692093846346337460743176EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
            "4955113521868881845667950383198145878459135270218EXAMPLE"
        }
      }
    ]
  }
}
```

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 `~/TrustPolicyForCWL.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": "AA0IIAH450GAB4HC5F431",
    "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 DestinationArn 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'PutSubscriptionFilter action 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::9999999999999999999999: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 dans [Étape 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 : créer 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 : créer 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 pour les 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 : créer une politique de filtrage des abonnements au niveau du compte](#).

### Étape 3 : créer 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, consultez [Prévention 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 Amazon 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 Amazon Kinesis Data Streams, et en utilisant la commande `aws kinesis get-records` pour récupérer certains enregistrements Amazon Kinesis Data Streams :

```
aws kinesis get-shard-iterator \
  --stream-name RecipientStream \
  --shard-id shardId-000000000000 \
  --shard-iterator-type TRIM_HORIZON

{
  "ShardIterator":
```

```
"AAAAAAAAAAGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGB9v4scv+3vaq+f
+0IK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
  --limit 10 \
  --shard-iterator
  "AAAAAAAAAAGU/
kLvNggvndHq2UIF0w5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3Igvq0b5dRA0yybNdRcdzvnC35KQANoHzzahKdRGB9v4scv+3vaq+f
+0IK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

### Note

Vous devrez peut-être réexécuter la `get-records` commande plusieurs fois avant qu'Amazon Kinesis Data Streams ne commence à renvoyer des données.

Vous devriez voir une réponse contenant un ensemble d'enregistrements Amazon Kinesis Data Streams. L'attribut de données de l'enregistrement Amazon 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": "3195310660696698337880902507980421114328961542429EXAMPLE",
```



```

    "timestamp": 1432826855000,
    "message": "{\"eventVersion\":\"1.03\", \"userIdentity\":{\"type\":\"Root
\"}\"}
  },
  {
    "id": "3195310660696698337880902507980421114328961542429EXAMPLE",
    "timestamp": 1432826855000,
    "message": "{\"eventVersion\":\"1.03\", \"userIdentity\":{\"type\":\"Root
\"}\"}
  },
  {
    "id": "3195310660696698337880902507980421114328961542429EXAMPLE",
    "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 Amazon 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": "{\n\"Version\": \"2012-10-17\", \"Statement\":
        [{\n\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\n\"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 `PutDestinationPolicy` pour 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 dans [Activer 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 dans [Étape 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 de journal** : définit une destination qui encapsule un flux Amazon Kinesis Data Streams et indique à CloudWatch Logs que le destinataire souhaite recevoir des données de 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. Cependant, la AWS ressource pointée par 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 : créer 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 actif [Étape 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": "AR0AR3BXASEKYJYWF243H",
"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 : créer 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    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n      \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action\n\" : \"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 à 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:SourceArn](#) [aws: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:service:*: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 Amazon Kinesis Data Streams [Étape 2 : créer une destination](#) et Firehose montrent comment vous pouvez utiliser la clé de contexte de condition globale pour éviter `aws:SourceArn` le problème de confusion lié aux 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 davantage de 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'`PutAccountPolicyAPI` 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'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, consultez [CloudWatch 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 ("").

### Exemple: 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

### Exemple: 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

### Exemple: 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 mot `REQUEST`, 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`

### Exemple: 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`

### Exemple: 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`

Exemple: 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 correspondre 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](#).

### Exemple: 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
```

#### Exemple: 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 vous publiez des journaux directement sur Amazon S3 ou Firehose, des frais de CloudWatch livraison s'appliquent. Si vous envoyez des journaux à Amazon S3, les **AWS\_REGION-S3-Egress-Bytes** frais apparaissent dans Cost Explorer ou sur votre facture. Si vous envoyez des journaux à Firehose, des **AWS\_REGION-FH-Egress-Bytes** frais apparaissent. Pour plus d'informations sur la tarification des journaux vendus, consultez l'onglet Journaux [sur 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 | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|-----------------------------------------------------|-----------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Journaux d'accès Amazon API Gateway</a> | Journaux vendés | <a href="#">Pris en charge</a>                                      |                                                                   |                                                              |                                                              |

| Source des journaux                                                      | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|--------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
|                                                                          |                        | <a href="#">[Autorisations V1]</a>                                  |                                                                   |                                                              |                                                              |
| <a href="#">AWS AppSync journaux</a>                                     | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Journaux Amazon Aurora MySQL</a>                             | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Amazon Bedrock Journalisation des bases de connaissances</a> | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Amazon Bedrock Journalisation de l'agent</a>                 | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Amazon Bedrock AgentCore Exécution</a>                       | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            | <a href="#">Pris en charge [Autorisations V2]</a>            |



| Source des journaux                                                                                  | Type de journal | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|------------------------------------------------------------------------------------------------------|-----------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Amazon Bedrock AgentCore Passerelle</a>                                                  | Journaux vendés | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            | <a href="#">Pris en charge [Autorisations V2]</a>            |
| <a href="#">Amazon Bedrock AgentCore Identité</a>                                                    | Journaux vendés | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            | <a href="#">Pris en charge [Autorisations V2]</a>            |
| <a href="#">Amazon Bedrock AgentCore Mémoire</a>                                                     | Journaux vendés | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            | <a href="#">Pris en charge [Autorisations V2]</a>            |
| <a href="#">Amazon Bedrock AgentCore Outils</a>                                                      | Journaux vendés | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            | <a href="#">Pris en charge [Autorisations V2]</a>            |
| <a href="#">Journaux de métriques de qualité multimédia Amazon Chime et journaux de messages SIP</a> | Journaux vendés | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   |                                                              |                                                              |
| <a href="#">CloudFront: journaux d'accès</a>                                                         | Journaux vendés | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |

| Source des journaux                                                                | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|------------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">AWS CloudHSM journaux d'audit</a>                                      | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">CloudWatch De toute évidence, journaux des événements d'évaluation</a> | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 |                                                              |                                                              |
| <a href="#">CloudWatch Journaux du moniteur Internet</a>                           | Journaux vendus        |                                                                     | <a href="#">Pris en charge [Autorisations V1]</a>                 |                                                              |                                                              |
| <a href="#">CloudTrail journaux</a>                                                | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">AWS CodeBuild journaux</a>                                             | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Amazon CodeWhisperer journaux d'événements</a>                         | Journaux vendus        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |

| Source des journaux                                     | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|---------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Amazon Cognito journaux</a>                 | Journaux vendés        | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   |                                                              |                                                              |
| <a href="#">Journaux Amazon Connect</a>                 | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">AWS DataSync journaux</a>                   | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Agent AWS DevOps journaux</a>               | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Journaux Amazon ElastiCache (Redis OSS)</a> | Journaux vendés        | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">AWS Elastic Beanstalk Journaux</a>          | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |

| Source des journaux                                                              | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|----------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Journaux Amazon Elastic Container Service</a>                        | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Journaux du mode automatique d'Amazon Elastic Kubernetes Service</a> | Journaux vendus        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Journaux de plan de contrôle d'Amazon Elastic Kubernetes Service</a> | Journaux vendus        | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">AWS Elemental MediaPackage journaux d'accès</a>                      | Journaux vendus        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">AWS Elemental MediaTailor journaux</a>                               | Journaux vendus        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Résolution des entités AWS Journaux</a>                              | Journaux vendus        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |

| Source des journaux                                                 | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|---------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Amazon EventBridge Enregistrement des canalisations</a> | Journaux vendés        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Amazon EventBridge bus événementiels</a>                | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">AWS Fargate Journaux</a>                                | Journaux personnalisés | <a href="#">Pris en charge</a>                                      |                                                                   |                                                              |                                                              |
| <a href="#">Journaux d'expériences AWS Fault Injection Service</a>  | Journaux vendés        |                                                                     | <a href="#">Pris en charge [Autorisations V1]</a>                 |                                                              |                                                              |
| <a href="#">Amazon FinSpace</a>                                     | Journaux vendés        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">AWS Global Accelerator journaux de flux</a>             | Journaux vendés        |                                                                     | <a href="#">Pris en charge [Autorisations V1]</a>                 |                                                              |                                                              |

| Source des journaux                                                       | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|---------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">AWS Glue journaux de tâches</a>                               | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Journaux d'erreurs d'IAM Identity Center</a>                  | Journaux vendus        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Journaux de conversation Amazon Interactive Video Service</a> | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">AWS IoT Journaux</a>                                          | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">AWS IoT FleetWise journaux</a>                                | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">AWS Lambda Journaux</a>                                       | Journaux vendus        | Pris en charge                                                      | Pris en charge                                                    | Pris en charge                                               |                                                              |
| <a href="#">Journaux Amazon Macie</a>                                     | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |

| Source des journaux                                            | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|----------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Journaux Amazon SES</a>                            | Journaux vendus        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">AWS Mainframe Modernization</a>                    | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Journaux Amazon Managed Service for Prometheus</a> | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   |                                                              |                                                              |
| <a href="#">Journaux de l'agent Amazon MSK</a>                 | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Journaux Amazon MSK Connect</a>                    | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Journaux Amazon MQ</a>                             | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |

| Source des journaux                                              | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">AWS Journaux du Network Firewall</a>                 | Journaux vendés        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">AWS Journaux du proxy Network Firewall</a>           | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Journaux d'accès de Network Load Balancer</a>        | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">OpenSearch journaux</a>                              | Journaux personnalisés | <a href="#">Pris en charge</a>                                      |                                                                   |                                                              |                                                              |
| <a href="#">Journaux d'ingestion d'Amazon OpenSearch Service</a> | Journaux vendés        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| Journaux <a href="#">AWS PCS</a>                                 | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |



| Source des journaux                                                                    | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|----------------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Journaux du connecteur Amazon Q Business</a>                               | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Journaux de conversation Amazon Q Business</a>                             | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Journaux de discussion et de commentaires Amazon Quick</a>                 | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">Journaux ServicePostgre SQL de la base de données relationnelle Amazon</a> | Journaux personnalisés | <a href="#">Pris en charge</a>                                      |                                                                   |                                                              |                                                              |
| AWS Bûches en <a href="#">tissu RTB</a>                                                | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">AWS Security Hub (CSPM)</a>                                                | Journaux vendés        | <a href="#">Pris en charge [Autorisations V2]</a>                   |                                                                   |                                                              |                                                              |

| Source des journaux                                                    | Type de journal | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|------------------------------------------------------------------------|-----------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Journaux des requêtes DNS publiques Amazon Route 53</a>    | Journaux vendus | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Journaux de requête Amazon Route 53 Resolver</a>           | Journaux vendus | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 |                                                              |                                                              |
| <a href="#">Événements Amazon SageMaker AI</a>                         | Journaux vendus | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   |                                                              |                                                              |
| <a href="#">Événements destinés aux employés d'Amazon SageMaker AI</a> | Journaux vendus | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   |                                                              |                                                              |
| <a href="#">AWS Journaux VPN de site à site</a>                        | Journaux vendus | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Journaux d'Amazon Simple Email Service</a>                 | Journaux vendus | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |

| Source des journaux                                                                                    | Type de journal        | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|--------------------------------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Journaux Amazon Simple Notification Service</a>                                            | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Journaux des politiques de protection des données d'Amazon Simple Notification Service</a> | Journaux personnalisés | Pris en charge                                                      |                                                                   |                                                              |                                                              |
| <a href="#">Fichiers de flux de données des instances Spot EC2</a>                                     | Journaux vendus        |                                                                     | <a href="#">Pris en charge [Autorisations V1]</a>                 |                                                              |                                                              |
| <a href="#">AWS Step Functions Journaux Express Workflow et Standard Workflow</a>                      | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   |                                                              |                                                              |
| <a href="#">Journaux d'audit et journaux d'état de Storage Gateway</a>                                 | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   |                                                                   |                                                              |                                                              |
| <a href="#">AWS Transfer Family journaux</a>                                                           | Journaux vendus        | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |

| Source des journaux                                             | Type de journal | <a href="#">the section called “Logs envoyés à CloudWatch Logs”</a> | <a href="#">the section called “Journaux envoyés à Amazon S3”</a> | <a href="#">the section called “Logs envoyés à Firehose”</a> | <a href="#">the section called “Traces envoyées à X-Ray”</a> |
|-----------------------------------------------------------------|-----------------|---------------------------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| <a href="#">Accès vérifié par AWS journaux</a>                  | Journaux vendés | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Journaux de flux d'Amazon Virtual Private Cloud</a> | Journaux vendés | <a href="#">Pris en charge</a>                                      | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Journaux d'accès Amazon VPC Lattice</a>             | Journaux vendés | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge [Autorisations V1]</a>            |                                                              |
| <a href="#">Serveur de routage Amazon VPC</a>                   | Journaux vendés | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |
| <a href="#">AWS WAF Journaux</a>                                | Journaux vendés | <a href="#">Pris en charge [Autorisations V1]</a>                   | <a href="#">Pris en charge [Autorisations V1]</a>                 | <a href="#">Pris en charge</a>                               |                                                              |
| <a href="#">Amazon WorkMail journaux d'audit</a>                | Journaux vendés | <a href="#">Pris en charge [Autorisations V2]</a>                   | <a href="#">Pris en charge [Autorisations V2]</a>                 | <a href="#">Pris en charge [Autorisations V2]</a>            |                                                              |

## 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 précédent 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:DescribeLogGroups` ou `logs:DescribeResourcePolicies`, ou l'autorisation `logs:PutResourcePolicy`, veuillez 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'autorisation `logs:CreateLogDelivery` 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 autorisations `logs:DescribeLogGroups` et `logs:PutResourcePolicy`, 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. Pour les abonnements nouvellement créés, les politiques de ressources sont configurées au niveau du groupe de journaux et leur taille maximale est de 51 200 octets. Si une politique de ressources existante au niveau du compte accorde déjà des autorisations par le biais de caractères génériques, aucune politique distincte au niveau du groupe de journaux ne sera créée. Pour vérifier la politique de ressources au niveau du groupe de journaux pour un groupe de journaux spécifique, utilisez la `describe-resource-policies` commande avec

le `--resource-arn` paramètre défini sur l'ARN du groupe de journaux et le `--policy-scope` paramètre défini sur `RESOURCE`

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

La limite de politique de ressources du groupe de journaux est de 51 200 octets. Une fois cette limite atteinte, AWS ne peut pas ajouter de nouvelles autorisations. Cela oblige les clients à modifier manuellement la politique afin d'accorder au `delivery.logs.amazonaws.com` service les autorisations principales sur les `logs:PutLogEvents` actions `logs:CreateLogStream` et. Les clients doivent utiliser un préfixe de nom de groupe de journaux avec des caractères génériques, tels que ce nom de groupe de journaux, `/aws/vendedlogs/*` et utiliser ce nom de groupe de journaux pour la création de futures livraisons.

## 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/aws/
vendedlogs/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": [
            "0123456789"
          ]
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:logs:us-east-1:111122223333:*"
          ]
        }
      }
    }
  ]
}
```



```
]
}
```

## 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 `s3:ListBucket` autorisation n'a pas été accordée `delivery.logs.amazonaws.com`. Pour éviter ces erreurs dans vos CloudTrail journaux, vous devez accorder `s3:ListBucket` autorisation `delivery.logs.amazonaws.com` et inclure les `Condition` paramètres indiqués dans `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 [PutDeliveryDestinationPolicy](#) dans 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'[DeleteDelivery](#)opération.
2. **DeliverySource**Supprimez-le à l'aide de l'[DeleteDeliverySource](#)opération.
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'[DeleteDeliveryDestinations](#)opération.

## 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:PutResourcePolicy``logs: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. Pour les abonnements nouvellement créés, les politiques de ressources sont configurées au niveau du groupe de journaux et leur taille maximale est de 51 200 octets. Si une politique de ressources existante au niveau du compte accorde déjà des autorisations par le biais de caractères génériques, aucune politique distincte au niveau du groupe de journaux ne sera créée. Pour vérifier la politique de ressources au niveau du groupe de journaux pour un groupe de journaux spécifique, utilisez la `describe-resource-policies` commande avec le `--resource-arn` paramètre défini sur l'ARN du groupe de journaux et le `--policy-scope` paramètre défini sur `RESOURCE`

### 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:*"
            ]
        }
    }
}
]
}

```

La limite de politique de ressources du groupe de journaux est de 51 200 octets. Une fois cette limite atteinte, AWS ne peut pas ajouter de nouvelles autorisations. Cela oblige les clients à modifier manuellement la politique afin d'accorder au `delivery.logs.amazonaws.com` service les autorisations principales sur les `logs:PutLogEvents` actions `logs:CreateLogStream` et. Les clients doivent utiliser un préfixe de nom de groupe de journaux avec des caractères génériques, tels que ce nom de groupe de journaux, `/aws/vendedlogs/*` et utiliser ce nom de groupe de journaux pour la création de futures livraisons.

## 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/aws/
vendedlogs/*"
    ],
    "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 `s3:ListBucket` autorisation n'a pas été accordée `delivery.logs.amazonaws.com`. Pour éviter ces erreurs dans vos CloudTrail journaux, vous devez accorder `s3:ListBucket` autorisation `delivery.logs.amazonaws.com` et inclure les Condition paramètres indiqués dans `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-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": "AllowUpdatesToResourcePolicyXRay",
      "Effect": "Allow",

```

```

        "Action": [
            "xray:PutResourcePolicy",
            "xray:ListResourcePolicies",
            "xray:GetTraceSegmentDestination"
        ],
        "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 l'utilisateur qui configure le suivi dispose `xray:PutResourcePolicy` d'`xray:ListResourcePolicies` autorisations sur le compte, crée AWS automatiquement la politique de ressources lorsque vous commencez à envoyer des traces à X-Ray. La politique créée dépend du service source :

### Amazon Bedrock AgentCore ressources

AWS crée une politique de ressources par type de ressource. La politique utilise des modèles génériques s'étendant jusqu'aux limites du compte, couvrant toutes les ressources du même type de Amazon Bedrock AgentCore ressources du compte. Par exemple, si une ressource Amazon Bedrock AgentCore mémoire est activée pour la livraison de traces, la politique couvre toutes les ressources de mémoire de ce compte, y compris les ressources de mémoire créées dans le futur.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "xray:PutTraceSegments",
      "Resource": "*",
      "Condition": {
        "StringEquals": {

```

```

        "aws:SourceAccount": "123456789012"
      },
      "ForAllValues:ArnLike": {
        "logs:LogGeneratingResourceArns": "arn:aws:bedrock-agentcore:us-
east-1:123456789012:memory/*"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:delivery-
source:*"
      }
    }
  ]
}

```

## Autres AWS services

Pour les autres services qui prennent en charge la fourniture de traces, AWS crée une politique de ressources limitée à la ressource source spécifique.

## JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "xray:PutTraceSegments",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ForAllValues:ArnLike": {
          "logs:LogGeneratingResourceArns": "arn:aws:bedrock:us-
east-1:123456789012:knowledge-base/KnowledgeBaseId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:delivery-
source:xray-test"
        }
      }
    }
  ]
}

```

```
}  
}  
}  
]  
}
```

## 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'AllowVendedLogDeliveryForResourceaction 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. Dans l'exemple suivant, la politique a été mise à jour pour activer les journaux vendus depuis Amazon SES.

### JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "ServiceLevelAccessForLogDelivery",  
      "Effect": "Allow",  
      "Action": [  
        "ses:AllowVendedLogDeliveryForResource"  
      ],  
      "Resource": "arn:aws:ses: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": [
```

```

    "x"
  ]
}
]
}

```

## 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/XXXXXXXXXXXX
```

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-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": [
            "123456789012"
          ]
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:logs:us-east-1:123456789012: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:BBBBBBBBBBBBB: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 à 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 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 de contexte de condition globale pour éviter `aws:SourceArn` le problème de confusion lié aux adjoints.

# 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 S3 du même compte ou d'un autre compte.

Vous pouvez effectuer les opérations 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

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, consultez [Traitement 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 organiser les données exportées, spécifiez un préfixe pour chaque tâche d'exportation qui sera utilisé comme préfixe clé Amazon S3 pour tous les objets exportés. Par exemple : `prod/app-logs/2026-01-03/` ou `log-group-name/backup/`

## 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]+/\x00/' | 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]+/\x00/' | 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 \(CLI\)](#)
- [Annuler une tâche d'exportation \(CLI\)](#)

## 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 `amzn-s3-demo-bucket`.

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 sur le même compte \(console\)](#)
- [Exportation entre comptes \(console\)](#)

## Exportation sur le même compte (console)

Si le compartiment Amazon S3 se trouve dans le même compte que les journaux à exporter, suivez les instructions de cette section.

### Rubriques

- [Création d'un compartiment Amazon S3 \(console\)](#)
- [Configurer les autorisations d'accès \(console\)](#)
- [Définir des autorisations sur un compartiment Amazon S3 \(console\)](#)
- [\(Facultatif\) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS \(console\)](#)
- [Création d'une tâche d'exportation \(console\)](#)

## Création d'un compartiment Amazon S3 (console)

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 Amazon S3 doit résider dans la même région que les données de journal à exporter. CloudWatch Logs ne prend pas en charge l'exportation de données vers des compartiments Amazon S3 d'une autre région.

Pour créer un compartiment Amazon 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 journaux.
6. Choisissez Créer.

## Configurer les autorisations d'accès (console)

Pour créer la tâche d'exportation, vous devez être connecté avec le rôle AmazonS3ReadOnlyAccess IAM 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.

## Définir des autorisations sur un compartiment Amazon S3 (console)

Par défaut, tous les objets et les compartiments Amazon S3 sont privés. Seul le propriétaire de la ressource, Compte AWS celui qui a 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 les exportations vers les compartiments Amazon S3, nous vous demandons désormais de spécifier la liste des comptes sources autorisés à exporter les données de journal 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 Amazon 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.
2. Choisissez Permissions, Bucket policy.
3. Dans Bucket Policy Editor (Éditeur de stratégie de compartiment), ajoutez la politique ci-dessous. Remplacez `amzn-s3-demo-bucket` 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 Amazon 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) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS (console)

Cette étape n'est nécessaire que si vous exportez vers un compartiment Amazon 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 managed 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 [Création d'un compartiment S3 \(CLI\)](#) 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.

## Création d'une tâche d'exportation (console)

Dans cette procédure, vous créez la tâche d'exportation pour exporter les journaux à partir 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 [Configurer les autorisations d'accès \(console\)](#).
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 le 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 entre comptes (console)

Si le compartiment Amazon S3 se trouve dans un compte différent de celui des journaux à exporter, suivez les instructions de cette section.

### Rubriques

- [Création d'un compartiment Amazon S3 pour l'exportation entre comptes \(console\)](#)
- [Configurer les autorisations d'accès pour l'exportation entre comptes \(console\)](#)
- [Définir des autorisations sur un compartiment S3 pour l'exportation entre comptes \(console\)](#)
- [\(Facultatif\) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS pour l'exportation entre comptes \(console\)](#)
- [Création d'une tâche d'exportation pour l'exportation entre comptes \(console\)](#)

## Création d'un compartiment Amazon S3 pour l'exportation entre comptes (console)

Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Toutefois, si vous souhaitez utiliser un bucket existant, vous pouvez ignorer cette procédure.

 Note

Le compartiment Amazon S3 doit résider dans la même région que les données de journal à exporter. CloudWatch Logs ne prend pas en charge l'exportation de données vers des compartiments Amazon S3 d'une autre région.

Pour créer un compartiment Amazon 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 journaux.
6. Choisissez Créer.

## Configurer les autorisations d'accès pour l'exportation entre comptes (console)

Tout d'abord, vous devez créer une nouvelle politique IAM pour permettre à CloudWatch Logs d'avoir l'`s3:PutObject` action pour le compartiment Amazon S3 de destination dans le compte de destination.

Outre les `s3:PutObject` actions, les actions supplémentaires incluses dans la politique varient selon que le compartiment de destination utilise le AWS KMS chiffrement ou qu'il a été ACLs activé à l'aide du paramètre [S3 Object Ownership](#).

- Si vous utilisez le chiffrement KMS, ajoutez les `kms:Decrypt` actions `kms:GenerateDataKey` et pour la ressource clé
- Si ACLs elles sont activées sur le bucket, ajoutez l'`s3:PutObjectAcl` action pour la ressource du bucket.

Modifiez `amzn-s3-demo-bucket` le nom de votre compartiment S3 de destination dans les politiques suivantes.

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:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

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"
  }
]
}

```

S' ACLs ils sont activés sur le compartiment de destination, ajoutez s3 : PutObjectAcl au bloc s3 : PutObject Action dans les politiques ci-dessus.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}

```

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 une tâche d'exportation, vous devez être connecté avec un rôle IAM auquel est attachée la politique AmazonS3ReadOnlyAccess gérée, la stratégie IAM créée ci-dessus, et également avec 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.

## Définir des autorisations sur un compartiment S3 pour l'exportation entre comptes (console)

Par défaut, tous les objets et les compartiments S3 sont privés. Seul le propriétaire de la ressource, Compte AWS celui qui a 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.
2. Choisissez Permissions, Bucket policy.
3. Dans Bucket Policy Editor (Éditeur de stratégie de compartiment), ajoutez la politique ci-dessous. Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment S3. Assurez-vous de spécifier le point de terminaison correct de la région, tel que `us-east-1`, pour Principal.

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:::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:*"
      ]
    }
  },
  {
    "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"
      }
    }
  }
]

```

```
}
```

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) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS pour l'exportation entre comptes (console)

Cette procédure 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 `us-east-1` par la région de vos journaux, `account-ARN` par l'ARN du compte propriétaire de la clé KMS, `123456789012` par le numéro de compte propriétaire de la clé KMS, `key_id` par l'identifiant de la clé KMS et `role_name` par le rôle utilisé pour créer la tâche d'exportation.

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-1:123456789012: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 [Création d'un compartiment S3 \(CLI\)](#) 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.

## Création d'une tâche d'exportation pour l'exportation entre comptes (console)

Dans cette procédure, vous créez la tâche d'exportation pour exporter les journaux à partir 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 [Configurer les autorisations d'accès \(console\)](#).
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 `amzn-s3-demo-bucket`. 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 sur le même compte \(CLI\)](#)
- [Exportation entre comptes \(CLI\)](#)

## Exportation sur le même compte (CLI)

Si le compartiment Amazon S3 se trouve dans le même compte que les journaux à exporter, suivez les instructions de cette section.

## Rubriques

- [Création d'un compartiment S3 \(CLI\)](#)
- [Configurer les autorisations d'accès \(CLI\)](#)
- [Définir des autorisations sur un compartiment S3 \(CLI\)](#)
- [\(Facultatif\) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS \(CLI\)](#)
- [Création d'une tâche d'exportation \(CLI\)](#)

## Création d'un compartiment S3 (CLI)

Nous vous recommandons d'utiliser un bucket créé spécifiquement pour CloudWatch Logs. Toutefois, si vous souhaitez utiliser un bucket existant, vous pouvez ignorer cette procédure.



 Note

Le compartiment S3 doit résider dans la même région que les données de 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 amzn-s3-demo-bucket --create-bucket-configuration  
LocationConstraint=us-east-2
```

Voici un exemple de sortie.

```
{  
  "Location": "/amzn-s3-demo-bucket"  
}
```

## Configurer les autorisations d'accès (CLI)

Pour créer la tâche d'exportation ultérieurement, vous devez être connecté avec le rôle `AmazonS3ReadOnlyAccess` IAM 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.

## Définir des autorisations sur un compartiment S3 (CLI)

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 `amzn-s3-demo-bucket` 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-east-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-policy](#) commande. 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 amzn-s3-demo-bucket --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) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS (CLI)

Cette procédure 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-ARN: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 *amzn-s3-demo-bucket* remplaçant par le nom du compartiment vers lequel vous exportez les journaux.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-configuration file://bucketencryption.json
```

Si la commande ne renvoie pas d'erreur, le processus est réussi.

## Création d'une tâche d'exportation (CLI)

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 [Configurer les autorisations d'accès \(CLI\)](#).
2. À l'invite de commandes, utilisez la [create-export-task](#) commande 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 "amzn-s3-demo-bucket" --destination-prefix "export-task-output"
```

Voici un exemple de sortie.

```
{
  "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

## Exportation entre comptes (CLI)

Si le compartiment Amazon S3 se trouve dans un compte différent de celui des journaux à exporter, suivez les instructions de cette section.

### Rubriques

- [Création d'un compartiment S3 pour l'exportation entre comptes \(CLI\)](#)

- [Configurer les autorisations d'accès pour l'exportation entre comptes \(CLI\)](#)
- [Définissez des autorisations sur un compartiment S3 pour l'exportation entre comptes \(CLI\)](#)
- [\(Facultatif\) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS pour l'exportation entre comptes \(CLI\)](#)
- [Création d'une tâche d'exportation pour l'exportation entre comptes \(CLI\)](#)

## Création d'un compartiment S3 pour l'exportation entre comptes (CLI)

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 de 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 amzn-s3-demo-bucket --create-bucket-configuration  
LocationConstraint=us-east-2
```

Voici un exemple de sortie.

```
{  
  "Location": "/amzn-s3-demo-bucket"  
}
```

## Configurer les autorisations d'accès pour l'exportation entre comptes (CLI)

Tout d'abord, vous devez créer une nouvelle politique IAM pour permettre à CloudWatch Logs d'avoir `s3:PutObject` pour le compartiment Amazon S3 de destination dans le compte de destination.



Outre les `s3:PutObject` actions, les actions supplémentaires incluses dans la politique varient selon que le compartiment de destination utilise le AWS KMS chiffrement ou qu'il a été ACLs activé à l'aide du paramètre [S3 Object Ownership](#).

- Si vous utilisez le chiffrement KMS, ajoutez les `kms:Decrypt` actions `kms:GenerateDataKey` et pour la ressource clé
- Si ACLs elles sont activées sur le bucket, ajoutez l'`s3:PutObjectAcl` action pour la ressource du bucket.

Modifiez `amzn-s3-demo-bucket` le nom de votre compartiment S3 de destination dans les politiques suivantes.

La politique que vous créez varie selon que le compartiment de destination utilise ou non le chiffrement AWS KMS . 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:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

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"
    }
]
}

```

S' ACLs ils sont activés sur le compartiment de destination, ajoutez s3 : PutObjectAcl au bloc s3 : PutObject Action dans les politiques ci-dessus.

## JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}

```

Pour créer une tâche d'exportation, vous devez être connecté avec un rôle IAM auquel est attachée la politique AmazonS3ReadOnlyAccess gérée, la stratégie IAM créée ci-dessus, et également avec 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.

## Définissez des autorisations sur un compartiment S3 pour l'exportation entre comptes (CLI)

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 politique d'accès suivante, en remplaçant `amzn-s3-demo-bucket` le nom de votre compartiment S3 de destination, `Principal` au point de terminaison de la région dans laquelle vous exportez les données de journal, par exemple `us-east-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:::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:*"
        ]
      }
    }
  },
  {
    "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-policy](#) commande. 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 amzn-s3-demo-bucket --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) Exportation vers un compartiment Amazon S3 de destination chiffré avec SSE-KMS pour l'exportation entre comptes (CLI)

Cette procédure 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 `us-east-1` par la région de vos journaux.
  - `account-ARN` Remplacez-le par l'ARN du compte propriétaire de la clé KMS.
  - `123456789012` Remplacez-le par le numéro de compte auquel appartient la clé KMS.
  - `key_id` avec l'identifiant de la clé KMS.
  - `role_name` avec le rôle utilisé pour créer la tâche d'exportation.

### 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": "account-ARN"
        },
        "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-1:123456789012: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 *amzn-s3-demo-bucket* remplaçant par le nom du compartiment vers lequel vous exportez les journaux.



```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-configuration file://bucketencryption.json
```

Si la commande ne renvoie pas d'erreur, le processus est réussi.

## Création d'une tâche d'exportation pour l'exportation entre comptes (CLI)

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 [Configurer les autorisations d'accès \(CLI\)](#).
2. À l'invite de commandes, utilisez la [create-export-task](#) commande suivante pour créer la tâche d'exportation.

```
aws logs create-export-task --profile CWExportUser --task-name "my-log-group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --to 1441494000000 --destination "amzn-s3-demo-bucket" --destination-prefix "export-task-output"
```

Voici un exemple de sortie.

```
{  
  "taskId": "cda45419-90ea-4db5-9833-aade86253e66"  
}
```

## Décrire les tâches d'exportation (CLI)

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-tasks](#) commande 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": "amzn-s3-demo-bucket",
  "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 (CLI)

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-task](#) commande suivante :

```
aws logs --profile CWExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

Vous pouvez utiliser la [describe-export-tasks](#) commande 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, consultez [Classes 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.

## Conditions préalables

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 souhaitez 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": [  
    {  
      "Effect": "Allow",  
      "Action": "logs:CreateLogGroup",  
      "Resource": "arn:aws:logs:*:*:log-group:*",  
      "Condition": {"Bool": {"aws:MultiValueCognitoAuthenticationGroups": "true"}},  
      "Principal": {"AWS": "arn:aws:iam::*:role/*"}  
    },  
    {  
      "Effect": "Allow",  
      "Action": "logs:CreateLogStream",  
      "Resource": "arn:aws:logs:*:*:log-group:*:log-stream:*",  
      "Condition": {"Bool": {"aws:MultiValueCognitoAuthenticationGroups": "true"}},  
      "Principal": {"AWS": "arn:aws:iam::*:role/*"}  
    },  
    {  
      "Effect": "Allow",  
      "Action": "logs:PutLogEvents",  
      "Resource": "arn:aws:logs:*:*:log-group:*:log-stream:*",  
      "Condition": {"Bool": {"aws:MultiValueCognitoAuthenticationGroups": "true"}},  
      "Principal": {"AWS": "arn:aws:iam::*:role/*"}  
    },  
    {  
      "Effect": "Allow",  
      "Action": "logs:DescribeLogStreams",  
      "Resource": "arn:aws:logs:*:*:log-group:*:log-stream:*",  
      "Condition": {"Bool": {"aws:MultiValueCognitoAuthenticationGroups": "true"}},  
      "Principal": {"AWS": "arn:aws:iam::*:role/*"}  
    },  
    {  
      "Effect": "Allow",  
      "Action": "logs:FilterLogEvents",  
      "Resource": "arn:aws:logs:*:*:log-group:*:log-stream:*",  
      "Condition": {"Bool": {"aws:MultiValueCognitoAuthenticationGroups": "true"}},  
      "Principal": {"AWS": "arn:aws:iam::*:role/*"}  
    },  
    {  
      "Effect": "Allow",  
      "Action": "logs:DeleteLogStream",  
      "Resource": "arn:aws:logs:*:*:log-group:*:log-stream:*",  
      "Condition": {"Bool": {"aws:MultiValueCognitoAuthenticationGroups": "true"}},  
      "Principal": {"AWS": "arn:aws:iam::*:role/*"}  
    },  
    {  
      "Effect": "Allow",  
      "Action": "logs:DeleteLogGroup",  
      "Resource": "arn:aws:logs:*:*:log-group:*",  
      "Condition": {"Bool": {"aws:MultiValueCognitoAuthenticationGroups": "true"}},  
      "Principal": {"AWS": "arn:aws:iam::*:role/*"}  
    }  
  ]  
}
```

```
{
  "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 `AWSLambdaVPCAccessExecutionRole` politique 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, consultez [Utilisation 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](#)
  - [Configuration d'Amazon ECS Service Connect](#)
  - [Création de votre première fonction Lambda](#)
  - [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 à ce sujet 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/7c9ecccc2-38cb-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 [AssociateKmsKey](#) à 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit 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 à ce sujet 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 [CancelExportTask](#) à 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit 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 à ce sujet 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 [CreateExportTask](#) à 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

## Utilisation **CreateLogGroup** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser CreateLogGroup.

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 :

- [Configuration d'Amazon ECS Service Connect](#)

## .NET

### SDK pour .NET

#### 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](#).

```
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 [CreateLogGroup](#) à la section Référence des AWS SDK pour .NET API.

## CLI

### AWS CLI

La commande suivante permet de créer un groupe de journaux nommé `my-logs` :

```
aws logs create-log-group --log-group-name my-logs
```

- Pour plus de détails sur l'API, reportez-vous [CreateLogGroup](#) à la section Référence des AWS CLI commandes.

## 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 { 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 [CreateLogGroup](#) à 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 du kit 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 à ce sujet 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 [CreateLogStream](#) à la section Référence des AWS SDK pour .NET API.

## CLI

### AWS CLI

La commande suivante crée un flux de journaux 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, reportez-vous [CreateLogStream](#) à la section Référence des AWS CLI commandes.

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 du kit SDK.

### Utilisation **DeleteLogGroup** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser DeleteLogGroup.

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 les exemples de code suivants :

- [Configuration d'Amazon ECS Service Connect](#)
- [Création de votre première fonction Lambda](#)

## .NET

### SDK pour .NET

#### 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](#).

```
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 [DeleteLogGroup](#) à 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, reportez-vous [DeleteLogGroup](#) à la section Référence des AWS CLI commandes.

## 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 { 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 [DeleteLogGroup](#) à 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 du kit 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 à ce sujet 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 [DeleteSubscriptionFilter](#) à la section Référence des AWS SDK pour C++ API.

## Java

### SDK pour Java 2.x

#### 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 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 [DeleteSubscriptionFilter](#) à 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 { 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 [DeleteSubscriptionFilter](#) à 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 = {
  filterName: "FILTER",
  logGroupName: "LOG_GROUP",
};

cwl.deleteSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Pour plus d'informations, consultez le [Guide du développeur AWS SDK pour JavaScript](#).
- Pour plus de détails sur l'API, reportez-vous [DeleteSubscriptionFilter](#) à la section Référence des AWS SDK pour JavaScript API.

## Kotlin

### SDK pour Kotlin

#### 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](#).

```
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 [DeleteSubscriptionFilter](#) la 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit 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 à ce sujet 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 [DescribeExportTasks](#) à 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit 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 à ce sujet 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 [DescribeLogGroups](#) à 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, reportez-vous [DescribeLogGroups](#) à la section Référence des AWS CLI commandes.

## 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 {
  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 [DescribeLogGroups](#) à 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 du kit SDK.

## Utilisation **DescribeLogStreams** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeLogStreams`.

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 :

- [Création de votre première fonction Lambda](#)

### CLI

#### AWS CLI

La commande suivante affiche tous les flux de journaux 2015 commençant par le préfixe `my-logs` dans le groupe 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, reportez-vous [DescribeLogStreams](#) à la section Référence des AWS CLI commandes.

## Java

### SDK pour Java 2.x

#### 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](#).

Recherche les flux de journaux au sein d'un groupe de journaux spécifié qui correspondent à un préfixe donné.

```
/**
 * 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 CloudWatchLogsSearch {

    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <logGroupName> <logStreamName>

            Where:
                logGroupName - The name of the log group (for example,
                WeathertopJavaContainerLogs).
                logStreamName - The name of the log stream (for example,
                weathertop-java-stream).
                pattern - the pattern to use (for example, INFO)

            "";

        if (args.length != 3) {
            System.out.print(usage);
            System.exit(1);
        }

        String logGroupName = args[0] ;
        String logStreamName = args[1] ;
        String pattern = args[2] ;

        CloudWatchLogsClient cwlClient = CloudWatchLogsClient.builder()
            .region(Region.US_EAST_1)
            .build();

        searchLogStreamsAndFilterEvents(cwlClient, logGroupName, logStreamName,
        pattern);
    }

    /**
     * Searches for log streams with a specific prefix within a log group and
     * filters log events based on a specified pattern.
     *
     * @param cwlClient      the CloudWatchLogsClient used to interact with AWS
     * CloudWatch Logs

```

```
* @param logGroupName    the name of the log group to search within
* @param logStreamPrefix the prefix of the log streams to search for
* @param pattern          the pattern to filter log events by
*/
public static void searchLogStreamsAndFilterEvents(CloudWatchLogsClient
cwlClient, String logGroupName, String logStreamPrefix, String pattern) {
    DescribeLogStreamsRequest describeLogStreamsRequest =
DescribeLogStreamsRequest.builder()
        .logGroupName(logGroupName)
        .logStreamNamePrefix(logStreamPrefix)
        .build();

    DescribeLogStreamsResponse describeLogStreamsResponse =
cwlClient.describeLogStreams(describeLogStreamsRequest);
    List<LogStream> logStreams = describeLogStreamsResponse.logStreams();

    for (LogStream logStream : logStreams) {
        String logStreamName = logStream.logStreamName();
        System.out.println("Searching in log stream: " + logStreamName);

        FilterLogEventsRequest filterLogEventsRequest =
FilterLogEventsRequest.builder()
            .logGroupName(logGroupName)
            .logStreamNames(logStreamName)
            .filterPattern(pattern)
            .build();

        FilterLogEventsResponse filterLogEventsResponse =
cwlClient.filterLogEvents(filterLogEventsRequest);

        for (FilteredLogEvent event : filterLogEventsResponse.events()) {
            System.out.println(event.message());
        }

        System.out.println("-----"); //
        Separator for better readability
    }
}
```

Imprime les métadonnées relatives au flux de journaux le plus récent d'un groupe de journaux spécifié.

```
/**
 * 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 [DescribeLogStreams](#) à 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit 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 à ce sujet 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 = cw1.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 [DescribeSubscriptionFilters](#) à la section Référence des AWS SDK pour C++ API.

## Java

### SDK pour Java 2.x

#### 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 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 [DescribeSubscriptionFilters](#) à 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 { 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 [DescribeSubscriptionFilters](#) à 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 = {
  logGroupName: "GROUP_NAME",
  limit: 5,
};

cwl.describeSubscriptionFilters(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.subscriptionFilters);
  }
});
```

- Pour plus d'informations, consultez le [Guide du développeur AWS SDK pour JavaScript](#).
- Pour plus de détails sur l'API, reportez-vous [DescribeSubscriptionFilters](#) à la section Référence des AWS SDK pour JavaScript API.

## Kotlin

### SDK pour Kotlin

#### 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](#).

```
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 [DescribeSubscriptionFilters](#) la 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

## Utilisation **GetLogEvents** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser GetLogEvents.

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 :

- [Création de votre première fonction Lambda](#)

### CLI

#### AWS CLI

La commande suivante extrait les événements du journal à partir d'un flux de journaux nommé 20150601 dans le groupe de journaux my-logs :

```
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, reportez-vous [GetLogEvents](#) à la section Référence des AWS CLI commandes.

## Java

### SDK pour Java 2.x

#### 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 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.DescribeLogStreamsRequest;
import
    software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.GetLogEventsRequest;
import software.amazon.awssdk.services.cloudwatchlogs.model.GetLogEventsResponse;

import java.time.Instant;
import java.time.temporal.ChronoUnit;

/**
 * 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 = "WeathertopJavaContainerLogs" ; //args[0];
        String logStreamName = "weathertop-java-stream" ; //args[1];

        Region region = Region.US_EAST_1 ;
        CloudWatchLogsClient cloudWatchLogsClient =
CloudWatchLogsClient.builder()
            .region(region)
            .build();

        getCWLogEvents(cloudWatchLogsClient, logGroupName, logStreamName);
        cloudWatchLogsClient.close();
    }

    public static void getCWLogEvents(CloudWatchLogsClient cloudWatchLogsClient,
                                     String logGroupName,
                                     String logStreamPrefix) {

        try {
            // First, find the exact log stream name
            DescribeLogStreamsRequest describeRequest =
DescribeLogStreamsRequest.builder()
                .logGroupName(logGroupName)

```

```
        .logStreamNamePrefix(logStreamPrefix)
        .limit(1) // get the first matching stream
        .build();

    DescribeLogStreamsResponse describeResponse =
cloudWatchLogsClient.describeLogStreams(describeRequest);

    if (describeResponse.logStreams().isEmpty()) {
        System.out.println("No matching log streams found for prefix: " +
logStreamPrefix);
        return;
    }

    String exactLogStreamName =
describeResponse.logStreams().get(0).logStreamName();
    System.out.println("Using exact log stream: " + exactLogStreamName);

    long startTime = Instant.now().minus(7,
ChronoUnit.DAYS).toEpochMilli();
    long endTime = Instant.now().toEpochMilli();

    GetLogEventsRequest getLogEventsRequest =
GetLogEventsRequest.builder()
        .logGroupName(logGroupName)
        .logStreamName(exactLogStreamName) // <-- exact name, not
prefix
        .startTime(startTime)
        .endTime(endTime)
        .startFromHead(true)
        .build();

    GetLogEventsResponse response =
cloudWatchLogsClient.getLogEvents(getLogEventsRequest);

    if (response.events().isEmpty()) {
        System.out.println("No log events found in the past 7 days.");
    } else {
        response.events().forEach(e -> System.out.println(e.message()));
    }

} catch (CloudWatchException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
```



```
}  
}
```

- Pour plus de détails sur l'API, reportez-vous [GetLogEvents](#) à 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit 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 :

- [Exécution d'une requête volumineuse](#)

### .NET

SDK pour .NET (v4)

#### 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](#).

```
/// <summary>  
/// Gets the results of a CloudWatch Logs Insights query.  
/// </summary>  
/// <param name="queryId">The ID of the query.</param>  
/// <returns>The query results response.</returns>  
public async Task<GetQueryResultsResponse?> GetQueryResultsAsync(string  
queryId)  
{
```

```
try
{
    var request = new GetQueryResultsRequest
    {
        QueryId = queryId
    };

    var response = await
        _amazonCloudWatchLogs.GetQueryResultsAsync(request);
    return response;
}
catch (ResourceNotFoundException ex)
{
    _logger.LogError($"Query not found: {ex.Message}");
    return null;
}
catch (Exception ex)
{
    _logger.LogError($"An error occurred while getting query results:
{ex.Message}");
    return null;
}
}
```

- Pour plus de détails sur l'API, reportez-vous [GetQueryResults](#) à la section Référence des AWS SDK pour .NET 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](#).

```
/**
 * 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 [GetQueryResults](#) à la section Référence des AWS SDK pour JavaScript API.

## Python

### Kit SDK for 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 _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 [GetQueryResults](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

## SAP ABAP

### Kit SDK pour SAP ABAP

#### 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](#).

```
TRY.  
    oo_result = lo_cwl->getqueryresults(  
        iv_queryid = iv_query_id ).  
  
    " Display query status and result count  
    DATA(lv_status) = oo_result->get_status( ).  
    DATA(lt_results) = oo_result->get_results( ).  
    DATA(lv_result_count) = lines( lt_results ).  
  
    MESSAGE |Query status: { lv_status }. Retrieved { lv_result_count } log  
event(s).| TYPE 'I'.  
    CATCH /aws1/cx_cwlinvalidparameterex.  
        MESSAGE 'Invalid parameter.' TYPE 'E'.  
    CATCH /aws1/cx_cwlresourcenotfoundex.  
        MESSAGE 'Resource not found.' TYPE 'E'.  
    CATCH /aws1/cx_cwlserviceunavailex.  
        MESSAGE 'Service unavailable.' TYPE 'E'.  
ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous [GetQueryResults](#) à la section de référence du AWS SDK pour l'API SAP ABAP.

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 du kit 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 à ce sujet 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 [PutSubscriptionFilter](#) à la section Référence des AWS SDK pour C++ API.

## Java

### SDK pour Java 2.x

#### 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 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 [PutSubscriptionFilter](#) à 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 [PutSubscriptionFilter](#) à 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 plus d'informations, consultez le [Guide du développeur AWS SDK pour JavaScript](#).
- Pour plus de détails sur l'API, reportez-vous [PutSubscriptionFilter](#) à 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 du kit 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 une 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 [StartLiveTail](#) à la section Référence des AWS SDK pour .NET API.

## Go

### Kit SDK pour 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 occurred during streaming: %v", err)
    } else if event == nil {
        log.Println("Stream is Closed")
        return
    } else {
        log.Fatalf("Unknown event type: %T", e)
    }
}
}
}
}

```

Démarrez une 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 après 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 [StartLiveTail](#) à 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 une 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 après 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 [StartLiveTail](#) à 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 une 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 après 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 [StartLiveTail](#) à 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 une 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 unsubscribe 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 [StartLiveTail](#) la section AWS SDK pour la référence de l'API Kotlin.

## Python

### Kit SDK for Python (Boto3)

Joignez les fichiers requis.

```
import boto3
import time
from datetime import datetime
```

Démarrez une 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](#) le 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit 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 :

- [Exécution d'une requête volumineuse](#)

### .NET

#### SDK pour .NET (v4)

##### 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](#).

```
/// <summary>
/// Starts a CloudWatch Logs Insights query.
/// </summary>
/// <param name="logGroupName">The name of the log group to query.</param>
/// <param name="queryString">The CloudWatch Logs Insights query string.</
param>
/// <param name="startTime">The start time for the query (seconds since
epoch).</param>
/// <param name="endTime">The end time for the query (seconds since epoch).</
param>
/// <param name="limit">The maximum number of results to return.</param>
/// <returns>The query ID if successful, null otherwise.</returns>
public async Task<string?> StartQueryAsync(
    string logGroupName,
    string queryString,
    long startTime,
    long endTime,
    int limit = 10000)
{
    try
    {
        var request = new StartQueryRequest
        {
            LogGroupName = logGroupName,
            QueryString = queryString,
            StartTime = startTime,
            EndTime = endTime,
            Limit = limit
        };

        var response = await _amazonCloudWatchLogs.StartQueryAsync(request);
        return response.QueryId;
    }
    catch (InvalidParameterException ex)
    {
        _logger.LogError($"Invalid parameter for query: {ex.Message}");
        return null;
    }
    catch (ResourceNotFoundException ex)
    {
        _logger.LogError($"Log group not found: {ex.Message}");
        return null;
    }
}
```

```

        catch (Exception ex)
        {
            _logger.LogError($"An error occurred while starting query:
{ex.Message}");
            return null;
        }
    }
}

```

- Pour plus de détails sur l'API, reportez-vous [StartQuery](#) à la section Référence des AWS SDK pour .NET 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](#).

```

/**
 * 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 [StartQuery](#) à la section Référence des AWS SDK pour JavaScript API.

## Python

### Kit SDK for 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 [StartQuery](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

## SAP ABAP

### Kit SDK pour SAP ABAP

#### 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](#).

```

TRY.
    " iv_log_group_name = '/aws/lambda/my-function'
    " iv_query_string = 'fields @timestamp, @message | sort @timestamp desc |
limit 20'
    " iv_start_time and iv_end_time must be in Unix epoch milliseconds (ms
since Jan 1, 1970 00:00:00 UTC)
    oo_result = lo_cwl->startquery(

```

```

        iv_loggroupname = iv_log_group_name
        iv_starttime     = iv_start_time
        iv_endtime       = iv_end_time
        iv_querystring   = iv_query_string
        iv_limit         = iv_limit ).

" Display the query ID for tracking
DATA(lv_query_id) = oo_result->get_queryid( ).
MESSAGE |Query started successfully with ID: { lv_query_id }| TYPE 'I'.
CATCH /aws1/cx_cwlinvalidparameterex.
    MESSAGE 'Invalid parameter.' TYPE 'E'.
CATCH /aws1/cx_cwllimitexceededex.
    MESSAGE 'Limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cwlmalformedqueryex.
    MESSAGE 'Malformed query.' TYPE 'E'.
CATCH /aws1/cx_cwlresourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cwlserviceunavailex.
    MESSAGE 'Service unavailable.' TYPE 'E'.
ENDTRY.

```

- Pour plus de détails sur l'API, reportez-vous [StartQuery](#) à la section de référence du AWS SDK pour l'API SAP ABAP.

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 du kit 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

- [Configuration d'Amazon ECS Service Connect](#)
- [Création de votre première fonction Lambda](#)
- [Utiliser CloudWatch les journaux pour exécuter une requête volumineuse](#)
- [Utilisent des événements planifiés pour invoquer une fonction Lambda](#)

## Configuration d'Amazon ECS Service Connect

L'exemple de code suivant illustre comment :

- Création de l'infrastructure VPC
- Configuration de la journalisation
- Création du cluster ECS
- Configuration des rôles IAM
- Créez le service avec Service Connect
- Vérification du déploiement
- nettoyer des ressources ;

### Bash

AWS CLI avec le script Bash

#### 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](#).

```
#!/bin/bash

# ECS Service Connect Tutorial Script v4 - Modified to use Default VPC
# This script creates an ECS cluster with Service Connect and deploys an nginx
# service
# Uses the default VPC to avoid VPC limits

set -e # Exit on any error

# Configuration
```

```
SCRIPT_NAME="ECS Service Connect Tutorial"
LOG_FILE="ecs-service-connect-tutorial-v4-default-vpc.log"
REGION=${AWS_DEFAULT_REGION:-${AWS_REGION:-$(aws configure get region 2>/dev/
null)}}}
if [ -z "$REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Set one with: aws configure set region us-east-1"
    exit 1
fi
ENV_PREFIX="tutorial"
CLUSTER_NAME="${ENV_PREFIX}-cluster"
NAMESPACE_NAME="service-connect"

# Generate random suffix for unique resource names
RANDOM_SUFFIX=$(openssl rand -hex 6)

# Arrays to track created resources for cleanup
declare -a CREATED_RESOURCES=()

# Logging function
log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    log "ERROR: Script failed at line $1"
    log "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Set up error handling
trap 'handle_error $LINENO' ERR

# Function to add resource to tracking array
track_resource() {
    CREATED_RESOURCES+=("$1")
    log "Tracking resource: $1"
}

# Function to check if command output contains actual errors
check_for_errors() {
    local output="$1"
```

```

    local command_name="$2"

    # Check for specific AWS CLI error patterns, not just any occurrence of
    "error"
    if echo "$output" | grep -qi "An error occurred\|InvalidParameterException\|
AccessDenied\|ValidationException\|ResourceNotFoundException"; then
        log "ERROR in $command_name: $output"
        return 1
    fi
    return 0
}

# Function to get AWS account ID
get_account_id() {
    ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
    log "Using AWS Account ID: $ACCOUNT_ID"
}

# Function to wait for resources to be ready
wait_for_resource() {
    local resource_type="$1"
    local resource_id="$2"

    case "$resource_type" in
        "cluster")
            log "Waiting for cluster $resource_id to be active..."
            local attempt=1
            local max_attempts=30
            while [ $attempt -le $max_attempts ]; do
                local status=$(aws ecs describe-clusters --clusters
"$resource_id" --query 'clusters[0].status' --output text)
                if [ "$status" = "ACTIVE" ]; then
                    log "Cluster is now active"
                    return 0
                fi
                log "Cluster status: $status (attempt $attempt/$max_attempts)"
                sleep 10
                ((attempt++))
            done
            log "ERROR: Cluster did not become active within expected time"
            return 1
            ;;
        "service")
            log "Waiting for service $resource_id to be stable..."

```

```

        aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
"$resource_id"
        ;;
        "nat-gateway")
        log "Waiting for NAT Gateway $resource_id to be available..."
        aws ec2 wait nat-gateway-available --nat-gateway-ids "$resource_id"
        ;;
    esac
}

# Function to use default VPC infrastructure
setup_default_vpc_infrastructure() {
    log "Using default VPC infrastructure..."

    # Get default VPC
    VPC_ID=$(aws ec2 describe-vpcs --filters "Name=isDefault,Values=true" --query
'Vpcs[0].VpcId' --output text)
    if [[ "$VPC_ID" == "None" || -z "$VPC_ID" ]]; then
        log "ERROR: No default VPC found. Please create a default VPC first."
        exit 1
    fi
    log "Using default VPC: $VPC_ID"

    # Get default subnets
    SUBNETS=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID"
"Name=default-for-az,Values=true" --query 'Subnets[].SubnetId' --output text)
    SUBNET_ARRAY=($SUBNETS)

    if [ ${#SUBNET_ARRAY[@]} -lt 2 ]; then
        log "ERROR: Need at least 2 subnets for ECS Service Connect. Found:
${#SUBNET_ARRAY[@]}"
        exit 1
    fi

    PUBLIC_SUBNET1=${SUBNET_ARRAY[0]}
    PUBLIC_SUBNET2=${SUBNET_ARRAY[1]}

    log "Using subnets: $PUBLIC_SUBNET1, $PUBLIC_SUBNET2"

    # Create security group for ECS tasks
    SG_OUTPUT=$(aws ec2 create-security-group \
        --group-name "${ENV_PREFIX}-ecs-sg-${RANDOM_SUFFIX}" \
        --description "Security group for ECS Service Connect tutorial" \
        --vpc-id "$VPC_ID" 2>&1)

```



```

    check_for_errors "$SG_OUTPUT" "create-security-group"
    SECURITY_GROUP_ID=$(echo "$SG_OUTPUT" | grep -o '"GroupId": "[^"]*"' | cut -d'"' -f4)
    track_resource "SG:$SECURITY_GROUP_ID"
    log "Created security group: $SECURITY_GROUP_ID"

    # Add inbound rules to security group
    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 80 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 443 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    log "Default VPC infrastructure setup completed"
}

# Function to create CloudWatch log groups
create_log_groups() {
    log "Creating CloudWatch log groups..."

    # Create log group for nginx container
    aws logs create-log-group --log-group-name "/ecs/service-connect-nginx" 2>&1
    | grep -v "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-nginx created"
            track_resource "LOG_GROUP:/ecs/service-connect-nginx"
        else
            log "Log group /ecs/service-connect-nginx already exists"
        fi
    }

    # Create log group for service connect proxy
    aws logs create-log-group --log-group-name "/ecs/service-connect-proxy" 2>&1
    | grep -v "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-proxy created"
            track_resource "LOG_GROUP:/ecs/service-connect-proxy"
        else

```

```

        log "Log group /ecs/service-connect-proxy already exists"
    fi
}

# Function to create ECS cluster with Service Connect
create_ecs_cluster() {
    log "Creating ECS cluster with Service Connect..."

    CLUSTER_OUTPUT=$(aws ecs create-cluster \
        --cluster-name "$CLUSTER_NAME" \
        --service-connect-defaults namespace="$NAMESPACE_NAME" \
        --tags key=Environment,value=tutorial 2>&1)
    check_for_errors "$CLUSTER_OUTPUT" "create-cluster"

    track_resource "CLUSTER:$CLUSTER_NAME"
    log "Created ECS cluster: $CLUSTER_NAME"

    wait_for_resource "cluster" "$CLUSTER_NAME"

    # Track the Service Connect namespace that gets created
    # Wait a moment for the namespace to be created
    sleep 5
    NAMESPACE_ID=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_ID" && "$NAMESPACE_ID" != "None" ]]; then
        track_resource "NAMESPACE:$NAMESPACE_ID"
        log "Service Connect namespace created: $NAMESPACE_ID"
    fi
}

# Function to create IAM roles
create_iam_roles() {
    log "Creating IAM roles..."

    # Check if ecsTaskExecutionRole exists
    if aws iam get-role --role-name ecsTaskExecutionRole >/dev/null 2>&1; then
        log "IAM role ecsTaskExecutionRole exists"
    else
        log "Creating ecsTaskExecutionRole..."
        aws iam create-role \

```

```

        --role-name ecsTaskExecutionRole \
        --assume-role-policy-document '{
            "Version": "2012-10-17",
            "Statement": [{
                "Effect": "Allow",
                "Principal": {"Service": "ecs-tasks.amazonaws.com"},
                "Action": "sts:AssumeRole"
            }]
        }' >/dev/null 2>&1
    aws iam attach-role-policy \
        --role-name ecsTaskExecutionRole \
        --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy >/dev/null 2>&1
    track_resource "ROLE:ecsTaskExecutionRole"
    log "Created ecsTaskExecutionRole"
    sleep 10
fi

# Check if ecsTaskRole exists, create if not
if aws iam get-role --role-name ecsTaskRole >/dev/null 2>&1; then
    log "IAM role ecsTaskRole exists"
else
    log "IAM role ecsTaskRole does not exist, will create it"

    # Create trust policy for ECS tasks
    cat > /tmp/ecs-task-trust-policy.json << EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

    aws iam create-role \
        --role-name ecsTaskRole \
        --assume-role-policy-document file:///tmp/ecs-task-trust-policy.json
>/dev/null

```

```

        track_resource "IAM_ROLE:ecsTaskRole"
        log "Created ecsTaskRole"

        # Wait for role to be available
        sleep 10
    fi
}

# Function to create task definition
create_task_definition() {
    log "Creating task definition..."

    # Create task definition JSON
    cat > /tmp/task-definition.json << EOF
{
    "family": "service-connect-nginx",
    "networkMode": "awsvpc",
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "256",
    "memory": "512",
    "executionRoleArn": "arn:aws:iam:${ACCOUNT_ID}:role/ecsTaskExecutionRole",
    "taskRoleArn": "arn:aws:iam:${ACCOUNT_ID}:role/ecsTaskRole",
    "containerDefinitions": [
        {
            "name": "nginx",
            "image": "public.ecr.aws/docker/library/nginx:latest",
            "portMappings": [
                {
                    "containerPort": 80,
                    "protocol": "tcp",
                    "name": "nginx-port"
                }
            ],
            "essential": true,
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/service-connect-nginx",
                    "awslogs-region": "${REGION}",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ]
}
    EOF
}

```

```

    ]
}
EOF

TASK_DEF_OUTPUT=$(aws ecs register-task-definition --cli-input-json file:///
tmp/task-definition.json 2>&1)
check_for_errors "$TASK_DEF_OUTPUT" "register-task-definition"

TASK_DEF_ARN=$(echo "$TASK_DEF_OUTPUT" | grep -o '"taskDefinitionArn":
"[^"]*"' | cut -d'"' -f4)
track_resource "TASK_DEF:service-connect-nginx"
log "Created task definition: $TASK_DEF_ARN"

# Clean up temporary file
rm -f /tmp/task-definition.json
}

# Function to create ECS service with Service Connect
create_ecs_service() {
    log "Creating ECS service with Service Connect..."

    # Create service definition JSON
    cat > /tmp/service-definition.json << EOF
{
    "serviceName": "service-connect-nginx-service",
    "cluster": "${CLUSTER_NAME}",
    "taskDefinition": "service-connect-nginx",
    "desiredCount": 1,
    "launchType": "FARGATE",
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": ["${PUBLIC_SUBNET1}", "${PUBLIC_SUBNET2}"],
            "securityGroups": ["${SECURITY_GROUP_ID}"],
            "assignPublicIp": "ENABLED"
        }
    },
    "serviceConnectConfiguration": {
        "enabled": true,
        "namespace": "${NAMESPACE_NAME}",
        "services": [
            {
                "portName": "nginx-port",
                "discoveryName": "nginx",
                "clientAliases": [

```

```

        {
            "port": 80,
            "dnsName": "nginx"
        }
    ]
}
],
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "/ecs/service-connect-proxy",
        "awslogs-region": "${REGION}",
        "awslogs-stream-prefix": "ecs-service-connect"
    }
}
},
"tags": [
    {
        "key": "Environment",
        "value": "tutorial"
    }
]
}
EOF

```

```

SERVICE_OUTPUT=$(aws ecs create-service --cli-input-json file:///tmp/service-
definition.json 2>&1)
check_for_errors "$SERVICE_OUTPUT" "create-service"

track_resource "SERVICE:service-connect-nginx-service"
log "Created ECS service: service-connect-nginx-service"

wait_for_resource "service" "service-connect-nginx-service"

# Clean up temporary file
rm -f /tmp/service-definition.json
}

# Function to verify deployment
verify_deployment() {
    log "Verifying deployment..."

    # Check service status
    SERVICE_STATUS=$(aws ecs describe-services \

```

```

    --cluster "$CLUSTER_NAME" \
    --services "service-connect-nginx-service" \
    --query 'services[0].status' --output text)
log "Service status: $SERVICE_STATUS"

# Check running tasks
RUNNING_COUNT=$(aws ecs describe-services \
    --cluster "$CLUSTER_NAME" \
    --services "service-connect-nginx-service" \
    --query 'services[0].runningCount' --output text)
log "Running tasks: $RUNNING_COUNT"

# Get task ARN
TASK_ARN=$(aws ecs list-tasks \
    --cluster "$CLUSTER_NAME" \
    --service-name "service-connect-nginx-service" \
    --query 'taskArns[0]' --output text)

if [[ "$TASK_ARN" != "None" && -n "$TASK_ARN" ]]; then
    log "Task ARN: $TASK_ARN"

    # Try to get task IP address
    TASK_IP=$(aws ecs describe-tasks \
        --cluster "$CLUSTER_NAME" \
        --tasks "$TASK_ARN" \
        --query 'tasks[0].attachments[0].details[?
name==`privateIPv4Address`.value' \
        --output text 2>/dev/null || echo "")

    if [[ -n "$TASK_IP" && "$TASK_IP" != "None" ]]; then
        log "Task IP address: $TASK_IP"
    else
        log "Could not retrieve task IP address"
    fi
fi

# Check Service Connect namespace
NAMESPACE_STATUS=$(aws servicediscovery list-namespaces \
    --filters Name=TYPE,Values=HTTP \
    --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

if [[ -n "$NAMESPACE_STATUS" && "$NAMESPACE_STATUS" != "None" ]]; then
    log "Service Connect namespace '$NAMESPACE_NAME' is active"

```

```

else
    log "Service Connect namespace '$NAMESPACE_NAME' not found or not active"
fi

# Display Service Connect configuration
log "Service Connect configuration:"
aws ecs describe-services \
    --cluster "$CLUSTER_NAME" \
    --services "service-connect-nginx-service" \
    --query 'services[0].serviceConnectConfiguration' 2>/dev/null || true
}

# Function to display created resources
display_resources() {
    echo ""
    echo "======"
    echo "CREATED RESOURCES"
    echo "======"
    for resource in "${CREATED_RESOURCES[@]}; do
        echo "- $resource"
    done
    echo "======"
    echo ""
}

# Function to cleanup resources
cleanup_resources() {
    log "Starting cleanup process..."

    # Delete resources in reverse order of creation
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource="${CREATED_RESOURCES[i]}"
        resource_type=$(echo "$resource" | cut -d':' -f1)
        resource_id=$(echo "$resource" | cut -d':' -f2)

        log "Cleaning up $resource_type: $resource_id"

        case "$resource_type" in
            "SERVICE")
                aws ecs update-service --cluster "$CLUSTER_NAME" --service
"$resource_id" --desired-count 0 2>&1 | grep -qi "error" && log "Warning: Failed
to scale down service $resource_id"
                aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
"$resource_id" 2>/dev/null || true

```



```

        aws ecs delete-service --cluster "$CLUSTER_NAME" --service
"$resource_id" --force 2>&1 | grep -qi "error" && log "Warning: Failed to delete
service $resource_id"
        ;;
    "TASK_DEF")
        TASK_DEF_ARNS=$(aws ecs list-task-definitions --family-prefix
"$resource_id" --query 'taskDefinitionArns' --output text 2>/dev/null)
        for arn in $TASK_DEF_ARNS; do
            aws ecs deregister-task-definition --task-definition "$arn"
>/dev/null 2>&1 || true
        done
        ;;
    "ROLE")
        aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
        aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
"error" && log "Warning: Failed to delete role $resource_id"
        ;;
    "IAM_ROLE")
        aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
        aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
"error" && log "Warning: Failed to delete role $resource_id"
        ;;
    "CLUSTER")
        aws ecs delete-cluster --cluster "$resource_id" 2>&1 | grep -qi
"error" && log "Warning: Failed to delete cluster $resource_id"
        ;;
    "SG")
        for attempt in 1 2 3 4 5; do
            if aws ec2 delete-security-group --group-id "$resource_id"
2>/dev/null; then
                break
            fi
            log "Security group $resource_id still has dependencies,
retrying in 30s ($attempt/5)..."
            sleep 30
        done
        ;;
    "LOG_GROUP")
        aws logs delete-log-group --log-group-name "$resource_id" 2>&1 |
grep -qi "error" && log "Warning: Failed to delete log group $resource_id"

```

```

        ;;
        "NAMESPACE")
        # First, delete any services in the namespace
        NAMESPACE_SERVICES=$(aws servicediscovery list-services \
            --filters Name=NAMESPACE_ID,Values="$resource_id" \
            --query 'Services[].Id' --output text 2>/dev/null || echo "")

        if [[ -n "$NAMESPACE_SERVICES" && "$NAMESPACE_SERVICES" !=
"None" ]]; then
            for service_id in $NAMESPACE_SERVICES; do
                aws servicediscovery delete-service --id "$service_id" >/
dev/null 2>&1 || true
                sleep 2
            done
        fi

        # Then delete the namespace
        aws servicediscovery delete-namespace --id "$resource_id" >/dev/
null 2>&1 || true
        ;;
    esac

    sleep 2 # Brief pause between deletions
done

# Clean up temporary files
rm -f /tmp/ecs-task-trust-policy.json
rm -f /tmp/task-definition.json
rm -f /tmp/service-definition.json

log "Cleanup completed"
}

# Main execution
main() {
    log "Starting $SCRIPT_NAME v4 (Default VPC)"
    log "Region: $REGION"
    log "Log file: $LOG_FILE"

    # Get AWS account ID
    get_account_id

    # Setup infrastructure using default VPC
    setup_default_vpc_infrastructure

```

```
# Create CloudWatch log groups
create_log_groups

# Create ECS cluster
create_ecs_cluster

# Create IAM roles
create_iam_roles

# Create task definition
create_task_definition

# Create ECS service
create_ecs_service

# Verify deployment
verify_deployment

log "Tutorial completed successfully!"

# Display created resources
display_resources

# Ask user if they want to clean up
echo ""
echo "=====
echo "CLEANUP CONFIRMATION"
echo "=====
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    cleanup_resources
    log "All resources have been cleaned up"
else
    log "Resources left intact. You can clean them up later by running the
cleanup function."
    echo ""
    echo "To clean up resources later, you can use the AWS CLI commands or
the AWS Management Console."
    echo "Remember to delete resources in the correct order to avoid
dependency issues."
fi
```

```
}  
  
# Make script executable and run  
chmod +x "$0"  
main "$@"
```

- Pour plus de détails sur l'API, consultez les rubriques suivantes dans la Référence des commandes de l'AWS CLI .

- [AttachRolePolicy](#)
- [AuthorizeSecurityGroupIngress](#)
- [CreateCluster](#)
- [CreateLogGroup](#)
- [CreateRole](#)
- [CreateSecurityGroup](#)
- [CreateService](#)
- [DeleteCluster](#)
- [DeleteLogGroup](#)
- [DeleteNamespace](#)
- [DeleteRole](#)
- [DeleteSecurityGroup](#)
- [DeleteService](#)
- [DeregisterTaskDefinition](#)
- [DescribeClusters](#)
- [DescribeServices](#)
- [DescribeSubnets](#)
- [DescribeTasks](#)
- [DescribeVpcs](#)
- [DetachRolePolicy](#)
- [GetCallerIdentity](#)
- [GetRole](#)
- [ListNamespaces](#)
- [ListServices](#)

- [ListTaskDefinitions](#)
- [ListTasks](#)
- [RegisterTaskDefinition](#)
- [UpdateService](#)
- [Attente](#)

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 du kit SDK.

## Création de votre première fonction Lambda

L'exemple de code suivant illustre comment :

- Création d'un rôle IAM pour Lambda
- Créer un code de fonction
- Création d'une fonction Lambda
- Testez votre fonction Lambda
- nettoyer des ressources ;

### Bash

AWS CLI avec le script Bash

#### 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](#).

```
#!/bin/bash
# AWS Lambda - Create Your First Function
# This script creates a Lambda function, invokes it with a test event,
# views CloudWatch logs, and cleans up all resources.
#
# Source: https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html
#
```

```

# Resources created:
#   - IAM role (Lambda execution role with basic logging permissions)
#   - Lambda function (Python 3.13 or Node.js 22.x runtime)
#   - CloudWatch log group (created automatically by Lambda on invocation)

set -eE

#####
# Setup
#####

UNIQUE_ID=$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 8 | head -n 1)
FUNCTION_NAME="my-lambda-function-${UNIQUE_ID}"
ROLE_NAME="lambda-execution-role-${UNIQUE_ID}"
LOG_GROUP_NAME="/aws/lambda/${FUNCTION_NAME}"

TEMP_DIR=$(mktemp -d)
LOG_FILE="${TEMP_DIR}/lambda-gettingstarted.log"

exec > >(tee -a "$LOG_FILE") 2>&1

declare -a CREATED_RESOURCES

#####
# Helper functions
#####

cleanup_resources() {
    # Disable error trap to prevent recursion during cleanup
    trap - ERR
    set +eE

    echo ""
    echo "Cleaning up resources..."
    echo ""

    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        local RESOURCE="${CREATED_RESOURCES[$i]}"
        local TYPE="${RESOURCE%%:*}"
        local NAME="${RESOURCE#*:}"

        case "$TYPE" in
            log-group)
                echo "Deleting CloudWatch log group: ${NAME}"
            ;;
        esac
    done
}

```

```

        aws logs delete-log-group \
            --log-group-name "$NAME" 2>&1 || echo "  WARNING: Could not
delete log group ${NAME}."
        ;;
    lambda-function)
        echo "Deleting Lambda function: ${NAME}"
        aws lambda delete-function \
            --function-name "$NAME" 2>&1 || echo "  WARNING: Could not
delete Lambda function ${NAME}."
        echo "  Waiting for function deletion to complete..."
        local DELETE_WAIT=0
        while aws lambda get-function --function-name "$NAME" > /dev/null
2>&1; do
            sleep 2
            DELETE_WAIT=$((DELETE_WAIT + 2))
            if [ "$DELETE_WAIT" -ge 60 ]; then
                echo "  WARNING: Timed out waiting for function
deletion."
                break
            fi
        done
        ;;
    iam-role-policy)
        local ROLE_PART="${NAME%*|*}"
        local POLICY_PART="${NAME#*|}"
        echo "Detaching policy from role: ${ROLE_PART}"
        aws iam detach-role-policy \
            --role-name "$ROLE_PART" \
            --policy-arn "$POLICY_PART" 2>&1 || echo "  WARNING: Could
not detach policy from role ${ROLE_PART}."
        ;;
    iam-role)
        echo "Deleting IAM role: ${NAME}"
        aws iam delete-role \
            --role-name "$NAME" 2>&1 || echo "  WARNING: Could not delete
IAM role ${NAME}."
        ;;
    esac
done

if [ -d "$TEMP_DIR" ]; then
    rm -rf "$TEMP_DIR"
fi

```

```

    echo ""
    echo "Cleanup complete."
}

handle_error() {
    echo ""
    echo "=====
    echo "ERROR: Script failed at $1"
    echo "=====
    echo ""
    if [ ${#CREATED_RESOURCES[@]} -gt 0 ]; then
        echo "Attempting to clean up ${#CREATED_RESOURCES[@]} resource(s)..."
        cleanup_resources
    fi
    exit 1
}

trap 'handle_error "line $LINENO"' ERR

wait_for_resource() {
    local DESCRIPTION="$1"
    local COMMAND="$2"
    local TARGET_VALUE="$3"
    local TIMEOUT=300
    local ELAPSED=0
    local INTERVAL=5

    echo "Waiting for ${DESCRIPTION}..."
    while true; do
        local RESULT
        RESULT=$(eval "$COMMAND" 2>&1) || true
        if echo "$RESULT" | grep -q "$TARGET_VALUE"; then
            echo "  ${DESCRIPTION} is ready."
            return 0
        fi
        if [ "$ELAPSED" -ge "$TIMEOUT" ]; then
            echo "ERROR: Timed out waiting for ${DESCRIPTION} after ${TIMEOUT}
seconds."
            return 1
        fi
        sleep "$INTERVAL"
        ELAPSED=$((ELAPSED + INTERVAL))
    done
}

```



```
#####
# Region pre-check
#####

CONFIGURED_REGION=$(aws configure get region 2>/dev/null || true)
if [ -z "$CONFIGURED_REGION" ] && [ -z "$AWS_DEFAULT_REGION" ] && [ -z
"$AWS_REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Run 'aws configure set region <region>' or export AWS_DEFAULT_REGION."
    exit 1
fi

#####
# Runtime selection
#####

echo ""
echo "=====
echo "AWS Lambda - Create Your First Function"
echo "=====
echo ""
echo "Select a runtime for your Lambda function:"
echo "  1) Python 3.13"
echo "  2) Node.js 22.x"
echo ""
echo "Enter your choice (1 or 2): "
read -r RUNTIME_CHOICE

case "$RUNTIME_CHOICE" in
    1)
        RUNTIME="python3.13"
        HANDLER="lambda_function.lambda_handler"
        CODE_FILE="lambda_function.py"
        cat > "${TEMP_DIR}/${CODE_FILE}" << 'PYTHON_EOF'
import json
import logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
def lambda_handler(event, context):
    length = event['length']
    width = event['width']
    area = calculate_area(length, width)
    print(f'The area is {area}')
```

```

    logger.info(f'CloudWatch logs group: {context.log_group_name}')
    return json.dumps({'area': area})
def calculate_area(length, width):
    return length * width
PYTHON_EOF
    echo "Selected runtime: Python 3.13"
    ;;
2)
    RUNTIME="nodejs22.x"
    HANDLER="index.handler"
    CODE_FILE="index.mjs"
    cat > "${TEMP_DIR}/${CODE_FILE}" << 'NODEJS_EOF'
export const handler = async (event, context) => {
    const area = event.length * event.width;
    console.log(`The area is ${area}`);
    console.log('CloudWatch log group: ', context.logGroupName);
    return JSON.stringify({area});
};
NODEJS_EOF
    echo "Selected runtime: Node.js 22.x"
    ;;
*)
    echo "ERROR: Invalid choice. Please enter 1 or 2."
    exit 1
    ;;
esac

#####
# Step 1: Create IAM execution role
#####

echo ""
echo "=====
echo "Step 1: Create IAM execution role"
echo "=====
echo ""

TRUST_POLICY='{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            }
        }
    ]
}'

```

```

        },
        "Action": "sts:AssumeRole"
    }
]
}'

echo "Creating IAM role: ${ROLE_NAME}"
ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document "$TRUST_POLICY" \
    --query 'Role.Arn' \
    --output text 2>&1)
echo "$ROLE_OUTPUT"
ROLE_ARN="$ROLE_OUTPUT"
CREATED_RESOURCES+=("iam-role:${ROLE_NAME}")
echo "Role ARN: ${ROLE_ARN}"

echo ""
echo "Attaching AWSLambdaBasicExecutionRole policy..."
aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>&1
CREATED_RESOURCES+=("iam-role-policy:${ROLE_NAME}|arn:aws:iam::aws:policy/
service-role/AWSLambdaBasicExecutionRole")
echo "Policy attached."

# IAM roles can take a few seconds to propagate
echo "Waiting for IAM role to propagate..."
sleep 10

#####
# Step 2: Create Lambda function
#####

echo ""
echo "=====
echo "Step 2: Create Lambda function"
echo "=====
echo ""

echo "Creating deployment package..."
ORIGINAL_DIR=$(pwd)
cd "$TEMP_DIR"

```

```

zip -j function.zip "$CODE_FILE" > /dev/null 2>&1
cd "$ORIGINAL_DIR"

echo "Creating Lambda function: ${FUNCTION_NAME}"
echo "  Runtime: ${RUNTIME}"
echo "  Handler: ${HANDLER}"
echo ""

CREATE_OUTPUT=$(aws lambda create-function \
    --function-name "$FUNCTION_NAME" \
    --runtime "$RUNTIME" \
    --role "$ROLE_ARN" \
    --handler "$HANDLER" \
    --architectures x86_64 \
    --zip-file "fileb://${TEMP_DIR}/function.zip" \
    --query '[FunctionName, FunctionArn, Runtime, State]' \
    --output text 2>&1)
echo "$CREATE_OUTPUT"
CREATED_RESOURCES+=("lambda-function:${FUNCTION_NAME}")

wait_for_resource "Lambda function to become Active" \
    "aws lambda get-function-configuration --function-name ${FUNCTION_NAME} --
query State --output text" \
    "Active"

#####
# Step 3: Invoke the function
#####

echo ""
echo "=====
echo "Step 3: Invoke the function"
echo "=====
echo ""

TEST_EVENT='{ "length": 6, "width": 7}'
echo "Invoking function with test event: ${TEST_EVENT}"
echo ""

echo "$TEST_EVENT" > "${TEMP_DIR}/test-event.json"

INVOKE_OUTPUT=$(aws lambda invoke \
    --function-name "$FUNCTION_NAME" \
    --payload "fileb://${TEMP_DIR}/test-event.json" \

```

```

    --cli-read-timeout 30 \
    "${TEMP_DIR}/response.json" 2>&1)
echo "$INVOKE_OUTPUT"

RESPONSE=$(cat "${TEMP_DIR}/response.json")
echo ""
echo "Function response: ${RESPONSE}"
echo ""

if echo "$INVOKE_OUTPUT" | grep -qi "functionerror"; then
    echo "WARNING: Function returned an error."
fi

#####
# Step 4: View CloudWatch logs
#####

echo ""
echo "=====
echo "Step 4: View CloudWatch Logs"
echo "=====
echo ""

echo "Log group: ${LOG_GROUP_NAME}"
echo ""

echo "Waiting for CloudWatch logs to be available..."

LOG_STREAMS=""
for i in $(seq 1 6); do
    LOG_STREAMS=$(aws logs describe-log-streams \
        --log-group-name "$LOG_GROUP_NAME" \
        --order-by LastEventTime \
        --descending \
        --query 'logStreams[0].logStreamName' \
        --output text 2>/dev/null) || true
    if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then
        break
    fi
    LOG_STREAMS=""
    sleep 5
done

if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then

```

```

    echo "Latest log stream: ${LOG_STREAMS}"
    echo ""
    echo "--- Log events ---"
    LOG_EVENTS=$(aws logs get-log-events \
        --log-group-name "$LOG_GROUP_NAME" \
        --log-stream-name "$LOG_STREAMS" \
        --query 'events[].message' \
        --output text 2>&1) || true
    echo "$LOG_EVENTS"
    echo "--- End of log events ---"
else
    echo "No log streams found yet. Logs may take a moment to appear."
    echo "You can view them in the CloudWatch console:"
    echo "  Log group: ${LOG_GROUP_NAME}"
fi

CREATED_RESOURCES+=("log-group:${LOG_GROUP_NAME}")

#####
# Summary and cleanup
#####

echo ""
echo "=====
echo "SUMMARY"
echo "=====
echo ""
echo "Resources created:"
echo "  IAM role:      ${ROLE_NAME}"
echo "  Lambda function:  ${FUNCTION_NAME}"
echo "  CloudWatch logs:  ${LOG_GROUP_NAME}"
echo ""
echo "=====
echo "CLEANUP CONFIRMATION"
echo "=====
echo ""
echo "Do you want to clean up all created resources? (y/n): "
read -r CLEANUP_CHOICE

if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
    cleanup_resources
else
    echo ""
    echo "Resources were NOT deleted. To clean up manually, run:"

```

```
echo ""
echo " # Delete the Lambda function"
echo " aws lambda delete-function --function-name ${FUNCTION_NAME}"
echo ""
echo " # Delete the CloudWatch log group"
echo " aws logs delete-log-group --log-group-name ${LOG_GROUP_NAME}"
echo ""
echo " # Detach the policy and delete the IAM role"
echo " aws iam detach-role-policy --role-name ${ROLE_NAME} --policy-arn
arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
echo " aws iam delete-role --role-name ${ROLE_NAME}"
echo ""

if [ -d "$TEMP_DIR" ]; then
    rm -rf "$TEMP_DIR"
fi
fi

echo ""
echo "Done."
```

- Pour plus de détails sur l'API, consultez les rubriques suivantes dans la Référence des commandes de l'AWS CLI .
  - [AttachRolePolicy](#)
  - [CreateFunction](#)
  - [CreateRole](#)
  - [DeleteFunction](#)
  - [DeleteLogGroup](#)
  - [DeleteRole](#)
  - [DescribeLogStreams](#)
  - [DetachRolePolicy](#)
  - [GetFunction](#)
  - [GetFunctionConfiguration](#)
  - [GetLogEvents](#)
  - [Invoke](#)

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 du kit SDK.

## 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.

.NET

SDK pour .NET (v4)

### 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](#).

Il s'agit du flux de travail principal qui illustre le scénario de requêtes volumineuses.

```
using System.Diagnostics;
using System.Text.RegularExpressions;
using Amazon.CloudFormation;
using Amazon.CloudFormation.Model;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
using CloudWatchLogsActions;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;

namespace CloudWatchLogsScenario;

public class LargeQueryWorkflow
{
    /*
        Before running this .NET code example, set up your development environment,
        including your credentials.
        This .NET code example performs the following tasks for the CloudWatch Logs
        Large Query workflow:
```



1. Prepare the Application:
  - Prompt the user to deploy CloudFormation stack and generate sample logs.
  - Deploy the CloudFormation template for resource creation.
  - Generate 50,000 sample log entries using CloudWatch Logs API.
  - Wait 5 minutes for logs to be fully ingested.
2. Execute Large Query:
  - Perform recursive queries to retrieve all logs using binary search.
  - Display progress for each query executed.
  - Show total execution time and logs found.
3. Clean up:
  - Prompt the user to delete the CloudFormation stack and all resources.
  - Destroy the CloudFormation stack and wait until removed.

\*/

```
public static ILogger<LargeQueryWorkflow> _logger = null!;
public static CloudWatchLogsWrapper _wrapper = null!;
public static IAmazonCloudFormation _amazonCloudFormation = null!;

private static string _logGroupName = "/workflows/cloudwatch-logs/large-
query";
private static string _logStreamName = "stream1";
private static long _queryStartDate;
private static long _queryEndDate;

public static bool _interactive = true;
public static string _stackName = "CloudWatchLargeQueryStack";
private static string _stackResourcePath = "../..../..../..../scenarios/
features/cloudwatch_logs_large_query/resources/stack.yaml";

public static async Task Main(string[] args)
{
    using var host = Host.CreateDefaultBuilder(args)
        .ConfigureLogging(logging =>
            logging.AddFilter("System", LogLevel.Debug)
                .AddFilter("Microsoft", LogLevel.Information))
        .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonCloudWatchLogs>()
                .AddAWSService<IAmazonCloudFormation>()
                .AddTransient<CloudWatchLogsWrapper>()
            )
        .Build();
```

```

    if (_interactive)
    {
        _logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<LargeQueryWorkflow>();

        _wrapper = host.Services.GetRequiredService<CloudWatchLogsWrapper>();
        _amazonCloudFormation =
host.Services.GetRequiredService<IAmazonCloudFormation>();
    }

    Console.WriteLine(new string('-', 80));
    Console.WriteLine("Welcome to the CloudWatch Logs Large Query
Scenario.");
    Console.WriteLine(new string('-', 80));
    Console.WriteLine("This scenario demonstrates how to perform large-scale
queries on");
    Console.WriteLine("CloudWatch Logs using recursive binary search to
retrieve more than");
    Console.WriteLine("the 10,000 result limit.");
    Console.WriteLine();

    try
    {
        Console.WriteLine(new string('-', 80));
        var prepareSuccess = await PrepareApplication();
        Console.WriteLine(new string('-', 80));

        if (prepareSuccess)
        {
            Console.WriteLine(new string('-', 80));
            await ExecuteLargeQuery();
            Console.WriteLine(new string('-', 80));
        }

        Console.WriteLine(new string('-', 80));
        await Cleanup();
        Console.WriteLine(new string('-', 80));
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "There was a problem with the scenario,
initiating cleanup...");
        _interactive = false;
        await Cleanup();
    }

```

```
    }

    Console.WriteLine("CloudWatch Logs Large Query scenario completed.");
}

/// <summary>
/// Runs the scenario workflow. Used for testing.
/// </summary>
public static async Task RunScenario()
{
    Console.WriteLine(new string('-', 80));
    Console.WriteLine("Welcome to the CloudWatch Logs Large Query
Scenario.");
    Console.WriteLine(new string('-', 80));
    Console.WriteLine("This scenario demonstrates how to perform large-scale
queries on");
    Console.WriteLine("CloudWatch Logs using recursive binary search to
retrieve more than");
    Console.WriteLine("the 10,000 result limit.");
    Console.WriteLine();

    try
    {
        Console.WriteLine(new string('-', 80));
        var prepareSuccess = await PrepareApplication();
        Console.WriteLine(new string('-', 80));

        if (prepareSuccess)
        {
            Console.WriteLine(new string('-', 80));
            await ExecuteLargeQuery();
            Console.WriteLine(new string('-', 80));
        }

        Console.WriteLine(new string('-', 80));
        await Cleanup();
        Console.WriteLine(new string('-', 80));
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "There was a problem with the scenario,
initiating cleanup...");
        _interactive = false;
        await Cleanup();
    }
}
```

```

    }

    Console.WriteLine("CloudWatch Logs Large Query scenario completed.");
}

/// <summary>
/// Prepares the application by creating the necessary resources.
/// </summary>
/// <returns>True if the application was prepared successfully.</returns>
public static async Task<bool> PrepareApplication()
{
    Console.WriteLine("Preparing the application...");
    Console.WriteLine();

    try
    {
        var deployStack = !_interactive || GetYesNoResponse(
            "Would you like to deploy the CloudFormation stack and generate
sample logs? (y/n) ");

        if (deployStack)
        {
            if (!_interactive)
            {
                Console.Write(
                    $"Enter a path for the CloudFormation stack
resource .yaml file (or press Enter for default '{_stackResourcePath}'): ");
                string? inputPath = Console.ReadLine();
                if (!string.IsNullOrEmpty(inputPath))
                {
                    _stackResourcePath = inputPath;
                }
            }

            _stackName = PromptUserForStackName();

            var deploySuccess = await DeployCloudFormationStack(_stackName);

            if (deploySuccess)
            {
                Console.WriteLine();
                Console.WriteLine("Generating 50,000 sample log entries...");
                var generateSuccess = await GenerateSampleLogs();
            }
        }
    }
}

```

```

        if (generateSuccess)
        {
            Console.WriteLine();
            Console.WriteLine("Sample logs created. Waiting 5 minutes
for logs to be fully ingested...");
            await WaitWithCountdown(300);

            Console.WriteLine("Application preparation complete.");
            return true;
        }
    }
}
else
{
    _logGroupName = PromptUserForInput("Enter the log group name ",
_logGroupName);
    _logStreamName = PromptUserForInput("Enter the log stream name ",
_logStreamName);

    var startDateMs = PromptUserForLong("Enter the query start date
(milliseconds since epoch): ");
    var endDateMs = PromptUserForLong("Enter the query end date
(milliseconds since epoch): ");

    _queryStartDate = startDateMs / 1000;
    _queryEndDate = endDateMs / 1000;

    Console.WriteLine("Application preparation complete.");
    return true;
}
}
catch (Exception ex)
{
    _logger.LogError(ex, "An error occurred while preparing the
application.");
}

Console.WriteLine("Application preparation failed.");
return false;
}

/// <summary>
/// Deploys the CloudFormation stack with the necessary resources.
/// </summary>

```

```
/// <param name="stackName">The name of the CloudFormation stack.</param>
/// <returns>True if the stack was deployed successfully.</returns>
private static async Task<bool> DeployCloudFormationStack(string stackName)
{
    Console.WriteLine($"\\nDeploying CloudFormation stack: {stackName}");

    try
    {
        var request = new CreateStackRequest
        {
            StackName = stackName,
            TemplateBody = await File.ReadAllTextAsync(_stackResourcePath)
        };

        var response = await _amazonCloudFormation.CreateStackAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"CloudFormation stack creation started:
{stackName}");

            bool stackCreated = await
WaitForStackCompletion(response.StackId);

            if (stackCreated)
            {
                Console.WriteLine("CloudFormation stack created
successfully.");
                return true;
            }
            else
            {
                _logger.LogError($"CloudFormation stack creation failed:
{stackName}");
                return false;
            }
        }
        else
        {
            _logger.LogError($"Failed to create CloudFormation stack:
{stackName}");
            return false;
        }
    }
}
```

```
        catch (AlreadyExistsException)
        {
            _logger.LogWarning($"CloudFormation stack '{stackName}' already
exists. Please provide a unique name.");
            var newStackName = PromptUserForStackName();
            return await DeployCloudFormationStack(newStackName);
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while deploying the
CloudFormation stack: {stackName}");
            return false;
        }
    }
}

/// <summary>
/// Waits for the CloudFormation stack to be in the CREATE_COMPLETE state.
/// </summary>
/// <param name="stackId">The ID of the CloudFormation stack.</param>
/// <returns>True if the stack was created successfully.</returns>
private static async Task<bool> WaitForStackCompletion(string stackId)
{
    int retryCount = 0;
    const int maxRetries = 30;
    const int retryDelay = 10000;

    while (retryCount < maxRetries)
    {
        var describeStacksRequest = new DescribeStacksRequest
        {
            StackName = stackId
        };

        var describeStacksResponse = await
_amazonCloudFormation.DescribeStacksAsync(describeStacksRequest);

        if (describeStacksResponse.Stacks.Count > 0)
        {
            if (describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.CREATE_COMPLETE)
            {
                return true;
            }
        }
    }
}
```

```

        if (describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.CREATE_FAILED ||
            describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.ROLLBACK_COMPLETE)
        {
            return false;
        }
    }

    Console.WriteLine("Waiting for CloudFormation stack creation to
complete...");
    await Task.Delay(retryDelay);
    retryCount++;
}

_logger.LogError("Timed out waiting for CloudFormation stack creation to
complete.");
return false;
}

/// <summary>
/// Generates sample logs directly using CloudWatch Logs API.
/// Creates 50,000 log entries spanning 5 minutes.
/// </summary>
/// <returns>True if logs were generated successfully.</returns>
private static async Task<bool> GenerateSampleLogs()
{
    const int totalEntries = 50000;
    const int entriesPerBatch = 10000;
    const int fiveMinutesMs = 5 * 60 * 1000;

    try
    {
        // Calculate timestamps
        var startTimeMs = DateTimeOffset.UtcNow.ToUnixTimeMilliseconds();
        var timestampIncrement = fiveMinutesMs / totalEntries;

        Console.WriteLine($"Generating {totalEntries} log entries...");

        var entryCount = 0;
        var currentTimestamp = startTimeMs;
        var numBatches = totalEntries / entriesPerBatch;

        // Generate and upload logs in batches

```



```

        for (int batchNum = 0; batchNum < numBatches; batchNum++)
        {
            var logEvents = new List<InputLogEvent>();

            for (int i = 0; i < entriesPerBatch; i++)
            {
                logEvents.Add(new InputLogEvent
                {
                    Timestamp =
DateTimeOffset.FromUnixTimeMilliseconds(currentTimestamp).UtcDateTime,
                    Message = $"Entry {entryCount}"
                });

                entryCount++;
                currentTimestamp += timestampIncrement;
            }

            // Upload batch
            var success = await _wrapper.PutLogEventsAsync(_logGroupName,
                _logStreamName, logEvents);
            if (!success)
            {
                _logger.LogError($"Failed to upload batch {batchNum + 1}/
{numBatches}");
                return false;
            }

            Console.WriteLine($"Uploaded batch {batchNum + 1}/{numBatches}");
        }

        // Set query date range (convert milliseconds to seconds for query
API)
        _queryStartDate = startTimeMs / 1000;
        _queryEndDate = (currentTimestamp - timestampIncrement) / 1000;

        Console.WriteLine($"Query start date:
{DateTimeOffset.FromUnixTimeSeconds(_queryStartDate):yyyy-MM-
ddTHH:mm:ss.fffZ}");
        Console.WriteLine($"Query end date:
{DateTimeOffset.FromUnixTimeSeconds(_queryEndDate):yyyy-MM-ddTHH:mm:ss.fffZ}");
        Console.WriteLine($"Successfully uploaded {totalEntries} log
entries");

        return true;

```

```
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "An error occurred while generating sample
logs.");
        return false;
    }
}

/// <summary>
/// Executes the large query workflow.
/// </summary>
public static async Task ExecuteLargeQuery()
{
    Console.WriteLine("Starting recursive query to retrieve all logs...");
    Console.WriteLine();

    var queryLimit = PromptUserForInteger("Enter the query limit (max 10000)
", 10000);
    if (queryLimit > 10000) queryLimit = 10000;

    var queryString = "fields @timestamp, @message | sort @timestamp asc";

    var stopwatch = Stopwatch.StartNew();
    var allResults = await PerformLargeQuery(_logGroupName, queryString,
_queryStartDate, _queryEndDate, queryLimit);
    stopwatch.Stop();

    Console.WriteLine();
    Console.WriteLine($"Queries finished in
{stopwatch.Elapsed.TotalSeconds:F3} seconds.");
    Console.WriteLine($"Total logs found: {allResults.Count}");

    // Check for duplicates
    Console.WriteLine();
    Console.WriteLine("Checking for duplicate logs...");
    var duplicates = FindDuplicateLogs(allResults);
    if (duplicates.Count > 0)
    {
        Console.WriteLine($"WARNING: Found {duplicates.Count} duplicate log
entries!");
        Console.WriteLine("Duplicate entries (showing first 10):");
        foreach (var dup in duplicates.Take(10))
        {
```

```

        Console.WriteLine($" [{dup.Timestamp}] {dup.Message} (appears
{dup.Count} times)");
    }

    var uniqueCount = allResults.Count - duplicates.Sum(d => d.Count -
1);
    Console.WriteLine($"Unique logs: {uniqueCount}");
}
else
{
    Console.WriteLine("No duplicates found. All logs are unique.");
}
Console.WriteLine();

var viewSample = !_interactive || GetYesNoResponse("Would you like to see
a sample of the logs? (y/n) ");
if (viewSample)
{
    Console.WriteLine();
    Console.WriteLine($"Sample logs (first 10 of {allResults.Count}):");
    for (int i = 0; i < Math.Min(10, allResults.Count); i++)
    {
        var timestamp = allResults[i].Find(f => f.Field ==
"@timestamp")?.Value ?? "N/A";
        var message = allResults[i].Find(f => f.Field ==
"@message")?.Value ?? "N/A";
        Console.WriteLine($"[{timestamp}] {message}");
    }
}
}

/// <summary>
/// Performs a large query using recursive binary search.
/// </summary>
private static async Task<List<List<ResultField>>> PerformLargeQuery(
    string logGroupName,
    string queryString,
    long startTime,
    long endTime,
    int limit)
{
    var queryId = await _wrapper.StartQueryAsync(logGroupName, queryString,
startTime, endTime, limit);
    if (queryId == null)

```

```
{
    return new List<List<ResultField>>();
}

var results = await PollQueryResults(queryId);
if (results == null || results.Count == 0)
{
    return new List<List<ResultField>>();
}

var startDate =
    DateTimeOffset.FromUnixTimeSeconds(startTime).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
var endDate = DateTimeOffset.FromUnixTimeSeconds(endTime).ToString("yyyy-
MM-ddTHH:mm:ss.fffZ");
Console.WriteLine($"Query date range: {startDate} ({startTime}s) to
{endDate} ({endTime}s). Found {results.Count} logs.");

if (results.Count < limit)
{
    Console.WriteLine($" -> Returning {results.Count} logs (less than
limit of {limit})");
    return results;
}

Console.WriteLine($" -> Hit limit of {limit}. Need to split and
recurse.");

// Get the timestamp of the last log (sorted to find the actual last one)
var lastLogTimestamp = GetLastLogTimestamp(results);
if (lastLogTimestamp == null)
{
    Console.WriteLine($" -> No timestamp found in results. Returning
{results.Count} logs.");
    return results;
}

Console.WriteLine($" -> Last log timestamp: {lastLogTimestamp}");

// Parse the timestamp and add 1 millisecond to avoid querying the same
log again
var lastLogDate = DateTimeOffset.Parse(lastLogTimestamp + " +0000");
Console.WriteLine($" -> Last log as DateTimeOffset: {lastLogDate:yyyy-
MM-ddTHH:mm:ss.fffZ} ({lastLogDate.ToUnixTimeSeconds()}s)");
```

```
        var offsetLastLogDate = lastLogDate.AddMilliseconds(1);
        Console.WriteLine($" -> Offset timestamp (last
+ 1ms): {offsetLastLogDate:yyyy-MM-ddTHH:mm:ss.fffZ}
({offsetLastLogDate.ToUnixTimeSeconds()}s)");

        // Convert to seconds, but round UP to the next second to avoid
        overlapping with logs in the same second
        // This ensures we don't re-query logs that share the same second as the
        last log
        var offsetLastLogTime = offsetLastLogDate.ToUnixTimeSeconds();
        if (offsetLastLogDate.Millisecond > 0)
        {
            offsetLastLogTime++; // Move to the next full second
            Console.WriteLine($" -> Adjusted to next full second:
{offsetLastLogTime}s
({DateTimeOffset.FromUnixTimeSeconds(offsetLastLogTime):yyyy-MM-
ddTHH:mm:ss.fffZ})");
        }

        Console.WriteLine($" -> Comparing:
offsetLastLogTime={offsetLastLogTime}s vs endTime={endTime}s");
        Console.WriteLine($" -> End time as date:
{DateTimeOffset.FromUnixTimeSeconds(endTime):yyyy-MM-ddTHH:mm:ss.fffZ}");

        // Check if there's any time range left to query
        if (offsetLastLogTime >= endTime)
        {
            Console.WriteLine($" -> No time range left to query. Offset time
({offsetLastLogTime}s) >= end time ({endTime}s)");
            return results;
        }

        // Split the remaining date range in half
        var (range1Start, range1End, range2Start, range2End) =
SplitDateRange(offsetLastLogTime, endTime);

        var range1StartDate =
DateTimeOffset.FromUnixTimeSeconds(range1Start).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var range1EndDate =
DateTimeOffset.FromUnixTimeSeconds(range1End).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
```

```

        var range2StartDate =
            DateTimeOffset.FromUnixTimeSeconds(range2Start).ToString("yyyy-MM-
            ddTHH:mm:ss.fffZ");
        var range2EndDate =
            DateTimeOffset.FromUnixTimeSeconds(range2End).ToString("yyyy-MM-
            ddTHH:mm:ss.fffZ");

        Console.WriteLine($" -> Splitting remaining range:");
        Console.WriteLine($"      Range 1: {range1StartDate} ({range1Start}s) to
        {range1EndDate} ({range1End}s)");
        Console.WriteLine($"      Range 2: {range2StartDate} ({range2Start}s) to
        {range2EndDate} ({range2End}s)");

        // Query both halves recursively
        Console.WriteLine($" -> Querying range 1...");
        var results1 = await PerformLargeQuery(logGroupName, queryString,
        range1Start, range1End, limit);
        Console.WriteLine($" -> Range 1 returned {results1.Count} logs");

        Console.WriteLine($" -> Querying range 2...");
        var results2 = await PerformLargeQuery(logGroupName, queryString,
        range2Start, range2End, limit);
        Console.WriteLine($" -> Range 2 returned {results2.Count} logs");

        // Combine all results
        var allResults = new List<List<ResultField>>(results);
        allResults.AddRange(results1);
        allResults.AddRange(results2);

        Console.WriteLine($" -> Combined total: {allResults.Count} logs
        ({results.Count} + {results1.Count} + {results2.Count})");

        return allResults;
    }

    /// <summary>
    /// Gets the timestamp string of the most recent log from a list of logs.
    /// Sorts timestamps to find the actual last one.
    /// </summary>
    private static string? GetLastLogTimestamp(List<List<ResultField>> logs)
    {
        var timestamps = logs
            .Select(log => log.Find(f => f.Field == "@timestamp")?.Value)
            .Where(t => !string.IsNullOrEmpty(t))

```

```

        .OrderBy(t => t)
        .ToList();

    if (timestamps.Count == 0)
    {
        return null;
    }

    return timestamps[timestamps.Count - 1];
}

/// <summary>
/// Splits a date range in half.
/// Range 2 starts at midpoint + 1 second to avoid overlap.
/// </summary>
private static (long range1Start, long range1End, long range2Start, long
range2End) SplitDateRange(long startTime, long endTime)
{
    var midpoint = startTime + (endTime - startTime) / 2;
    // Range 2 starts at midpoint + 1 to avoid querying the same second twice
    return (startTime, midpoint, midpoint + 1, endTime);
}

/// <summary>
/// Polls for query results until complete.
/// </summary>
private static async Task<List<List<ResultField>>>> PollQueryResults(string
queryId)
{
    int retryCount = 0;
    const int maxRetries = 60;
    const int retryDelay = 1000;

    while (retryCount < maxRetries)
    {
        var response = await _wrapper.GetQueryResultsAsync(queryId);
        if (response == null)
        {
            return null;
        }

        if (response.Status == QueryStatus.Complete)
        {
            return response.Results;
        }
    }
}

```

```

    }

    if (response.Status == QueryStatus.Failed ||
        response.Status == QueryStatus.Cancelled ||
        response.Status == QueryStatus.Timeout ||
        response.Status == QueryStatus.Unknown)
    {
        _logger.LogError($"Query failed with status: {response.Status}");
        return null;
    }

    await Task.Delay(retryDelay);
    retryCount++;
}

_logger.LogError("Timed out waiting for query results.");
return null;
}

/// <summary>
/// Cleans up the resources created during the scenario.
/// </summary>
public static async Task<bool> Cleanup()
{
    var cleanup = !_interactive || GetYesNoResponse(
        "Do you want to delete the CloudFormation stack and all resources?
(y/n) ");

    if (cleanup)
    {
        try
        {
            var stackDeleteSuccess = await
DeleteCloudFormationStack(_stackName, false);
            return stackDeleteSuccess;
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "An error occurred while cleaning up the
resources.");
            return false;
        }
    }
}

```



```
        Console.WriteLine($"Resources will remain. Stack name: {_stackName}, Log
group: {_logGroupName}");
        _logger.LogInformation("CloudWatch Logs Large Query scenario is
complete.");
        return true;
    }

    /// <summary>
    /// Deletes the CloudFormation stack and waits for confirmation.
    /// </summary>
    private static async Task<bool> DeleteCloudFormationStack(string stackName,
bool forceDelete)
    {
        var request = new DeleteStackRequest
        {
            StackName = stackName,
        };

        if (forceDelete)
        {
            request.DeletionMode = DeletionMode.FORCE_DELETE_STACK;
        }

        await _amazonCloudFormation.DeleteStackAsync(request);
        Console.WriteLine($"CloudFormation stack '{stackName}' is being deleted.
This may take a few minutes.");

        bool stackDeleted = await WaitForStackDeletion(stackName, forceDelete);

        if (stackDeleted)
        {
            Console.WriteLine($"CloudFormation stack '{stackName}' has been
deleted.");
            return true;
        }
        else
        {
            _logger.LogError($"Failed to delete CloudFormation stack
'{stackName}'.");
            return false;
        }
    }

    /// <summary>
```

```
/// Waits for the stack to be deleted.
/// </summary>
private static async Task<bool> WaitForStackDeletion(string stackName, bool
forceDelete)
{
    int retryCount = 0;
    const int maxRetries = 30;
    const int retryDelay = 10000;

    while (retryCount < maxRetries)
    {
        var describeStacksRequest = new DescribeStacksRequest
        {
            StackName = stackName
        };

        try
        {
            var describeStacksResponse = await
_amazonCloudFormation.DescribeStacksAsync(describeStacksRequest);

            if (describeStacksResponse.Stacks.Count == 0 ||
                describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.DELETE_COMPLETE)
            {
                return true;
            }

            if (!forceDelete && describeStacksResponse.Stacks[0].StackStatus
== StackStatus.DELETE_FAILED)
            {
                return await DeleteCloudFormationStack(stackName, true);
            }
        }
        catch (AmazonCloudFormationException ex) when (ex.ErrorCode ==
"ValidationError")
        {
            return true;
        }

        Console.WriteLine($"Waiting for CloudFormation stack '{stackName}' to
be deleted...");
        await Task.Delay(retryDelay);
        retryCount++;
    }
}
```

```

    }

    _logger.LogError($"Timed out waiting for CloudFormation stack
'{stackName}' to be deleted.");
    return false;
}

/// <summary>
/// Waits with a countdown display.
/// </summary>
private static async Task WaitWithCountdown(int seconds)
{
    for (int i = seconds; i > 0; i--)
    {
        Console.Write($"\\rWaiting: {i} seconds remaining... ");
        await Task.Delay(1000);
    }
    Console.WriteLine("\\rWait complete.                ");
}

/// <summary>
/// Helper method to get a yes or no response from the user.
/// </summary>
private static bool GetYesNoResponse(string question)
{
    Console.WriteLine(question);
    var ynResponse = Console.ReadLine();
    var response = ynResponse != null && ynResponse.Equals("y",
StringComparison.InvariantCultureIgnoreCase);
    return response;
}

/// <summary>
/// Prompts the user for a stack name.
/// </summary>
private static string PromptUserForStackName()
{
    if (_interactive)
    {
        Console.Write($"Enter a name for the CloudFormation stack (press
Enter for default '{_stackName}'): ");
        string? input = Console.ReadLine();
        if (!string.IsNullOrEmpty(input))
        {

```

```

        var regex = "[a-zA-Z][-a-zA-Z0-9]*";
        if (!Regex.IsMatch(input, regex))
        {
            Console.WriteLine($"Invalid stack name. Using default:
{_stackName}");
            return _stackName;
        }
        return input;
    }
}
return _stackName;
}

/// <summary>
/// Prompts the user for input with a default value.
/// </summary>
private static string PromptUserForInput(string prompt, string defaultValue)
{
    if (_interactive)
    {
        Console.Write($"{prompt}(press Enter for default '{defaultValue}'):
");
        string? input = Console.ReadLine();
        return string.IsNullOrEmpty(input) ? defaultValue : input;
    }
    return defaultValue;
}

/// <summary>
/// Prompts the user for an integer value.
/// </summary>
private static int PromptUserForInteger(string prompt, int defaultValue)
{
    if (_interactive)
    {
        Console.Write($"{prompt}(press Enter for default '{defaultValue}'):
");
        string? input = Console.ReadLine();
        if (string.IsNullOrEmpty(input) || !int.TryParse(input, out var
result))
        {
            return defaultValue;
        }
        return result;
    }
}

```

```
    }
    return defaultValue;
}

/// <summary>
/// Prompts the user for a long value.
/// </summary>
private static long PromptUserForLong(string prompt)
{
    if (_interactive)
    {
        Console.Write(prompt);
        string? input = Console.ReadLine();
        if (long.TryParse(input, out var result))
        {
            return result;
        }
    }
    return 0;
}

/// <summary>
/// Finds duplicate log entries based on timestamp and message.
/// </summary>
private static List<(string Timestamp, string Message, int Count)>
FindDuplicateLogs(List<List<ResultField>> logs)
{
    var logSignatures = new Dictionary<string, int>();

    foreach (var log in logs)
    {
        var timestamp = log.Find(f => f.Field == "@timestamp")?.Value ?? "";
        var message = log.Find(f => f.Field == "@message")?.Value ?? "";
        var signature = $"{timestamp}|{message}";

        if (logSignatures.ContainsKey(signature))
        {
            logSignatures[signature]++;
        }
        else
        {
            logSignatures[signature] = 1;
        }
    }
}
```

```
        return logSignatures
            .Where(kvp => kvp.Value > 1)
            .Select(kvp =>
            {
                var parts = kvp.Key.Split('|');
                return (Timestamp: parts[0], Message: parts[1], Count:
kvp.Value);
            })
            .OrderByDescending(x => x.Count)
            .ToList();
    }
}
```

- Pour plus de détails sur l'API, consultez les rubriques suivantes dans la Référence des API du kit AWS SDK pour .NET .
  - [GetQueryResults](#)
  - [StartQuery](#)

## 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](#).

Il s'agit du 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 de détails sur l'API, consultez les rubriques suivantes dans la Référence des API du kit AWS SDK pour JavaScript .
  - [GetQueryResults](#)
  - [StartQuery](#)

## Python

### Kit SDK for 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 supérieures à 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 dont le nombre dépasse 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 de détails sur l'API, consultez les rubriques suivantes dans la Référence des API du kit AWS SDK for Python (Boto3).
  - [GetQueryResults](#)
  - [StartQuery](#)

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 du kit 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

### Kit SDK for 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, consultez [Utilisation CloudWatch des journaux avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.



# Chiffrez les tables de recherche dans les CloudWatch journaux à l'aide de AWS Key Management Service

Les données de la table de recherche sont toujours cryptées dans CloudWatch les journaux. 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 de la table de recherche au repos. En guise d'alternative, vous pouvez utiliser AWS Key Management Service pour ce chiffrement. Dans ce cas, le chiffrement est effectué à l'aide d'une AWS KMS clé. L'utilisation du chiffrement AWS KMS est activée au niveau de la table de recherche, en associant une clé KMS à une table de recherche, soit lorsque vous créez la table de recherche, soit lorsque vous la mettez à jour.

## Important

CloudWatch Logs ne prend en charge que les clés KMS symétriques. N'utilisez pas de clé asymétrique pour chiffrer les données de vos tables de recherche. Pour plus d'informations, consultez [Utilisation des clés symétriques et asymétriques](#).

Une fois que vous avez associé une clé KMS à une table de recherche, toutes les données stockées dans la table de recherche sont chiffrées à l'aide de cette clé. 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'une table de recherche, CloudWatch Logs chiffre les données à l'aide de la méthode de chiffrement par défaut de CloudWatch Logs. Toutefois, si la clé est désactivée ou supprimée avant que vous ne la dissociiez, CloudWatch Logs ne pourra pas lire les données chiffrées avec cette clé.

Pour obtenir des informations générales sur la manière dont CloudWatch AWS KMS Logs chiffre les données des journaux, consultez [Chiffrez les données du journal dans CloudWatch Logs à l'aide de AWS Key Management Service](#).

# Comment CloudWatch Logs l'utilise AWS KMS pour les tables de recherche

CloudWatch Logs utilise le chiffrement des AWS KMS enveloppes pour protéger les données des tables de recherche. Lorsque vous associez une clé KMS à une table de recherche, CloudWatch Logs envoie une `GenerateDataKey` demande à AWS KMS. AWS KMS génère une clé de chiffrement des données (DEK) unique et renvoie à la fois une copie en texte brut et une copie cryptée de la DEK. CloudWatch Logs utilise le DEK en texte brut pour chiffrer les données de la table de recherche, puis stocke le DEK chiffré à côté des données chiffrées. Le DEK en texte brut n'est pas stocké et est supprimé de la mémoire après utilisation.

Lorsque CloudWatch Logs a besoin de lire les données de la table de recherche, il envoie une `Decrypt` demande AWS KMS au DEK crypté. AWS KMS déchiffre le DEK et renvoie le DEK en texte brut à CloudWatch Logs, qui l'utilise ensuite pour déchiffrer les données de la table de recherche.

CloudWatch Logs utilise le contexte de chiffrement suivant lorsqu'il adresse des demandes à AWS KMS :

```
{
  "aws:logs:arn": "arn:aws:logs:region:account-id:lookup-table:lookup-table-name"
}
```

Vous pouvez utiliser ce contexte de chiffrement dans les politiques IAM et les politiques AWS KMS clés pour contrôler l'accès à la clé KMS. Pour de plus amples informations, veuillez consulter [AWS KMS clés et contexte de chiffrement](#).

## Autorisations nécessaires

Pour utiliser AWS KMS le chiffrement avec les tables de recherche, le principal IAM doit disposer des AWS KMS autorisations suivantes sur la clé KMS :

- `kms:Decrypt`
- `kms:GenerateDataKey`

L'`kms:Decrypt` autorisation est requise lorsque vous appelez `GetLookupTable` une table de recherche chiffrée à l'aide d'une clé KMS, afin que CloudWatch Logs puisse déchiffrer les données

en votre nom. L'`kms:Decrypt` autorisation est également requise sur la clé (la clé KMS utilisée pour chiffrer la table de recherche) lors d'un appel `StartQuery` avec une requête qui utilise la `lookup` commande sur une table de recherche cryptée. L'`kms:GenerateDataKey` autorisation est requise lors de l'appel `CreateLookupTable` ou `UpdateLookupTable` avec une clé KMS, afin que CloudWatch Logs puisse générer une clé de chiffrement des données pour chiffrer les données de la table de recherche.

En outre, le service CloudWatch Logs doit être autorisé à utiliser la clé KMS. Vous accordez ces autorisations en ajoutant une déclaration de politique à la politique de clé KMS, comme décrit dans la section suivante.

## Étape 1 : Création d'une AWS KMS clé

Pour créer une clé KMS symétrique, 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"
    ]
  }
}
```

Enregistrez l'ARN de la clé. Vous en aurez besoin dans les étapes suivantes.

## 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. Vous devez accorder au service CloudWatch Logs l'autorisation principale d'utiliser la clé, et également accorder au rôle appelant l'autorisation d'utiliser la clé.

Tout d'abord, enregistrez la politique par défaut pour votre clé KMS à `policy.json` l'aide de la [get-key-policy](#) commande suivante :

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./policy.json
```

Ouvrez le `policy.json` fichier dans un éditeur de texte et ajoutez l'instruction suivante pour accorder au service CloudWatch Logs l'autorisation principale d'utiliser la clé. Cet exemple utilise une `Condition` section qui correspond au contexte de chiffrement pour restreindre la clé à une table de recherche spécifique.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "logs.region.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-id:lookup-table:lookup-table-name",
      "aws:SourceAccount": "account-id",
      "aws:SourceArn": "arn:aws:logs:region:account-id:lookup-table:lookup-table-name"
    }
  }
}
```

Ajoutez ensuite des autorisations au rôle qui appellera les CloudWatch journaux `CreateLookupTable` ou `UpdateLookupTable` APIs. CloudWatch Enregistre `kms:ViaService`

les utilisations pour passer AWS KMS des appels au nom du client. Pour plus d'informations, voir [kms : ViaService](#).

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account-id:role/role-name"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "logs.region.amazonaws.com"
      ]
    }
  }
}
```

Enfin, ajoutez la politique mise à jour à l'aide de la [put-key-policy](#) commande suivante :

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

## Étape 3 : associer une clé KMS à une table de recherche

Vous pouvez associer une clé KMS à une table de recherche lorsque vous la créez à l'aide de l>CreateLookupTableAPI ou que vous mettez à jour une table de recherche existante à l'aide de l'UpdateLookupTableAPI. Les deux APIs font partie du AWSLogsConfigService.

Pour associer la clé KMS à une table de recherche lorsque vous la créez

Utilisez l>CreateLookupTableAPI et spécifiez le kmsKeyArn paramètre avec l'ARN de votre clé KMS :

```
aws logs create-lookup-table \
  --lookup-table-name my-lookup-table \
```

```
--kms-key-arn "arn:aws:kms:region:account-id:key/key-id"
```

Pour associer la clé KMS à une table de recherche existante

Utilisez l'UpdateLookupTableAPI et spécifiez le kmsKeyArn paramètre avec l'ARN de votre clé KMS :

```
aws logs update-lookup-table \  
  --lookup-table-name my-lookup-table \  
  --kms-key-arn "arn:aws:kms:region:account-id:key/key-id"
```

## Considérations

- Après avoir associé ou dissocié une clé KMS d'une table de recherche, 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, les données de votre table de recherche cryptée dans CloudWatch Logs ne peuvent plus être récupérées.
- Pour effectuer les étapes décrites dans cette rubrique, vous devez disposer des autorisations suivantes :kms:CreateKey, kms:GetKeyPolicykms:PutKeyPolicy, et des autorisations CloudWatch Logs appropriées pour appeler CreateLookupTable ouUpdateLookupTable.

# 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 répondre à 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 d'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 Gestion des identités et des accès AWS (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.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. 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 AES-GCM (Advanced Encryption Standard Galois/Counter Mode) 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 [Gestion des identités et des accès AWS \(IAM\)](#) 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 sans autorisation. 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 | <p>Les deux éléments suivants sont utilisés.</p> <p>Le second, avec le : * à la fin, est ce qui est renvoyé par la commande <code>describe-log-groups</code> CLI et l'<code>DescribeLogGroupsAPI</code>.</p> <p><code>arn:aws:logs : ::log-group : <i>region account-id log_group_name</i></code></p> <p><code>arn:aws:logs : ::log-group : : * <i>region account-id log_group_name</i></code></p> <p>Utilisez la première version, sans la suite : *, dans les situations suivantes :</p> <ul style="list-style-type: none"><li>• Dans le champ de <code>logGroupIdentifier</code> saisie dans de nombreux cas CloudWatch Logs APIs.</li><li>• Sur le <code>resourceArn</code> terrain lors du balisage APIs</li><li>• Dans IAM les politiques, lorsque vous spécifiez des autorisations pour <a href="#">TagResource</a>, <a href="#">UntagResource</a>, et <a href="#">ListTagsForResource</a>.</li></ul> <p>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.</p> |
| Flux de journaux   | <p><code>arn:aws:logs : ::log-group ::log-stream : <i>region account-id log_group_name log-stream-name</i></code></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Type de ressource | Format ARN                                                                         |
|-------------------|------------------------------------------------------------------------------------|
| Destination       | arn:aws:logs : ::destination : <i>region</i><br><i>account-id destination_name</i> |

Pour plus d'informations ARNs, consultez le Guide de [ARNs](#) l'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, consultez [Utilisation 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'[PutDestination](#) API, et vous pouvez ajouter une politique de ressources à la destination à l'aide de l'[PutDestinationPolicy](#) API. 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, consultez [CloudWatch 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 CloudWatch console](#)
- [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, consultez [CloudWatch Référence des autorisations de journalisation](#).

## Autorisations requises pour utiliser la CloudWatch console

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 dans [AWS 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 `CloudWatchLogsFullAccess` politique 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 [CloudWatchLogsFullAccess](#) le Guide de référence des politiques AWS gérées.

### `CloudWatchLogsReadOnlyAccess`

La `CloudWatchLogsReadOnlyAccess` politique accorde un accès en lecture seule aux journaux. 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 [CloudWatchLogsReadOnlyAccess](#)le 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 [CloudWatchOpenSearchDashboardsFullAccess](#)le 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 [CloudWatchOpenSearchDashboardAccess](#)le 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 [CloudWatchLogsCrossAccountSharingConfiguration](#)le Guide de référence des politiques AWS gérées.

## CloudWatchLogsAPIKeyAccès

La politique CloudWatchLogsAPIKeyd'accès permet l'authentification par clé de l'API CloudWatch Logs et l'ingestion cryptée des journaux. Cette politique accorde des autorisations pour s'authentifier à l'aide de jetons porteurs et pour écrire les événements du journal dans les CloudWatch journaux, avec des AWS KMS autorisations supplémentaires pour le déchiffrement et la génération de clés de données lorsque les journaux sont chiffrés.

Cette stratégie accorde les autorisations suivantes :

- logs— Permet aux principaux de s'authentifier via des jetons porteurs de clés de l'API et d'écrire les événements du journal dans les flux de CloudWatch journaux.
- kms— Permet aux principaux de lire les métadonnées des AWS KMS clés, de générer des clés de données pour le chiffrement et de déchiffrer les données. Ces autorisations prennent en charge le chiffrement CloudWatch des journaux en permettant au service de chiffrer les données des journaux à l'aide de clés gérées par le client AWS KMS . L'accès est limité aux opérations appelées via le service CloudWatch Logs.

Pour plus de détails sur la politique, y compris la dernière version du document de politique JSON, consultez [CloudWatchLogsAPIKeyAccess](#) dans le 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.

| Modifier                                                           | Description                                                                                                                                          | Date            |
|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">CloudWatchLogsAPIKeyAccès</a><br>— Nouvelle politique. | CloudWatch Les journaux ont ajouté une nouvelle politique gérée CloudWatchLogsAPIKeyAccess.<br><br>Cette politique permet l'authentification par clé | 17 février 2026 |

| Modifier                                                                                   | Description                                                                                                                                                                                                                                                                               | Date            |
|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
|                                                                                            | de l'API CloudWatch Logs et l'ingestion cryptée des journaux, en accordant les autorisations d'authentification à l'aide de jetons porteurs et d'écriture des événements de journal dans Logs. CloudWatch                                                                                 |                 |
| <a href="#">CloudWatchLogsFullAccess</a><br>- Mettre à jour vers une politique existante.  | CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsFullAccess.<br><br>Des autorisations pour les actions d'administration de l'observabilité ont été ajoutées pour permettre l'accès en lecture seule aux pipelines de télémétrie et aux intégrations de tables S3.     | 2 décembre 2025 |
| <a href="#">CloudWatchLogsReadOnlyAccess</a> - Mettre à jour vers une politique existante. | CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsReadOnlyAccess.<br><br>Des autorisations pour les actions d'administration de l'observabilité ont été ajoutées pour permettre l'accès en lecture seule aux pipelines de télémétrie et aux intégrations de tables S3. | 2 décembre 2025 |



| Modifier                                                                                   | Description                                                                                                                                                                                                                                                                      | Date        |
|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <a href="#">CloudWatchLogsFullAccess</a><br>- Mettre à jour vers une politique existante.  | CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsFullAccess.<br><br>Des autorisations pour <code>cloudwatch:GenerateQueryResultsSummary</code> ont été ajoutées pour permettre la génération d'un résumé en langage naturel des résultats de la requête.     | 20 mai 2025 |
| <a href="#">CloudWatchLogsReadOnlyAccess</a> - Mettre à jour vers une politique existante. | CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsReadOnlyAccess.<br><br>Des autorisations pour <code>cloudwatch:GenerateQueryResultsSummary</code> ont été ajoutées pour permettre la génération d'un résumé en langage naturel des résultats de la requête. | 20 mai 2025 |

| Modifier                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                              | Date              |
|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">CloudWatchLogsFullAccess</a><br>- Mettre à jour vers une politique existante. | CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsFullAccess.<br><br>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.                                                                                                                                          | 1er décembre 2024 |
| <a href="#">CloudWatchOpenSearchDashboardsFullAccess</a> —<br>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, et pour créer, gérer et supprimer des tableaux de bord de journaux vendus dans ces intégrations. Pour de plus amples informations, veuillez consulter <a href="#">Analyser avec Amazon OpenSearch Service</a> . | 1er décembre 2024 |

| Modifier                                                                                  | Description                                                                                                                                                                                                                                                                                                                  | Date              |
|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">CloudWatchOpenSearchDashboardAccess</a> —<br>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 <a href="#">Analyser avec Amazon OpenSearch Service</a> . | 1er décembre 2024 |
| <a href="#">CloudWatchLogsFullAccess</a><br>- Mettre à jour vers une politique existante. | CloudWatch Logs a ajouté une autorisation à CloudWatchLogsFullAccess.<br><br>L'cloudwatch:GenerateQuery autorisation a été ajoutée afin que les utilisateurs soumis à cette politique puissent générer une chaîne de requête <a href="#">CloudWatch Logs Insights</a> à partir d'une invite en langage naturel.              | 27 novembre 2023  |

| Modifier                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Date             |
|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">CloudWatchLogsReadOnlyAccess</a> - Mettre à jour vers une politique existante. | <p>CloudWatch a ajouté une autorisation à CloudWatchLogsReadOnlyAccess.</p> <p>L'cloudwatch:GenerateQuery autorisation a été ajoutée afin que les utilisateurs soumis à cette politique puissent générer une chaîne de requête <a href="#">CloudWatch Logs Insights</a> à partir d'une invite en langage naturel.</p>                                                                                                                                                  | 27 novembre 2023 |
| <a href="#">CloudWatchLogsReadOnlyAccess</a> : mise à jour d'une politique existante       | <p>CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsReadOnlyAccess.</p> <p>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 <a href="#">Utilisation de Live Tail pour visualiser les journaux en temps quasi réel</a>.</p> | 6 juin 2023      |

| Modifier                                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                         | Date             |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">CloudWatchLogsCrossAccountSharingConfiguration</a> : nouvelle politique  | <p>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.</p> <p>Pour plus d'informations, consultez la <a href="#">CloudWatch section Observabilité entre comptes</a></p>                                                                                                          | 27 novembre 2022 |
| <a href="#">CloudWatchLogsReadOnlyAccess</a> : mise à jour d'une politique existante | <p>CloudWatch Les journaux ont ajouté des autorisations à CloudWatchLogsReadOnlyAccess.</p> <p>Les <code>oam:ListAttachedLinks</code> autorisations <code>oam:ListSinks</code> 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.</p> | 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 `CloudWatchLogsReadOnlyAccess` politique 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 `StopQuery` et `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)                                                                                            |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">CancelExportTask</a>              | <code>logs:CancelExportTask</code><br><br>Exigé pour annuler une tâche d'exportation en attente ou en cours d'exécution.          |
| <a href="#">CreateExportTask</a>              | <code>logs:CreateExportTask</code><br><br>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)                                                                                                          |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">CreateLogGroup</a>                | <code>logs:CreateLogGroup</code><br><br>Exigé pour créer un nouveau groupe de journaux.                                                         |
| <a href="#">CreateLogStream</a>               | <code>logs:CreateLogStream</code><br><br>Exigé pour créer un nouveau flux de journaux dans un groupe de journaux.                               |
| <a href="#">DeleteDestination</a>             | <code>logs:DeleteDestination</code><br><br>Exigé pour supprimer une destination de journal et désactive tous les filtres d'abonnement connexes. |
| <a href="#">DeleteLogGroup</a>                | <code>logs:DeleteLogGroup</code><br><br>Exigé pour supprimer un groupe de journaux et tous les événements du journal archivés associés.         |
| <a href="#">DeleteLogStream</a>               | <code>logs:DeleteLogStream</code><br><br>Exigé pour supprimer un flux de journaux et tous les événements du journal archivés associés.          |
| <a href="#">DeleteMetricFilter</a>            | <code>logs:DeleteMetricFilter</code><br><br>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)                                                                                                        |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">DeleteQueryDefinition</a>         | <code>logs:DeleteQueryDefinition</code><br><br>Nécessaire pour supprimer une définition de requête enregistrée dans CloudWatch Logs Insights. |
| <a href="#">DeleteResourcePolicy</a>          | <code>logs:DeleteResourcePolicy</code><br><br>Nécessaire pour supprimer une politique de ressources CloudWatch Logs.                          |
| <a href="#">DeleteRetentionPolicy</a>         | <code>logs:DeleteRetentionPolicy</code><br><br>Exigé pour supprimer la politique de rétention d'un groupe de journaux.                        |
| <a href="#">DeleteSubscriptionFilter</a>      | <code>logs:DeleteSubscriptionFilter</code><br><br>Exigé pour supprimer le filtre d'abonnement associé à un groupe de journaux.                |
| <a href="#">DescribeDestinations</a>          | <code>logs:DescribeDestinations</code><br><br>Exigé pour afficher toutes les destinations associées au compte.                                |
| <a href="#">DescribeExportTasks</a>           | <code>logs:DescribeExportTasks</code><br><br>Exigé pour afficher toutes les tâches d'exportation associées au compte.                         |

| CloudWatch Journalise les opérations de l'API | Autorisations requises (actions d'API)                                                                                                                                 |
|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">DescribeLogGroups</a>             | <code>logs:DescribeLogGroups</code><br><br>Exigé pour afficher tous les groupes de journaux associés au compte.                                                        |
| <a href="#">DescribeLogStreams</a>            | <code>logs:DescribeLogStreams</code><br><br>Exigé pour afficher tous les flux de journaux associés à un groupe de journaux.                                            |
| <a href="#">DescribeMetricFilters</a>         | <code>logs:DescribeMetricFilters</code><br><br>Exigé pour afficher toutes les métriques associées à un groupe de journaux.                                             |
| <a href="#">DescribeQueryDefinitions</a>      | <code>logs:DescribeQueryDefinitions</code><br><br>Nécessaire pour voir la liste des définitions de requêtes enregistrées dans CloudWatch Logs Insights.                |
| <a href="#">DescribeQueries</a>               | <code>logs:DescribeQueries</code><br><br>Obligatoire pour voir la liste des requêtes CloudWatch Logs Insights planifiées, en cours d'exécution ou récemment exécutées. |
| <a href="#">DescribeResourcePolicies</a>      | <code>logs:DescribeResourcePolicies</code><br><br>Obligatoire pour consulter la liste des politiques relatives CloudWatch aux ressources des journaux.                 |

| CloudWatch Journalise les opérations de l'API | Autorisations requises (actions d'API)                                                                                                                                                     |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">DescribeSubscriptionFilters</a>   | <code>logs:DescribeSubscriptionFilters</code><br><br>Exigé pour afficher tous les filtres d'abonnement associés à un groupe de journaux.                                                   |
| <a href="#">FilterLogEvents</a>               | <code>logs:FilterLogEvents</code><br><br>Exigé pour trier les événements du journal par modèle de filtres de groupes de journaux.                                                          |
| <a href="#">GetLogEvents</a>                  | <code>logs:GetLogEvents</code><br><br>Exigé pour récupérer les événements du journal à partir d'un flux de journaux.                                                                       |
| <a href="#">GetLogGroupFields</a>             | <code>logs:GetLogGroupFields</code><br><br>Obligatoire pour récupérer la liste des champs qui sont inclus dans les événements du journal d'un groupe de journaux.                          |
| <a href="#">GetLogRecord</a>                  | <code>logs:GetLogRecord</code><br><br>Obligatoire pour récupérer des informations à partir d'un seul événement du journal.                                                                 |
| <a href="#">GetLogObject</a>                  | <code>logs:GetLogRecord</code><br><br>Nécessaire pour récupérer le contenu d'une grande partie des événements du journal qui ont été ingérés via l' <code>PutOpenTelemetryLogs</code> API. |

| CloudWatch Journalise les opérations de l'API                                               | Autorisations requises (actions d'API)                                                                                                                                                              |
|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">GetQueryResults</a>                                                             | <code>logs:GetQueryResults</code><br><br>Nécessaire pour récupérer les résultats des requêtes CloudWatch Logs Insights.                                                                             |
| <code>ListEntitiesForLogGroup</code><br><br>(autorisation CloudWatch réservée à la console) | <code>logs:ListEntitiesForLogGroup</code><br><br>Obligatoire pour trouver les entités associées à un groupe de journaux. Nécessaire pour explorer les journaux associés dans la CloudWatch console. |
| <code>ListLogGroupsForEntity</code><br><br>(autorisation CloudWatch réservée à la console)  | <code>logs:ListLogGroupsForEntity</code><br><br>Nécessaire pour trouver les groupes de journaux associés à une entité. Nécessaire pour explorer les journaux associés dans la CloudWatch console.   |
| <a href="#">ListTagsLogGroup</a>                                                            | <code>logs:ListTagsLogGroup</code><br><br>Exigé pour afficher toutes les étiquettes associées à un groupe de journaux.                                                                              |
| <a href="#">ListLogGroups</a>                                                               | <code>logs:DescribeLogGroups</code><br><br>Exigé pour afficher tous les groupes de journaux associés au compte.                                                                                     |
| <a href="#">PutDestination</a>                                                              | <code>logs:PutDestination</code><br><br>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)                                                                                                                                  |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">PutDestinationPolicy</a>          | <code>logs:PutDestinationPolicy</code><br><br>Exigé pour créer ou mettre à jour une politique d'accès associée à une destination de journal existante.                  |
| <a href="#">PutLogEvents</a>                  | <code>logs:PutLogEvents</code><br><br>Exigé pour charger un lot d'événements du journal dans un flux de journaux.                                                       |
| <a href="#">PutMetricFilter</a>               | <code>logs:PutMetricFilter</code><br><br>Exigé pour créer ou mettre à jour un filtre de métrique et l'associer à un groupe de journaux.                                 |
| <a href="#">PutQueryDefinition</a>            | <code>logs:PutQueryDefinition</code><br><br>Nécessaire pour enregistrer une requête dans CloudWatch Logs Insights, y compris les requêtes enregistrées avec paramètres. |
| <a href="#">PutResourcePolicy</a>             | <code>logs:PutResourcePolicy</code><br><br>Nécessaire pour créer une politique de ressources CloudWatch Logs.                                                           |
| <a href="#">PutRetentionPolicy</a>            | <code>logs:PutRetentionPolicy</code><br><br>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)                                                                                                                                                                                           |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">PutSubscriptionFilter</a>         | <code>logs:PutSubscriptionFilter</code><br><br>Exigé pour créer ou mettre à jour un filtre d'abonnement et l'associer à un groupe de journaux.                                                                                   |
| <a href="#">StartQuery</a>                    | <code>logs:StartQuery</code><br><br>Nécessaire pour démarrer CloudWatch les requêtes Logs Insights. Pour exécuter une requête enregistrée avec des paramètres, vous devez également <code>logs:DescribeQueryDefinitions</code> . |
| <a href="#">StopQuery</a>                     | <code>logs:StopQuery</code><br><br>Nécessaire pour arrêter une requête CloudWatch Logs Insights en cours.                                                                                                                        |
| <a href="#">TagLogGroup</a>                   | <code>logs:TagLogGroup</code><br><br>Exigé pour ajouter ou mettre à jour des étiquettes de groupe de journaux.                                                                                                                   |
| <a href="#">TestMetricFilter</a>              | <code>logs:TestMetricFilter</code><br><br>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 Gestion des identités et des accès AWS 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 en savoir plus, 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 `AWSServiceRoleForLogDelivery` un service

- Arrêtez d'envoyer des logs directement aux streams Firehose.

Pour supprimer manuellement le rôle lié au service à l'aide d'IAM

Utilisez la console IAM, le AWS CLI, ou l' AWS API pour supprimer le rôle lié au `AWSServiceRoleForLogDelivery` service. 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 à un service dans toutes les AWS régions où le service est disponible. Pour plus d'informations, consultez la section [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.

| Modifier                                                                                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                        | Date            |
|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| AWSServiceRoleForLogDelivery politique <a href="#">de rôle liée au service</a> — <a href="#">Mise à jour d'une politique</a> existante | <p>CloudWatch Les journaux ont modifié les autorisations de la politique IAM associée au rôle lié au <code>AWSServiceRoleForLogDelivery</code> service. La modification suivante a été apportée :</p> <ul style="list-style-type: none"><li>La clé de condition <code>firehose:ResourceTag/LogDeliveryEnabled</code>: "true" a été modifiée en <code>aws:ResourceTag/LogDeliveryEnabled</code>: "true" .</li></ul> | 15 juillet 2021 |

| Modifier                                                        | 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. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez [AWS la documentation de sécurité](#).

## 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.

## 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.

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 [Démarez](#) 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. Pour vous connecter à un point de terminaison FIPS, le service à choisir est `com.amazonaws.Region.logs-fips`. 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.

Certains CloudWatch journaux APIs, tels que `StartLiveTail` et `GetLogObject`, sont hébergés sous un point de terminaison et un point de terminaison VPC différents : `stream-logs.Region.amazonaws.com` Pour créer un point de terminaison VPC d'interface pour ceux-ci APIs, le service à choisir est `com.amazonaws.Region.stream-logs`. Pour vous connecter à un point de terminaison FIPS, le service à choisir est `com.amazonaws.Region.stream-logs-fips`.

## 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 instance Amazon EC2 qui se trouve dans votre VPC. Pour plus d'informations sur la connexion, consultez [Connexion à votre instance Linux](#) ou [Connexion à votre instance Windows](#) dans la documentation Amazon EC2.
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 panneau 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 CloudTrail](#) un 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 dans 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'`GenerateQuery` action 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"
}
```

# 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      | <p>Nombre d'opérations API spécifiées effectuées dans votre compte.</p> <p>CallCount est une métrique d'utilisation du service CloudWatch Logs. Pour de plus amples informations, veuillez consulter <a href="#">CloudWatch Statistiques d'utilisation du service de journalisation</a>.</p> <p>Dimensions valides : Classe, Ressource, Service, Type</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p>                                                                                                                                                                                                                                                                                                                   |
| DeliveryErrors | <p>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. <code>AccessDeniedException</code> <code>ResourceNotFoundException</code></p> <p>Dimensions valides : LogGroupName DestinationType, FilterName, PolicyLevel</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> |

| Métrique           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DeliveryThrottling | <p>Nombre d'événements de journal pour lesquels CloudWatch Logs a été limité lors du transfert des données vers la destination de l'abonnement.</p> <p>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 <code>AccessDeniedException</code> ou <code>ResourceNotFoundException</code>.</p> <p>Dimensions valides : LogGroupName DestinationType, FilterName, PolicyLevel</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> |
| EMFDisabledErrors  | <p>Nombre d'événements de journal au format EMF qui ont été ignorés en raison d'une politique de compte actif de type METRIC_EXTRACTION_POLICY correspondant au groupe de journaux. Pour plus d'informations sur les politiques d'extraction de métriques, consultez <a href="#">PutAccountPolicy</a>.</p> <p>Dimensions valides : LogGroupName</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p>                                                                                                                                                                                                                                                                                                                           |

| Métrique            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EMFParsingErrors    | <p>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 <a href="#">Spécification : format des métriques intégrées</a>.</p> <p>Dimensions valides : LogGroupName</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| EMFValidationErrors | <p>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 <code>MetricDatum</code>. Pour plus d'informations sur le format de métrique CloudWatch intégré, voir <a href="#">Spécification : format de métrique intégré</a>. Pour plus d'informations sur le type de données <code>MetricDatum</code>, consultez <a href="#">MetricDatum</a> le Amazon CloudWatch API Reference.</p> <div> <p> <b>Note</b></p> <p>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.</p> </div> <p>Dimensions valides : LogGroupName</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> |

| Métrique            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ErrorCount          | <p>Le nombre d'opérations API effectuées dans votre compte qui ont donné lieu à des erreurs.</p> <p>ErrorCount est une métrique d'utilisation du service CloudWatch Logs. Pour de plus amples informations, veuillez consulter <a href="#">CloudWatch Statistiques d'utilisation du service de journalisation</a>.</p> <p>Dimensions valides : Classe, Ressource, Service, Type</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> |
| ForwardedBytes      | <p>Volume des événements du journal en octets compressés transférés vers la destination de l'abonnement.</p> <p>Dimensions valides : LogGroupName, DestinationType, FilterName</p> <p>Statistique valide : somme</p> <p>Unités : octets</p>                                                                                                                                                                                                  |
| Forwarded LogEvents | <p>Nombre d'événements du journal envoyés à destination de l'abonnement.</p> <p>Dimensions valides : LogGroupName DestinationType, FilterName, PolicyLevel</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p>                                                                                                                                                                                                                      |

| Métrique              | Description                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IncomingBytes         | <p>Volume des événements du journal en octets non compressés transférés 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.</p> <p>Dimensions valides : LogGroupName</p> <p>Statistique valide : somme</p> <p>Unités : octets</p>                                            |
| IncomingLogEvents     | <p>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.</p> <p>Dimensions valides : LogGroupName</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p>                                                                                                   |
| LogEventsWithFindings | <p>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 <a href="#">Aider à protéger les données sensibles des journaux grâce au masquage</a>.</p> <p>Dimensions valides : aucune</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> |

| Métrique      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ThrottleCount | <p>Le nombre d'opérations API effectuées dans votre compte qui ont été limitées en raison de quotas d'utilisation.</p> <p>ThrottleCount est une métrique d'utilisation du service CloudWatch Logs. Pour de plus amples informations, veuillez consulter <a href="#">CloudWatch Statistiques d'utilisation du service de journalisation</a>.</p> <p>Dimensions valides : Classe, Ressource, Service, Type</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> |

## 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.                                                                                                                                                                                    |
| 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 | <p>Nombre d'erreurs rencontrées lors de la transformation des événements du journal avec le transformateur spécifié.</p> <p>Unité : aucune</p> <p>Statistique valide : somme</p> |
| TransformedBytes     | <p>Volume de sortie des événements du journal transformé, en octets non compressés.</p> <p>Unité : octets</p> <p>Statistique valide : somme</p>                                  |



| Métrique             | Description                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------------------|
| TransformedLogEvents | Le nombre d'événements du journal transformé.<br><br>Unité : aucune<br><br>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 transformateurs au niveau du compte. Actuellement, la seule valeur valide pour cette dimension est AccountPolicy |

## Mesures et dimensions de centralisation

Pour une surveillance centralisée sur plusieurs comptes et régions, vous pouvez utiliser la centralisation CloudWatch des journaux pour consolider les données des journaux et les indicateurs dans un emplacement central. Pour de plus amples informations, veuillez consulter [Centralisation des journaux inter-comptes et inter-régions](#).

CloudWatch Logs publie les métriques de centralisation suivantes CloudWatch dans l'espace de AWS/Logs noms. Ces mesures vous aident à surveiller la réplication des données des journaux des comptes source vers les comptes de destination lorsque vous utilisez CloudWatch les règles de centralisation des journaux.

### Note

CloudWatch commence à générer des rapports sur les mesures de centralisation peu de temps après avoir créé une règle de centralisation. Les métriques sont publiées dans la

mesure du possible et ne suivent que les nouveaux événements du journal qui arrivent après la création de la règle de centralisation.

| Métrique                | Description                                                                                                                                                                                                                                                                                                                                                           | Publiée dans          |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| IncomingCopiedBytes     | <p>Volume de données de journal en octets non compressés qui a été répliqué dans le compte de destination. Cette métrique s'applique uniquement aux nouveaux événements de journal qui arrivent après la création de la règle de centralisation.</p> <p>Dimensions valides : SourceRegion, SourceAccount</p> <p>Statistique valide : somme</p> <p>Unités : octets</p> | Compte de destination |
| IncomingCopiedLogEvents | <p>Le nombre d'événements du journal qui ont été répliqués dans le compte de destination. Cette métrique s'applique uniquement aux nouveaux événements de journal qui arrivent après la création de la règle de centralisation.</p> <p>Dimensions valides : SourceRegion, SourceAccount</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p>                  | Compte de destination |
| OutgoingCopiedBytes     | <p>Volume de données de journal en octets non compressés qui a été envoyé depuis les comptes source vers le compte de destination. Cette métrique s'applique uniquement aux nouveaux événements du journal copiés vers</p>                                                                                                                                            | Compte source         |

| Métrique                | Description                                                                                                                                                                                                                                                                                                                                                          | Publiée dans  |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
|                         | <p>la destination après la création de la règle de centralisation.</p> <p>Dimensions valides : DestinationRegion</p> <p>Statistique valide : somme</p> <p>Unités : octets</p>                                                                                                                                                                                        |               |
| OutgoingCopiedLogEvents | <p>Le nombre d'événements de journal envoyés depuis les comptes source vers le compte de destination. Cette métrique s'applique uniquement aux nouveaux événements du journal copiés vers la destination après la création de la règle de centralisation.</p> <p>Dimensions valides : DestinationRegion</p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> | Compte source |

| Métrique                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Publiée dans          |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| CentralizationError     | <p>Nombre d'erreurs rencontrées lors de la réplication des données du journal. Des erreurs peuvent survenir pour diverses raisons, telles que des problèmes d'autorisation liés aux clés KMS, des limites de quota de groupes de journaux ou des incohérences entre les niveaux de journalisation. Pour identifier les groupes de journaux ou les flux de journaux spécifiques dont la réplication a échoué et les raisons de leur échec, surveillez la <code>ErrorType</code> dimension.</p> <p>Dimensions valides : <code>ErrorType</code></p> <p>Statistique valide : somme</p> <p>Unités : aucune</p> | Compte de destination |
| CentralizationThrottled | <p>Le nombre de fois où le traitement de centralisation a été limité. La limitation ralentit le traitement de la centralisation mais n'empêche pas la réplication des données du journal.</p> <p>Dimensions valides : <code>DestinationRegion</code></p> <p>Statistique valide : somme</p> <p>Unités : aucune</p>                                                                                                                                                                                                                                                                                         | Compte source         |

Les dimensions suivantes sont utilisées par les métriques de centralisation.

| Dimension    | Description                                                |
|--------------|------------------------------------------------------------|
| SourceRegion | AWS Région d'où proviennent les données du journal source. |

| Dimension         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SourceAccount     | L'ID du AWS compte d'où proviennent les données du journal source.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| DestinationRegion | AWS Région dans laquelle les données du journal sont répliquées.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| ErrorType         | <p>Type d'erreur rencontré lors de la centralisation. Les valeurs possibles incluent :</p> <ul style="list-style-type: none"><li>• <code>LogGroupQuotaExceeded</code> : le compte de destination a atteint son quota de groupes de journaux.</li><li>• <code>InvalidKMS</code> : La clé KMS est manquante, supprimée, n'est pas activée ou est asymétrique au lieu d'être symétrique.</li><li>• <code>AccessDenied</code> : L'accès a été refusé lors de la tentative de réplication des données du journal. Cela peut être dû à des autorisations insuffisantes, telles que des autorisations de clé KMS incorrectes, des autorisations IAM manquantes ou des politiques de ressources restrictives.</li><li>• <code>LogTierMismatch</code> : Le niveau de journalisation des groupes de journaux source et de destination ne correspond pas.</li><li>• <code>InvalidLogStream</code> : paramètre non valide détecté lors de la création ou de l'écriture d'un flux de journal, par exemple lorsque le nom du flux de journal dépasse la longueur maximale.</li><li>• <code>InvalidLogGroup</code> : paramètre non valide détecté lors de la création ou de la configuration d'un groupe de journaux dans le compte de destination.</li><li>• <code>DestinationEncryptionMismatch</code> : le groupe de journaux de destination existe déjà avec une configuration de chiffrement KMS différente de celle spécifiée par la règle de centralisation.</li></ul> |

# 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, consultez [CloudWatch 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                                                                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>CallCount</code>  | <p>Nombre d'opérations spécifiées effectuées dans votre compte.</p> <p><code>CallCount</code> est publiée dans les espaces de noms <code>AWS/Usage</code> et <code>AWS/Logs</code>.</p> |
| <code>ErrorCount</code> | <p>Le nombre d'opérations API effectuées dans votre compte qui ont donné lieu à des erreurs.</p> <p><code>ErrorCount</code> est publiée uniquement dans <code>AWS/Logs</code>.</p>      |

| Métrique      | Description                                                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ThrottleCount | <p>Le nombre d'opérations API effectuées dans votre compte qui ont été limitées en raison de quotas d'utilisation.</p> <p>ThrottleCount est publiée uniquement dans AWS/Logs.</p> |

## Dimensions

| Dimension       | Description                                                                                                                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Account metrics | <p>Utilisez cette dimension pour obtenir la somme de la métrique pour tous les CloudWatch journaux APIs.</p> <p>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.</p> |
| Service         | Nom du AWS service contenant la ressource. Pour les métriques d'utilisation des CloudWatch journaux, la valeur de cette dimension est Logs.                                                                                                                                                         |
| Class           | Classe de ressource faisant l'objet du suivi. 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 <a href="#">Actions</a> . Par exemple, PutLogEvents .                                                                                                                                        |

# 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                                    | Ajusté | 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 par account/per région                                         |
| AssociateSourceToLimite d'TableIntegration accélération S3 du nombre de transactions par seconde | Chaque région prise en charge : 5             | Non    | Le nombre maximum d'TableIntegration appels AssociateSourceTo S3 par seconde et par account/per région                 |
| AssociateSourceToTableIntegration Limite S3 du nombre de transactions par seconde en rafale      | Chaque Région prise en charge : 10            | Non    | Le nombre maximum d'TableIntegration appels AssociateSourceTo S3 par seconde et par account/per région dans une rafale |



| Nom                                                                                     | Par défaut                                    | Ajusté | Description                                                                                        |
|-----------------------------------------------------------------------------------------|-----------------------------------------------|--------|----------------------------------------------------------------------------------------------------|
| 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-taskappels par seconde par account/per région                    |
| CancellImportTask limite d'accélération des transactions par seconde                    | Par région prise en charge : 1                | Non    | Le nombre maximum d'CancellImportTask appels par seconde par account/per région                    |
| CancellImportTask limite d'accélération du nombre de transactions par seconde en rafale | Par région prise en charge : 1                | Non    | Le nombre maximum d'CancellImportTask appels par seconde et par account/per région dans une rafale |
| CreateExportTask limite d'accélération des transactions par seconde                     | Chaque région prise en charge : 5 par seconde | Non    | Le nombre maximum d'create-export-taskappels par seconde par account/per région                    |
| CreateImportTask limite d'accélération des transactions par seconde                     | Par région prise en charge : 1                | Non    | Le nombre maximum d'CreateImportTask appels par seconde par account/per région                     |
| CreateImportTask limite d'accélération du nombre de transactions par seconde en rafale  | Par région prise en charge : 1                | Non    | Le nombre maximum d'CreateImportTask appels par seconde et par account/per région dans une rafale  |

| Nom                                                                           | Par défaut                                     | Ajusté              | Description                                                                                |
|-------------------------------------------------------------------------------|------------------------------------------------|---------------------|--------------------------------------------------------------------------------------------|
| CreateLogGroup limite d'accélération des transactions par seconde             | Chaque Région prise en charge : 10 par seconde | <a href="#">Oui</a> | Le nombre maximum d'create-log-groupappels par seconde par account/per région              |
| CreateLogStream limite d'accélération des transactions par seconde            | Chaque région prise en charge : 50 par seconde | <a href="#">Oui</a> | Le nombre maximum d'create-log-streamappels par seconde 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                                   |
| DeleteDataProtectionPolicy limite d'accélération des transactions par seconde | Chaque région prise en charge : 5              | Non                 | Le nombre maximum d'delete-data-protection-policyappels par seconde par account/per région |
| 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 | <a href="#">Oui</a> | Le nombre maximum d'delete-log-groupappels par seconde par account/per région              |
| DeleteLogStream limite d'accélération des transactions par seconde            | Chaque région prise en charge : 15 par seconde | <a href="#">Oui</a> | Le nombre maximum d'delete-log-streamappels par seconde par account/per région             |

| Nom                                                                         | Par défaut                                    | Ajusté | Description                                                                                      |
|-----------------------------------------------------------------------------|-----------------------------------------------|--------|--------------------------------------------------------------------------------------------------|
| 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 par account/per région               |
| 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 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 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 suppression-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 par account/per région              |

| Nom                                                                                             | Par défaut                                     | Ajusté              | Description                                                                                                  |
|-------------------------------------------------------------------------------------------------|------------------------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------|
| DescribeImportTaskBatches limite d'accélération des transactions par seconde                    | Chaque Région prise en charge : 10             | Non                 | Le nombre maximum d' DescribeImportTask Batches appels par seconde par account/per région                    |
| DescribeImportTaskBatches limite d'accélération du nombre de transactions par seconde en rafale | Chaque Région prise en charge : 10             | Non                 | Le nombre maximum d' DescribeImportTask Batches appels par seconde et par account/per région dans une rafale |
| DescribeImportTasks limite d'accélération des transactions par seconde                          | Chaque Région prise en charge : 10             | Non                 | Le nombre maximum d' DescribeImportTasks appels par seconde par account/per région                           |
| DescribeImportTasks limite d'accélération du nombre de transactions par seconde en rafale       | Chaque Région prise en charge : 10             | Non                 | Le nombre maximum d' DescribeImportTasks appels par seconde et par account/per région dans une rafale        |
| DescribeLogGroups limite d'accélération des transactions par seconde                            | Chaque Région prise en charge : 10 par seconde | <a href="#">Oui</a> | Le nombre maximum d' describe-log-group s'appels par seconde par account/per région                          |
| DescribeLogStreams limite d'accélération des transactions par seconde                           | Chaque région prise en charge : 25 par seconde | <a href="#">Oui</a> | Le nombre maximum d' describe-log-streams appels par seconde par account/per région                          |

| Nom                                                                                              | Par défaut                                        | Ajusté | Description                                                                                                                 |
|--------------------------------------------------------------------------------------------------|---------------------------------------------------|--------|-----------------------------------------------------------------------------------------------------------------------------|
| DescribeMetricFilters limite d'accélération des transactions par seconde                         | Chaque région prise en charge : 5 par seconde     | Non    | Le nombre maximum d' describe-metric-filters appels par seconde 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-filters appels par seconde par account/per région                                |
| DisassociateSourceFromTableIntegration accélération S3 du nombre de transactions par seconde     | Chaque région prise en charge : 5                 | Non    | Le nombre maximum d'TableIntegration appels DisassociateSourceFrom S3 par seconde et par account/per région                 |
| DisassociateSourceFromTableIntegration Limite S3 du nombre de transactions par seconde en rafale | Chaque Région prise en charge : 10                | Non    | Le nombre maximum d'TableIntegration appels DisassociateSourceFrom S3 par seconde et par account/per région dans une rafale |
| Taille de l'événement                                                                            | Chaque région prise en charge : 1 024 kilo-octets | Non    | Taille maximale des événements du journal en Ko                                                                             |

| Nom                                                                        | Par défaut                                                                                                                                                                                                                                                  | Ajuste | Description                                                                             |
|----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----------------------------------------------------------------------------------------|
| FilterLogEvents limite d'accélération des transactions par seconde         | us-east-1 : 25 par seconde<br>ap-northeast-3 : 5 par seconde<br>ap-southeast-3 : 5 par seconde<br>ca-west-1 : 5 par seconde<br>eu-central-1 : 5 par seconde<br>il-central-1 : 5 par seconde<br>Chacune des autres régions prises en charge : 10 par seconde | Non    | Le nombre maximum d'filter-log-eventsappels par seconde par account/per région          |
| GetDataProtectionPolicy limite d'accélération des transactions par seconde | Chaque région prise en charge : 5                                                                                                                                                                                                                           | Non    | Le nombre maximum d'get-data-protection-policyappels par seconde par account/per région |

| Nom                                                                | Par défaut                                                                                                                                                                                                                                                                                                                   | Ajusté | Description                                                                    |
|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------------------------------------------------------------------------------|
| GetLogEvents limite d'accélération des transactions par seconde    | us-west-2 : 10 par seconde<br>ap-northeast-3 : 10 par seconde<br>ap-southeast-3 : 10 par seconde<br>ca-west-1 : 10 par seconde<br>eu-central-1 : 10 par seconde<br>eu-west-1 : 10 par seconde<br>eu-west-3 : 30 par seconde<br>il-central-1 : 10 par seconde<br>Chacune des autres régions prises en charge : 25 par seconde | Non    | Le nombre maximum d'get-log-eventsappels par seconde par account/per région    |
| GetQueryResults limite d'accélération des transactions par seconde | Chaque Région prise en charge : 10                                                                                                                                                                                                                                                                                           | Non    | Le nombre maximum d'get-query-resultsappels par seconde par account/per région |

| Nom                                                                                           | Par défaut                                    | Ajusté              | Description                                                                                                          |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------|---------------------|----------------------------------------------------------------------------------------------------------------------|
| 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                                     |
| ListSourcesForLimite d'TableIntegration accélération S3 du nombre de transactions par seconde | Chaque région prise en charge : 5             | Non                 | Le nombre maximum d'TableIntegration appels ListSourcesFor S3 par seconde et par account/p er région                 |
| ListSourcesForTableIntegration Limite S3 du nombre de transactions par seconde en rafale      | Chaque Région prise en charge : 10            | Non                 | Le nombre maximum d'TableIntegration appels ListSourcesFor S3 par seconde et par account/p er région dans une rafale |
| 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 par account/per région                                 |
| ListTagsLogGroup limite d'accélération des transactions par seconde                           | Chaque région prise en charge : 5 par seconde | Non                 | Le nombre maximum d' list-tags-log-groupappels par seconde par account/ per région                                   |
| Limite de sessions simultanées de Live Tail                                                   | Chaque région prise en charge : 15            | <a href="#">Oui</a> | Le nombre maximum de sessions Live Tail actives simultanées par compte                                               |
| Groupes de journaux                                                                           | Chaque Région prise en charge : 1 000 000     | <a href="#">Oui</a> | Le nombre maximum de groupes de journaux qu'un compte peut avoir                                                     |



| Nom                                                                                                | Par défaut                                    | Ajusté | Description                                                                                                       |
|----------------------------------------------------------------------------------------------------|-----------------------------------------------|--------|-------------------------------------------------------------------------------------------------------------------|
| 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                                                                 |
| PutBearerTokenAuthentication limite d'accélération des transactions par seconde                    | Chaque région prise en charge : 5             | Non    | Le nombre maximum d' put-bearer-token-authentication appels par seconde par account/per région                    |
| PutBearerTokenAuthentication limite d'accélération du nombre de transactions par seconde en rafale | Chaque Région prise en charge : 10            | Non    | Le nombre maximum d' put-bearer-token-authentication appels par seconde et par account/per région dans une rafale |
| PutDataProtectionPolicy limite d'accélération des transactions par seconde                         | Chaque région prise en charge : 5             | Non    | Le nombre maximum d' put-data-protection-policy appels par seconde par account/per région                         |
| 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'accélération des transactions par seconde                            | Chaque région prise en charge : 5 par seconde | Non    | Le nombre maximum d' put-destination-policy appels par seconde par account/per région                             |

| Nom                                                                      | Par défaut                                        | Ajusté              | Description                                                                                                                                      |
|--------------------------------------------------------------------------|---------------------------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| PutLogEvents limite d'accélération des transactions par seconde          | Chaque région prise en charge : 5 000 par seconde | <a href="#">Oui</a> | Le nombre maximum d'put-log-eventsappels par seconde et par account/p er région                                                                  |
| 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/p er région                                                               |
| PutRetentionPolicy limite d'accélération des transactions par seconde    | Chaque région prise en charge : 5 par seconde     | Non                 | Le nombre maximum d'put-retention-poli cyappels 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-filterappe ls 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-trans formateur 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 |

| Nom                                                                 | Par défaut                                    | Ajusté | Description                                                                                 |
|---------------------------------------------------------------------|-----------------------------------------------|--------|---------------------------------------------------------------------------------------------|
| StartQuery limite d'accélération des transactions par seconde       | Chaque Région prise en charge : 10            | Non    | Le nombre maximum d'appels de requête de démarrage par seconde et par région account/per    |
| 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/p<br>er 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/p<br>er région      |
| TestTransformer limite d'accélération des transactions par seconde  | Chaque région prise en charge : 5 par seconde | Non    | Le nombre maximum d'appels de transformateurs 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-groupappels par seconde et par account/per région             |

| Nom                                                              | Par défaut                                    | Ajustable           | Description                                                                     |
|------------------------------------------------------------------|-----------------------------------------------|---------------------|---------------------------------------------------------------------------------|
| 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 |
| enregistre les détecteurs d'anomalies                            | Chaque région prise en charge : 500           | <a href="#">Oui</a> | 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 panneau 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            |
|----------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">Requêtes enregistrées avec paramètres pour CloudWatch Logs Insights</a>    | Vous pouvez désormais créer des modèles de requêtes réutilisables avec des paramètres nommés dans CloudWatch Logs Insights. Définissez un modèle unique et transmettez différentes valeurs au moment de l'exécution au lieu de conserver plusieurs requêtes enregistrées quasiment identiques. Pour plus d'informations, consultez la section <a href="#">Utilisation de requêtes enregistrées avec des paramètres</a> . | 26 mars 2026    |
| <a href="#">Politique d'CloudWatchLogsAPIKeyaccès aux nouveaux CloudWatch journaux</a> | CloudWatch Logs a publié une nouvelle politique gérée CloudWatchLogsAPIKeyAccess qui permet l'authentification par clé d'API CloudWatch Logs et l'ingestion cryptée des journaux. Pour plus d'informations, consultez la section <a href="#">CloudWatch Journalisation des mises à jour des politiques AWS gérées</a> .                                                                                                  | 17 février 2026 |

[CloudWatchLogsRead  
OnlyAccess Politique mise à  
jour](#)

Des autorisations pour les actions d'administration de l'observabilité ont été ajoutées pour [CloudWatchLogsRead OnlyAccess](#) permettre l'accès en lecture seule aux pipelines de télémétrie et aux intégrations de tables S3.

2 décembre 2025

[CloudWatchLogsFullAccess  
Politique mise à jour](#)

Des autorisations pour les actions d'administration de l'observabilité ont été ajoutées pour [CloudWatchLogsFull Access](#) permettre l'accès en lecture seule aux pipelines de télémétrie et aux intégrations de tables S3.

2 décembre 2025

[Nouvelle fonctionnalité de  
gestion des CloudWatch  
journaux dans Logs](#)

CloudWatch consolide la gestion des journaux en un seul service doté de fonctionnalités de gouvernance intégrées sans stocker ni conserver plusieurs copies des mêmes données dans différents outils et magasins de données. Pour plus d'informations, consultez la section [Gestion des journaux](#).

2 décembre 2025

[Nouvelle fonctionnalité  
d'ingestion et de normalisation  
des données dans CloudWatch  
Logs](#)

CloudWatch collecte automatiquement les journaux AWS vendus sur l'ensemble des comptes et des AWS régions grâce à l'intégration aux AWS Organizations et à des connecteurs prédéfinis pour les sources tierces. Pour plus d'informations, consultez [Transformation des journaux lors de l'ingestion](#).

2 décembre 2025

[Nouvelle fonctionnalité  
d'analyse et de requête basée  
sur les sources de données  
dans CloudWatch Logs  
Insights](#)

CloudWatch Insights introduit des facettes pour les requêtes. Les facettes sont utiles pour analyser les journaux car elles vous permettent de filtrer et de regrouper vos données de manière interactive sans exécuter de requêtes. Une facette est un champ de vos journaux (tel que ServiceName ou StatusCode) qui permet le filtrage, l'agrégation et l'analyse entre les groupes de journaux. Pour plus d'informations, voir [Utiliser des facettes pour regrouper et explorer les journaux](#).

2 décembre 2025



## [Intégration des nouvelles tables S3](#)

[L'intégration de S3 Tables](#) vous CloudWatch permet d'accéder aux données des journaux ingérées à CloudWatch l'aide de moteurs d'analyse tels qu'Amazon Athena, Amazon Redshift et d'outils tiers prenant en charge la connexion à des boutiques compatibles avec Apache Iceberg. Pour plus d'informations, consultez [Access logs with S3 Tables Integration](#).

2 décembre 2025

## [CloudWatch Logs : ajout de la prise en charge de la centralisation des journaux entre comptes et entre régions](#)

CloudWatch Logs a ajouté la prise en charge de la centralisation des journaux entre comptes et entre régions, ce qui permet à CloudWatch Logs de rassembler les données des journaux des comptes AWS Organizations membres dans un compte central à des fins d'analyse. Pour plus d'informations, consultez la [section Centralisation des journaux entre comptes et entre régions](#).

17 septembre 2025

[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 AWS PCS des journaux](#)

CloudWatch Logs Insights ajoute la prise en charge AWS PCS des journaux. Pour plus d'informations, voir [Activer la journalisation à partir AWS des services](#). Les nouvelles sources de log pour AWS PCS, `PCS_SCHEDULER_LOGS` et `PCS_JOBCOMP_LOGS` sont documentées dans [PutDeliverySource](#)

12 juin 2025

[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 politique  
s gérées pour prendre en  
charge les résumés en  
langage naturel](#)

Des autorisations pour  
cloudwatch:Generat  
eQueryResultsSumma  
ry ont été ajoutées  
CloudWatchLogsFullAccesset  
CloudWatchLogsRead  
OnlyAccesspermettent de  
générer un résumé en langage  
naturel des résultats de  
la requête. Pour consulter  
le contenu des politique  
s, consultez [CloudWate  
hLogsFullAccesset CloudWate  
hLogsReadOnlyAccess](#) dans  
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

[CloudWatchOpenSearchDashboardsAccesset CloudWatchOpenSearchDashboardsFullAccesssont de nouvelles politiques IAM](#)

CloudWatch Les journaux ont ajouté deux nouvelles politiques IAM, CloudWatchOpenSearchDashboardsFullAccesset CloudWatchOpenSearchDashboardsAccess. CloudWatchOpenSearchDashboardsFullAccessaccorde l'autorisation de créer et de gérer des intégrations avec OpenSearch Service. CloudWatchOpenSearchDashboardsAccessautorise 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

### [CloudWatchLogsRead OnlyAccesspolitique mise à jour](#)

CloudWatch Logs a ajouté l'`cloudwatch:GenerateQuery` autorisation de `CloudWatchLogsRead OnlyAccess`, 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 régulières 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 régulières 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 régulières flexibles intégrées à des modèles de filtres. Pour plus d'informations, consultez la section [Syntaxe du modèle de filtre](#) dans le guide de l'utilisateur 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 protection des données au niveau du compte. Ces politiques 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 éventuellement filtrer le flux d'événements du journal affiché en fonction de termes spécifiés , et également mettre en évidence les événements du journal contenant des termes spécifiés. Pour plus d'informations, 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:Stop LiveTail 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 de 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.

| Modifier                                   | 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 <a href="#">Utilisation des CloudWatch journaux avec les points de terminaison VPC de l'interface</a> . | 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 <a href="#">Qu'est-ce qu'Amazon CloudWatch Logs ?</a> ou <a href="#">Journalisation des requêtes DNS</a> 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 <a href="#">Étiqueter les groupes de journaux dans Amazon CloudWatch Logs</a> .                                                                                                                                                | 13 décembre 2016    |

| Modifier                                                                                           | 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 <a href="#">Transition des métriques aux journaux</a> .                                                                                                                                             | 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 <a href="#">Afficher les données du journal envoyées à CloudWatch Logs</a> . | 29 août 2016        |
| Ajout de la prise en charge d'Amazon CloudWatch Logs et des nouvelles métriques de CloudWatch Logs | Ajout AWS CloudTrail du support pour CloudWatch Logs et les journaux. Pour de plus amples informations, veuillez consulter <a href="#">Logging CloudWatch Logs, API et opérations de console dans AWS CloudTrail</a> .                                                                                                              | 10 mars 2016        |
| Ajout de la prise en charge de l'exportation des 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 <a href="#">Exporter les données du journal vers Amazon S3</a> .                                                                                                              | 7 décembre 2015     |

| Modifier                                                                                          | 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 du support pour 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 <a href="#">Qu'est-ce qu'Amazon CloudWatch Logs ?</a> . | 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.